
# ENHANCING DATA SECURITY IN CLOUD STORAGE USING RESIDUE NUMBER SYSTEM AND ADVANCED ENCRYPTION STANDARD

**DAMILARE NIMAT AKOGUN**

**18/27/MCS005**

**December, 2020**

# SCHOOL OF POSTGRADUATE STUDIES (SPGS)


## ENHANCING DATA SECURITY IN CLOUD STORAGE USING RESIDUE NUMBER SYSTEM AND ADVANCED ENCRYPTION STANDARD

**A M.Sc. THESIS SUBMITTED**

**BY**

**DAMILARE NIMAT AKOGUN**

**18/27/MCS005**

**In Partial Fulfilment of the requirement for the award of Master of Science in Computer Science**

**DEPARTMENT OF COMPUTER SCIENCE**

**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY,**

**KWARA STATE UNIVERSITY, MALETE, NIGERIA**

***December, 2020.***

## DECLARATION PAGE

I hereby declare that this thesis titled Enhancing Data Security in Cloud Storage using Residue Number System and Advanced Encryption Standard is a record of my research. It has neither been presented nor accepted in any previous application for higher degree.

_____ _____

DAMILARE NIMAT AKOGUN Signature / Date

## CERTIFICATION PAGE

This is to certify that this thesis by Damilare Nimat Akogun has been read and approved as meeting the requirements of the Department of Computer Science for the award of the degree of Master of Science in Computer Science.

…… ……
Prof. Kazeem Alagbe Gbolagade — Signature / Date
Main Supervisor

……. ……
Dr. Ronke Seyi Babatunde — Signature / Date
Co-Supervisor

…… ……
Prof. Kazeem Alagbe Gbolagade — Signature / Date
Head of Department

…… ……
Prof. Hamza Abdulraheem — Signature / Date
Dean
School of Postgraduate Studies (SPGS)

…… ……
Dr. Oluwakemi Christiana Abikoye — Signature / Date
External Examiner

## DEDICATION

I dedicate this thesis to Almighty Allah who has been my guide throughout the duration of my postgraduate programme at Kwara State University, Malete and also to my parents Alhaji J.R and Alhaja M.M Akogun who have both provided financial and psychological assistance.

## ACKNOWLEDGMENT

I thank and praise almighty Allah for saving my life up to this moment, for giving me financial stability and for giving me the experience and understanding needed to make the programme successful.

I am honored to work with my supervisor, Professor Kazeem Alagbe Gbolagade. I deeply appreciate his efforts for his total mentoring, guidance, support, warm-heartedness and simple RNS guidance which was crucial to the success of this research.

I thank all the lecturers of Computer Science department; Dr. R.M Isiaka, Dr. Mrs. Ronke Babatunde, Dr. Mrs. Ajao, Dr. A.N Babatunde, Mr. S.O Abdulsalam, Mrs. Shakirat Yusuf, Mrs. B.F Balogun, Mr. D.P Oyinloye, Mr. A.F Kadri and Mr. D. Popoola.

I would like to appreciate the effort of my boss; Dr. Mrs. Ronke Babatunde for her academic support and motherly advice. The efforts are highly appreciated towards the success of this research.

I would also like to appreciate the effort of Engr. Kamaldeen Jimoh for his guidance and contributions to the success of this research.

Finally, I will like to thank my whole family; my father, Alhaji J.R Akogun and to my precious mother, Alhaja M.M Akogun and my siblings for their invaluable support and motivation during this entire period.

**LIST OF TABLES** **Pages**

**LIST OF FIGURES** **Pages**

**LIST OF ALGORITHMS** **Pages**

## Abstract

Cloud computing is a technology by internet, where a large amount of data being pooled by different users is stored. It provides users with a range of services and is highly cost-effective and versatile. The data being stored comes from various organizations, individuals, and communities etc. Thus, security and privacy of data is of utmost importance to all of its users regardless of the nature of the data being stored. Cloud storage security concerns have shown to be the biggest challenge that could impact its large benefits. Data security is a major issue in cloud computing environment; this is becoming a serious problem because the data is stored in a variety of ways over the cloud. The present methods used for encrypting the files in cloud are not highly efficient. The use of multiple encryption technique gives the importance of data security, privacy protection, nature of attacks and other issues that may corrupt the data. Therefore, it is essential to apply effective encryption methods to increase data security. This research presents an enhanced data security in cloud storage with the use of multilevel cryptography so as to protect the confidentiality and privacy of data. The multilevel cryptography utilizes Residue Number System (RNS) and Advanced Encryption Standard (AES-256). It focuses on storing data in the cloud in an encrypted format to avoid data contact from an unauthorized access. The principle of cryptography in RNS was introduced so as to enhance the security of data stored on the cloud. American Standard Code for Information Interchange (ASCII) table was used to generate the decimal numbers of text data, RNS was used for the forward and reverse conversions with the moduli set $\{2^n, 2^{2n}-1, 2^{2n}+1\}$. AES-256 algorithm was also used to attain high level of security. The system was deployed on Heroku cloud platform and its performance is evaluated in reference with encryption time, decryption time, encryption throughput and security level. Other metrics used to compare the developed approach with existing algorithms and multilevel approaches are key length, cryptographic strength and possibility of attacks. The experimental results show the efficacy of the system as the security level of the approach is higher than other existing approaches. Also, the multilevel approach (RNS-AES) utilized only less time for encryption of data and showed that as the file size increases, the throughput also increases which gives a higher throughput.



**Keywords**: cloud storage, data security, data privacy, multilevel cryptography, Residue Number System, Advanced Encryption Standard.

# CHAPTER ONE
# INTRODUCTION

## 1.1 Background to the Study

Cloud computing has evolved steadily over the past few years as a widely accepted computing paradigm as it is expected to play a major role in the future Internet of Things, enabling on-demand provisioning of applications, platforms, and computing infrastructures (Seth, Dalal, & Kumar, 2019). According to the National Institute of Standards and Technology ( NIST), cloud computing is a paradigm that enables ubiquitous, easy, on-demand network access to a common pool of configurable computing resources that can be provided easily and released with minimal management effort or service provider involvement. It provides clients with a virtual computing infrastructure on top of which they can store data and run applications. The main core concepts that the cloud computing paradigm has been built around include on demand computing resources, elastic scaling, lower operational expenses, and promoting a pay-per-use business model for Information Technology (IT) and computing services. It is the fastest growing technology which offers various services over the internet. This technology has changed the face of traditional computing technologies and offers many benefits to the IT enterprises. It provides an adaptable, flexible and convenient way for sharing the data that brings plethora of benefits for both the industry and community (Sirisha & Basha, 2018). Security is the primary challenge facing cloud suppliers and their clients in the cloud area network.

Cloud infrastructure provides a popular data service known as cloud storage, but data security has always remained a major issue in IT and cloud, and is of special concern since data are scattered at different places all over the globe (Seth et al., 2019). The cloud storage can equally save an organization's time and money but trusting the system is very much important because the true asset of every organization is the data they outsource to the cloud server. The entire data reside on a collection of networked resources, allowing the data to be accessed through virtual machines. Often there is a natural resistance for individual users or industry to directly outsource the data to be shared on to the cloud server as the data often might contain confidential information such as transaction data, research data, health data, etc. (Sirisha & Basha, 2018). The security issues of users need to be addressed in order to make the cloud environment reliable and trustworthy. In cloud computing, a trustworthy environment is a basic prerequisite to win confidence of a user to adopt such technology (Nishit, Tarun, Varun, & Vrince, 2018).

Data is a list or collection of raw facts and figures. Security is the protection from danger or threat (Izhar, Kaushal, Fatima & Qadeer, 2018). Data security relates to data protection against unauthorized access. The aim of data security is to safeguard or protect any type of data or information in order to ensure privacy. It is definitely needed to safeguard information from being compelled to be brute (Izhar et al., 2017). Cloud data may be attacked in two ways. One is outsider attack and the other is insider attack (Kartit, marraki, Azougaghe, & Mustapha, 2016). Insider as an administrator may have the ability to hack the user's data. It is very difficult to detect an insider attack. Cloud users should be

very careful when storing their data in cloud storage. Even though the data is accessed by the third party, the actual data should not be gotten (Kartit et al., 2016).

Cryptography refers to the use of mathematics for data security (Izhar et al., 2017). It is a technique of transformation and transmission of confidential data in an encoded manner so that only permitted and intended users can obtain or work on it. It is a Greek word where "crypto" means hidden and "graphy" means writing, so cryptography means "hidden" or "secret" writing. It introduces triads like confidentiality, non-repudiation, integrity and authenticity within ongoing data communication (Anu, Shree & Ahlawat, 2017). Cryptography is a method used for encryption and decryption. Encryption is the conversion of original data i.e. Plain text into a private code (Cipher text). To encrypt data, there are two types of techniques which are commonly referred to as symmetric, also known as conventional cryptography or single key encryption (same key is used in both encryption and decryption) and asymmetric, also known as public key cryptography or public key encryption in which different keys are used in encryption and decryption (Izhar et al., 2017).

Residue Number System (RNS) is a data representation system that allows representing an integer number as a set of smaller numbers. It is an integer number system with the competency to support parallel, carry-free addition, borrow-free subtraction and single step multiplication without unfinished product. For good application of RNS, data conversion must be very fast so that the conversion outlay doesn't nullify advantages of RNS (Aremu & Gbolagade, 2017). Data Conversion in the residue number system is

generally based on either the Chinese Remainder Theorem (CRT) or the Mixed Radix Conversion (MRC) which can be classified as forward and reverse conversions. The forward conversion involves converting a binary or decimal number to its RNS equivalent while the reverse conversion involves converting an RNS number to a binary or decimal number. In any case, RNS is very useful for certain applications such as speech processing, computer security (cryptography), image processing, communications engineering, and transformation in which severe arithmetic operations are addition and multiplication.

