A Proposed New Framework for Securing Cloud Data on Multiple Infrastructures using Erasure Coding, Dispersal Technique and Encryption

Frimpong Twum
Kwame Nkrumah University of Science and Technology
Department of Computer Science
Kumasi - Ghana

J. B. Hayfron-Acquah
Kwame Nkrumah University of Science and Technology
Department of Computer Science
Kumasi - Ghana

J. K. Panford
Kwame Nkrumah University of Science and Technology
Department of Computer Science
Kumasi - Ghana

ABSTRACT

Cloud computing is a technology that has come to save organizations from investing in and owning high cost IT infrastructure including its management and maintenance. The technology enables an organization to outsource its IT needs to the care of a remote third party Cloud Service Provider (CSP) while focusing on its core business processes. It enables the usage of IT resources remotely as a service on subscription basis at a per usage fee on demand. The service models available are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). These service models are deployed in one of four cloud deployment models as Public, Private, Community or Hybrid cloud. Despite the technology's numerous benefits, it also poses serious security threats to vital business data assets as the subscriber has to surrender control over its management and maintenance to a remote CSP. The threats include: the CSP using the data for their own gains, the location of the data not known to the subscriber, the ownership of the data (for example, on contract termination or in the event of conflict or dispute), and also the subscriber not knowing who has unauthorized access to their data resource. The challenge therefore, is how to create a secure and vigorous data security solution that can mitigate these threats and alleviate the cloud subscriber fear to freely enjoy using cloud computing services. Hence, this study proposes a Six-level Cloud Data Distribution Intermediary (CDDI) Framework that enables the cloud subscriber to effectively secure its data against these threats. The framework employs Erasure Coding (based on the Galois Field Theory and Reed Solomon Coding), and a Data Dispersion technique with a Transposition Encryption technique based on Rubik's cube transformation. In addition, it also uses this study's proposed Erasure Coding technique based on checksum dubbed "Checksum Data Recovery" (CDR). The CDDI framework when implemented on the cloud subscriber's gateway system will encrypt and split the subscriber's data into chunks of data fragments which are distributed randomly to the subscribers selected multiple CSP storage infrastructures. This alleviates threats of data usage, location, ownership, and access, identified.

General Terms

Cloud Computing, Erasure Coding, Encryption, Decryption, Cipher, Security, Privacy, File Split Architecture, Cloud Computing Framework

Keywords

Cloud Computing, Erasure Coding, Reed Solomon Coding, Galois Field Theory, Checksum, Data Dispersal Technique, Encryption, Decryption, Cloud Computing Framework

1. INTRODUCTION

Cloud computing technology is an invention in the ever changing computing technology that has come to save organizations from setting up, owning, and maintaining high cost computing equipment and other ICT infrastructure. Benefits includes cost savings (in terms of hardware, software, personnel, etc.), ability to access resources from anywhere at any-time provided there is an Internet enabled device and connectivity to the Internet, and paying per usage among others. Cloud computing gives users huge storage capacity via storage facilities hosted on the Internet that are usually owned and managed by third party Cloud Service Providers (CSP's). These storage facilities usually are publicly accessible referred to as Public Cloud, or may be configured for an individual subscriber's private use referred to as Private Cloud, or configured explicitly for a group of organizations usage referred to as Community Cloud, or may be a composite of two or more of the specific cloud deployment models referred to as Hybrid Cloud. The CSP has access and control over the data whether encrypted or un-encrypted as the responsibility for the data maintenance (such as data backups and data restore) is usually mandated to them. Although the cloud tenant outsourcing its IT functions enables them to focus on their core business processes, they also put their vital data resources at risk in the hands of the third party provider who may use it for their own gains. For example, selling the data to a competitor, or using it for other purposes other than has been agreed. Cloud computing at the onset came with security challenges as a result of its resource pooling and multi-tenancy characteristics where multiple customers share the same resources, same application, same databases or in some cases same tables (Youssef and Alageel, 2012; Khatri et. al, 2013). As an example, a cloud provider computing resources may be pooled to serve multiple subscribers and this may put data at risk of getting into unauthorized hands through accidental or intentional disclosure. Thus, the CSP may accidentally or deliberately leak data or other vital resources to a competitor as they serve multiple subscribers (Khan and Yasiri, 2016; Shapland, 2017). A study by Trigueros-Preciado et. al.,(2013) found cloud computing security to be of a supreme concern to subscribers and this discovery in 2017 remains unchanged as confirmed by Ahmed (2017) study. The Treacherous 12 (2017) survey identifies data security breaches such as: The two Yahoo! data breaches reported in September and December 2016 (affecting 3 billion user accounts, leading to a drop of $350 million in the acquisition price of Yahoo! which was earlier valued at $4.8 billion) (McMillan and Knutson, 2017), Data loss such as malicious CSPs or malicious users intentionally corrupting the user's data inside the cloud by modifying or deleting

(Chauhan, 2015; Sailaja and Usharani, 2017), Malicious insiders such as the theft of 1.5 million T-Mobile customers' data by an employee at their Czech offices (Wei, 2016), Denial-of-Service (DoS) attacks such as the Australian Bureau of Statistics denial of service (ABS, 2016) as concerns of cloud computing security. Another issue that arises from the use of Cloud Storage as a Service is the use of customer data for marketing and personal profiting such as leaking it to competitors (Chauhan, 2015). Ahmed (2017) study established cloud computing poses security threats to the subscriber in terms of: (1) Who has access to the data/resource (accessibility), (2) What other use is the data/resource been used for (usage), (3) Where the data/resource is located (location), (4) Who has ownership over the data/resources outsourced to the cloud (ownership), (5) How accuracy of the data outsourced for cloud storage can be ensured (accuracy). These threats raise questions as follows: (1) How can the cloud subscriber prevent unauthorized access to their data? (2) In what ways can a cloud subscriber prevents their data from been used for other purposes by the CSP? (3) In what ways can the cloud subscriber ensure that their outsourced data is not vulnerable as a result of the data location since different countries have different data privacy laws? (4) In what ways can the cloud subscriber ensure that they have sole ownership of their data outsourced for cloud storage? (5) In what ways can the integrity of data outsourced for cloud storage be maintained?

In relation to ownership there is the risk in terms of what happens to the data on contract termination or in the event of conflict between the cloud subscriber and the cloud provider. For example, when a CSP refuses to grant a subscriber access to their data in the event of a dispute over say the subscriber’s subscription payments. With the issue of location, accessibility, and usage of the data resource, cloud computing distributes data across servers setup and managed by CSPs across the globe and this makes it difficult for the cloud subscriber to find in which country(s) their data is been stored, who has access to the data, and for what unauthorized use (Rao and Selvamani, 2015). Finally data outsourced for cloud storage can be altered in transmission by man-in-the-middle (MITM) attack or modified inside cloud provider’s storage facilities by a malicious insider attack (Sailaja and Usharani, 2017). These issues are making it unattractive for organizations and individuals to subscribe to cloud services. Although traditional counter security measures such as using encryption techniques (for confidentiality), using hash functions (for integrity), and using firewall, anti-virus, intrusion detection and prevention systems (for availability) have been employed, they have been inadequate to securely protect vital organization data against attacks. Malicious attackers have found ways of going round them to compromised vital business data asset using network security attacks as Dos/DDoS, U2R attack, R2U attack, Probing attack, MITM attack, Message replay attack, and Brute-Force analysis attack (Khandelwal, 2017). According to Wang (2009), cloud computing technology distributes data on multiple servers belonging to a single CSP but the challenge as noted by Ahmed (2017) is implementing a distributed protocol architecture that assures of a robust secured cloud data security in a defense-in-depth design.

Aim of Research - This research therefore seeks to propose a cloud data security solution framework whereby data outsourced for cloud storage is first sliced into chunks of data fragments and then encrypted on the subscriber’s gateway system before being distributed to multiple different CSP’s storage nodes (storage servers).

