Journal of Advances in Mathematics and Computer Science

xx(x): ....., 2021; Article no.JAMCS.xxxxx

ISSN: 2456-9968

(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

An enhanced Error Detection and Correction Scheme for Enterprise Resource Planning (ERP) Data Storage

Arnold Mashud Abukari[*1], Edem Kwedzo Bankas[2] and Mohammed Muniru Iddrisu[3]

[1] *Tamale Technical University, Tamale, Ghana*

[2,3] *C.K. Tedam University of Technology and Applied Sciences, Navrongo, Ghana*

Original Research Article

Received: XX December 20XX
Accepted: XX December 20XX
Online Ready: XX December 20XX

Abstract

In this research paper, a Redundant Residue Number System (n,k) code is introduced to enhance Cloud ERP Data storage. The research findings have been able to demonstrate the application of Redundant Residue Number System (RRNS) in the concept of Cloud ERP Data storage. The scheme contributed in addressing data loss challenges during data transmission. The proposed scheme also addressed and improved the probability of failure to access data compared to other existing systems. The proposed scheme adopted the concept of Homomorphic encryption and secret sharing whiles applying Redundant Residue Number System to detect and correct errors. The moduli set used is $\{2^m,\ 2^m+1,\ 2^{m+1}-1,\ 2^{m+1}+1,\ 2^{m+1}+k,\ 2^{2m}-k,\ 2^{2m}+1\}$ where k is the number of the information moduli set used. The information moduli set is $\{2^m,\ 2^m+1,\ 2^{m+1}-1\}$ and the redundant moduli is $\{2^{m+1}+1,\ 2^{m+1}+k,\ 2^{2m}-k,\ 2^{2m}+1\}$. The proposed scheme per the simulation results using python reveals that it performs far better in terms of data loss and failure to access data related concerns. The proposed scheme performed better between 41.2% for data loss to about 99% for data access based on the combination of $(2,4)$ and $(2,5)$ data shares respectively in a (k,n) settings.

Keywords: Cloud; Homomorphic Encryption; Data Storage; ERP; Error Detection and Correction; Data loss; Data access

2010 Mathematics Subject Classification: 53C25; 83C05; 57N16

1 Introduction

The security issues regarding Cloud Enterprise Resource Planning (ERP) Data has more connection with trust related issues between the Cloud ERP service provider and the user or client. The user or

Journal of Advances in Mathematics and Computer Science

xx(x): ....., 2021; Article no.JAMCS.xxxxx

ISSN: 2456-9968

(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

the client access the Cloud ERP solution through the help of web services.

The web services are used to connect the Cloud ERP provider and the client or user via the internet. Cloud service providers have over the years attempted to provide security solutions in the form of data encryption, authentication, authorisation and fraud detection. Rittinghouse and Ransome (2016) identified three (3) classification of security issues in Cloud ERP data namely physical security, transmission security and storage security.

The storage of Cloud ERP Data is very essential in ensuring data privacy and security especially data transmission. The detection and correction of the cloud ERP Data is one of the surest ways to ensure the right data is transmitted.

2 Cloud ERP Emergence

The concept of Cloud ERP and Cloud computing has been influenced and made possible by three (3) technological advancement in the world of Information Technology (Schubert and Adisa, 2011). These very advanced technologies are complementary and influential achievements that ha contributed greatly towards the acheivement of cloud computing. Schubert and Adisa (2011) identified these achievements as Ajax Technology, the concept of multitenancy and virtualisation.

AJAX Technology: The term AJAX stands for Asynchronous JavaScript and XML. A Technology for creating better, more interractive and faster web related applications. The Ajax technology makes it possible for an application to be hosted by an external hosting application locally.

This Technology allows the client to establish communication with a server and to change web pages dynamically without reloading onto the hosting application. Linthicum (2009) argues that the Ajax Technology helps create what he termed as ”rich client” and has boosted the use of clients and devices.

AJAX Technology is argued as the most viable Rich Internet Application (RIA) with its very tremendous industry achievements as well as used as a reliable backbone to several tool kit and emerging frameworks. In the AJAX Technology, a user sends a JavaScript call in the form of XMLHttpRequest

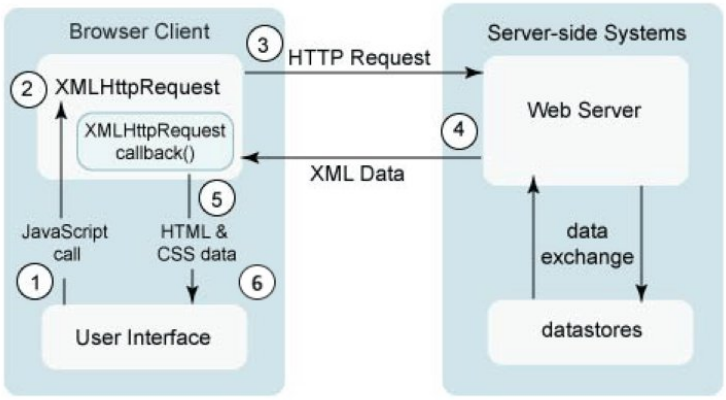

Figure 1: Ajax Technology

Journal of Advances in Mathematics and Computer Science

xx(x): ....., 2021; Article no.JAMCS.xxxxx

ISSN: 2456-9968

(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

to the browser. The browser sends an HTTPRequest to a web server. The Web servers looks into its database or datastore, fetches the requested data and send to the browser in the form of XML Data. The browser upon receiving the XML Data through its XMLHttpRequestcallback() module, it then forward the data to the user using HTML and CSS Data.

The Concept of Multitenancy: The Multitenancy concept in cloud computing is seen as a software architecture with a single software instance built with the potential of serving multiple and distinct user groups. A typical example is Cloud ERP which is a Software-as-a-service (SaaS).

Multitenancy has shared hosting ability, a situation in which a server or software is used by different customers or tenants. The application or software to be shared by customers are customised for each customer or tenant based on their needs and subscriptions. Velte et al (2010) described the concept of multitenancy as the ”shared use of an installation of a single software program by multiple client companies using their own private and individual data spaces”.
Virtualization: Babcock (2010) describes virtualization as the sharing of physical computer resources.