Data privacy and security are the two key factors for building the trust of the user and making the cloud effective. Therefore this research aims to provide an enhanced approach to ensure the storage of data securely in cloud computing with the use of Residue Number System (forward and reverse conversions) and Advanced Encryption Standard (AES-256) algorithm.

## 1.2 Statement of Problem

Cloud storage is a form of networked data storage where data files are stored on multiple virtual servers. The servers used for cloud storage are hosted by third party companies who operate large data centres (Papri, Vishal, & Pravin, 2017). The cloud providers are maintaining the user's data in the cloud environment. Outsourcing data to cloud server implies that data is out control of users (Rady, Abdelkader, & Ismail, 2019). This may cause users' hesitation since the outsourced or stored data usually contain vital,

valuable and sensitive information. Many people use cloud for data storage and one of the most cogent issues is the security and privacy of data because the entire data reside in a collection of interconnected resource pools that allow the virtual machine to access those data. Security threats faced by cloud data storage can come from different sources. A cloud service provider can be self-interested, untrusted and possibly malicious. There may also exist an attacker or intruder who compromises a number of cloud data storage servers while remaining undetected by the cloud service provider (Rady et al., 2019). The present methods used for encrypting the files in cloud are not highly efficient. These methods are mostly single level cryptography which are easy for hackers or intruders to gain unauthorized access to the data. Therefore, while outsourcing data to cloud server, effective measures (multilevel cryptography) need to be put in place so as to ensure its security and also control how the data is accessed.

Several works have been done on data security and privacy in cloud computing including (Kumar, Shekhar, & Sing, 2018). It made use of a binary tree for data security in cloud storage as each node holds an alphabet, number, and special character, and each link has a binary value 0 or 1. Though this approach is simple, but it is still easy for hackers and intruders to crack or decrypt. Also the binary tree used did not cater for some of the special characters and no encryption algorithm was used. Hence the need for an enhanced data security technique for cloud storage. This research provides an enhanced approach with multilevel cryptography by using ASCII, RNS and AES-256 for the reliability of cloud

environment. The data will be encrypted before it is being stored or outsourced. If all data stored in the cloud is in an encrypted format that would effectively solve issues like data security, privacy, confidentiality and third party control.

### 1.3 Aim and Objectives

The aim of this research is to develop an enhanced data security system for cloud storage.

To achieve the above aim, the following objectives are to:

i. Formulate a multilevel model for encryption and decryption process using Residue Number System and Advanced Encryption Standard.
ii. Implement the multilevel model in (i).
iii. Deploy the system on a cloud platform.
iv. Evaluate and compare the performance and robustness of the system for cloud storage.

### 1.4 Scope of the Study

This research focuses on data security in cloud data storage. It is limited to the use of cryptography in Residue Number System and a symmetric encryption algorithm (AES-256) to improve on the security and privacy level in the cloud environment. The implemented data security system is limited to text data type only in order to protect the

confidentiality and integrity of the data stored on the cloud. However, this data security could be adopted by individuals and organizations so as to provide high level of security.

## 1.5 Significance of the Study

The significance of this research is that the security, integrity and confidentiality of the data are ensured by providing authorized access to the data stored in the cloud. The enhanced approach tagged RNS-AES is not easy for hackers, intruders or unauthorized users to gain access to the users' data stored on the cloud. If by mistake, a hacker tries to guess the secret AES key used for encryption and decryption, reverse conversion with the correct moduli set is still needed to be performed in order to get the actual data. Also, due to the mathematical complexity, the third party unauthorized access will not be very easy. It is expected that using multilevel cryptography provides more security for Cloud Storage than using single level cryptography. Hence this enhanced approach secures storage of data in cloud computing.

## 1.6 Definition of Terms

**Cryptography** – It is a technique of transformation and transmission of confidential data in an encoded manner so that only permitted and intended users can obtain or work on it. It is a Greek word where "crypto" means hidden and "graphy" means writing, so cryptography means "hidden" or "secret" writing (Anu, Shree & Ahlawat, 2017).

**Data security** – It relates to data protection against unauthorized access. The aim of data security is to safeguard or protect any type of data or information in order to ensure privacy (Izhar et al., 2017).

**Cloud storage** – It is a cloud computing model in which data is stored on remote servers accessed from the internet, or "cloud". It is maintained, operated and managed by a cloud storage service provider on a storage servers that are built on virtualization techniques (What is Cloud Storage?, n.d.).

**Encryption** – It is the conversion of original data i.e. Plain text into a Cipher text (Izhar et al., 2017).

**Decryption** – It is the conversion of Cipher text into human readable form i.e. Plain text.

**Data Integrity** – It refers to the confidence that the data stored in the cloud is not fiddled or breached by unauthorized parties (Kumar, Raj, & Jelciana, 2018).

**Data Confidentiality** – It refers to data privacy where the owner's sensitive data is not revealed to unauthorized parties on any occasion (Kumar, Raj, & Jelciana, 2018).

# CHAPTER TWO
# LITERATURE REVIEW

## 2.1 Overview of Cloud Computing

Cloud computing is an internet based computing technology where virtual shared servers provide software, infrastructure, platform, devices and other resources for hosting customers on a pay as per you use basis (Khan, Shekhar, & Khare, 2018). All information that a digitized system has to offer is provided as a service in the cloud computing model. Users can access these services available on the "Internet cloud" without having any previous know-how on managing the resources involved. It serves most of the hardware and software facilities required for companies for storing, creating, managing, running consumer applications on cloud in lease or rent basis, it provides resources as a service to multiple consumers by virtualization. This technology helps many Information Technology (IT) organizations to start up business without huge economic barriers, slowly move to leading organization in the industry (Sirisha & Basha, 2018).

The word 'Cloud Computing' comes from two words, that is 'Cloud' which refers to the internet and 'Computing' which means technology based on computers (Sharma, Gupta, & Khatri, 2019). Here, Internet is a storage or warehouse where the virtualized resources are stored which then are provided as services. Cloud computing is as a result of usage of technology for day to day activities through internet. It came into the foreground as there were advances in virtualization, distributed computing with server clusters and increase in the availability of broadband internet access (Kanagavalli & Vagdevi, 2014).

Various cloud service providers are Amazon, Google, IBM, Microsoft, and Salesforce.com, which offer their cloud infrastructure for services (Sirisha & Basha, 2018). The cloud computing buzz began way back in 2006 with the launch of Amazon EC2, gained traction in 2007 and currently characterized by having an on demand access to elastic sources through a tenancy model (Kanagavalli & Vagdevi, 2014).


***Figure 2.1: Cloud Computing (Chinthagunta & Vidyamadhuri, 2017)***

Cloud Computing has several attractive benefits for organizations and end users which are:

**Reduced Cost**: Cloud computing allows a smaller effort and initial cost to start an IT company. Multiple consumers in the world share cloud computing services (Sirisha & Basha, 2018). Due to large numbers of consumers, it reduces service costs. It charges the

amount depending on the use of infrastructure, platform and other services, which helps consumers to lower costs by specifying the exact requirements. Companies can easily increase or decrease their service demand based on their company's market performance.

**Scalability and Flexibility**: Cloud computing can help businesses get started with a small set-up and grow fairly quickly to a large condition, then scale back if necessary. Furthermore, cloud computing flexibility enables companies to use additional resources at peak times to meet consumer demands. Cloud computing is also ready to fulfill any peak time requirement by setting up servers, storages, etc. with high capacity. This facility helps consumers meet all kinds of requirements regardless of project size.

**Backup and Recovery**: Since all the data is stored in the cloud, it is relatively much easier to back it up and restore it than to store the same on a physical device. It also has many techniques to recover from any kind of disaster; most cloud service providers adopt the most efficient and new techniques to cope with any kind of disaster. Cloud providers can get any type of technical and other support very quickly, regardless of their geographical limitations, than any individual organizations set up.

**Broad Network Access:** Cloud services are delivered via an open network (Internet), which can be accessed anywhere in the world at any time (Sirisha & Basha, 2018). Various devices such as mobile phones, laptops, Personal Digital Assistants (PDAs) etc. with different platforms can access these facilities. By using their mobile devices, consumers can access their files and other applications from anywhere at any time. This has increased cloud computing technology's adoption rate.

**Multi-sharing**: Cloud Computing provides services to single and multiple users through the use of virtualization and multi-tenancy by sharing architecture and other applications over the Internet. Multiple users and applications can work more efficiently with cost reductions by sharing common infrastructure with the cloud working in a distributed and shared mode.

**Measured Service:** Cloud schemes automatically monitor and optimize resource use by leveraging metering capabilities at a suitable level of abstraction for the type of service (e.g., storage, processing, bandwidth, and active user accounts). Use of resources can be tracked, controlled and reported, offering transparency of the service used for both the supplier and the consumer (Kartit et al. 2016).

**Collaboration:** Major projects or applications are delivered through the efforts of multiple working groups of people. Cloud computing offers a convenient way to effectively work together groups of people on a common project or applications.

**Deliver New Services:** Multi-national companies such as Amazon, Google, IBM, Microsoft, Salesforce.com, etc. provide cloud services. These organizations, at the release time itself, can easily deliver any new application or product.