Specific Research Objectives:

- To enhance security of data outsourced for cloud storage by ensuring the data is useful to only the data owner
- To propose a cloud data security solution framework that alleviates the cloud subscriber’s fear of their data/resource accessibility, usage, location ownership, and accuracy.

2. LITERATURE REVIEW

OpenCirrus (2017) identifies four major challenges of cloud computing as follows: Data Security and Privacy, Data Ownership, Lack of Standardization, and Lack of resources and expertise. Out of the four, ensuring data security and privacy is noted as the biggest challenge today. This challenge was attributed to the fact that some CSP’s may behave un-ethically by making money through using personal information of subscribers entrusted with them for advertisements and other purposes for which the data owner’s permission has not been sought. Or the CSP may use the information to learn more about their subscribers for their own interest. In addition given that personal information may be transferred by a CSP to another third party organization (say a data center) probably located in another country un-knowingly to the cloud subscriber, it is paramount to ensure that the information transferred is useful only to authorized persons. There is however the risk of the information falling into the hands of un-authorized persons or risks of the information being kept by the CSP or its allied partners for other purposes even when an agreement has been ended or annulled (Sailaja and Usharani, 2017). Data ownership is seen as another major challenge of cloud computing. Different countries have privacy and security laws, acts, and regulations that govern the protection of data, for example, the Asia Pacific Economic Cooperation (APEC) privacy framework, the Organization for Economic Corporation and Development (OECD) privacy framework, the European Economic Area (EEA) data protection laws, and etc. Each of these laws according to a CSA (2011) report places the burden of ensuring the protection and security of personal data on the custodian of the data. In cloud computing the data custodian is the cloud provider and most cloud contracts have clauses that make the custodian of the data the owner (OpenCirrus, 2017). This security challenge is a concern to subscribers and hence preventing widespread adoption of cloud computing. Cloud computing technology presents new challenges to securing data and other resources usage than traditional IT hosting service. Cloud computing characteristics of multi-tenancy, resource pooling, rapid elasticity, on-demand self-service, and broad network access, require new data security approach. Although cloud computing comes with significant concerns about security, privacy, data integrity, intellectual properties, research suggest that cloud based service models provides better security to clients data and other resources than traditional IT models (OPC, 2011). This though is not as a result of use of superior counter security measures but because privacy and security laws as well as government acts and regulations compels cloud providers to put in place privacy protection and also use security mechanisms to secure subscribers data. Cloud providers therefore ought to implement security mechanisms in overlapping layers to prevent, detect, and respond to unauthorised intrusion or unauthorised usage of resources to enhance subscriber confidence in cloud services. In the same manner, it is also vital cloud subscribers knows the security measures employed by their chosen CSP in combating the Confidentiality Intergrity Availability (CIA) security traits (Shah and Anandane, 2013). As the cloud subscriber/tenant usually have no physical control over cloud infrastructure in most cloud setup, contract agreements, service level agreements (SLA's) and providers documentation become vital in managing risks than in a traditional enterprise owned hosting environment (Hussain et. al., 2017).

CSP's LOCATION as a data security issue: With respect to cloud data security, it is extremely important the cloud tenant (client subscriber or client provider e.g. a SaaS cloud provider hosting its application on IaaS cloud providers infrastructure) knows where their data resource outsourced for cloud storage is located (Mahmood, 2011; Raisian and Yahaya, 2015). By getting to know the location, a subscriber for example may be able to check the privacy and security laws, acts, and regulations that governs the protection of data in that country and know the extent of their enforcement (Sailaja and Usharani, 2017).

ACCESS as a data security issue: In addition to knowing the location of data in the cloud, it is important that the cloud subscriber knows who has access to the data and how the data is accessed in other to be assured of the data security (Rao and Selvamani, 2015; Ahmed, 2017).

OWNERSHIP as data security issues: After knowing the location of the data and also who has access, cloud subscribers must take keen interest in determining ownership of data outsourced for cloud storage. According to Gray (2014), ownership of cloud data depends on where the data was created. Thus, whether the data is created on the cloud provider's infrastructure or created on the cloud subscriber's system before upload. Service providers usually prevent access to their clients' data. For example, LinkedIn does not permit other services to access all of its user personal data such as the email address through their API. Also, Facebook's end-user-agreement states that the company stores data for as long as it is necessary and not as long as users want to keep their data. This implies users lose control and ownership of their data (FileCloud, 2016). In summary, with cloud computing, data is distributes across servers setup and managed by CSPs across the globe and hence are subjected to different privacy and security laws, acts, and regulations. This distribution of servers across many countries makes it difficult for the cloud subscriber to find out the location of their data, determine who has access to their data, find out what un-authorized usage is their data been used for, determine ownership of outsourced data on the cloud, and ensure data accuracy.

Some current existing solutions for ensuring cloud data security:

Rao and Selvamani (2015) propose that encrypting the data using the RSA cryptographic algorithm is the best solution to secure cloud resources. This claim is arguable as research shows encrypted files can be decrypted using a brute-force analysis attack. Also, O'Reilly (2017) note that hard drive based encryption are not safe and hence cloud subscribers ought to be mindful of how their CSP encrypt their data. The CSP encrypting data on their server using software is much secure and recommended than using a drive-based encryption where the provider installs hard drives that automatically encrypt the cloud subscriber's data. Security researchers in 2015 for example uncovered flaws in a particular hard drive product line that enabled viewing encrypted data (O'Reilly, 2017)

Another solution employed by CSP's has been to split data into pieces, encrypt the pieces and then distribute them to their distributed servers e.g. Google File System – GFS (Jain, 2013; Roshoan, 2014; Carson, 2016; Strickland, 2017; Techopedia, 2017), Apache Hadoop Distributed File System – HDFS (Natarajan, 2012; Hadoop, 2013; DeZyre, 2016), Backblaze B2 (Backblaze, 2015a; Backblaze, 2015b, Backblaze, 2017). Though this approach secures the data to some extent it did not protect data from been decrypted, deleted, or altered by a malicious insider. Hence a client-side encryption approach that gives access, management, and control of the encryption keys only to the cloud subscriber was proposed to prevent data breaches caused by a malicious insider (Shah and Anandane, 2013; Chou, 2013). Client-Side encryption encrypts data at the subscriber's premises before the data is sent to the CSP. This solution although gives subscribers some level of assurance of their data security, O'Reilly, (2017) noted that encrypting data slices and sending them to a single CSP's storage facilities still poses a threat as the data slices can be re-assembled and decrypted, deleted, or modified by the CSP. Most major CSP's including Google, Dropbox, BackBlaze B2, Box, and etc. however uses this approach of splitting, encrypting and distributing data slices on their own storage facilities. This study therefore proposes a solution that distributes slices of encrypted data to multiple CSP's storage nodes thereby preventing a single CSP from having access to all of the data pieces.

3. CONCEPTUAL FRAMEWORK OF THE PROPOSED NEW CLOUD SECURITY MODEL

The proposed cloud security model is designed to use encryption, hashing, and erasure coding technique based on Reed Solomon Coding and the Galois Field Theory. The model: apply erasure coding technique to first sliced data objects outsourced for cloud storage into chunks of data fragments, apply an encryption algorithm to encrypt the chunks of data fragment, apply data dispersion technique to shuffle the encrypted data fragments and distribute to multiple CSP's storage nodes, To ensure efficiency especially during data retrieval as different CSP's storage nodes host the data fragments and hence may be operating at different data rates, a buffering technique is used to buffer the data fragments from the fast storage nodes as a waiting mechanism until the data fragments from the delayed storage nodes are received and assembled for onward delivery to the subscriber.

By employing above measures, this study hope to address the cloud security issues identified and assure the cloud subscriber of the security of their data as the encrypted data fragments will be of no value to a CSP.

