Protection of Big Data Privacy on Multiple Cloud Providers by Asymmetric Security Scheme

Parinya Suwansrikham* and Kun She*
*School of Information and Software Engineering
University of Electronic Science and Technology of China, Chengdu, Sichuan, China
E-mail: parinsu@hotmail.com
E-mail: kun@uestc.edu.cn

***Abstract*—Big data is the name that defines data which has enormous size and unstructured. Due to the file size is pretty huge. It is impracticable to store a large file in one storage volume. However, cloud computing is a solution to this impossible. Data owner can store the file in a cloud storage provider (CSP). Nevertheless, the new dilemma has arisen. Relying on single cloud storage may generate trouble for the customer. A CSP may stop its service anytime. Moreover, the CSP is the third party that user have to trust without verification. In that case, the privacy or unauthorized accessing of data may be violated without notice. To overcome this risk, we propose secure data storage scheme for big data storing on multiple CSPs. The one big data file is split into chunks and distributed to multiple cloud storage provider. After splitting the file, metadata is generated. Metadata is a place to keep chunks information, includes; chunk locations, access paths, username and password of the data owner, methods to connect each CSP. The metadata is encrypted and transferred to the user who requests to access the file. The user utilizes the metadata and chunks of the file to compose the original file. This method will minimize the risk of privacy. The goal of this paper is to provide the method to protect the privacy of data stored on multiple cloud storage providers. Furthermore, we discuss and analyze how this data storage scheme promote the protection of big data privacy.**

I. INTRODUCTION

Big data is another new word and also be an enthusiastic matter in this modern information technology. Big data is generated from various devices and sources, such as business operational data, scientific data, social networking, web logs, video streaming, sensor data, smartphone data. The first example is imagery from Google Maps offers over 20 petabytes (PB) of imagery. The second example is Video streaming from Netflix. It has over 3.14 PB of video in the master copies. [1]

Also, the format of big data is not in primitive forms. Data is structured, semi-structured, or unstructured. Structured data is well organized structure, such as, XML or JavaScript Object Notation (JSON). Semi-structured data is not defined in row and column as in structured data but it contains some structure, for instance, HTML, XML with a schema. Unstructured data is data that is raw text files and contain no structure, for example, server log file, a Portable Document Format (PDF) file, e-mail.

Some data is generated in streams. A data stream is a sequence of digitally encoded signals used to represent information in transmission". Reference [2] also defines big data is data that has grown to a size that requires new techniques to store, organize, and analyze the data. Therefore, big data is usually characterized by "5V" –volume, velocity, and variety.

Volume defines the size of data. As mentioned above, the data is on the scale of terabytes or petabytes. The data may be a massive bulk of file or be in the form of streaming data. We can consider streaming data has infinite size. It is no end until the streaming stops.

Velocity describes the speed of big data characteristic. Due to advancement and breakthrough in network technology, the speed of communication increases dramatically, from 128 kbps speed in ISDN network to Gbps level in gigabit LAN area network. Velocity does not only describe the speed of communication, but it is also defined the speed of generating, capturing and sharing of the data.

Variety means a reproduction of new data types from various sources, such as social networks, machine devices, mobile devices, also sensors that are embedded or integrated with other hardware. The data generating is not producing only basic data types, namely texts, voices, videos. There is sophisticated data type generated, such as geospatial data. It is the data that identifies the geographic location of features and boundaries on Earth, such as natural or constructed features, oceans. Spatial data is usually collected as coordinates and topology.

Due to the unique "3V" characteristic, the traditional processing is not suitable to operate with big data. It requires new theory, methods, operations to solve big data problem or even get the benefit from it.

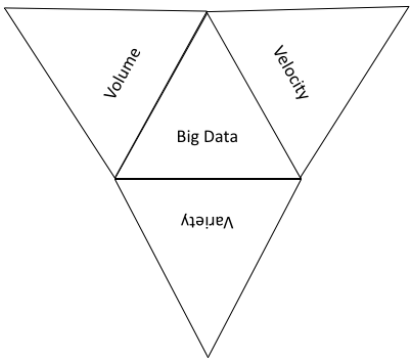

Fig.1 The 3 Vs of big data

Figure 1 summarized the 3 Vs characteristic of big data. In some sources define characteristics of big data as 5 Vs: volume, variety, velocity, veracity, and value. In this paper, we consider 3 Vs is enough to describe big data characteristics.