## 2.2 Cloud Computing Service Delivery Models

Cloud service models describe how cloud services are made available to clients. According to the United States (US) National Institute of Standards and Technology (NIST), there are basically three service delivery models of cloud computing which are the

Software as a Service (SaaS), the Platform as a Service (PaaS) and the Infrastructure as a Service (IaaS). These service models are based on modern data centers and provide them as services by enabling consumers to pay for just what they use (pay per usage). Data centers are generally made up of multiple servers connected to each other; and are clustered in thickly populated bands where the possibility of a natural disaster is negligible (Tinankoria & Babak, 2017) .

### 2.2.1 Software as a Service

Software as a Service (SaaS) is the model in which an application is hosted as a service to customers who access it via the Internet (Ewang, 2019). In the SaaS model, the user can access the infrastructure of the provider via an interface. Web browsers are the most widely used interfaces. In this model a single instance on the service provider's end supports multiple access instance on the client's side. A major advantage of this model is that software licensing expenses are not borne by the consumer. On the service side, however, licensing costs are greatly reduced as only one software instance is required to support access to multiple customers or consumers. Customers do not have the rights for handling the cloud infrastructure in this model (Alajmi, Ali, Adzhar, & Mohammed, 2018).

SaaS greatly enhances developers' ability to scale their software on demand and meet customer's requirements. Nevertheless, SaaS systems are subject to issues regarding confidentiality, safety and reliability (Ewang, 2019). SaaS implementations vary from earlier distributed computing approaches because SaaS was explicitly designed to use web

tools, such as the browser, which makes them native to the internet. It was also designed with a multitenant back end in mind that enables the use of an application by multiple customers.

Since the software is managed at a central location, customers can access their applications wherever they have web or internet access (Ewang, 2019). Some of these applications include:

- Online word processing and spreadsheet tools
- Customer Relationship Management (CRM)
- Workflow management
- Video conferencing
- IT service management
- Web content delivery services (Salesforce CRM, Google Docs, etc.)

Another group of SaaS are Web 2.0 applications such as: metadata management, social networking, blogs, wiki services, and portal services.

### 2.2.2 Platform as a Service

Platform as a Service (PaaS) is a form of cloud computing which enables the consumer to develop and deploy their applications on the service provider's infrastructure (Nishit et al., 2018). A development platform is delivered as a product on the platform as a service. The platform allows customers to build their applications running on the

infrastructure of the service provider. The platform supports programming languages like Python, Net and Java among other support tools that enable custom applications to be created by clients (Alajmi, Ali, Adzhar, & Mohammed, 2018). PaaS services include application design, development, testing, deployment, and hosting. Other services include team collaboration, web service integration, database integration, security, scalability, storage, state management, and versioning. While the customer has no control over the basic infrastructure such as storage and other hardware, they have control over the type of end-user application that they have deployed. With PaaS, developers can build Web applications without installing any tools on their computers and then deploy those applications without any specialized systems administration skills (Kartit et al., 2016). Examples of platform as a service model include Google App Engine, Force.com and Microsoft Azure.

#### 2.2.2.1 Heroku cloud

Heroku is a cloud platform as a service (PaaS) that supports some programming languages. It is one of the cloud platforms that have been developed since June 2007. It supports programming languages like Java, PHP, Python, Ruby, Go, Scala, and Clojure. It is based on a managed container system with integrated data service and more powerful ecosystem to develop and run modern apps (Bih-Hwang, Ervin, & Muhammad, 2018). Heroku is an integration of tools and developer tools which runs on Dynos app. Dynos are the heart of the Heroku platform. They make it easy to develop and run apps more flexible

and measurable. They also make it easy to manage the infrastructure, so it can be used to run great applications. With Heroku, developers can build system using the programming languages that they like and deploy easily.

### 2.2.3 Infrastructure as a service

The Infrastructure as a service (IaaS) model provides infrastructure components to clients or consumers. Servers, storage systems, networking equipment, data center space etc. are pooled and made available to handle workloads (Ewang, 2019). The customer's capacity is to lease power, space, networks, and other basic computing tools where the customer is able to deploy and run arbitrary code that can include operating systems and applications. The customer does not operate or monitor the underlying cloud infrastructure, but has control over operating systems, space, applications deployed and probably select network components (e.g. firewalls, load balancers, virtual machines, storage, networks, etc.).

In IaaS, a third-party service provider hosts services such as hardware, application, network, data, etc., on behalf of consumer and charges on a usage basis. It also liberates its users from time-consuming and side lined tasks like system maintenance, backups, and system recovery planning (Nishit et al., 2018). Some examples of IaaS are Amazon EC2, Amazon S3, Go Grid, Rackspace, IBM Smart Cloud etc. Amazon Web Services is one of largest IaaS providers. The IaaS model is the simplest for cloud service providers to provision (Ewang, 2019).

Understanding how the three models relate is essential. IaaS forms the foundation of all the models with PaaS following onto which SaaS builds itself on (Alajmi et al., 2018). IaaS offers the client with more security controls. The vulnerability in this model occurs in the virtualization techniques. Virtualization security is, therefore, essential while working with IaaS. The Figure 2.2 indicates how the service delivery models relate and the subscriber (consumer) as well as service provider functions in the different models.

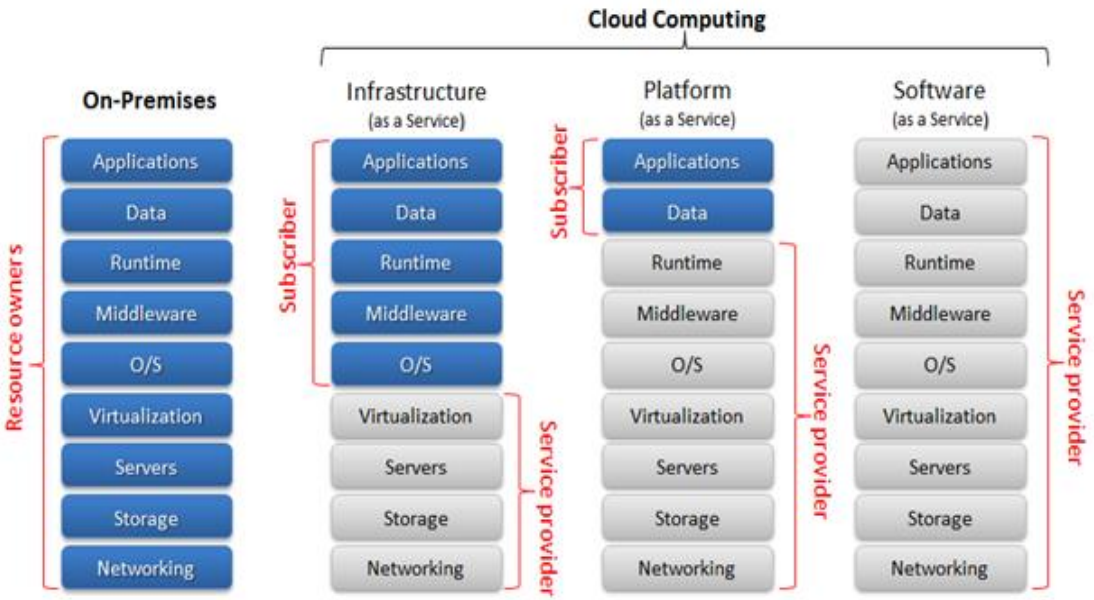


***Figure 2.2: Cloud Computing Service Delivery Models (Alajmi, Ali, Adzhar, & Mohammed, 2018)***

## 2.3 Cloud Computing Deployment Models

Cloud deployment models indicate how the cloud services are made available to users or consumers. It specifies the relationship between the provider and consumer. According to NIST, there are four deployment models in cloud computing.

### 2.3.1 Private cloud

A private cloud service refers to a cloud solution where the infrastructure is provisioned and hosted on a private platform in the customer data center exclusively for use by a single organization (Ewang, 2019). It is dedicated to a particular organization and not shared with other organizations. The organization often acts as a cloud service provider to internal business units that obtain all the benefits of a cloud without having to provision their own infrastructure. A private cloud may be owned, managed, and operated by the organization, a third party, or a combination (Ewang, 2019).

It is also known as Internal Cloud; includes a cloud-based environment that is distinct and secure in which only the specified user can work or operate. This allows only the approved users and gives the company more and more direct control of their data. The private cloud system is similar to the more traditional model of individual local access networks (LANs) used by companies in the past but with the added virtualization advantages (Narayana, Sailesh, & Jayashree, 2017).

A private cloud gives many benefits over a public cloud to an enterprise. The company has greater leverage over the cloud's various resources (Ewang, 2019). In the

private cloud, scalable resources and virtual applications provided by the cloud vendor are pooled together and available for cloud users to share and use. Only the organization and designated stakeholders may have access to operate on a specific Private cloud (Kartit el., 2016). This differs from the public cloud in that the company itself handles all cloud services and applications, close to the functionality of the Intranet. The private cloud is known as the most secured cloud as it data processes are controlled and managed in the company exclusive of any limitation of bandwidth network, security disclosures, and legitimate requirements (Tinankoria & Babak, 2017). Examples of Private cloud are Eucalyptus, Ubuntu Enterprise Cloud, Amazon VPC (Virtual Private Cloud), VMware Cloud Infrastructure, Microsoft ECI data Center.

### 2.3.2 Public cloud

Public clouds are owned and operated by third parties; they offer superior economies of scale to consumers, because infrastructure costs are spread across a variety of users, offering the single customer or consumer an appealing "pay-as-you-go" model at low cost (Narayana, Sailesh, & Jayashree, 2017). All customers share the same infrastructure pool with the cloud service being operated and enabled by restricted configuration, security protections and availability variances. One of the benefits of a public cloud is that they may be larger than an enterprises cloud, thus providing the ability to scale seamlessly, on demand.