Finally the study foresees performance to be likely affected as security is strengthened and hence caters for performance by employing the use of Metadata server to keep track of the data fragments and where they are distributed so as to ensure accuracy of the cloud subscriber data resources

In effect, the study hope the proposed model will assure cloud subscribers of the confidentiality, integrity and availability of their data resources outsourced for cloud storage.

4. METHODOLOGY

In this section the methodology for the proposed new Cloud Data Distribution Intermediary (CDDI) framework is presented. The study employs the design research methodology which enables the development and delivery of new solutions that help to understand human needs and meet them thereby improving livelihoods (Lee, 2012). The CDDI framework uses erasure coding, data dispersal, and encryption to secure files. The framework has a client side (CDDI Client) which is implemented on the cloud subscriber's gateway system to encrypt and split the subscriber data into chunks of data fragments and distribute them randomly to the subscriber selected multiple CSP storage infrastructures. There is also a server side (CDDI Metadata Server) that holds metadata information for all the files the subscriber uploads to the cloud. The CDDI framework seek to secure subscriber data from threats such as: the CSP using subscriber data, the subscriber not knowing where (which countries) their data is located, the CSP claiming ownership of the subscriber data, and the subscriber not knowing who has unauthorized access to their data.

4.1 The Proposed Architecture

Fig. 1 is the overall architecture of the proposed solution for securing data outsourced for cloud storage.

Subscriber → Cloud Data Distribution Intermediary (CDDI) → Cloud Storage Provider

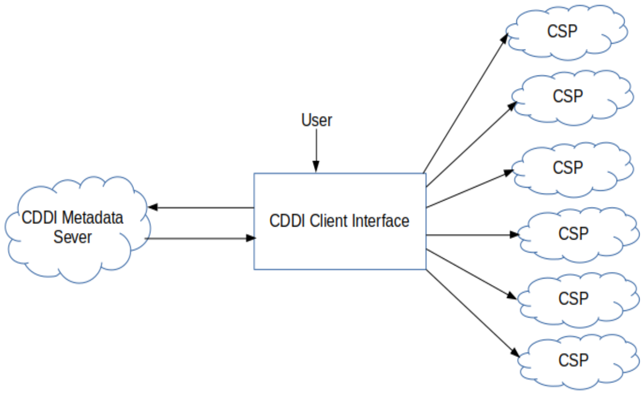

Fig. 1 - Architecture for the Proposed CDDI Framework for Cloud Data Storage

4.2 The CDDI

The study proposes a new indirect model of interaction between the cloud subscriber and the CSP. It is proposed that a software framework intermediary is introduced into the data transfer transaction to inject a high degree of security. The intermediary would be responsible for ensuring that the subscriber data is protected at various levels from the diverse cloud security issues outlined earlier in sections 1 and 2. Based on the manner in which the intermediary operates, the study refers to the intermediary as a ***Cloud Data Distribution Intermediary (CDDI)***. The operation of the CDDI involves the following processes: Receive data from the user for storage in the cloud, Obfuscate the name of the file to hide its purpose from malicious persons snooping on the network and hackers who may have gained access to the file information in the subscriber cloud storage account, Encrypt the subscriber data to hide its content from unauthorized persons who may obtain it, Distribute the encrypted content of the file in unique pieces to a number of CSP's to prevent the problem of one CSP having access to the entire data, and Save metadata on each file uploaded in order to retrieve the file when required by the cloud subscriber.

4.2.1 Components of the CDDI Framework

The CDDI framework comprises of a number of modules as outlined (refer to the conceptual framework**)** with each performing a tasks that helps to secure the file being stored on the cloud.

4.2.1.1 File Name Obfuscation Module

The file name is hashed using a hashing algorithm as the first layer of system security, to obscure the identity of the file being uploaded. This step makes it difficult for people or software that are sniffing on the network from discovering the true purpose of the file while it is in transit. Similarly, any intruder to the subscriber cloud account would likewise be confounded by the irregular file name pattern.

4.2.1.2 Data Obfuscation Module

As a second layer of security, the contents of the file are transposed using the encryption function of this study's proposed transposition cipher algorithm which is based on the rotations of the Rubik's cube to generate the cipher text (Twum et.al., 2019).

4.2.1.3 Data Distribution Module

The greatest strength of the proposed CDDI framework comes from its use of resilient techniques to distribute the contents of the subscriber's file to multiple CSP's storage nodes. By so doing, the CDDI is able to mitigate the issue of data ownership on the cloud, as no one CSP has sufficient data to rebuild the file and therefore any claim of data ownership on the part of a CSP is rendered null by their inability to make any use of the portions of the file in their custody. The data distribution module comprises of two sub-modules, namely:

File Splitting and Erasure Protection Module (FSEPM): This module is responsible for breaking the encrypted file into a pre-determined number of pieces or shards for subsequent upload to the cloud. To guard against data loss, data corruption and as well as Cloud Storage Service down-time, this module makes use of two very resilient techniques to ensure that in most cases, the full file is available to the subscriber when the file is requested. The techniques that the CDDI makes use of, are **Reed-Solomon encoding** (Twum et. al, 2016a; Twum et. al, 2016b), **Reed-Solomon decoding** (Twum et. al, 2017) and also this study's newly developed **Checksum Data Recovery (CDR) technique**.

Shards Dispersal Module (SDM): This module is responsible for ensuring that there is no an observable pattern in how the file shards are sent to the cloud by scrambling the original order of the shards.

4.2.1.4 Checksum Data Recovery (CDR)

The study developed a data recovery technique based on checksum by making use of the unique property of the bitwise XOR operator. The technique uses a thorough computation of checksums on sections of the file that is being uploaded. The checksum data can then be used later to recover deleted portions of the file as well as detect and correct errors in the file.

Overall architecture of the proposed new CDR technique

The CDR has 3 main modules: Data, Compute Parities, and LocateError. The CDR technique divides data into several modules (data shards) (Fig. 2), and parity information (checksum) for each module is computed and stored (Fig. 3). Fig. 4 presents the flow diagram for the CDR technique whiles fig. 5 depicts the activity diagram.

Definition of Terms: Data and Module - A given file of size 'X' (which can be in KB, MB, GB, TB etc.) is computed into a 3-dimensional array (3D) and each entry of the 3D array is a 2-dimensional array (2D) of size 4x4 matrix called a module or data shard (Fig. 2). Thus, a module is a 4x4 matrix that has a byte of the data in each entry. This implies a module has a size of 16 bytes. Therefore to obtain a module, the file of size 'X' is first converted into bytes of data (say 'Y' bytes) and then divided by 16. As the file size may not exactly be a perfect multiple of 16 byte, this may result with Y/16 modules remaining Y mod 16 bytes of the data. The remainder (i.e. the Y mod 16) is padded with 0 byte(s) to make up a module (16 bytes). Each module within the data has its own metadata (Row and Column Parity information) which is independent on other modules (Fig. 3). Therefore a corrupted module does not depended on other modules for recovery. Thus, the metadata for each module are independent, implying that a corrupted metadata does not affect other module metadata. The above data representation depicted by fig. 2 shows that data can be grouped into modules up to 'n'.

Row Parities are the parities of each row of a module. It is computed by performing the XOR of a value in a row with the values succeeding it in the same row as shown by fig. 3.

Column Parities are the parities of each column in a module. It is computed by performing the XOR of a value with the values succeeding it in the same column as shown by fig. 3.

Shard dispersal module: To improve security and avoid prediction of the destination of the shards, the derived shards after splitting are scrambled before they are forwarded to the various cloud storages. In other words, the transmission of the shards is obscured so that the shards will not be uploaded orderly but rather shuffled among the users chosen CSP's storage infrastructures. This is achieved by following the underlying algorithm.

Shard dispersal algorithm

- Create an array list containing integer numbers that correspond to the number of the derived shards after file splitting.
- Shuffle the integer numbers in the new list.
- Append these numbers from the shuffled list to the name of the file and use the new name to get the individual shards to be ready for upload.