Due to its unique characteristic, we cannot use the classic data processing method with big data anymore. Big data is hard to store, process, and analyze. The size of big data is enormous. It cannot be fitted in only one storage device. One way to store big data is to keep it in the data center. It cannot be processed or analyzed by classical algorithms or methods. If we use the classical algorithm to process or analyze big data, it would be impossible to archive answer in reasonable time. Even though we get the answer, it would come with the high resource usage.

Cloud computing is one of the paradigm shifts in modern information technology era. The cloud computing transforms service for enterprise applications and has become a vital architecture to perform large-scale and complex computing. The significant benefits of cloud computing are virtualized resources, parallel processing, security, and data service integration with scalable data storage.

[3] NIST defined the cloud as "A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources such as, networks, servers, storage, applications, and services, that can be rapidly provisioned and released with minimal management effort or service provider interaction." From NIST definition of cloud computing, we can see that cloud computing is a model that can be easily and economically to use from both consumers and providers point of view.

There are 3 basic service models of cloud computing. First is software as a service (SaaS). This model user access to software that host on cloud through browser. The example of SaaS is e-mail service. Second service is platform as a service (PaaS). Service providers usually provide all the details for the computing platform, connectivity, elasticity/scalability, backup. The third service model is infrastructure as a service (IaaS). This model let user control not only software and platform but also let user control more physical details of cloud. User may control Virtual Machine of even servers. The example for IaaS is Web Hosting.

[4] Cloud storage is one kind of service that provides from a service provider. The example of cloud storage providers nowadays is Dropbox, Google Drive, OneDrive. Cloud storage presents practically limitless storage capacity for users. Now, it is a trend that large numbers of organizations are storing their data into cloud storage. However, only utilizing one cloud storage provider is more apparent to experience from single-point failures and vendor lock-in. As a consequence, the multi cloud storage that relies on multiple cloud storage providers to locate data at some level. It can avoid vendor lock-in and improve fault-tolerance. These features are considered advantageous to systems or applications such as data backup, document archiving, or electronic health recording, which need to keep a large amount of data.

In such circumstances, the technique called erasure coding is employed for further development as it can significantly diminish the price of data storage compared to full replication. The total quantity of data that must convey over the network can be lessened. With erasure coding, each data object is equally divided into k blocks; then these blocks are used to generate n - k encoded data blocks (n blocks in total, $n > k$). This parameter is called as a parameter (n,k) of erasure coding. These n data blocks will be uploaded into n cloud storage providers. Each provider is holding just one data block. Any k blocks from the n ones can be used to reconstruct the original data file. This technique is applied in some of the previous works, RACS [5], HAIL [6], DepSky [7], Triones [4]. However, this technique is very costly in term of performance. It requires large memory usage, CPU time.

Multi cloud storage is the combination of public, private or managed clouds including managed services or service providers. Multi cloud data systems can enhance data sharing, and this aspect will be significant of great help to data users. Most business organizations share their data with either their clients or suppliers and consider data sharing as a priority.

Cloud computing and big data are associated to each other perfectly. Cloud computing offers a lot of benefits, such as cost saving and scalability. Big data facilitate users by providing the ability to use specialty computing to process distributed queries across multiple set of data and return resultant assortments promptly. Big data utilizes distributed storage technology based on cloud computing rather than local storage attached to a computer or electronic device. Big data evaluation is driven by fast-growing cloud-based applications developed using virtualized technologies. A way to store big data is depositing data in cloud storage instead of spending substantial funds on building a massive data center in the organization. Cloud storage is a crucial application of cloud computing. Consumers can store their files on cloud storage provider (CSP). CSP implements storage-as-a-service which supports database technologies, both SQL and NoSQL. After omitting their files on CSP, data owners deliver the maintaining storage duty to the CSP.

In this research, we propose big data allocation on multi cloud to overcome vendor lock-in, data security, and improve privacy protection. The asymmetric security idea is applied in this work. The data consists of 2 parts: public and secret parts. The public part is data part which uploaded and stored on multi cloud storage. The content of this part is allowed to leak or in at hand of the attacker. The content that indicates privacy of data owner would be screened out before splitting. The secret part is critical data part. Its content would not allow leaking or fall at hand of the attacker. The content consists of a method to reconstruct the original file, location, path of each chunk that store on multi cloud storage providers. The secret part is kept by the data owner. We called the secret part as metadata.