A public cloud is one that renders the database services accessible over the Internet to the general public or to a large industry community. The infrastructure is not owned by the user, but by an organization providing cloud services. The customer has no knowledge or power of the location of the cloud services (Ewang, 2019). Most organizations share the core infrastructure, but the information & software use of each entity is logically separated so that only approved users are allowed access. Public clouds are less secure than the other cloud models because it places an additional burden of ensuring all applications and data accessed on the public cloud are not subjected to malicious attacks. Public cloud service is generally cheaper as there is little or no capital expenditures needed (Ewang, 2019). The accepted examples of public clouds are Amazon Elastic Cloud Compute, Google App Engine, Blue Cloud by IBM and Azure Services Platform by Windows.

### 2.3.3 Hybrid cloud

According to The National Institute of Standards and Technology (NIST), "a hybrid cloud is a combination of public and private clouds bound together by either standardized or proprietary technology that enables data and application portability." The goal is to combine services and data from a variety of cloud models to create a unified, automated, and well-managed computing environment (Ewang, 2019). With a Hybrid Cloud, service providers can utilize third party Cloud Providers in a full or partial manner thus increasing the flexibility of computing. Combining public services with private clouds and the data center as a hybrid is the new definition of corporate computing. Not all companies that use

some public and some private cloud services have a hybrid cloud. Rather, a hybrid cloud is an environment where the private and public services are used together to create value.

Hybrid cloud can also be known as an integrated cloud service utilizing both private and public clouds to perform distinct functions within the same organization (Narayana et al., 2017). It permits the user to increase the capacity or the capability by aggregation, assimilation or customization with another cloud package/service. In a hybrid cloud, the resources are managed and provided either in-house or by external providers. Hybrid cloud service providers are Microsoft Hybrid Cloud (Azure), VMware Hybrid Cloud, Amazon Web Services (AWS) Cloud, Rackspace Hybrid Cloud, EMC Hybrid Cloud and HP Hybrid Cloud.

### 2.3.4 Community cloud

The NIST describes the community cloud as "a cloud infrastructure that is shared by several organizations and supports a specific community, such as healthcare, that has shared concerns around mission, policy and compliance considerations." The primary goal of a community cloud is to have participating organizations realize the benefits of a public cloud, such as shared infrastructure costs and a pay-as-you-go billing structure, with the added level of privacy, security, and policy compliance usually associated with a private cloud (Ewang, 2019). The community cloud infrastructure may be provided on-premises or at a third party's data center, and may be managed by the participating organizations or a third party

A community cloud falls between public and private clouds with respect to the target set of consumers. It is somewhat similar to a private cloud, but the infrastructure and computational resources are exclusive to two or more organizations that have common privacy, security, and regulatory considerations, rather than a single organization (Narayana et al., 2017). Academic clouds are an example of community cloud.

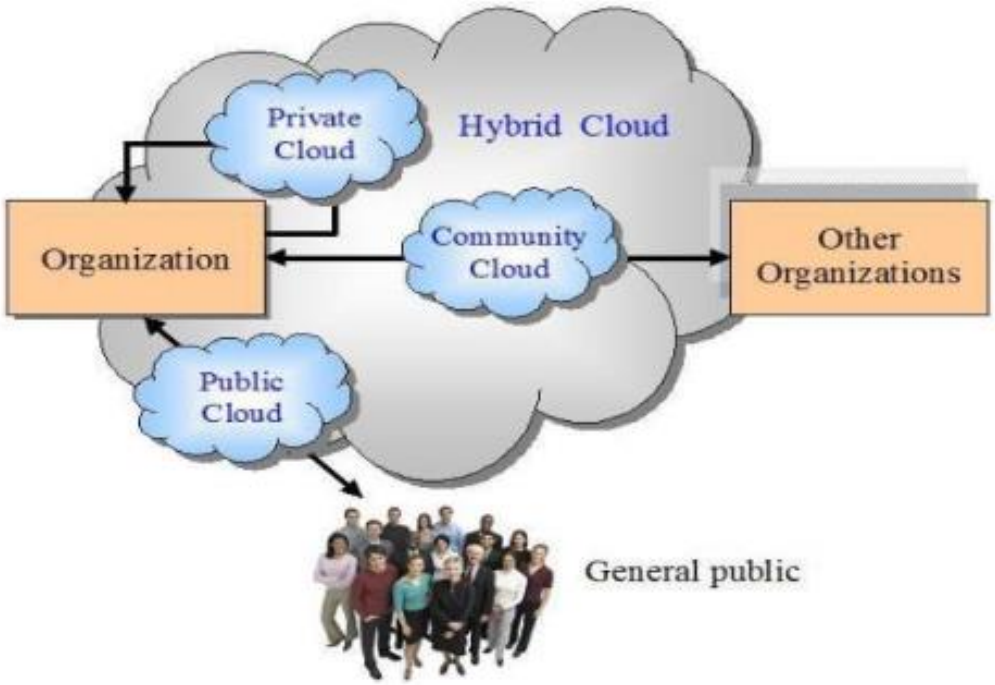


***Figure 2.3: Cloud Computing Deployment Models (Chinthagunta & Vidyamadhuri, 2017)***

## 2.4 Security Challenges in Cloud Computing

Cloud computing offers great advantages over other traditional IT solutions but it raises serious security concerns (Khandelwal & Saini, 2019). As it is expanding its wings in all the directions of technology it is also increasing the risks of security. Security is one of the main concerns rising above the flexible and interesting services provided by cloud computing (Nishit et al., 2018). In addition, when determining whether to move data, software, and other related resources to cloud environments, security and privacy are important considerations for an organization (Al-assam, Hassan, & Zeadally, 2019). Service agreements between clients and Cloud Service Providers (CSPs) tend to include details on how to access and utilize cloud services, service duration, and data storage and management when the contract ends. However, the main challenge is how to guarantee that the data is accessible by authorized users only. Data security is the most tedious job in cloud computing while analyzing these difficulties. According to a survey carried out by Khandelwal & Saini (2019), over 72% of Chief Technical Officers thought that the main reason why cloud computing services were not used was information security and privacy issues. When data owners decide to use the cloud environment, they rely entirely on third parties to make decisions about their data. Therefore, it is very important for data owners to have the right technologies or methods to prevent CSPs from utilizing such data without their permission. Both technical and nontechnical methods have to provide effective means to fulfil this goal (Al-assam et al., 2019). The following are some of the critical issues or challenges in cloud computing.

- Data Breaches and Loss
- Account or Service Hijacking
- Denial of Service
- Malicious Insiders
- Hypervisor Vulnerabilities

### 2.4.1 Data Breaches

Data breach is defined as the leakage of sensitive customer or organization data to unauthorized user (Kazim & Zhu, 2015). Data breach from organization can have a huge impact on its business regarding finance, trust and loss of customers. This may happen accidently due to flaws in infrastructure, application designing, operational issues, insufficiency of authentication, authorization, and audit controls (Kazim & Zhu, 2015). Moreover, it can also occur due to other reasons such as the attacks by malicious users who have a virtual machine (VM) on the same physical system as the one they want to access in unauthorized way. Multiple users and organizations from different parts of the globe share the cloud environment; their precious information is stored in one location (Khandelwal & Saini, 2019). Any break or problem on cloud may expose these sensitive data to the users of other organizations sharing same storage. Because of multi-tenancy, customers using different applications on virtual machines could share same database and any corruption event that happens to it is going to affect others sharing the same database.

Data breach was even one of the popular attacks before the introduction of cloud computing and after the introduction of cloud computing cases of data breaches increased exponentially (Nishit et al., 2018). Apple's iCloud users faced a data leakage attack recently in which an attempt was made to gain access to their private data. Such attacks have also been done at other companies cloud such as Microsoft, Yahoo and Google. An example of data breach is cross VM side channel attack introduced by (Zhang, Juels, Reiter, & Ristenpart, 2012) that extracts cryptographic keys of other VMs on the same system and can access their data.

### 2.4.2 Data loss

Data loss is the second major cloud security issue which means a data loss that occur in any device. It happens when data may be logically or physically detached from the organization or user either unintentionally or intentionally (Rex & Britto, 2015). Data loss is a sensitive issue for any organization, like data breach, and can have a devastating effect on its business (Kazim & Zhu, 2015). It is mostly caused by malicious hackers or attackers, data corruption, data deletion, loss of data encryption key, faults in storage system, or natural disasters. In 2013, 44% of cloud service providers were subjected to brute force attacks resulting in data loss and data leakage. Similarly, cloud applications were also targeted at malware attacks that resulted in data destruction.

### 2.4.3 Account or service hijacking

Account hijacking involves the stealing of user credentials to get an access to his account, data or other computing services (Kazim & Zhu, 2015). These stolen credentials can be used to access and compromise cloud services. The network attacks including phishing, fraud, Cross Site Scripting (XSS), botnets, and software vulnerabilities such as buffer overflow result in account or service hijacking.

In this kind of attack, the attacker can make a phishing attack to fool the client into giving his/her credentials. This kind of attack is very common in computing world; the attacker can get access to cloud and then can steal sensitive information from the cloud (Nishit et al., 2018). This can lead to the compromise of user privacy as the attacker can eavesdrop on all his operations, modify data, and redirect his network traffic. (Khandelwal & Saini, 2019) Proposed many solutions to avoid account or service hijacking such as:

- Prevents the sharing of credentials by customers.
- Using a two-factor authentication system.
- Monitor all operations in order to detect unauthorized access.