3.3 Metadata Module

The proposed new CDDI framework makes use of metadata to save data concerning each file uploaded by a cloud subscriber. Two different types of metadata are used. One keeps record of the framework user uploaded files (***user metadata***), and the other keeps track of the uploaded shards to the multiple CSP's (***file metadata***). The user metadata has the names of all files that a user has uploaded using the CDDI framework. It also has the user hash value which is used for encrypting each file the user intends to encrypt and upload. With the user metadata, the list of files that a user has uploaded can be retrieved and rendered in a view to the user. It thus relieves the user the burden of keeping track of uploaded files. The other metadata, dubbed 'file metadata', contains data relating to each of the files. It stores details for each data shard belonging to a file. From the file metadata, a shard's position in the sequence of the shard chunks can be determined. The destination cloud account is also saved in the file metadata. Again, it has other details such as date of upload, and the number of columns which is essential for the Reed-Solomon algorithm.

5. IMPLEMNTATION

The study proposes a new Six-level Cloud Data Distribution Intermediary (CDDI) Framework that addresses the study objectives as shown in fig. 6.

6. DISCURSIONS, FINDINGS AND CONCLUSIONS

This section discusses the study findings and presents the conclusions.

6.1 Discussions

Below is a summary of how the CDDI framework compares with existing frameworks in terms of architecture, ensuring confidentiality, providing integrity, and controlling access to data and other resources outsourced for cloud storage.

6.1.1 Architecture

The Google File System (GFS) and Apache Hadoop are distributed file storage systems whereas Backblaze offers cloud storage and backup service. The CDDI framework is a distributed cloud backup service. GFS is Google's proprietary distributed file storage system upon which the Google Drive cloud storage is built. GFS and Apache Hadoop make use of clusters of commodity machines for data storage and computations (Hadoop 2013; Roshan 2014). Backblaze utilizes a single data centre to hold all of the backed-up data (Backblaze, 2017). However, the proposed CDDI framework makes use of multiple existing cloud storage service providers such as Google Drive, Dropbox and Box, to store shards of a single file.

6.1.2 Confidentiality

Google File System and Apache Hadoop are designed to support constant data access by applications that perform computations with the stored data. As such the recommendation is for the data to be plain or raw (Hadoop, 2013; Roshan, 2014). Any external party that accesses the data is able to get the data in its plain format. While this is not a strict requirement, it is the recommended design. Confidentiality is traded off for computational ease in these systems. Backblaze uses a combination of Advanced Encryption Standard (AES) and Secure Socket Layer (SSL) to secure the data that is transmitted to and saved on their servers, thus ensuring confidentiality (Backblaze, 2017). The proposed CDDI framework uses a custom-made encryption algorithm based on the motions of a Rubik's Cube to obfuscate the data (Twum et. al., 2019). In addition, the filename is also hashed to obscure the purpose of the file. The combination of the data encryption and filename hashing greatly enhances the confidentiality of the system.

6.1.3 Access Control

Google File System, Apache Hadoop and Backblaze all store data shards in a location where the service provider has access to all the pieces of the file data. As such, while it may be possible that the data owner may be unaware of the storage location of their file shards, the service providers have all that information available to them (Chima, 2016). On the other hand, the proposed CDDI framework distributes the data shards randomly to multiple service providers without storing the credentials required to access them. This means that no one service provider knows the location of all the file shards, ensuring that only the data owner has full access to the data. To access file shards on Apache Hadoop and Backblaze, the data owner must supply a single set of login credentials. The service providers require no login credentials to access the files saved on their servers and since all the file shards are hosted on their storage servers, they have full access to the files that are uploaded to their servers. In contrast, with the proposed CDDI framework a separate set of login credentials is required for each of the cloud storage service providers that the user subscribes to. Any user or service provider that wishes to access files when the CDDI is implemented will need to know all of the login credentials. This greatly increases the confidentiality factor of the proposed CDDI framework.

6.1.4 *Integrity*

The GFS and HDFS file systems are designed for appending to files but not altering the file contents. This helps to prevent altering the content of the files which are saved on the file system, thus securing the integrity of the data. On the other hand, Backblaze uses Reed-Solomon erasure coding via Vandermonde matrix to guard against data loss. While this algorithm protects against data loss when a storage cabinet goes offline, it does nothing to prevent alteration of the data in a shard. As long as the shard is present, it is included in the downloaded file. This means that the data in the shard can be altered without detection by the data owner. This observation was noted through compiling and running the Backblaze open source Reed Solomon Erasure Coding Source Code (Backblaze, 2015b; Twum et. al, 2016a; Twum et. al, 2016b, Twum et. al, 2017). The proposed CDDI framework employs Reed-Solomon coding, and the proposed new CDR method that are implemented at the client side for error detection and correction. The CDDI client ensures that any alterations to the data can be detected and corrected.

6.1.5 *Ownership*

The proposed CDDI framework ensures that only the owner of the data (cloud subscriber) has sole ownership of their data resource stored on the cloud. This is in contrast to existing architectures implemented by storage CSP's such as Google Drive, Dropbox, or Box where ownership of the data becomes a contentious issue but in most cases the CSP claim ownership (Gray, 2014; FileCloud, 2016).

6.2 Findings

The finding from the study on how the proposed CDDI framework addresses issues of cloud data confidentiality, integrity, ownership, availability and authentication is presented by Tables 1. In addition Table 2 presents how the CDDI framework addresses other cloud security issues as Multi-Tenancy, Data Loss, Data Location and other computer network attacks such as Dos/DDos, Malicious Insider, Malware Injection, Man-in- the-middle (MITM), Message Replay and U2R and R2U.

6.3 Conclusions

This study's main purpose was to address the security challenges in relation to outsourcing data and other resources for third party CSP's storage particularly in terms of preventing the CSP from making use of the data. The study purpose has been achieved as the proposed CDDI framework is able to assure of the confidentiality, integrity, and as well able to effectively control and manage who has access to the data and can make use of it. The CDDI framework addresses the study objectives as follows:

6.3.1 *How can cloud data be secured to prevent unauthorized access?*

The CDDI framework when implemented distributed shards to multiple CSPs. The slices of data which each CSP receives from the data dispersal technique of the file uploading process are both incomplete and encrypted. This means that the CSP does not have access to the subscriber's full data. Only the cloud subscriber has access to the full data through applying the CDDI framework file download process.

6.3.2 *In what ways can a cloud subscriber prevent their data from being used for other purposes by the CSP?*

The CSP's are unable to make use of the data entrusted in their care as they receive incomplete and encrypted slices of the data when the CDDI framework is employed. They can only store the data but cannot use it for any other purpose.

6.3.3 *In what ways can the cloud subscriber ensure that their outsourced data is not vulnerable as a result of the data location since different countries have different data privacy laws?*

Unlike the current cloud data storage where the subscriber's data is vulnerable as it resides with only one provider (Chima, 2016), the CDDI framework randomly disperses slices of the resource to multiple CSPs. This prevents the CSPs from having access to all the files as well as guessing the locations of the slices which they do not have. Hence in countries where data privacy laws are liberal or not strictly enforced, the CSP is still unable to make use of the portions of the data entrusted with them.

6.3.4 *In what ways can the cloud subscriber ensure that they have sole ownership of their data outsourced for cloud storage?*

The scrambling and data dispersal features of the CDDI framework enforce single-ownership of the data. Thus: the framework ensures that the file is never whole and useful anywhere except on the data owner's computer.

6.3.5 *How can we ensure data outsourced for cloud storage is useful only to the data owner?*

In computer security, the CIA trade notes three security dimensions as Confidentiality, Integrity and Availability. The proposed CDDI framework uses a combination of hashing, encryption, scrambling, erasure coding and data dispersal to address these security dimensions and also addresses cloud subscribers concerns of data Ownership, data Usage, data Location, and other security issues that poses threats to data outsourced for cloud storage.