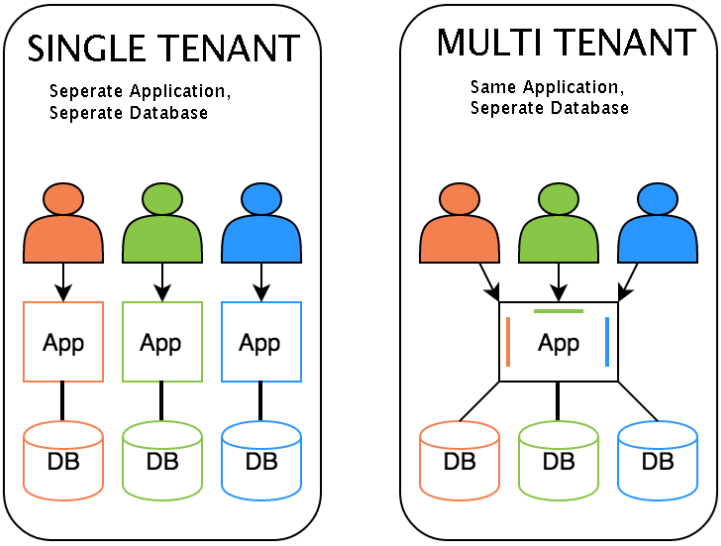

Figure 2: Multitenancy in Cloud

Loganayagi and Sijatha (2010) defines Virtualization as a technique that allows for the creation of abstract layer of a system resources whiles hiding the complexities of hardware and software working environment.

The implementation of virtualization is done with the help of hypervisor technology according to Rutkowska and Tereshkin (2008). In virtualization, a combination of both hardware and software engineering is used to create a virtual machine (VM).

This virtual machine (VM) allows multiple operating systems (OS) to run on one platform. The virtual creation of hardware, software, platform, an operating system, storage or a network device is all built in the concept of virtualization (Kretzschmar and Hanigk, 2010).

In virtualization, the physical resources of a server are logically separated and use as different

Journal of Advances in Mathematics and Computer Science

xx(x): ....., 2021; Article no.JAMCS.xxxxx

ISSN: 2456-9968

(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

isolated machines. The CPU, the RAM and the Hard Disk are all shared to the different isolated machines called virtual machines usually based on their requirements. Virtualization is the backbone of Cloud Computing which brings efficcient and numerous benefits to cloud service clients (Rashid and Chaturvedi, 2019).

According to Velte et al (2010), there are two types of Virtualization applicable to cloud computing

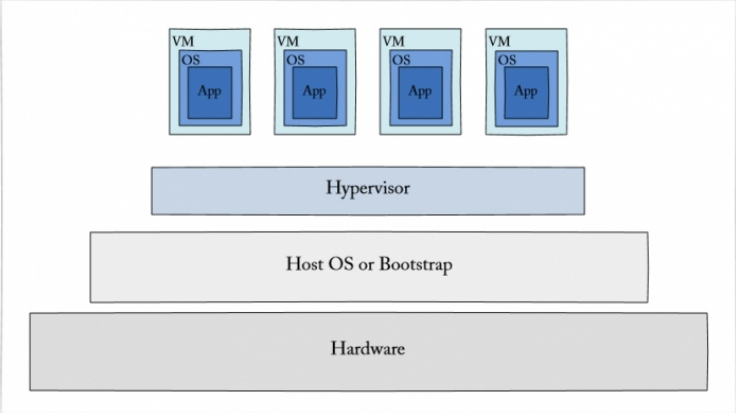

Figure 3: Virtualization Concept

environment. These are Full virtualization and Paravirtualization.

In the concept of Full virtualization, a complete copy of a one computer or computing device is installed or replicated in one machine or computer called the virtual Machine (VM). The virtual machine (VM) have all the replica software of the actual computer or server. This concept is implemented in a remote environment where a remote datacenter delivers a service in a virtualised fashion.

In the Full Virtualization concept, the sharing of a computer system among multiple clients or users is achieved and this is done by isolating the clients from each other. The concept of the Paravirtualization is heavily dependent on the ability of the hardware to allow multiple operating systems (OS) to operate or to be installed on a single machine. This is achieved through the efficient use of the system resources like CPU, Hard disks, memory and processors.

A typical example of a Paravirtualization is the VMWare software. In Paravirtualization, not all the services are available. The services are provided partially based on specific configurations. Disaster recovery, migration and capacity management are some of the advantages of the Paravirtualization (Velte et al, 2010).

3 Data Storage in Cloud

A cloud computing model that allows clients or cloud service users to store their data and access their data on the internet through a cloud service provider is known as cloud storage. The cloud service provider is responsible for managing and operating the data storage as a service to the client. The client usually purchase the data storage capacity as a service provided by the cloud service provider usually in a pay-as-you-go model.

Journal of Advances in Mathematics and Computer Science

xx(x): ....., 2021; Article no.JAMCS.xxxxx

ISSN: 2456-9968

(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

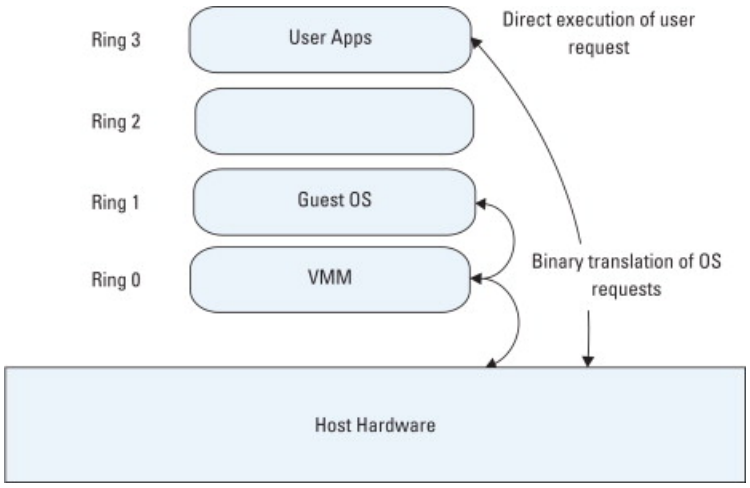

Figure 4: Full Virtualization

According to literature, there are basically three (3) types of cloud data storage which offers different advantages and applicable to different use cases. These three (3) cloud data storage types are object storage, file storage and block storage.