### 2.4.4 Denial of Service

Denial of Service (DOS) attack is nothing special but the simple attack that prevents the legitimate users from accessing cloud services due to service failure (Nishit et al., 2018). The attacker overloads the system by increasing utilization of resources (disks,

processors, and network). This disruption of service can cause a huge loss in terms of money to cloud service providers and can give attackers time to prepare for another attack while the service is down.

DOS attacks have been on rise in cloud computing in past 5 years and 81% of customers consider it as a significant threat in cloud (Kazim & Zhu, 2015). They are usually done by compromising a service that can be used to consume most cloud resources such as computation power, memory, and network bandwidth. This causes a delay in cloud operations, and sometimes cloud is unable to respond to other users and services.

### 2.4.5 Malicious Insiders

Malicious insiders are authorized employees who are appointed to manage and maintain cloud by cloud service providers (Khandelwal & Saini, 2019). They often steal or manipulate cloud sensitive data from organizations and pass it on to other organizations that share the same cloud. These malicious insiders can be charged for this unauthorized act and sometimes service providers are unable to take action against them. These kinds of attacks are carried out by attackers who are part of the organizations and can plan an attack from inside the organization, so the attacker can get all the information to the cloud as they have direct access to the data (Nishit et al., 2018). Other categories of malicious insiders involve hobbyist hackers who are administrators that want to get unauthorized sensitive information just for fun, and corporate espionage that involves stealing secret information

of business for corporate purposes that might be sponsored by national governments (Kazim & Zhu, 2015).

## 2.5 Cryptography

Cryptography is a technique of transforming and transmitting confidential data in an encoded way so that only authorized and intended users can obtain or work on it (Anu, Divya, & Seema, 2017). It is a Greek origin word in which “crypto” means hidden and “graphy” means writing, so cryptography means hidden or secret writing. It introduces triads like confidentiality, non-repudiation, integrity and authenticity within ongoing data communication. It is the science of using mathematics for making plain text information into an unreadable cipher text format called encryption and reconverting that cipher text back to plain text called decryption with the set of cryptographic algorithms as illustrated in Figure 2.4 (Akashdeep, GVB, Vinay, & Hanumat, 2016).

Cryptographic techniques are not only helping in keeping data in the computing world but before that also, from the time of Second World War cryptography techniques have evolved and integrated into computing world (Nishit, Tarun, Varun, & Vrince, 2018). It has expanded its roots in every field and now one cannot ignore security of data without implementation of cryptography algorithm. Cryptography is a combination of three types of algorithms. They are:

- Symmetric key algorithm
- Asymmetric key algorithm

- Hashing algorithm.

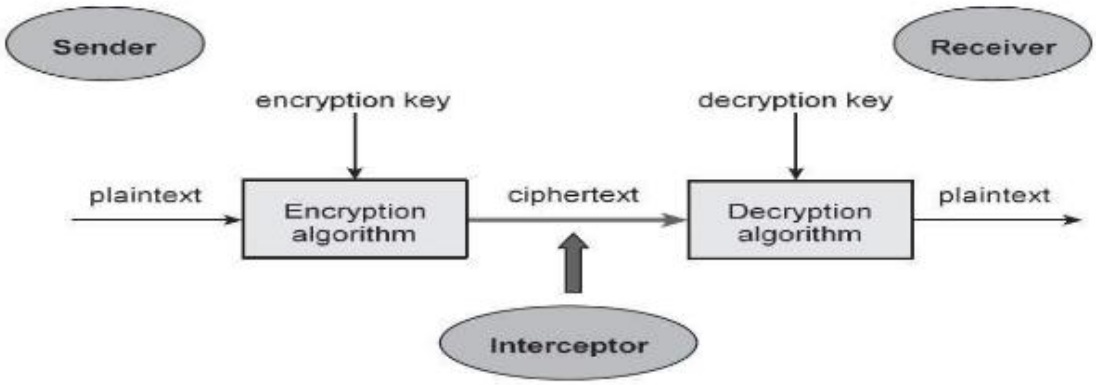


***Figure 2.4: Cryptography process (Anu, Divya, & Seema, 2017)***

The basic components of any cryptographic system are as follows:

a) Plain Text: It is the original intelligible source information or data that is input to algorithms

b) Cipher Text: It is the transformed and changed plain text which is not understandable while merely looking at it. It is obtained after applying encryption algorithm and encryption key over the plain text.

c) Encryption Algorithm: It is a mathematical step-by-step process used for converting plain text into cipher text based on some encryption key. Different examples of such algorithm are AES, DES, blowfish and serpent etc. It is used at sender's side.

d) Decryption Algorithm: It is exactly the reverse mathematical process of used encryption algorithm. It takes cipher text and decryption key to produce original plain text. It is used at receiver's side.

e) Encryption Key: This key is a value that is the lead aspect of the cryptographic system which is either known only to sender or to both sender and receiver. Safe guarding of this key is of great importance for making cryptographic system successful. This key is applied within encryption algorithm to generate cipher text out of plain text.

f) Decryption Key: This key is the value known to receiver and it may or may not be identical to encryption key. It is applied within decryption algorithm to generate the plain text back from received cipher text.

### 2.5.1 Symmetric Key Algorithms

Symmetric algorithms involve a single shared secret key to encrypt as well as decrypt data (Akashdeep, GVB, Vinay, & Hanumat, 2016). It is that type of encryption where both sender and receiver share identical key. It is also called secret-key or single key encryption. It can be implemented using either the technique of block cipher or the technique of stream cipher. Block Cipher encodes plain text block by block on fixed number of 64-bit units, while stream cipher encodes character by character (Nishit Mishra et al. 2018). The advantage of these types of algorithm is that they do not consume too much of computing power and it works with high speed while encrypting and decrypting

the data (Nishit Mishra et al., 2018). They are capable of processing large amount of data and from computing standpoint, they are not very power intensive which results in lower overhead on the systems (Akashdeep et al., 2016). Some examples of symmetric key encryption are AES (Advanced Encryption Standard), DES (Data Encryption Standard), Triple-DES, Blowfish etc.

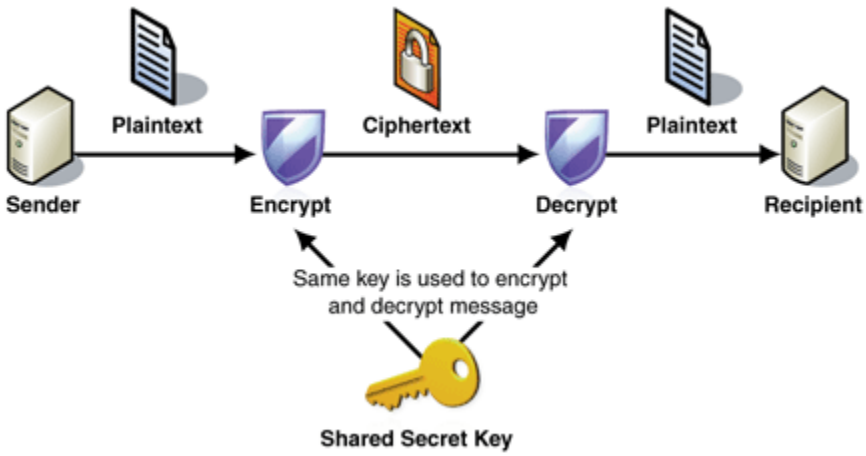


***Figure 2.5: Symmetric Key Encryption (Anu, Divya, & Seema, 2017)***

### 2.5.1.1 Data Encryption Standard

DES stands for Data Encryption Standard and it was developed in 1977. It was the first encryption standard to be recommended by NIST (National Institute of Standards and Technology). DES is 64 bits key size with 64 bits block size (Gowthami & Kousalya, 2017). The key length of this algorithm is 56 bits; however a 64 bits key is actually input.

Since that time, many attacks and methods have witnessed weaknesses of DES, which made it an insecure block cipher. The algorithm is stated in Algorithm 2.1.

**Algorithm 2.1: DES Algorithm**

```
functionDES_Encrypt (N, K) where M = (L, R)
        N ←IP (N)
        For round ←1 to 16 do
                Ki←SK (K, round)
                L ←L xor F(R, Ki)
Swap (L, R)
        End
        Swap (L, R)
        M ←IP^-1(N)
Return N
End
```

### 2.5.1.2 Advanced Encryption Standard

Advanced Encryption Standard (AES), also known as Rijindael named after Joan Daemen and Vincent Rijmen is used for securing information (Papri, Vishal, & Pravin, 2017). It is the new encryption standard recommended by NIST to replace DES. AES is a symmetric block cipher that uses the basic techniques of substitution and transposition. It is characterized as a block cipher such that the data to be encrypted (known as plaintext) is divided into sections called blocks. AES uses a 128-bit block size, in which data is divided into a four-by-four array containing 16 bytes. Since there are eight bits per byte, the total in each block is 128 bits. The size of the encrypted data remains the same: 128 bits of plain

text yields 128 bits of cipher text. AES has key length of 128, 192, or 256 bits, by default 256. This can encrypts data blocks of 128 bits in 10, 12 and 14 rounds depending on the key size (Gowthami & Kousalya, 2017). AES encryption is fast and flexible; it can be implemented on various platforms especially in small devices.

AES brings additional security because it uses a key expansion process in which the initial key is used to come up with a series of new keys called round keys (Understanding AES 256 Encryption, 2019). These round keys are generated over multiple rounds of modification, each of which makes it harder to break the encryption. First, the initial key is added to the block using a XOR ("exclusive or") cipher, which is an operation built into processor hardware. Then each byte of data is substituted with another, following a predetermined table. Next, the rows of the 4x4 array are shifted: bytes in the second row are moved one space to the left, bytes in the third row are moved two spaces, and bytes in the fourth are moved three. The columns are then mixed—a mathematical operation combines the four bytes in each column. Finally, the round key is added to the block (much like the initial key was), and the process is repeated for each round (Understanding AES 256 Encryption, 2019). This yields cipher text that is radically different from the plaintext. For AES decryption, the same process is carried out in reverse. The algorithm is stated in Algorithm 2.2.