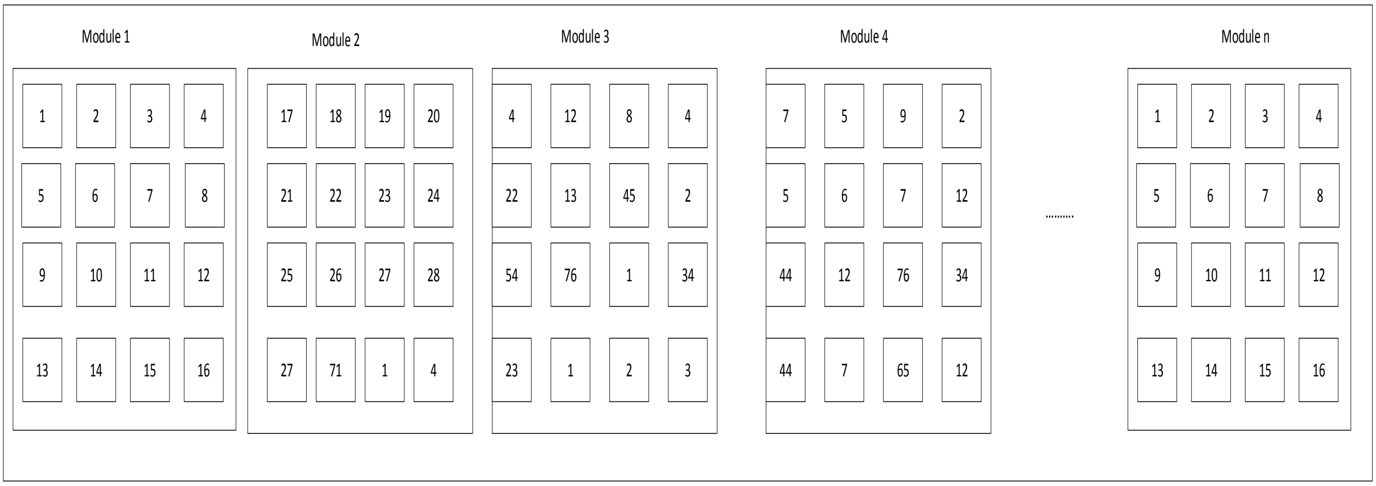

Fig. 2 - Modular representation of dat

Module Row Parities

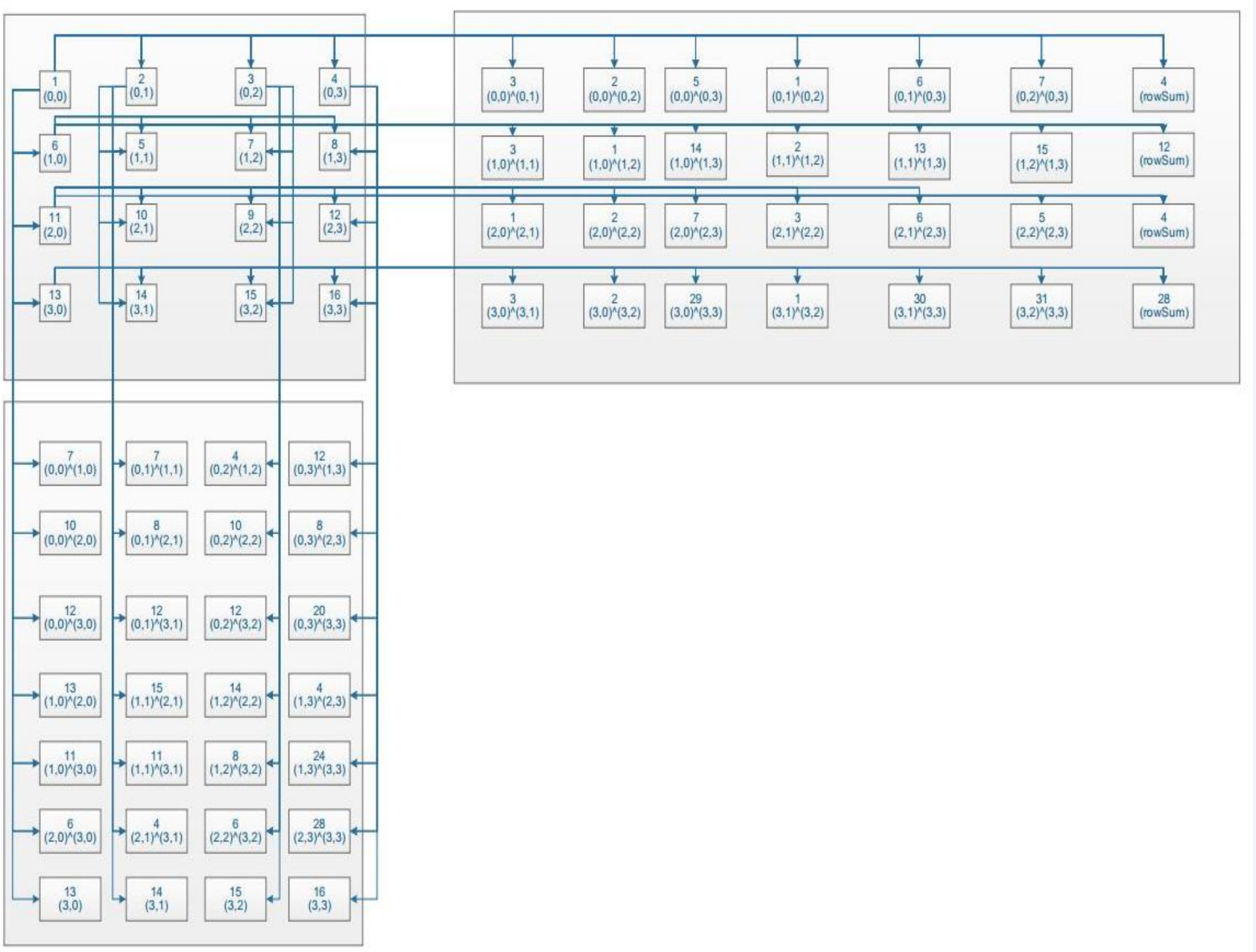

Fig. 3 - Module diagram of the proposed CDR technique

Column Parities

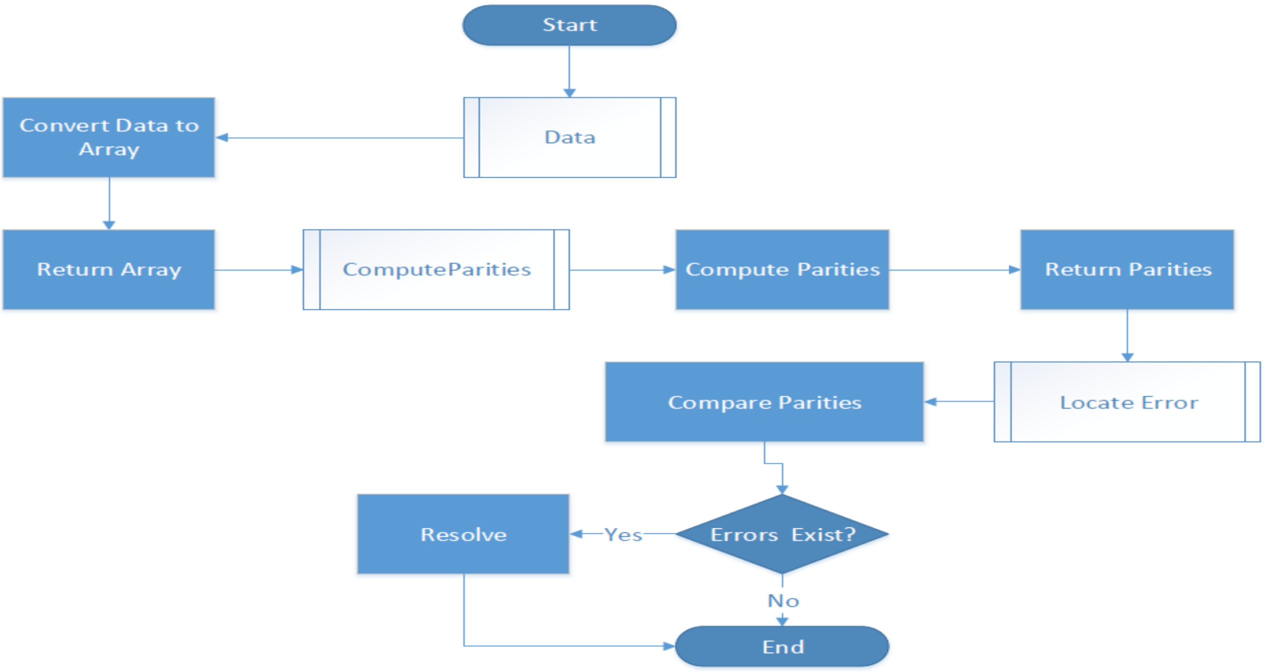

Fig. 4 - Flow diagram for the CDR technique

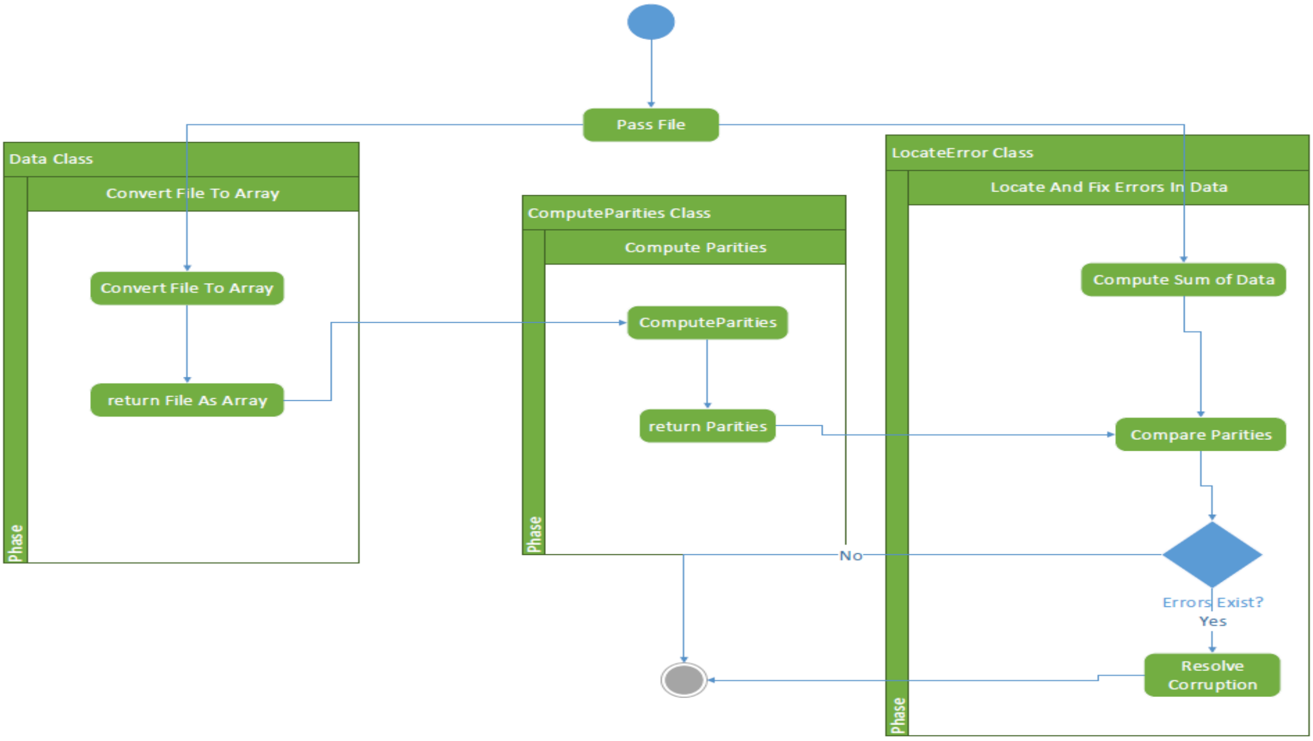

Fig. 5 - Activity diagram of the CDR technique

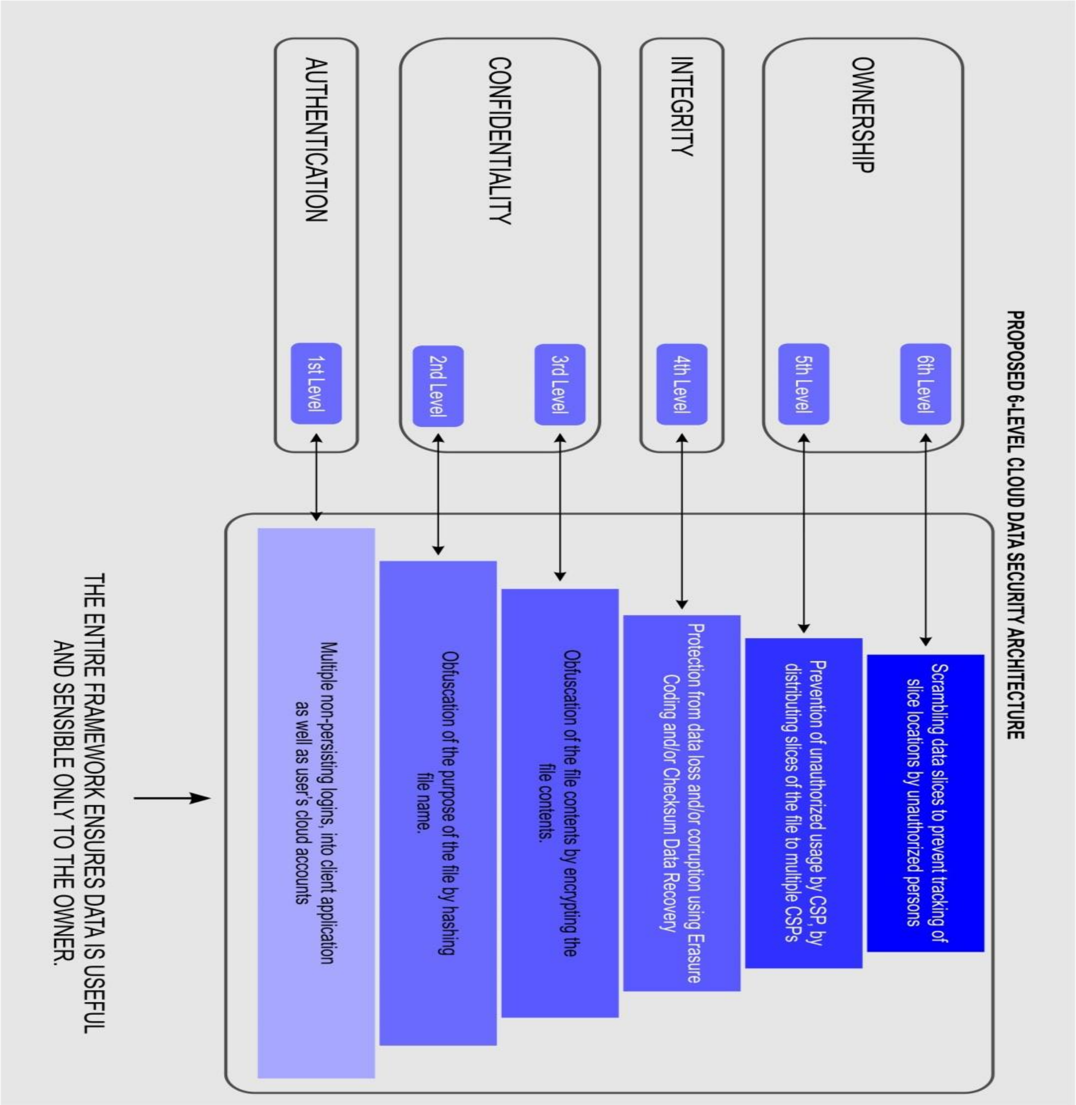
PROPOSED 6-LEVEL CLOUD DATA SECURITY ARCHITECTURE
OWNERSHIP
6th Level
5th Level
INTEGRITY
4th Level
CONFIDENTIALITY
3rd Level
2nd Level
AUTHENTICATION
1st Level
Scrambling data slices to prevent tracking of slice locations by unauthorized persons
Prevention of unauthorized usage by CSP, by distributing slices of the file to multiple CSPs
Protection from data loss and/or corruption using Erasure Coding and/or Checksum Data Recovery
Obfuscation of the file contents by encrypting the file contents.
Obfuscation of the purpose of the file by hashing file name.
Multiple non-persisting logins, into client application as well as user's cloud accounts
THE ENTIRE FRAMEWORK ENSURES DATA IS USEFUL AND SENSIBLE ONLY TO THE OWNER.