Alternatively, performing encryption on the data file is an inappropriate way. Due to the enormous volume or size of big data, it is almost impossible to encrypt the entire file. Even though, we can encrypt the big file. The encryption process is

time-consuming. Also, it is struggle for the user to decrypt the encrypted file back to original form. Instead of doing this, we can encrypt only the metadata file, which contains the place of chunks, access paths, and other related information. The size of metadata comparing with the whole file is significantly small. So, it is more practical to encrypt the metadata file rather than the big data file. Metadata contain the location of each chunk and access paths. If a data user requires accessing the file, data owner will send metadata manually or automatically to the data user. The data user utilizes metadata as a key and map to grant him access chunks on multiple cloud storage and retrieve chunks to his machine. We have the following contributions:

- We propose an approach for sharing mass distributed storage which prevent privacy of the data stored on cloud operators.
- We also propose security model to analyze the strength of the proposed system.

Our paper is organized as follows. In section II, we will show the related work. And in the section III, we will introduce the background. In section IV, we will introduce problem formulation. The analysis of our analysis and simulation will be introduced in section V. In section VI and the last section, we will introduce the conclusion and references respectively.

II. Related Works

Several preceding works are related to multi cloud storage and inspire us. In this section, we review that kind of works.

Data storing on the cloud do not just rely on security mechanism of the cloud provider. Many works add security mechanism into data to ensure that stored data in the cloud is safe. The most popular security mechanism that use in cloud storage is data encryption. Ciphertext Policy Attribute Based Encryption (CP-ABE) is one of the popular encryption technique that use in cloud storage. Many researchers apply this technique [8] also extended version of CP-ABE [9]. Data user can search on encrypted data instead of download all data decrypt it then performs search operation. Which is the inefficient way.

Many works are *Multi cloud storage based on erasure coding*. RACS [5] was the first work to attempt forward the idea of applying erasure coding in the multi cloud storage. RACS apply RAID idea on cloud storage scale. The purpose of RACS was to avoid vendor lock-in avoidance and increase the service availability. HAIL [6] aimed at protecting the availability and security of data by using erasure coding in the multi cloud storage. HAIL let a client knows that the stored file is accessible and retrievable. DepSky [7] is a system that enhances the availability, integrity, and confidentiality of information stored in the cloud. The enhancement performed through the encryption, encoding, and replication of the data on different clouds that form a cloud-of-clouds. Depsky also focuses on encryption and encoding at a reasonable cost and access latency. Triones [4] is multiple cloud storage system with erasure coding to achieve specific benefits including fault-tolerance improving or vendor lock-in avoiding. Its systematic model the data placement in multi cloud storage by using erasure coding. Triones focus on the optimization issue in general. The key to optimizing the multi cloud storage is to choose providers and erasure coding parameters effectively.

The allocation of file fragments on the distributed system also improve security and privacy of data on cloud storage. It can be categorized into 2 ways, pure fragmentation, and replicate fragmentation. Pure fragmentation means each piece of data has only 1 copy. Reference [10] proposes secure allocating processing (SAP) algorithm for the S-FAS. To improve the security level and consider its performance using the various feature of an extensive distribute system. The fragment of file in this proposed scheme has only 1 copy. In contrast, replicate fragmentation means each piece of data has more than 1 copy. Reference [11] develop a secure replica allocation plot called SecRA to enhance security, reliability, and performance of a cloud storage system. The cloud system in this scheme is vulnerabilities. SecRA combines the methods of replication and fragmentation with secret sharing in a heterogeneous cloud system, where storage nodes are constituted of various server classes regarding vulnerability characteristics. The number of copy of file fragment is vary. In order to assess the security, both works derived assurance model from reference [12].

III. Background

[13] Cloud storage is managed and operated by cloud storage providers in the form of service and interact with data. Also, cloud storage providers prepare the physical environment for their users. Users access storage service via cloud storage APIs or even web browsers. As the best infrastructure to accommodate big data, cloud storage has attracted attention from both industry and academia. However, various security and privacy concerns arising from the nature of cloud storage are preventing users from subscribing to this service. These concerns include:

- Untrustiness. Users can work on his data remotely via the APIs provided by cloud storage provider, not own client or application.
- Dynamic environment. Users and cloud storage providers may change the setting, configurations or services as requested or offered. This situation causes data moving around the organization or even increase the likelihood of revealing private data or information.
- Uncensored new services. Cloud storage providers may install or try their new services or features. This activity may add risk into sensitive information, which have no consent from users.

These concerns cause users reluctant to use cloud storage service storing their files. They are afraid their privacy would be revealed unintentionally. [14] Security and privacy issues of big data are confidentiality, integrity, availability, monitoring, and auditing. User privacy may be personal information or sensitive data stored on cloud storage.