3.1 Block Storage

Research reveals that block storage provides a fixed-sized raw data storage capacity where each storage volume is treated an independent disk drive. The independent disk drives are controlled by an external operating system as if they were mounted as a physical disk.

Storage Area Network (SAN), internet Small Computer Systems Interface (iSCSI) and local disks are some of the common examples of the block data storage devices. Researchers believes that the block data storage is the most commonly used type of data storage in cloud.

The block data storage approach is ideal for databases since databases usually need consistent input and output performance coupled with very low-latency in terms of network connectivity. The block storage approach also supports the Redundant Array of Independent Disks (RAID) volumes.

The RAID Volumes allows the combination of multiple disks usually organise in through mirroring or stripping. The block storage supports applications that requires server-side processing like PHP, .Net and Java. Mission-critical applications like MicrosoftSharePoint, Oracle, SAP and MicrosoftExchange are all supported in the block data storage approach.

Despite the advantages of the Block data Storage, one of the biggest challenge is that it can only be accessed through the existence of an operating system (OS). This is applicable in Infrastructure as a service (IaaS) model of cloud computing.

Journal of Advances in Mathematics and Computer Science

xx(x): ....., 2021; Article no.JAMCS.xxxxx

ISSN: 2456-9968

(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

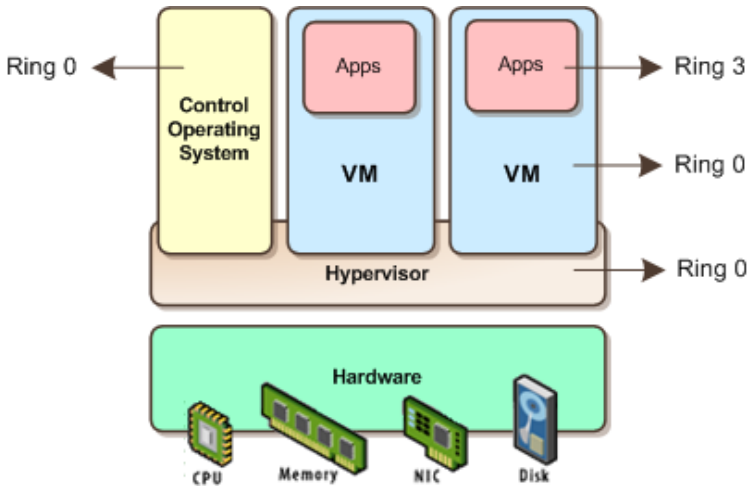

Figure 5: Paravirtualization

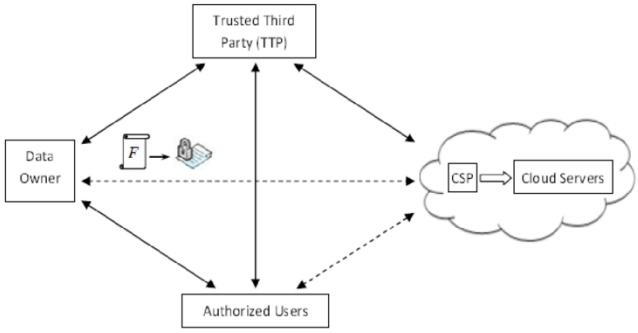

Figure 6: Data Storage in Cloud

3.2 Object Storage

As mentioned above, the inability to access data through the block data storage approach has played to the advantage of Object Data Storage to a large extent. Data stored on the Object Data storage system can be accessed through Application Programmable Interfaces (APIs) or HyperText Transfer Protocols (http/https).

The object data storage approach allows you to store any kind of data that can be accessed globally irrespective of your geographical location. Photos, log files, videos etc can be stored on the object data storage.

As the data stored on the Object data storage grows from terabytes into pentabytes range and beyond, it becomes more attractive as reported by cloud data storage researchers.

Journal of Advances in Mathematics and Computer Science

xx(x): ....., 2021; Article no.JAMCS.xxxxx

ISSN: 2456-9968

(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

3.3 File Storage

The file storage approach in the cloud data storage is deployed as a network attached storage system. This approach of cloud data storage mechanism uses a file system strategy to place and share data. it works very well when applied to a Local Area Network (LAN).

The performance of the file storage approach suffers when applied and made accessible to a Wide Area Network (WAN) since most file systems can not handle alot of files over the wide area network (WAN).

Figure 7: File Storage in Cloud

3.4 Reliability and Confidentiality in Cloud Data Storage

Reliability, confidentiality and scalability are some of the major challenges when the cloud is used for data storage. The concept of homomorphic encryption also throws additional challenge to the cloud service providers since the cloud is suppose to process data in their encrypted form without decrypting the data due to trust issues. All these concerns should be taken into concern when the cloud is to be used for data storage.

According to Ahmed et al (2017), technical characteristics of mobile devices such as energy consumption should also be considered when reliability issues in the cloud are being researched on.

Tchernykh et al. (2017) exhibited how data replication, redundant residue number systems(RRNS) and secret sharing schemes(SSS) under the uncertainty condition of the cloud can be used to achieve reliability and confidentiality in the distributed storage environment. Tchernykh et al (2017) also applied erasure codes as well as homomorphic encryption to achieve the reliability and confidentiality.

One of the alternative solutions proposed to build a reliable data storage solution for the cloud is

to apply error correction codes which is based on the Redundant Residue Number System (RRNS) as argued in (Dimakis et al,2010).

The application of regenerating codes were also used by Lin et al (2014) to help build a reliable storage system for the cloud. The challenges with the application of the Dimakis et al (2010) and Lin et al (2014) by using the error corrections based on RRNS and the regenerating codes do not allow the processing of encrypted data.

Achieving homomorphism propert is very essential for a reliable data storage in the cloud due to the untrusted nature of the cloud service providers. It is very essential to achieve homomorphism since it allows computations on encrypted data without decrypting the data and without additional computational cost according to (Rivest et al, 1978).

As argued earlier, one of the major breakthroughs in the concept of homomorphism is Gentry (2010) when he proposed the fully homomorphic encryption scheme for both addition and multiplication.