Fig. 6 – The Proposed Six-level Cloud Data Distribution Intermediary (CDDI) Framework

Table 1 - How the proposed CDDI framework address issues of Confidentiality, Integrity, Ownership, Availability and Authentication

Confidentiality	The proposed CDDI framework provides confidentiality of the subscriber's data at the second level (through obfuscation of the purpose of the file by hashing the file name) and third level (through obfuscation of the file content by encrypting the file content)
Integrity	The proposed CDDI framework provides protection from data loss or corruption using the Reed Solomon Coding or the CDR depending on the subscribers file priority level selected. Data integrity is achieved at the fourth level of the CDDI framework
Ownership and Availability	The CDDI ensures that the cloud subscriber has sole ownership of their data outsourced for cloud storage at levels five and six of the framework. In addition, the same levels of the framework together with the use of the metadata ensure that unavailability of a CSP that may be as a result of a DoS/DDoS attack does not prevent the subscriber from having access to their data. Thereby assuring subscribers the availability of their data.
Authentication	The CDDI framework uses multiple non-persistent logins to access the client interface as well as the subscriber's cloud accounts. The use of non-persistent logins means that even if a malicious person gains knowledge of one of the credentials, the person's ignorance of the remaining credentials will serve as a check to prevent access to the subscriber's data. Level one of the framework ensures this security feature.

Table 2 - How the CDDI framework addresses other cloud security issues

Multi-Tenancy	The CDDI framework addresses the issue of multi-tenancy threat by the use of the different CSP's storage facilities. The framework by breaking and distributing shards to multiple CSP's means that, the cloud security challenge of multi-tenancy which poses threat of a CSP maliciously leaking subscriber's data to a competitor deliberately or accidentally is eliminated. The CSP has no access to the subscriber's complete data as only a portion of the data is stored with them.
Data Loss	The CDDI framework prevents Data Loss (Erasure) through using the parity information stored on the metadata server and the use of the File Decoding Process via Reed Solomon Decoding method or the CDR method, depending on the user's choice of a priority level during the File Upload. The proposed framework compared to the existing cloud file architectures (google GFS, Apache Hadoop, and Backblaze B2) can recover the most data. Also the CDDI framework unlike the Backblaze B2 system is able to detect if an attacker alters the content of a shard and maintains the shard size.
Data Location	The current direct or indirect cloud architectures gives the provider access to the subscribers data as they know the locations of their storage facilities and can have access to them to retrieve the data even if encrypted (Chima, 2016). The CDDI framework distributing shards to multiple CSP's (with storage facilities located in different countries) prevents a single CSP from knowing the location of the subscriber's data and thereby addressing the issue of different privacy laws of different countries.
DoS/DDoS Attack	The CDDI framework distributing the data to different multiple CSP's storage facilities means that no single CSP has the subscriber's complete data. Hence a DoS/DDoS attack on one or more CSP's does not prevent the subscriber from accessing their outsourced data. The parity information stored on the CDDI metadata server together with the File downloading process can be used to recover the data even if several CSP's are attacked.
Malicious Insider Attack	By splitting subscribers data into shards and distributing to different multiple CSP's storage infrastructures, the CDDI framework protects subscribers data against an insider attack as an employee of a CSP only have access to scrambled portions of the subscribers data. The CDDI framework ensures that only the rightful owner of the data can make use of the data.
Malware Injection Attack	The CDDI framework addresses the threat of cloud malware injection attack where the attacker plant an evil virtual cloud machine in a CSP's cloud environment with the goal of intercepting subscribers data and taking full control. The framework using its metadata information about location of shards stored on the metadata server and the File downloading process can track and restore corrupted shards that may have been altered by the Malware Injection attacker. Also as the data received by the attacker is incomplete the attacker cannot make use of the data. In the event of this attack occurring, the CDDI framework treats the data sent to the evil cloud virtual machine as lost and recover using the metadata and either the Reed Solomon Decoding method or the CDR method depending on the priority selected for the upload.
MITM Attack and Message Replay Attack	The CDDI framework distributing the split shards to different CSP's infrastructures minimises the threat of MITM attack in the sense that the attacker will have to intercept all of the distributed file splits for the MITM attack to be effective. Since the shards are distributed to different multiple CSP's the intercepted data will be incomplete and un-useful to the attacker. Even if the attacker commits a Message Replay attack by changing the content of the intercepted shards, the CDDI framework metadata server can be used with the File Download Process to reconstruct the file to its original form.
U2R and R2U attacks	The U2R attack enables attacker to maliciously log into a system as a legitimate user using authorised system credentials and R2U attack enables an attacker to exploit a system vulnerabilities though sending probing packets to the system. The CDDI framework addresses threats from these attacks through the use of different multiple cloud storage facilities to store the distributed fragments of the subscriber's data. No single CSP has the subscriber's complete data and hence a successful U2R or R2U attacker only sees a portion of the subscribers data which will be scrambled and un-useful.

7. ACKNOWLEDGMENTS

I am thankful to Prof. William W. Oblitey for his immense contribution and guidance to conduct this study. Special appreciation goes to Dr. R. K. Boadi and Mr. Wofa Adu Gyamfi of KNUST Mathematics department; Mr. William Morgan-Darko, Mr. Yaw Darkwa, Mr. Samuel Acheampong, Mr. Dawuda Ahmed, Mr. Daniel Anane Agbemava, Mr. Benjamin Odoi Lartey, Mr. Benjamin Tei Partey, and Mr. Nuku Atta Kordzo Abiew, all of KNUST Computer Science department for their priceless support.

8. REFERENCES

[1] Youssef, A. E., Alageel, M., (2012), "A Framework for Securing Cloud Computing", International Journal of Computer Science Issues, Vol. 9, No. 4, pp. 487-500.

[2] Khatri, S. K., Singhal, H., Bahri, K. (2013), "Multi-Tenancy Engineering Architecture in SaaS", *International Journal of Computer Applications*. [Online]. Available From: http://research.ijcaonline.org/icrito/number1/icrito1309.pdf

[3] Khan, N., Yasiri, A. (2016), "Identifying Cloud Security Threats to Strengthen Cloud Computing Adoption Framework ", *Procedia Computer Science, ScienceDirect*, Vol. 94, pp. 485-490.