[15] From a data user perspective, 3 critical requirements have to satisfy in a cloud data service, concerning the security of outsourced data contents:

Data confidentiality – data confidentiality feature guarantees that data contents are concealed from unapproved users. Outsourced data are saved in distant cloud storage servers. It is out of the control of data owners. In this situation, only authorized users can obtain these data based on the allowed rights. On the other hands, other users, including cloud providers, should not obtain any information of outsourced contents. Moreover, it is better to strengthen a fine-grained access control.

• Data integrity – the data integrity attribute needs managing and assuring the correctness and completeness of data. A cloud client and data owner expect that their outsourced data in remote servers have to be exact and accurate. The outsourced data should not tamper, altered or maliciously destroyed. Consequently, once a part of outsourced data has been altered or destroyed, the cloud client or data owner should be notified or recognized these fraud activities. Also, the cloud provider should be able to reconstruct or repair the whole data contents from non-corrupted pieces of data.

• Privacy preservation – user privacy prevention examines both the protection of private information and assuring the unlinkability between various accesses to outsourced data. Personal data include the data or attributes that indicate or identify the user. Moreover, his access patterns need to be covered from unauthorized existences. This covering also prevent identity of user to be exposed.

These 3 critical perspectives are crucial and essential to guarantee that data hosting in remote cloud storage server is secure enough. Otherwise, the unauthorized users or attacker would retrieve or obtain stored data and harm privacy of data owner. The unauthorized user may impersonate as data owner and use his data in the wrong way.

IV. Problem Formulation

This section, we introduce the problem definition, followed by system model and system architecture

A. Problem Definition

In this work, we focus on the privacy protection of big data file. We assume that the attacker uses a method to unauthorized access cloud storages. The primary purpose of an attacker is to reconstruct original file from stolen chunks and secret part (metadata). Once the attacker has enough data parts, he can reconstruct the original file. If the attack is successful, the attacker will access and retrieve pieces of the file which store on multi cloud storage. Our scheme allows the attacker to retrieve *m* fragments or pieces of the file. *m* is also considered as a threshold value to guarantee file protection. If he retrieves fragment less than *m* fragments, he cannot reconstruct the original file.

This file is stored on cloud storage in a multi cloud environment. This environment involves 2 entities. The first entity is data owner, who stored their file on storage providers. The second entity is cloud service providers, who control resource and service for file or data storage.

If an attacker wants to reconstruct the original file, he has to fulfill 2 requirements. First, the attacker has to breach the security of cloud storage provider then access file location and get enough and right pieces of the target file. The fragments stored on cloud storage are the public part of the big data file. Even, we lost all fragments of the public part to the hand of the attacker. It could not harm data owner privacy.

Second, the attacker must possess secret part. This part would be stored locally inside data owner machine or remotely on dew server. A big data file is split into chunks then stored in cloud storage. One chunk is stored on one cloud storage provider. However, cloud storage provider does not only keep chunks of one user. Also, it keeps chunks or files of other users. Cloud attack may occur from unauthorized activities internally or externally. For our storage scheme, it is hard for the attacker to guess the ownership of each chunk.

B. System Model

A data owner possesses a large file *F*. Initially, data owner has to screen or filter privacy data out from file *F*. For instance, if the file is a medical record, the data owner may remove his name, address or any contents that indicate himself from the record. If it were an image, his face might be blur or crop out from the image file. These removed contents are stored in secret part. *(F_S)*. However, secret part is not only the sensitive data, but also including the location of chunks which stored in multi cloud storage, and connection methods. The secret part would be composed with the public part later in reconstruct process. The public part *(F_P)* is the remaining part and will be stored on multi cloud storage, as shown in Figure 2.

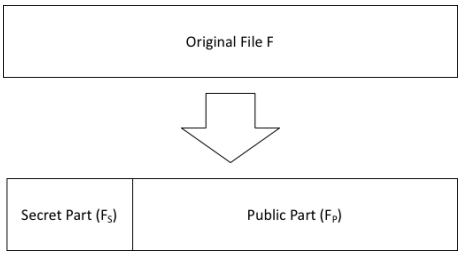

Fig.2 Secret part and public part

Data owner divides public part into equal *n* chunks, then upload each chunk to *n* cloud storage providers (CSPs). Figure 3 shows data file splitting and uploading from data owner. Each CSP contains part of file approximately $\frac{F}{n}$ bits.