However, Gentry (2010) scheme failed to deal with significant data redundancy as it's major challenge. Literature also states that Gentry (2010) scheme also was confronted with lack of tools to handle the arithmetic operations.

One of the proposed schemes that has been able to achieves or assures the safety of data storage in the cloud in terms of homomorphism, scalability, safety, reliability and confidentiality is the scheme proposed by Gomathisankaran et al. (2011).

The researchers in Gomathisankaran et al (2011) constructs the scheme using homomorphic secret sharing schemes by applying the Redundancy Residue Number Systems(RRNS). According to literature, the moduli sets are the secrets stored by the client of the cloud service provider.

The proposed solution by Gomathisankaran et al (2011) leads to increase in the load of the network and memory usages during data processing and this makes it inappropriate for implementation.

4 Related Works on Data Storage Security

There are different approaches to the construction of cloud computing systems for data storage and processing. These data storage mechanisms are usually based on the traditional grid computing and cloud computing strategies or paradigms according to Vouk (2008).

Data storage in the cloud computing and grid computing environments have infrastructure which have similar characteristics but with principal differences according to (Vouk 2008).As argued earlier, the use of the cloud for data storage comes with challenges. notably among those challenging factors are security of the data, reliability and scalability.

Mora et al (2012) stated that the security, reliability and scalability in storing data in the cloud under limited internet connection bandwidth are the challenges confronting the implementation of cloud computing in terms data storage. The argument made by Mora et al (2012) was also supported and backed by Ahmed and Rehmani (2017).

In an attempt to provide reliability and quick access to distributed data in the cloud computing environment, a group of researchers Chang et al (2008) proposed Bigtable system which was based on replication of unencrypted data without providing data security and privacy.

Journal of Advances in Mathematics and Computer Science

xx(x): ....., 2021; Article no.JAMCS.xxxxx

ISSN: 2456-9968

(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

The proposed bigtable system failed to handle data privacy and not applicable to Cloud ERP Data or data generated from Software As a Service (SaaS). In 2008, an alternative system was proposed based on splitting of the datasets to be sent to the cloud into independent chunks. These independent chunks are processed in parallel hence making processing on these chunks faster (Dean and Ghemawat, 2008).

As researchers continued finding solutions, Herodotou et al (2011) argued that the solution proposed by Dean and Ghemawat (2008) has a major drawback which they identified as low efficiency.

To provide security to data stored in the cloud computing environment, homomorphic encryption scheme introduced by Rivest et al (1978) is seen as an alternative solutions. The proposed homomorphic encryption scheme by Rivest et al (1978) allows the processing to be done on encrypted data.

However, significant achievement on homomorphic encryption was made by Gentry (2009) where he was able to proposed a fully homomorphic encryption. The challenge with the Gentry (2009) proposed homomorphic encryption scheme for cloud data storage is that, it was built on ideal lattice on a classical case which lead to large data reducndancy of the stored data hence not applicable to big data storage systems like cloud ERP data.

Ali et al (2021) presented a data protection model which is built using a hybrid of cryptographic algorithm. They applied Advanced Encryption Standard (AES), Blowfish and Message-Digest algorithm (MD5) for the data protection model. They argued that their solution provides speed and robust data encryption. Despite the hybrid cryptographic approach by Ali et al (2021), their scheme is unable to detect and correct errors but they had their attention on data protection.

Zagan and Danubianu (2021) described data lake as a new raw data storage technology that is increasingly common and gaining attention in the research world. Zagan and Danubianu (2021) identified four (4) different ways to implement data lake architecture namely Data Lake On-Premises, Cloud Data Lake, Multi-Cloud data lake and Hybrid Data lake. The Concept of Data lake is for storing raw data or files but do not have a mechanism to detect and correct errors during data transmission.

Homomorphic encryption schemes using Residue Number Systems have been proposed in recent years hence the need for this research.

4.1 Reliability and Confidentiality in Cloud Data Storage

Reliability, confidentiality and scalability are some of the major challenges when the cloud is used for data storage. The concept of homomorphic encryption also throws additional challenge to the cloud service providers since the cloud is suppose to process data in their encrypted form without decrypting the data due to trust issues. All these concerns should be taken into concern when the cloud is to be used for data storage.

According to Ahmed et al (2017), technical characteristics of mobile devices such as energy consumption should also be considered when reliability issues in the cloud are being researched on.

In 2016, Tchernykh et al, in (Tchernykh et al,2017) exhibited how data replication, redundant residue number systems(RRNS) and secret sharing schemes(SSS) under the uncertainty condition of the cloud can be used to achieve reliability and confidentiality in the distributed storage environment. Tchernykh et al (2017) also applied erasure codes as well as homomorphic encryption to achieve the

ISSN: 2456-9968

(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

reliability and confidentiality.

One of the alternative solutions proposed to build a reliable data storage solution for the cloud is to apply error correction codes which is based on the Redundant Residue Number System (RRNS) as argued in Dimakis et al. (2010).

The application of regenerating codes were also used by Lin et al (2014) to help build a reliable storage system for the cloud. The challenges with the application of the Dimakis et al (2010) and Lin et al (2014) by using the error corrections based on RRNS and the regenerating codes do not allow the processing of encrypted data.

Achieving homomorphism propert is very essential for a reliable data storage in the cloud due to the untrusted nature of the cloud service providers. It is very essential to achieve homomorphism since it allows computations on encrypted data without decrypting the data and without additional computational cost according to (Rivest et al, 1978).

As argued earlier, one of the major breakthroughs in the concept of homomorphism is Gentry (2010) when he proposed the fully homomorphic encryption scheme for both addition and multiplication.

However, Gentry (2010) scheme failed to deal with significant data redundancy as it's major challenge. Literature also states that Gentry (2010) scheme also was confronted with lack of tools to handle the arithmetic operations.

One of the proposed schemes that has been able to achieves or assures the safety of data storage in the cloud in terms of homomorphism, scalability, safety, reliability and confidentiality is the scheme proposed by (Gomathisankaran et al, 2011).

The researchers in Gomathisankaran et al (2011) constructs the scheme using homomorphic secret sharing schemes by applying the Redundancy Residue Number Systems(RRNS). According to literature, the moduli sets are the secrets stored by the client of the cloud service provider.