**Algorithm 2.2: AES Algorithm**

```
Cipher (byte [] input, byte [] output)
    {
Byte [4, 4] State;
Copy input [] into State [] AddRoundKey
For (round = 1; round < Nr-1; ++round)
        {
SubBytesShiftRowsMixColumnsAddRoundKey
         }
SubBytesShiftRowsAddRoundKey
Copy State [] to output []
  }
```

### 2.5.2 Asymmetric Key Algorithms

Asymmetric key algorithms are those algorithms that use different keys for encryption and decryption (Nishit Mishra et al., 2018). It is also known as public-key encryption (Anu, Divya, & Seema, 2017). It is a class of cryptographic algorithms which requires two separate keys, one of which is secret (or private) and one of which is public. Although different, the two parts of this key pair are mathematically linked. The public key is used to encrypt plaintext or to verify a digital signature; whereas the private key is used to decrypt cipher text or to create a digital signature (Kartit et al., 2016). Private keys are kept private and are used for the decryption of data by the receiver. This eliminates the need for sharing keys.

In cloud computing, asymmetric key algorithms are used to generate keys for encryption. The most common asymmetric key algorithm used in cloud computing is RSA and Diffie–Hellman key exchange.

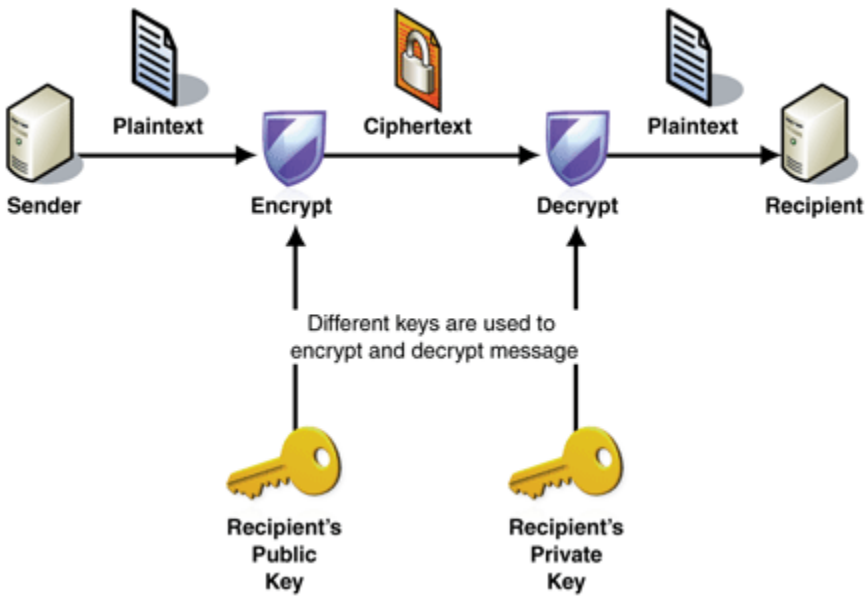


***Figure 2.6: Asymmetric Key Encryption (Anu, Divya, & Seema, 2017)***

### 2.5.2.1 RSA

The most common Public Key or asymmetric algorithm is RSA. This algorithm is named after its inventers; Rivest, Shamir and Adelman (Kartit et al., 2016). In this algorithm the public key is shared and distributed to all which is used for encryption of the data and the other key is a secret key which is also called a private key and used for

decryption. RSA algorithm is used to ensure the security of data in cloud computing (Papri et al., 2017). It is best suited for data traveling to/from Web and Cloud based environments.

RSA uses Euler's Theorem. Data conversion from plain text to cipher text is done using public key by the Cloud service provider and the cipher text to plain text decryption is done by the end user using private key as the Cloud service consumer (Akashdeep et al., 2016). Once the user data is encrypted with the public key, that cipher data can only be decrypted with the corresponding private key only. In this Algorithm, prime numbers are used to generate the public and private keys based on mathematical formulas and by multiplying the numbers together. The algorithm is stated in Algorithm 2.3.

**Algorithm 2.3: RSA Algorithm**

Key Generation: KeyGen(r, s)

**Input**: Two large primes – r, s

Compute n = r .s

F (n) = (r - 1) (r - 1)

Choose e such that gcd (e, f (n)) = 1

Determine d such that e.f = 1 mod f (n)

**Key:**

Public key = (e, n)

Secret key= (f, n)

**Encryption:**

$c = m^e \bmod n$

Where c is the cipher text and m is the plain text.

## 2.6 Residue Number System

Residue Number System (RNS) is an integer number system with the competency to support parallel, carry-free addition, borrow-free subtraction and single step multiplication without unfinished product (Aremu & Gbolagade, 2017). These features enable RNS utilization in Digital Signal Processing (DSP) applications such as convolution, fast fourier transform, digital filtering and image processing. The RNS is a very old number system, it was found 1500 years ago by a Chinese scholar Sun Tzu. Since the last five decades, RNS's features have been rediscovered and thus the interest in this system has been renewed. Researchers have used RNS in order to benefit from its features in designing high-speed and fault-tolerance applications (Younes, 2013).

The fundamental idea of the RNS is based on uniquely representing large binary numbers using a set of smaller residues, which results in carry-free, high-speed and parallel arithmetic. This system is based on modulus operation, where the divider is called modulo and the remainder of the division operation is called residue. The basic notation in RNS is shown in equation (2.1).

$$x_i = X \; mod \; m_i = \langle x_i \rangle_{m_i} \quad ; \quad 0 \leq x_i < m_i \tag{2.1}$$

Each integer in RNS is represented by a set of residues corresponding to a specified moduli set. The main condition is that the moduli within the moduli set should be relatively prime (Younes, 2013) as demonstrated in equation (2.2).

$$X \rightarrow \left(\langle x_1 \rangle_{m_1}, \langle x_2 \rangle_{m_2}, \ldots, \langle x_n \rangle_{m_n}\right); \quad GCD\left(m_i, m_j\right) = 1 \tag{2.2}$$

The RNS uniquely represents any integer *X* that locates in its dynamic range *M*, which is the product of the moduli within the moduli set as shown in equation 2.3.

$$M = \prod_{i=1}^{n} m_i \quad (2.3)$$

Both signed and unsigned integers can be represented in the RNS. For unsigned RNS, the range of the representable integers is

$$< 0 \leq X < M$$

For signed RNS, the range of representable integers is partitioned into two equal intervals

$$0 \leq X < [\frac{M}{2}] \qquad \text{For positive numbers}$$

$$[\frac{M}{2}] \leq X < M \qquad \text{For negative numbers}$$

The main aspect that distinguishes the RNS from other number systems is that the basic arithmetic operations are easy to implement, while operations such as division, root, comparison, overflow, scaling and sign detection are more complicated (Younes, 2013). The RNS is therefore extremely useful in applications involving a large number of addition and multiplication, as well as a minimum number of comparisons, divisions and scaling. In other words, in applications where additions and multiplications are important, the RNS is preferred. Such applications include DSP, image processing, voice processing, cryptography and transformation (Aremu & Gbolagade, 2017).

The main advantage of RNS is the lack of carry propagation between digits, resulting in the need for high-speed arithmetic in embedded processors. Another important feature of RNS is the independence of digits, so there is no propagation of an error in a

digit to other digits, which results in no propagation of errors, thereby providing fault tolerance systems (Younes, 2013). However, in complex-number arithmetic, the RNS can be very effective because it simplifies and reduces the number of multiplications required. All of these features increase the scientific movement towards RNS for DSP applications in particular.

The basic RNS processor's architecture is shown in Figure 2.7. It comprises of three main blocks, namely forward converter, residue arithmetic unit and reverse converter (Ananthalakshmi & Rajagopalan, 2019). The forward converter (binary to residue converter) converts the binary number to *n* equivalent RNS residues, corresponding to the *n* moduli. The *n* residues are then processed using *n* parallel Residue Arithmetic Units (RAUs); each of them corresponds to one modulo. The *n* outputs of these units represented in RNS are then converted back into their binary equivalent, by utilizing the reverse converter (residue to binary converter).

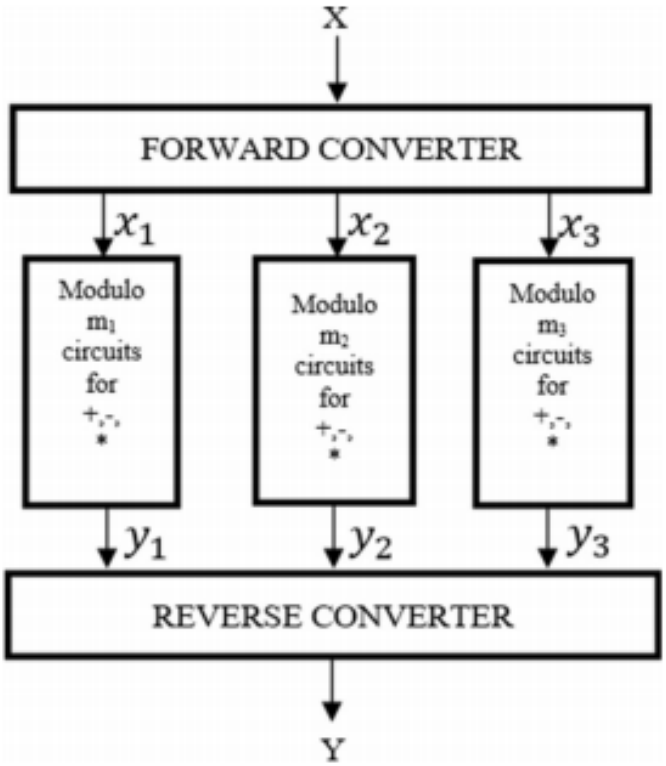


***Figure 2.7 Residue Number System Architecture (Ananthalakshmi & Rajagopalan, 2019)***

## 2.7 Data Conversion in Residue Number System

Data Conversion in the residue number system is generally based on either the Chinese Remainder Theorem (CRT) or the Mixed Radix Conversion (MRC) which can be classified as forward and reverse conversions. The forward conversion involves converting a binary or decimal number into its RNS equivalent while the reverse conversion is the inverse operation, which involves converting RNS number into binary or decimal (Aremu

& Gbolagade, 2017). For triumphant application of RNS, data conversion must be very fast so that the conversion outlay doesn't nullify the RNS advantages.