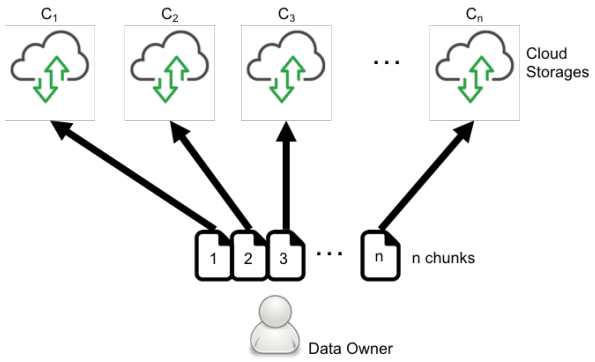

Fig.3 Splitting and Distributing chunks.

When data user wants to utilize the target data *F*. The data user has to ask permission from data owner. Once the data owner receives a request from data user, he would grant the request automatically or manually by sending the secret part to the data user. After the request is granted, the data user will make communication links to multi cloud storage following the information from the secret part. The device of data user might be any devices of users, such as mobile phone, tablet, laptop, or PC. We call them as 'user node'. CSPs and user nodes connect via Internet protocols. Each CSP is independent of each other CSP. This position means there is no file distribution contribute among CSPs themselves. Also, the same circumstances as in user nodes, there is no file contribution among user nodes. There is only file distribution occurs from CSPs to user nodes.

A data user request that requires downloading for F. For our model, we assume that the data is divisible into several pieces of data or chunks with equal size. Each chunk contains some important and sensitive information of the whole data. Thus, revealing the information of a single or few chunks does not make sense to malicious users.

V. Analysis and Simulation

A. Security Model Analysis

In this works, we separate data contained in a big data file into 2 parts: public and secret. The public part is insensitive or not important content. This part can be leaked or revealed to unauthorized users or attackers with no effect to the data owner. The public parts are stored in multiple cloud storages. The public part contains the content of the file such as text, database, pictures, voice or video. Our scheme will distribute the public part into several cloud storages. Each cloud storage provider holds a part of the public data part. On the other hand, the secret part is sensitive and important content. If it is leaked or revealed to unauthorized users, it would harm the data owner. The secret part is stored locally and protected by data owner.

Comparing this scheme with storing file in single cloud storage, if there is insider attack occurs in the storage provider, the attacker will get whole data file at once. In contrast, if there is insider attack of a cloud storage provider of our scheme, the attacker will get only a part of the data file. It is useless for the attacker that holds only one piece or some pieces of the data file. The attacker has to attack each of cloud storage provider in order to retrieve the complete data file.

Another part is the secret part (metadata). This part retains the storage index information for each data parts, file structure, or file header. It is similar to treasure map for user escorting to all data parts. Also, the secret part is the most crucial part of original file reconstruction. The user must have this part from data owner along with data chunks form each cloud storage in order to reconstruct original data file.

As the owner, he will hold the storage path (contain in secret part) confidentially. The data owner may store this part locally on his machine or another machine on behalf of him. Even the attacker retrieves some data parts, he cannot reconstruct or interpret the original data file without the secret part. The data owner will send to the user who requests him to access his file on multiple cloud storages.

The volume of the secret part is in a level of kilobytes to a few megabytes. It is acceptable and possible to encrypt secret part, for enhancing security, before transferring it to the authenticated users. Due to the enormous size of big data, it is impossible to encrypt the whole file. We have many choices of encryption algorithm to encrypt secret part. Also, we have many options to transmit encrypted secret part to request users.

We define the public part of a big data file as F_P. The F_P is split into n chunks or fragments $\{F_{P1}, F_{P2}, \dots, F_{Pn}\}$. In this research, each fragment has only one copy. The reason behind this idea is to save storage space which mean save the cost of service usage. Each copy is host on each cloud storage provider.

Let p_i is the probability that an unauthorized user is successfully attacking storage and access the target fragment of the file. Cloud storage does not store only one fragment of the file. It contains pieces of the file from different users or various sources. Let n_i is a number of fragment or pieces of the file stored on cloud storage i, and only one piece of target file store on cloud storage.

So,

$$p_i = \frac{1}{n_i} \tag{1}$$

However, only one successfully attacking is not enough. The attacker has to attack successfully n cloud storages that store all fragment of file F. We can define successfully attack whenever the attacker access n cloud storages and retrieve the right fragment of file f.

Define Z is an event that an attacker successfully attacks and retrieve the k fragments of the target file from k cloud storages. (1 correct fragment from 1 cloud storage) The probability of event Z is $P_{MC}(Z)$ which is

$$P_{MC}(Z) = \prod_{i=1}^{k} p_i = \prod_{i=1}^{k} \frac{1}{n_i} \tag{2}$$

$P_{MC}(Z)$ is probability of event Z in our multi cloud scheme.