The proposed solution by Gomathisankaran et al (2011) leads to increase in the load of the network and memory usages during data processing and this makes it inappropriate for implementation.

5 Proposed Data Storage Scheme

In our quest to ensure cloud ERP data are securely stored, we considered the key problems on data storage in the cloud which are error detection and error corrections. This research is seeking to improve the error detection and corrections in the Redundant Residue Number System (RRNS) approached by (Chervyakov et al, 2016).

The application of the redundant residue number system for cloud ERP data storage and processing seeks to provide a secure, reliable and scalable storage for the cloud service provider.

The proposed solution has the properties of homomorphism and secret sharing schemes properties which makes it very useful for securing the Cloud ERP Data as well as processing them in their encrypted form.

Journal of Advances in Mathematics and Computer Science

xx(x): ....., 2021; Article no.JAMCS.xxxxx

ISSN: 2456-9968

(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

5.1 Proposed Redundant Residue Number System (RRNS(n,k)) Code

The proposed Redundant Residue Number System code (RRNS(n,k)) is not only proposed for representation of Cloud ERP Data but also to ensure the protection and security of Cloud ERP Data in a multi-cloud environment. The Cloud ERP Data is split into $n-shares$ with corresponding moduli in two categories namely the information moduli (k) and the redundant moduli ($n-k$) in $n-shares$. The redundant moduli has the capability to ensure self-checking and detection and correction of errors.
Considering the Cloud ERP Data D split into data chunks of d_1,d_2, d_3,..., d_n with corresponding information moduli of p_1, p_2,...,p_k and a redundant moduli of p_{k+1}, p_{k+2}, ..., p_n. The information dynamic range and the redundant dynamic range are calculated below:

$$p_k = \prod_{i=1}^{k} p_i \tag{5.1}$$

$$p_{n-k} = \prod_{i=k+1}^{n-k} p_{k+j} \tag{5.2}$$

The intervals $[0, p_k-1]$ and $[p_k, p_{n-k}-1]$ represents the information(legitimate) and redundant(illegitimate) ranges of the Cloud ERP Data D. Any Cloud ERP Data chunk represented by a residue of the Redundant Residue Number System (RRNS) can be recovered by any combination of k number of the moduli sets and this is the basis for error detection and correction. A Redundant Residue Number System (RRNS) with k number of information moduli and $n-k$ number of redundant moduli is called a RRNS(n,k) code.
We consider the moduli set $\{2^m, 2^m+1, 2^{m+1}-1, 2^{m+1}+1, 2^{m+1}+k, 2^{2m}-k, 2^{2m}+1\}$ where k is the number of the information moduli set used. The information moduli set is $\{2^m, 2^m+1, 2^{m+1}-1\}$ and the redundant moduli is $\{2^{m+1}+1, 2^{m+1}+k, 2^{2m}-k, 2^{2m}+1\}$.
Any Cloud ERP Data expressed as an Integer belonging to the range legitimate or information moduli set are considered as legitimate and can be recovered by any k group of number of moduli and it's accompanied residue digits.
A group of Redundant Residue Number System (RRNS) codewords having integer representations of the Cloud ERP Data (D) is applied in the implementation with addition operations. The Redundant Residue Number System (RRNS) codewords are generated using the formula below:

$$C_i = D mod p_i \tag{5.3}$$

where C_i is the codeword associated with the $i-th$ moduli, p_i is the corresponding moduli and D is the Cloud ERP Data. The Redundant Residue Number System (RRNS) codewords are applied in the distribution of the Cloud ERP Data chunks to the cloud for secured storage.

5.2 Mathematical Illustration of the RRNS(n,k) Concept

Considering RRNS(7,3) where $n=7$ and $k=3$ and a value of $m=2$ used on the moduli set above, the information moduli of the set is presented as follows:
$p_1 = 4$
$p_2 = 5$
$p_3 = 7$
Let the redundant moduli be as follows:
$p_4 = 9$
$p_5 = 11$
$p_6 = 13$
$p_7 = 17$
The legitimate range of RRNS(7,3) will be [0,139] since $\prod_{i=1}^{k=3} p_i = 140$ and the illegitimate range will

Journal of Advances in Mathematics and Computer Science

xx(x): ....., 2021; Article no.JAMCS.xxxxx

ISSN: 2456-9968

(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

as well be [140, 21878].
Let the Cloud ERP Data D to be an integer where $D = 123$ and by appplying equation 3.18, the following codewords are generated from the encoding:
$123 mod 4 = 3$
$123 mod 5 = 3$
$123 mod 7 = 4$
$123 mod 9 = 6$
$123 mod 11 = 2$
$123 mod 13 = 6$
$123 mod 17 = 4$
Hence the RRNS(7,3) codewords for the Cloud ERP Data (D) is $3, 3, 4, 6, 2, 6, 4$

5.3 The Proposed Error Detection and Correction Model for ERP Data Storage

In our resolve to design a reliable and secured Cloud ERP Data storage model, we applied the error correction approach in the redundant residue number system(RRNS) whiles ensuring the secret sharing schemes and homomorphic encryption schemes properties are achieved.

The research propose that the cloud ERP Data be shared among a number of cloud service providers where the $ith - cloud$ is the redundant residue number system (RRNS) module as described earlier as C_i. The value of the C_i is of the form:

$$2^l - \varphi_i \tag{5.4}$$

where l is the length of the module C_i and φ_i is a small integer which are chosen based on the availability of computational resources of the cloud which is the $ith - cloud$ as it correspond with p_i of the redundant residue number system (RRNS).

In the secret sharing schemes settings where (k, n) settings are applied, we can recover r according to equation (2.18) where $r = n - k$ using any $k - clouds$ out the set of $n - clouds$ considered. Let the dynamic range of the redundant residue number system(RRNS) be p_i.