Every RNS system involves forward and reverse converters that convert weighted numbers into their equivalent RNS representation and vice versa, respectively. The structures of these converters can be memory-based, conventional-based or mix of both (Younes, 2013). The choice is actually determined by the dynamic range required for the application being designed. For applications with small dynamic ranges, such as digital image processing where the range of pixel values is [0,255], the memory-based converters are the most efficient. Contrary, for applications with large dynamic ranges (greater than 22 bits), such as cryptography and some Finite Impulse Response (FIR) filters, the combinational structure of the converters is preferred (Younes, 2013).

### 2.7.1 Forward Conversion

Conversion from decimal to residue number system representation is known as forward conversion and the most direct way to convert from a conventional representation to a residue representation, is to divide by each of the given moduli and then collect the remainders (Aremu & Gbolagade, 2017). This, however, is likely to be a costly operation if the number is represented in an arbitrary radix and the moduli are arbitrary. If on the other hand, the number is represented in radix-2 (or a radix that is a power of two) and the moduli are of a suitable form (e.g.$2^n$-1), then their procedures can be implemented

efficiently (Aremu & Gbolagade, 2017). The residues or remainders of the decimal number (X) are collected using the formula shown in equation (2.4)

$$r_i = |X|_{m_i} \tag{2.4}$$

Where $r_i$ are the residues and $m_i$ are the moduli sets

The forward converters receive weighted integer operands and produces residues. They are parallel architectures, for the different residues, composed of independent units, each one computing one residue based on adders (Amir & Leonel, 2017).

### 2.7.2 Reverse Conversion

Translating from residue representations back to conventional notations of decimal or binary number is called reverse conversion, usually after some residue-arithmetic operations. It is one of the more tricky RNS operations and has been a major, if not the major, limiting factor to a wider use of residue number systems (Aremu & Gbolagade, 2017). The reverse conversion is mainly done for storing the result in the memory or to communicate the obtained result to another system in the network. Since the system or node in the network may not recognize the residue number, binary number representation is suitable for communication purpose. Henceforth the reverse converter is very essential for adopting the RNS in practical applications (Ananthalakshmi & Rajagopalan, 2019). It is the important part of residue number system because the speed efficacy obtained in performing calculation in residue domain should not be degraded while converting it to the binary number system. Hence the design of speed efficient reverse converter has

significance that without speed efficacy, the use of residue number system that is speed and power efficient cannot be used in processors. Therefore it is necessary to have a high speed reverse conversion process. According to Ananthalakshmi & Rajagopalan (2019), there are three main steps to be noted in order to have a high speed reverse conversion process, which are:

- Judicious selection of the moduli set.
- A dynamic range suitable for the application.
- A conversion algorithm compatible with the properties of the selected moduli set.

The main techniques or algorithms for reverse conversion are based on the Chinese Remainder Theorem (CRT) and the Mixed-Radix Conversion (MRC) technique. All other techniques are just variations of these two, which variations occur from either the type of moduli-set chosen or from certain properties that can be easily adapted to suit the particular approach chosen (Aremu & Gbolagade, 2017). Every algorithm has its own advantages and disadvantages. The decision to use any of them is based on the used moduli set, the application being designed and the design's requirements (time, area, power). All reverse conversion methods depend on computing multiplicative inverses (Younes, 2013).

#### 2.7.2.1 Chinese Remainder Theorem

The residue number $\{x_1, x_2, x_3, \dots, x_n\}$ can be converted into the decimal number *X*, according to the Chinese Reminder Theorem (CRT) shown in equation (2.5).

$$X = \langle \sum_{i=1}^{n} M_i \langle M_i^{-1} \rangle_{m_i} x_i \rangle_M \tag{2.5}$$

Where $M = m_1 \times m_2 \times m_3 \times \ldots \times m_n$ , $M_i = \frac{M}{m_i}$, and $M_i^{-1}$ is the multiplicative inverse of $M_i$ relative to modulo $m_i$. The CRT-based converter can be implemented in parallel.

### 2.7.2.2 Mixed Radix Conversion

Mixed Radix Conversion (MRC) is an alternative scheme which does not involve the large modulo M (active range) calculations and also yield a low complexity of O(n) as compared to the CRT whose computational complexity of order $O(n^3)$ (Aremu & Gbolagade, 2017).

The conversion from RNS to binary using Mixed Radix Conversion (MRC) can be formulated as follows: given an *n*-digit binary number $X = (x_1, x_2, x_3, \ldots, x_n)$ in an RNS with the set of relatively prime integer moduli $\{m_i\}_{i=1,2,3,\ldots,n}$ one has to find a set of digits $\{v_1, v_2, v_3, \ldots, v_n\}$, which are the mixed radix digits (MRD), such that equation (2.6) holds true

$$X = v_1 + v_2 m_1 + v_3 m_1 m_2 + \cdots + v_n \prod_{i=1}^{n-1} m_i \tag{2.6}$$

The mixed radix digits $v_i$, $0 \leq v_i < m_i$ can be computed as shown in equation (2.7) to (2.9).

$$v_1 = x_1 \tag{2.7}$$

$$v_2 = \langle (x_2 - v_1) \times \langle m_1^{-1} \rangle_{m_2} \rangle_{m_2} \tag{2.8}$$

$$v_3 = \langle \Big( (x_3 - v_1) \times \langle m_1^{-1} \rangle_{m_3} - v_2 \Big) \times \langle m_2^{-1} \rangle_{m_3} \rangle_{m_3} \tag{2.9}$$

Thus, the MRC is sequential and involves modulo subtractions and modulo multiplication by multiplicative inverses of one modulus with respect to the remaining moduli.

## 2.8 ASCII

American Standard Code for Information Interchange (ASCII) is a 7-bit code, which means that 128 characters (27) are specified. The code consists of 33 non-printable and 95 printable characters and contains all letters, punctuation marks, numbers and control characters (ASCII Codes|Overview of all characters on the ASCII table, n.d.). The term describes fixed character encoding, assigning printable characters such as letters, numbers, punctuation marks, and non-printable ASCII character control codes. The ASCII code can be used to assess the representation of characters by electronic devices, such as personal computers (PCs) or smartphones. The American Standards Association (ASA, now referred to as the ANSI for the American National Standards Institute) had already accepted the ASCII in 1963 and provided binding requirements for how electronic devices should display characters. The ASCII characters are often divided into several groups.

- **Control characters (0–31 & 127)**: Control characters are non-printable characters. They are used to send commands to the PC or printer and are based on telex technology. You may set the line breaks or tabs with these characters. They're just out of use today.
- **Special characters (32–47/58–64/91–96/123–126)**: Special characters contain all printable characters that are not letters or numbers. They include punctuation or technical, mathematical characters. ASCII often contains space (a non-visible but printable character) and thus does not belong to the group of control characters, as one would suspect.

- **Numbers (30–39)**: These numbers contain ten Arabic numerals from 0–9.
- **Letters (65–90/97–122)**: Letters are split into two blocks, the first group containing the uppercase letters and the second group containing the lowercase.

Values are typically expressed in decimal, binary, octal and hexadecimal form in the ASCII code tables (ASCII-American Standard Code for Information Interchange, 2019). The first two are used as the most common number systems for humans and computers. On the other hand, the hexadecimal method has the advantage that it consists of 16 characters (0-9 + A-F) and can represent large numbers with fewer digits than the other two. This way, a byte can always be displayed with a maximum of two digits.

### 2.9 Review of Related works

Kumar, Shekhar, & Sing (2018) developed a data security and encryption technique for providing security to cloud users. They implemented a mechanism that used a binary tree in that each node holds an alphabet, number, and special character, and each link has a binary value 0 or 1. The limitation of the research is that the binary tree created did not cater for all special characters and no encryption algorithm was used. Thus the approach is simple.

Mohd & Prachi (2018) combined RSA encryption algorithm and SHA1 for the better security. RSA is for the encryption and decryption of the data and SHA1 is for generating hash value. It helped to provide security which only the authorized user can access it. The

drawbacks of RSA algorithm made this approach not suitable enough because the operation speed is slow and in terms of memory usage, high RAM is required.

Bih-Hwang, Ervin, & Muhammad (2018) proposed data security in cloud computing using AES under Heroku cloud. Heroku was implemented as a cloud platform, then, they implemented a website as an application to data security. The performance evaluation shows that AES cryptography can be used for data security. Moreover, delay calculation of data encryption shows that larger size of data increases the data delay time for encrypting data.