From equation (1) and (2), the probability that attacker successfully retrieve of target file F_P is depend on the inverse of number of file fragment or pieces store on cloud storage.

B. Simulation

We assume that cloud storages store fragments of file *F*. The attacker has to breach enough cloud storage servers. If he breaches enough servers of public part F_P, then attacker can reconstruct original file *F*. In this case we assume the attacker already possess secret part.

The simulation is study the effect of fragment number to probability that an attacker successfully attacks storage cloud and retrieve the fragments of the target file. To simplify (2), we assume that each cloud storage store *l* fragments of files. This assumption makes $n_i = l$. So, the equation (2) is simplified to (4),

$$P_{MC}(Z) = \prod_{i=1}^{k} \frac{1}{n_i} = \prod_{i=1}^{k} \frac{1}{l}$$

$$P_{MC}(Z) = \frac{1}{l^k} \quad (3)$$

Table 1 shows the probability of $P_{MC}(Z)$ calculated from (3).

Table 1 Value of $P_{MC}(Z)$ calculated from value of *k* and *l*

l	$P_{MC}(Z)$, k = 5	$P_{MC}(Z)$, k = 6	$P_{MC}(Z)$, k = 7	$P_{MC}(Z)$, k = 8	$P_{MC}(Z)$, k = 9	$P_{MC}(Z)$, k = 10
5	0.00032	0.000064	0.0000128	0.00000256	0.000000512	1.024E-07
6	0.000128601	2.14335E-05	3.57225E-06	5.95374E-07	9.9229E-08	1.65382E-08
7	5.9499E-05	8.49986E-06	1.21427E-06	1.73467E-07	2.47809E-08	3.54013E-09
8	3.05176E-05	3.8147E-06	4.76837E-07	5.96046E-08	7.45058E-09	9.31323E-10
9	1.69351E-05	1.88168E-06	2.09075E-07	2.32306E-08	2.58117E-09	2.86797E-10
10	0.00001	0.000001	0.0000001	0.00000001	0.000000001	1E-10
11	6.20921E-06	5.64474E-07	5.13158E-08	4.66507E-09	4.24098E-10	3.85543E-11
12	4.01878E-06	3.34898E-07	2.79082E-08	2.32568E-09	1.93807E-10	1.61506E-11
13	2.69329E-06	2.07176E-07	1.59366E-08	1.22589E-09	9.42996E-11	7.25382E-12
14	1.85934E-06	1.3281E-07	9.48645E-09	6.77604E-10	4.84003E-11	3.45716E-12
15	1.31687E-06	8.77915E-08	5.85277E-09	3.90184E-10	2.60123E-11	1.73415E-12

From Table 1, we try to change value of *k* and *l*. We vary size of fragments that store on each cloud (*l*) from 5 to 15 fragments per cloud. Also, number of fragments from compromised cloud storages *(k)* is also varying from 5 to 10 fragments.

Values from Table 1 is plotted as line graph shown in Figure 4.

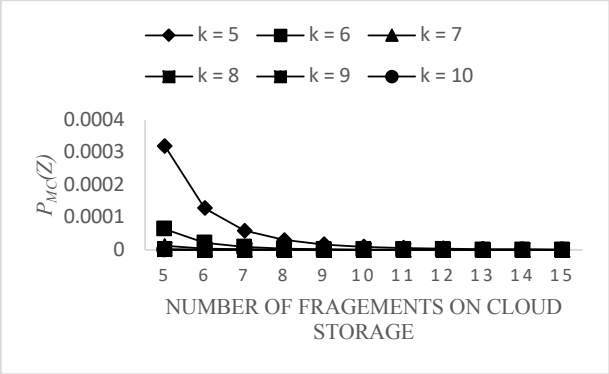

Fig.4 Probability of successfully attacking and attacker get the target fragments on multi cloud scheme.

From Figure 4 and Table 1, the initial probability of each value of *k* is quite low. For example, $P_{MC}(Z) = 0.000064$ when $k = 6$ and $l = 5$. This situation means the attacker hold 6 fragments of file from 6 breached cloud storages (1 fragment from 1 cloud storage), and each cloud storage stores 5 fragments. The probability that all 6 fragments belong to fragments of target public part, F_P, is 0.000064. As the value of *l* is increase the value of probability is plummet nearly zero. However, number of fragment greater than 5 is not significantly change in value of $P_{MC}(Z)$.

We compare our scheme with traditional file distribution scheme, which is client/server. In this scheme, we consider one cloud storage provider as a server. On the cloud storage, all of *n* fragments of target file are stored along with ordinary chunks of files. These *n* fragments are including in *l* stored fragments These fragments might be the files of different users that attacker does not interest to retrieve or undesired files.