Considering the the calculation of data size being an exponential function 2^n as stated by Eckstein (2007) where n is the number of bits. We substituted the number of bits n to be our Data Size and applied logarithm on the exponential function. We calculated the Cloud ERP Data size as:

$$D_s = \log_2 C_i \tag{5.5}$$

The data size of the Cloud ERP Data can also be calculated based on the number of $k - clouds$ and the length of the module l as shown:

$$D_s = k.l \tag{5.6}$$

This means we can compute:

$$\log_2 C_i = k.l \tag{5.7}$$

The ERP data is sent to the cloud in chunks and these chunks has properties and identifiers of the original ERP Data as shown in the above equations. The unique identifiers of the chunks of ERP data sent to the cloud are computed by applying the MD5 algorithm proposed by Wang and Yu (2005) and the SHA-3 algorithm proposed in 2014 by (Pritzker and Gallagher, 2014).

In the figure 3.3, the proposed model for our ERP Data chunk to be sent to the cloud is divided into three(3) key sections namely the header, Shadow of the original ERP Data and a digital signature.

The chunk's property which includes the chunk's index and the data size is captured in the header of

Journal of Advances in Mathematics and Computer Science

xx(x): ....., 2021; Article no.JAMCS.xxxxx

ISSN: 2456-9968

(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

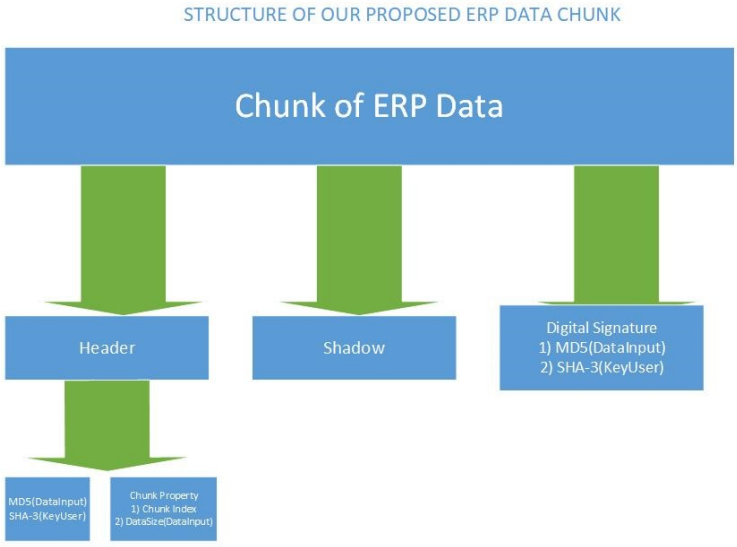

Figure 8: The Proposed ERP Data Chunk

the chunk of ERP data that is to be sent to the cloud. The digital signature uses the MD5 and SHA-3 as proposed by Wang and Yu (2005) and Pritzker and Gallagher (2014) respectively.

5.4 Proposed Parameters

Cloud ERP Solutions just like any cloud solutions or on-premise solutions is capable of crashing which could deny clients of the cloud service provider access to their data. In the case of a cloud ERP Data, it means the whole organisation or company can not work since all the data will be sitting in the crashed cloud. It has serious consequences to the organisations hosting their ERP Data in the cloud.

In 2009, Amazon, a cloud service provider was attacked by denial of service (DDoS) attack for several hours. Microsoft and Google also suffered series of cloud service outages in 2013 and this brought alot of losses to businesses.

Some of the key challenges leading to these lost in service by cloud service providers could be Technical failures and power outages as was reported by Amazon, Google and Microsoft. As reported by Munson (2015) some cloud service providers spends about 30,000 US Dollars to deal with DDoS attacks on a daily basis. Users could not access their data for about 11 hours in what is reported by Leswig (2016) as one the most powerful DDoS attacks.

An Information Technology security giants, Kapersky, reported in 2016 that the longest attack on a cloud lasted for about 197 hours. The 197 hours means it lasted for over eight (8) days according to Kapersky lab (2016).

Considering the report by Kapersky lab (2016) in the first quarter and the Geometric probability stated

Journal of Advances in Mathematics and Computer Science

xx(x): ....., 2021; Article no.JAMCS.xxxxx

ISSN: 2456-9968

(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

in Guntuboyina (2020), we can apply the definition of geometric probability of denial of access to the service offered by the cloud service provider as:

$$Pr(D_a) = \frac{N_f}{N_d} \tag{5.8}$$

Where D_a is the denial of access due to failure, N_f and N_d are the longest number of days a cloud service provider failed in a quarter and the number of days in a quarter respectively.

Considering the report by Kapersky lab (2016), the longest number of days a cloud service provider failed N_f is 8.2 days and the number of days in a quarter N_d is 90. Following equation 3.5, we have:

$$Pr(D_a) = \frac{8.2}{90} \approx 0.09 \tag{5.9}$$

The probability of denial of access to services provided by the cloud service providers is 0.09.
In order to determine and calculate the probability of failure to access data from the cloud service provider we applied the following formula:

$$Pr(D_f) = \frac{M_l}{N_o} \tag{5.10}$$

Where $Pr(D_f)$ is the probability that the client failed to access the data, M_l is the mean between the longest DDoS attack which is $M_l = (0 + 8.2)/2 = 4.1$ and N_o is the event that there was no DDoS attack. Applying these parameters, we have:

$$Pr(D_f) = \frac{4.1}{81.8} \approx 0.05 \tag{5.11}$$

As computed in equation 3.8, the probability of failure to access data $Pr(D_f) = 0.05$.
Researchers Gage (2013), WCO (2014) and Wu et al (2017) all reported that the probability of loss of information is estimated as $Pr(L_i) = \frac{3}{365} \approx 0.01$.

By applying the Bernoulli Probability formula and the law of probability addition on the secret sharing scheme(SSS) we propose the following formula:

$$Pr(k,n) = \sum_{i=r-k+1}^{n} C_n^i (Pr(L_i)^i * (1 - Pr(L_i))^{n-i} + Pr(D_f)^i * (1 - Pr(D_f))^{n-i}) \tag{5.12}$$

Inserting the values of $Pr(L_i)$ and $Pr(D_f)$ generates the following:

$$Pr(k,n) = \sum_{i=r-k+1}^{n} C_n^i (0.01^i * 0.99^{n-i} + 0.05^i * 0.95^{n-i}) \tag{5.13}$$

5.5 Data Redundancy with The Proposed Scheme

In the world of Information Technology and Information Systems, Data Redundancy is a condition that is created in a database or data storage technology system where a data is repeated in more than one data fields. Data redundancy is a common challenge confronting data storage technology systems like the cloud. Handling redundancy in the cloud is very essential in building the trust and confidence in the clients of the Cloud service provider.