[4] Shapland, R. (2017). Multi-Tenancy Cloud Security Requires Enterprise Awareness. Available from: http://searchcloudsecurity.techtarget.com/tip/Avoid-the-risks-of-multi-tenant-cloud-environments-through-awareness

[5] Trigueros-Preciado S., Perez-Gonzalez D., Solana-Gonzalez P., (2013). "Cloud computing in industrial SMEs: identification of the barriers to its adoption and effects of its application" *Electronic Mark*, Vol. 23, No. 2, pp. 105-114

[6] Ahmed, N. (2017), "Cloud Computing: Technology, Security Issues and Solutions", *IEEE*, [Online]. Available from: http://ieeexplore.ieee.org/document/7905258/

[7] The Treacherous 12, (2017). The Treacherous 12: Cloud Computing Top Threats in 2016 [Online]. Avalable from: http://www.storm-clouds.eu/services/2017/04/the-treacherous-12-cloud-computing-top-threats-in-2016/

[8] McMillan, R., Knutson, R. (2017). Yahoo Triples Estimate of Breached Accounts to 3 Billion. [Online]. Available from: https://www.wsj.com/articles/yahoo-triples-estimate-of-breached-accounts-to-3-billion-1507062804

[9] Chauhan, K. (2015). "Ensuing Data Storage Security in Cloud Computing", *International Journal of Computer Science and Information Technology Research*, Vol. 3, No. 2, pp. 283-287

[10] Sailaja, K. and Usharani, M. (2017), "Cloud Computing Security Issues, Challenges and its Solutions in Financial Sectors", *International Journal of Advanced Scientific Technologies, Engineering and Managemnt*, Vol. 3, No.1, pp. 190-196.

[11] Wei, W. (2016). Insider Breach: T-Mobile Czech Employee Steals and Sells 1.5 Million Users Data. Available from: https://thehackernews.com/2016/06/t-mobile-hacked.html

[12] ABS (2016). ABS Update – *2016 Online Census Form*. [Online]. Available from: http://www.abs.gov.au/ausstats/abs@.nsf/mediareleasesbyReleaseDate/617D51FA32D27BF9CA25800A0077B7BD?OpenDocument

[13] Rao, R. V. and Selvamani, K. (2015), "Data security challenges and its solutions in cloud computing", *Procedia Computer Science*, Vol. 48, pp. 204-209.

[14] Khandelwal, S. (2017). It's 3 Billion! Yes, Every Single Yahoo Account Was Hacked In 2013 Data Breach. Available from: https://thehackernews.com/2017/10/yahoo-email-hacked.html

[15] Wang, J. (2009). Computer Network Security Theory and Practice. Springer

[16] OpenCirrus (2017). *Cloud Computing Challenges In 2017*. [Online] Available from: http://www.opencirrus.org/cloud-computing-challenges-2017/

[17] CSA (2011). Security guidance for critical areas of focus in cloud computing V3.0. [Online] Available from: https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf

[18] OPC (2011). *Fact Sheet: Introduction to Cloud Computing.* [Online] Available from: https://www.priv.gc.ca/resource/fs-fi/02_05_d_51_cc_e.pdf

[19] Shah, H., Anandane, S. S., (2013), "Security Issues on Cloud Computing" *International Journal of Computer Science and Information Security*, Vol. 11, No. 8, pp. 25-33

[20] Hussain, S. A., Fatima, M., Atif, S., Imran, R., Raja, K. S., (2017). "Multilevel classification of security concerns in Cloud Computing*", Applied Computing and Informatics*, Vol. 13, pp. 57-65

[21] Mahmood, Z. (2011), "Data Location and Security Issues in Cloud Computing" *International Conference on Emerging Intelligent Data and Web Technologies*, *IEEE.* Available from: http://ieeexplore.ieee.org/document/6076420/

[22] Raisian, K. and Yahaya, J. (2015), "Security Issues Model on Cloud Computing: A Case of Malaysia", *International Journal of Advanced Computer Science and Applications*, Vol. 6, No. 8, pp.216-223.

[23] Gray, D. (2014). Data ownership in the cloud. [Online]. Available from: http://dataconomy.com/2014/03/data-ownership-in-the-cloud/

[24] FileCloud, (2016). Data Ownership in the Cloud – How does it affect you? [Online]. Available from: https://www.getfilecloud.com/blog/2016/11/data-ownership-in-the-cloud-how-does-it-affect-you/#.WgG7_I-0Pct

[25] O'Reilly, J. (2017). *7 Ways to Secure Cloud Storage*. [Online] Available from: https://www.networkcomputing.com/data-centers/7-ways-secure-cloud-storage/866645128

[26] Jain, R. (2013). Hadoop and HDFS for Beginners. [Online]. Available from: https://www.slideshare.net/rahuldausa/hadoop-hdfs-for-

beginners

[27] Roshan, B. (2014). General Architecture of the Google File System. [Online]. Available from: http://programming-project.blogspot.com/2014/04/general-architecture-of-google-file.html

[28] Carson, C. (2016). How much data does Google store? [Online]. Available from: https://www.cirrusinsight.com/blog/much-data-google-store

[29] Strickland, J. (2017). How the Google File System Works. [Online]. Available from: http://computer.howstuffworks.com/internet/basics/google-file-system5.htm

[30] Techopedia, (2017). Google File System (GFS). [Online]. Available from: https://www.techopedia.com/definition/26906/google-file-system-gfs

[31] Roshan, B. (2014). General Architecture of the Google File System. [Online]. Available from: http://programming-project.blogspot.com/2014/04/general-architecture-of-google-file.html

[32] Natarajan, R. (2012). Apache Hadoop Fundamentals – HDFS and MapReduce Explained with a Diagram. [Online]. Available from: http://www.thegeekstuff.com/2012/01/hadoop-hdfs-mapreduce-intro/comment-page-1/

[33] Hadoop, (2013). HDFS Architecture Guide [Online]. Available from: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

[34] DeZyre, (2016). Hadoop Architecture Explained – What it is and why it matters. [Online]. Available from:https://www.dezyre.com/article/hadoop-architecture-explained-what-it-is-and-why-it-matters/317

[35] BackBlaze, (2015a). Backblaze Open Sources Reed-Solomon Erasure Coding Source Code. [Online]. Available from: https://www.backblaze.com/blog/reed-solomon

[36] BackBlaze, (2015b). Backblaze Open Sources Reed-Solomon Erasure Coding Source Code. [Online]. Available from: https://www.backblaze.com/blog/vault-cloud-storage-architecture/

[37] BackBlaze, (2017). Cloud Storage that’s astonishingly easy and low-cost. [Online]. Available from: https://www.backblaze.com/

[38] Chou, Te-Shun (2013), "Security Threats on Cloud Computing Vulnerabilities", *International Journal of Computer Science and Information Technology*, Vol. 5, No. 3, pp. 79-88.

[39] Lee, P. (2012). Design Research: What is it? Why do it? [Online]. Available from: https://reboot.org/2012/02/19/design-research-what-is-it-and-why-do-it/

[40] Twum F., Hayfron-Acquah J. B, Morgan-Darko W., A Proposed Enhanced Transposition Cipher Algorithm Based on Rubik’s Cube Transformations, International Journal of Computer Applications, Vol. 182, No. 35, pp 18-26, January 2019.

[41] Twum, F., Hayfron-Acquah J. B., Oblitey W. W., Morgan-Darko W., Reed-Solomon Encoding: Simplified for Programmers, International Journal of Computer Science and Information Security, Vol 14, No. 11, November 2016

[42] Twum F., Hayfron-Acquah J. B., Oblitey W. W., Boadi R. K., A proposed algorithm for generating the Reed-Solomon Encoding Polynomial Coefficeints over GF(256) for RS[255,223]8,32, International Journal of Computer Applications, Vol. 156, No. 1, pp 24-39, December 2016.

[43] Twum, F., Hayfron-Acquah J. B., Oblitey W. W., Morgan-Darko W., Reed-Solomon Decoding Simplified for Programmers, International Journal of Computer Science and Information Security, Vol 15, No. 1, January 2017.

[44] Chima, R. (2016). Cloud Security – Who owns the data? [Online]. Available from: https://www.bbconsult.co.uk/blog/cloud-security-who-owns-the-data