Let $P_{CS}(Z)$ is probability of event *Z* in client/server scheme. The attacker may successfully breach the single cloud storage and retrieve *k* chunks from *l* chunks. The probability of successfully attack and attacker retrieve chunks of target file is,

$$P_{CS}(Z) = \frac{1}{\binom{l}{k}} \quad (4)$$

when $k \leq l$. $P_{CS}(Z)$ is inverse of combination of choose pattern *k* chunks from *l* chunks.

In order to reconstruct the original target file, the attacker must retrieve the correct $k = n$ chunks. This situation could happen only one in total number of methods of guessing *k* chunks out from *l* chunks as in (4).

For client/server scheme, Table 2 show the value of $P_{CS}(Z)$ by varying value of *k* and *l*.

Table 2 Value of $P_{CS}(Z)$ calculated from value of *k* and *l*

l	$P_{CS}(Z)$, k = 5	$P_{CS}(Z)$, k = 6	$P_{CS}(Z)$, k = 7	$P_{CS}(Z)$, k = 8	$P_{CS}(Z)$, k = 9	$P_{CS}(Z)$, k = 10
5	1					
6	0.166666667	1				
7	0.047619048	0.142857143	1			
8	0.017857143	0.035714286	0.125	1		
9	0.007936508	0.011904762	0.027777778	0.111111111	1	
10	0.003968254	0.004761905	0.008333333	0.022222222	0.1	1
11	0.002164502	0.002164502	0.003030303	0.006060606	0.018181818	0.090909091
12	0.001262626	0.001082251	0.001262626	0.002020202	0.004545455	0.015151515
13	0.000777001	0.000582751	0.000582751	0.000777001	0.001398601	0.003496503
14	0.0004995	0.000333	0.000291375	0.000333	0.0004995	0.000999001
15	0.000333	0.0001998	0.0001554	0.0001554	0.0001998	0.000333

From Table 2, the blank space indicates that value of denominator part in (4) is negative if value of $k > l$. Also, in the case of $k = l$, the probability from (4) $P_{CS}(Z) = 1$. In both situations, we consider that the attacker already successfully attacks a server in client/server scheme and retrieves enough fragments to reconstruct the target file.

At the same value of *k* and *l*, the probability $P_{MC}(Z)$ always less than $P_{CS}(Z)$. This situation implies the chance that attacker successfully breaches the security of cloud storages and retrieve k fragments of the target file from client/server scheme is easier than proposed scheme. For instance, at $k = 10$ and $l = 15$, the value of $P_{CS}(Z) = 0.000333$ while $P_{MC}(Z) = 1.73415*10^{-12}$.

Figure 5 and 6 show the value of $P_{CS}(Z)$ and $P_{MC}(Z)$ at the same value of *k* and *l*. We can notice that initial value of $P_{CS}(Z)$ begins at 1.00 and $P_{MC}(Z)$ nearly zero. Which means if the

attacker successfully attacks and retrieves $k = l$ fragments from client/server scheme, the attacker already gets the fragments of target file and ready to reconstruct original copy. However, this probability is decreases as we increase number of stored chunks on the cloud server. In contrary, even the attacker successfully attacks and retrieves $k = l$ fragments from proposed scheme, the probability that attacker retrieve the exact k fragments of the target file is extremely low.

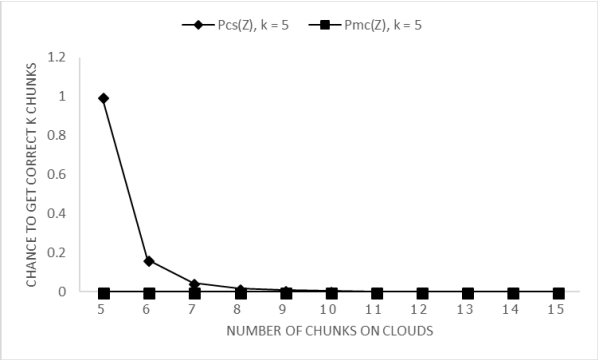

Fig.5 Comparing the probability of get correct k chunks of target file on client/server scheme and proposed scheme at $k = 5$