Considering worst case scenario where the number of bits that needs to be sent to the cloud or stored in the cloud is a derivative of the moduli set p_i as presented below:

$$\sum_{i=1}^{n} \log_2 p_i \tag{5.14}$$

The input data which is the Cloud ERP Data according to the characteristics of our proposed ERP data chunk, is a derivative of the moduli set p_i of the Redundant Residue Number System (RRNS) and the dynamic range and given as:

$$\sum_{i=1}^{k} \log_2 p_i \tag{5.15}$$

The redundancy of the ERP Data stored in the cloud can be expressed as a ratio of the stored ERP Data and it's original ERP Data size as:

$$\frac{\sum_{i=1}^{n} \log_2 p_i}{\sum_{i=1}^{k} \log_2 p_i} \tag{5.16}$$

For the equation 3.28 to be able effectively handle redundancy issues regarding the storage of the ERP data in the cloud, the Residue Number System (RNS) moduli set p_i must satisfy the condition:

$$2^{l-1} < p_1 < p_2 < p_3 < p_4 < ... < p_n < 2^l \tag{5.17}$$

And the following inequality for redundancy must be satisfied as well.

$$\frac{\sum_{i=1}^{n}(l_i - 1)}{\sum_{i=1}^{k} l_i} < \frac{\sum_{i=1}^{n} \log_2 p_i}{\sum_{i=1}^{k} \log_2 p_i} \leqslant \frac{\sum_{i=1}^{n} l_i}{\sum_{i=1}^{k} l_i} = \frac{n}{k} \tag{5.18}$$

Where l is the module length. From the above inequality, the usual method used to calculate data redundancy is $\frac{n}{k}$. We compare our scheme's data redundancy approach with this and the results presented.

6 Results and Discussions

6.1 Data loss analysis

The research applied and compared the proposed scheme with the results from (Gage, 2013), (WCO,2014) and (Wu et al, 2017). It was observed that the proposed scheme performs better in terms of the probability of data loss during the cloud ERP Data storage.

In the quest to determine the probability of data loss in the proposed scheme, the research applied the following equation:

$$Pr(k,n) = \sum_{i=n-k+1}^{n} C_n^i (Pr(L_i)^i * (1 - Pr(L_i))^{n-i} \tag{6.1}$$

Where $P_r(L_i)$ is estimated as 0.01, the proposed formula for calculating the probability of data loss using the proposed scheme is presented in the equation 4.2.

The figure 9 indicates that the proposed scheme with RRNS settings (k,n) of Probabilities from $(2,4),(3,4),(2,5),(3,5),(4,5)$ for data loss are quite lower than what is proposed in (Gage, 2013), (WCO,2014) and (Wu et al, 2017).

The research findings also observed that the probability of data loss from (n,n) configuration is quite higher compared to Gage (2013), WCO (2014) and Wu et al (2017) as indicated in figure 4.9.

6.2 Data Access Analysis

The research work applied $Pr(D_f) = 0.05$ as derived from Leswing (2016) and Kaspersky lab (2016) by taking the mean between the longest reported DDoS attacks on web related services the the

Journal of Advances in Mathematics and Computer Science

xx(x): ....., 2021; Article no.JAMCS.xxxxx

ISSN: 2456-9968

(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

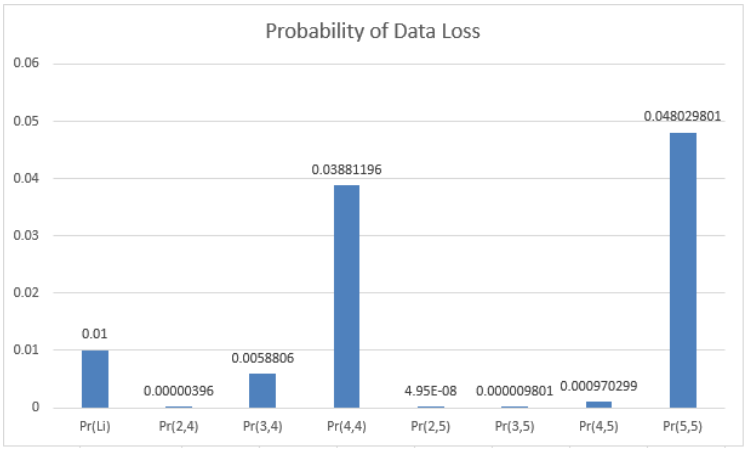

Figure 9: Probability of Data loss Analysis

period when there was no DDoS attacks. The formula below was applied:

$$Pr(k,n) = \sum_{i=n-k+1}^{n} C_n^i (Pr(D_f)^i * (1 - Pr(D_f))^{n-i} \tag{6.2}$$

The research findings compared the probability of failure to access data on the proposed scheme compared to the derived probability of 0.05. It was observed that the proposed scheme has very low

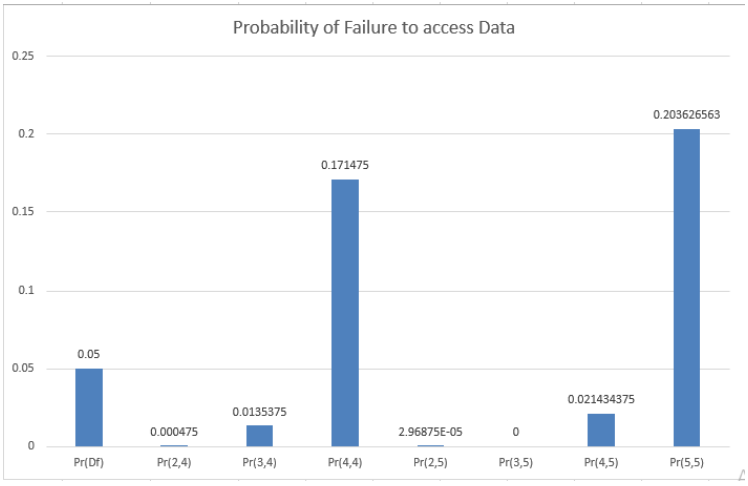

Figure 10: Probability of failure to access data Analysis

probability of failure to access data with configurations $(2,4),(3,4),(2,5),(3,5),(4,5)$.

However, applying the proposed scheme with configuration settings (n,n) results in a higher probability that there will be failure to access data when the proposed scheme is applied.

Journal of Advances in Mathematics and Computer Science

xx(x): ....., 2021; Article no.JAMCS.xxxxx

ISSN: 2456-9968

(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

7 Conclusion

In this research, the findings have been able to demonstrate the application of Redundant Residue Number System (RRNS) in the concept of Cloud ERP Data storage.

The proposed scheme in this research has ignited another new dimension of research in the area of Cloud ERP security. The probability of data loss and the probability of data access are essential elements in Cloud ERP data storage. Gage (2013),WCO (2014) and Wu et al (2017) are all existing schemes that were compared to our proposed scheme.

The proposed scheme has proven to be performing better compared to existing schemes in terms of probability of data loss for the Cloud ERP storage. This research findings indicates that the probability for data loss are quite lower in the proposed scheme.

The research further reveals that the proposed scheme has very low probability of failure to access data by clients.

The proposed system when implemented will improve the data loss by reducing the rate at which data is lost and also reduce the probability of failure to access ERP data in the cloud.
The test runs shows an improvement compared to Gage (2013), WCO (2014) and Wu et al (2017).

8 References

Ahmed,E. and Rehmani,M.H. (2017). P.Bonnet,Editorialtoaspecialsectiononinformationfusionininternetofthings,Inform.Sci.69, 194.

Ali, K. B., Tariq, A. K. M. and Zaid, A. A. (2021) A hybrid cryptography technique for data storage on cloud computing, Journal of Discrete Mathematical Sciences and Cryptography, 24:6, 1613-1624, DOI: 10.1080/09720529.2020.1859799

Babcock, C. (2010) Management Strategies for the Cloud Revolution: How cloud computing is transforming business and why you can't afford to be left behind, New York: McGraw-Hill.

Bankas, E.K. and Gbolagade,K.A. (2013).A residue to binary converter for a balanced moduli set $\{22n+1-1, 22n, 22n-1\}$.2013 International Joint Conference on Awareness Science and Technology and Ubi-Media Computing(iCAST 2013 and UMEDIA 2013). pp. 211-216. doi:10.1109/ICAwST.2013.6765435.

Dimakis,A. G., Godfrey, P. G., Wu,Y., Wainwright,M. J. and Ramchandran,K.(2010). BNetwork coding for distributed storage systems,[IEEE Trans. Inf. Theory, vol. 56, no. 9, pp. 4539–4551.

Gentry, C.(2010). Computing arbitrary functions of encrypted data. Commun ACM, 53(3):97-105

Gomathisankaran, M., Tyagi, A. and Namuduri, K.(2011).HORNS: A homomorphic encryption scheme for cloud computing using Residue Number System. 1-5.10.1109/CISS.2011.5766176

Kar, A., Sur, K., Godara, S., Basak, S., Mukherjee, D., Sukla, A., Das, R. and Choudhury, R. (2016). Secuirity in cloud storage: An enhanced technique of data storage in cloud using RNS. 1-4. 10.1109/UEMCON.2016.7777905.

Journal of Advances in Mathematics and Computer Science

xx(x): ....., 2021; Article no.JAMCS.xxxxx

ISSN: 2456-9968

(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

Kretzschmar, M. and Hanigk,S. (2010). "Security management interoperability challenges for Collaborative Clouds," 2010 4th International DMTF Academic Alliance Workshop on Systems and Virtualization Management, pp. 43-49, doi: 10.1109/SVM.2010.5674744.

Linthicum, D. (2009). Cloud Computing and SOA Convergence in Your Enterprise: A Step-by-Step Guide . s.l. : Addison-Wesley Professional, 2009. 978-0136009221.

Loganayagi, B. and Sujatha.S.(2010). "Creating virtual platform for cloud computing", IEEE International Conference on Computational Intelligence and Computing Research (ICCIC 2010); 28-29, pp.1-4.

Mora,A.C., Chen, Y., Fuchs, A., Lane, A., Lu,R. and Manadhata,P. (2012). Top ten big data security and privacy challenges. Cloud Security Alliance. $https://www.isaca.org/Groups/Professional-English/big-data/GroupDocuments/BigDataTopTen_v1.pdf(Accessed21June2017)$.

Navi, K., Molahosseini,A.S. and Esmeildoust,M. (2010). "How to Teach Residue Number System to Computer Scientists and Engineers",IEEE Transactions on Education, vol.53, no.3.

Rashid, A., and Chaturvedi, A.K. (2017). A Study on Resource Pooling, Allocation and Virtualization Tools used for Cloud Computing. International Journal of Computer Applications, 168, 7-11.

Rashid, A. and Chaturvedi, A. (2019). Cloud Computing Characteristics and Services: A Brief Review. International Journal of Computer Sciences and Engineering, 7(2), 421-426.
Rittinghouse, J.W. and Ransome, J.F. (2016) Cloud Computing: Implementation, Management, and Security. CRC Press, Boca Raton. Rutkowska, J., and Tereshkin, A. (2008). Bluepilling the Xen Hypervisor. Proceedings of the Black Hat Security Conference, Las Vegas, NV, USA.

Schubert, P., and Adisa, F. (2011). Cloud Computing for Standard ERP Systems: Reference Framework and Research Agenda.
Tchernykh, A., Babenko, M., Chervyakov, N., Miranda, V., Cortés-Mendoza, J., Du, Z., Navaux, P. and Avetisyan, A. (2017). Analysis of secured distributed cloud data storage based on multilevel RNS. 382-386. 10.1109/EIConRus.2018.8317112.

Velte,A.T., Velte, T.J. and Elsenpeter,R. (2010). Cloud Computing: A practical Approach, McGraw-Hill.

Vouk,M.A. (2008). Cloud Computing - Issues, research and implementations. CIT, Journal of Computing and Information Technology 16(4)235-384.

Zagan, E. and Danubianu, M. (2021). "Cloud DATA LAKE: The new trend of data storage," 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), pp. 1-4, doi: 10.1109/HORA52670.2021.9461293.