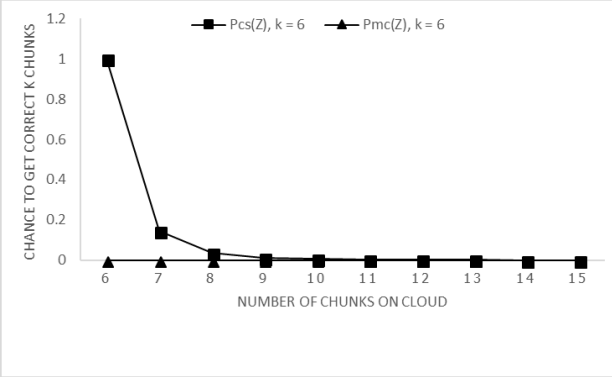

Fig.6 Comparing the probability of get correct k chunks of target file on client/server scheme and proposed scheme at $k = 6$

VI. CONCLUSIONS

This paper focused on protecting the privacy of big data file distribution via multi cloud storage. Data owner spilt his file into equal-size chunks and distributed them to multiple cloud storages. We study on the effect that can prevent unauthorized access to the target file, or even the attacker can get the fragments of files. Those stolen fragments are less likely to be the fragments of the target file. Moreover, the proposed scheme We found that cloud storage server store only 5 or 6 fragments or pieces of the file, this number can mislead an attacker. The attacker has fewer success opportunities to get the target or desired fragment on each cloud storage. The proposed scheme is secure and also avoid vendor lock-in

ACKNOWLEDGMENT

Tis work was supported by Sichuan Science and Technology Support Program under grant No.2016GZ0073.

REFERENCES

[1] M. Smiley, "Introduction to Big Data Analytics An Introduction to Big Data and Analytics," in *Encyclopedia of Cloud Computing*, 1st ed., Wiley-IEEE Press, 2016, pp. 744–756.

[2] S. Sagiroglu and D. Sinanc, "Big data: A review," *2013 Int. Conf. Collab. Technol. Syst.*, pp. 42–47, 2013.

[3] P. Mell and T. Grance, "The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology," *Natl. Inst. Stand. Technol. Inf. Technol. Lab.*, vol. 145, p. 7, 2011.

[4] M. Su, L. Zhang, Y. Wu, K. Chen, and K. Li, "Systematic Data Placement Optimization in Multi cloud Storage for Complex Requirements," *IEEE Trans. Comput.*, vol. 65, no. 6, pp. 1964–1977, 2016.

[5] H. Abu-Libdeh, L. Princehouse, and H. Weatherspoon, "RACS: a case for cloud storage diversity," *SoCC*, pp. 229–240, 2010.

[6] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," *Proc. 16th ACM Conf. Comput. Commun. Secur. - CCS '09*, vol. 489, p. 187, 2009.

[7] A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa, "DepSky:Dependable and Secure Storage in a Cloud-of-Clouds," *ACM Trans. Storage*, vol. 9, no. 4, pp. 1–33, 2013.

[8] M. Padhya and D. Jinwala, "A Novel Approach for Searchable CP-ABE with Hidden Ciphertext-Policy," *Inf. Syst. Secur.*, pp. 167–184, 2014.

[9] H. Su, Z. Zhu, L. Sun, and N. Pan, "Practical searchable CP-ABE in cloud storage," *2016 2nd IEEE Int. Conf. Comput. Commun. ICCC 2016 - Proc.*, pp. 180–185, 2017.

[10] Y. Tian *et al.*, "Secure fragment allocation in a distributed storage system with heterogeneous vulnerabilities," *Proc. - 6th IEEE Int. Conf. Networking, Archit. Storage, NAS 2011*, pp. 170–179, 2011.

[11] Y. Tian, X. Qin, and Y. Jia, "Secure Replica Allocation in Cloud Storage Systems with Heterogeneous Vulnerabilities," *2015 IEEE Int. Conf. Networking, Archit. Storage*, pp. 205–214, 2015.

[12] A. Mei, L. V. Mancini, and S. Jajodia, "Secure dynamic fragment and replica allocation in large-scale distributed file systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 14, no. 9, pp. 885–896, 2003.

[13] P. Li, S. Guo, T. Miyazaki, M. Xie, J. Hu, and W. Zhuang, "Privacy-Preserving Access to Big Data in the Cloud," *IEEE Cloud Comput.*, vol. 3, no. 5, pp. 34–42, 2016.

[14] H. K. Patil and R. Seshadri, "Big Data Security and Privacy Issues A Survey," *Int. Conf. Innov. Power Adv. Comput. Technol. [i-PACT2017]*, no. November, pp. 1–5, 2016.

[15] N. Kaaniche and M. Laurent, "Data security and privacy preservation in cloud storage environments based on cryptographic mechanisms," *Comput. Commun.*, vol. 111, pp. 120–141, 2017.