Jayapandian & Rahman (2017) depicted two efficient encryption algorithms that meet security demand in cloud. In order to maintain quality of service (QoS) and improve customer satisfaction, the authors proposed an efficient algorithm which combines the characteristics of both probabilistic and homomorphic encryption techniques, to provide high level of security. The scheme yields better encryption techniques reduce security attacks, increased throughput and improve the QoS. The limitation is that the implementation of probabilistic algorithm is a little complex and both algorithms consume high memory usage.

Kartit, marraki, Azougaghe, & Mustapha (2016) used both AES (Advanced Encryption Standard) and RSA algorithm. This algorithm suggests the encryption of the files to be

uploaded on the cloud. The existed file on the device was encrypted using AES algorithm. To enhance security; AES key was encrypted using RSA algorithm and was stored in intern server. The limitation of this research is that the download time is greater than the upload time. This is explained by the addition of key recovery time on server.

Karun & Uma (2015) made use of AES encryption algorithm and steganography to secure data in the cloud. The scheme revolves around the problem of data security and with the help of encryption at client side and steganography at server side provides a highly secure model that will not only solve the issue of data safety but also simple in its implementation and hence usage. The limitation of this paper is that the technique of image compression was not added to improve storage.

Belguith, Abderrazak & Rabah Attia (2015) proposed a new lightweight encryption algorithm which consists of combining symmetric algorithm to encrypt data and asymmetric one to distribute keys. This combination helps to benefit from the efficient security of asymmetric encryption and the rapid performance of symmetric encryption while conserving the rights of users to access data by a secured and authorized way. The limitation of this work is that the cloud provider is involved in the key distribution which is not supposed to be so. This work can be enhanced by proposing a new key distribution scheme which aims to give every authorized user the encryption key without the cloud provider interaction.

(Kumar, Jatinder, & Mamta, 2015) Kumar, Jatinder & Mamta (2015) emphasized on username authentication. The authors proposed an authentication and encryption techniques based on a binary tree diagram that contains alphabet values. They gave an algorithm that converts the entered alphabets into binary values according to the position in the binary tree. The limitation of this research is that the binary tree used does not take numbers or special character. Also the data was not compressed after adding the redundancy bits.

Shereek, Muda & Yasin (2014) used RSA algorithm to improve cloud computing security. In RSA, key size decides the strength of the cryptosystem, but the main problem of RSA is increasing key generation time. When large key size numbers are selected, the key generation time is also increased, this problem was solved by applying Fermat's little theorem during key generation process. The method adopted (RSA with Fermat's little theorem) did not solve all the drawbacks of RSA algorithm.

Feng, Chao & Chun (2014) introduced the homomorphic encryption mechanism and proposed a cloud computing data security scheme. The scheme ensures the transmission of data between the cloud and the user safely. This new security solution is fully fit for the processing and retrieval of the encrypted data. It is convenient for users and the third party agency to search date to dispose. The limitation of this research is that fully homomorphic encryption scheme has high computation problem so it needs further study.

# CHAPTER THREE

# RESEARCH METHODOLOGY

## 3.1 System Approach

Virtualization makes cloud computing very cost-effective and has made the research communities very promising. Today, the world of core business processes has a lot of dependency on cloud computing, but because it is internet based it is very vulnerable to attacks. Hence the security issue is of paramount concern. That is why the research community's attention has been drawn to cloud security. This research mainly focuses on the enhancement of data security in cloud storage; in this way, it fulfills the requirement of secure data storing in the cloud. With American Standard Code for Information Interchange (ASCII), the data owner will first convert the data into decimal then cryptography in RNS (forward and reverse conversions) is applied on the text data. Also Advanced Encryption Standard (AES-256) algorithm is implemented in the data security system.

### 3.1.1 Description of model

The method adopted in this research is based on a multilevel model (ASCII, RNS and AES) which involves data encryption and decryption process as shown in Figure 3.1. The combination of RNS and AES-256 makes the encryption and decryption stronger which in turn provides high level of security. ASCII conversion table is used to represent the data in its decimal equivalence.

The encryption scheme entails data conversion (forward conversion) with the use of Residue Number System (RNS) and encryption algorithm (AES-256). The text data to be stored in the cloud storage is represented in its ASCII equivalent or code. Then this ASCII equivalent of the text data is converted to residues with the moduli set **$\{2^n, 2^{2n}-1, 2^{2n}+1\}$** so as to generate the first level of encryption. To further increase the security, the residues are encrypted using Advanced Encryption Standard (AES-256) algorithm so as to generate the second level of encryption.

The decryption scheme entails decrypting the encrypted data with the AES-256 key (inverse AES algorithm) and Chinese Remainder Theorem algorithm (reverse conversion). The cipher text is read from the cloud storage and then decrypted with the AES key to generate the first level of decryption. Then the residues are converted to decimal using Chinese Remainder Theorem with the moduli set **$\{2^n, 2^{2n}-1, 2^{2n}+1\}$** so as to generate the second level of decryption. Lastly, the decimal values are converted to the ASCII characters so as to get the plain text.

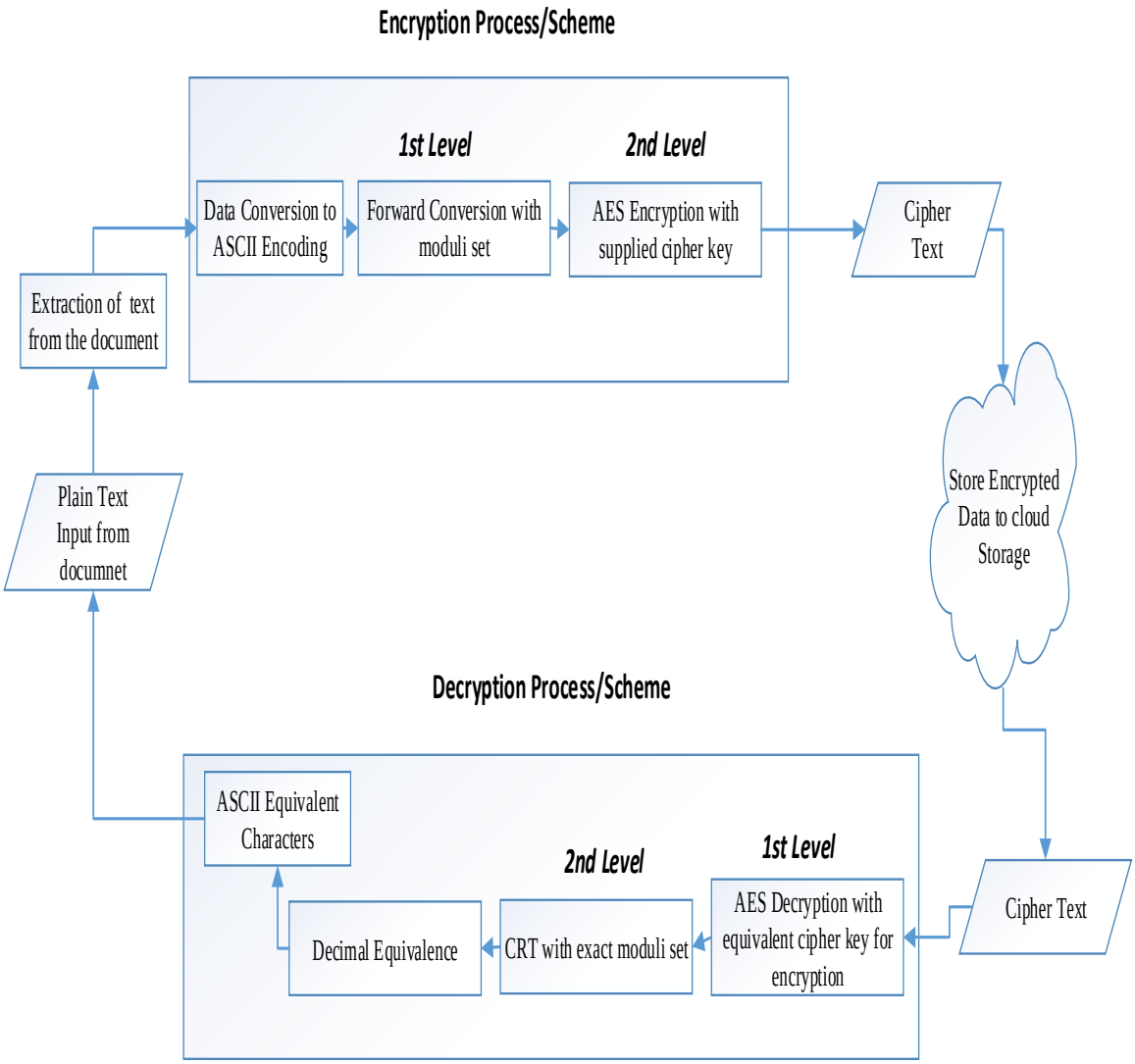


***Figure 3.1: System model***

### 3.2 ASCII Table Formation

The ASCII table comprises of binary, decimal, octal and hexadecimal values for each of the characters. The decimal value adopted and used for this project work is shown in Table 3.1. The alphabets, numbers and special characters were converted to its decimal equivalence.

***Table 3.1: ASCII Table***

| **ASCII symbol** | **Decimal value** | **Group** |
|---|---|---|
| SP | 32 | **Special Characters** |
| ! | 33 | |
| “ | 34 | |
| # | 35 | |
| $ | 36 | |
| % | 37 | |
| & | 38 | |
| ‘ | 39 | |
| ( | 40 | |
| ) | 41 | |
| * | 42 | |
| + | 43 | |
| , | 44 | |
| - | 45 | |
| . | 46 | |
| / | 47 | |
| : | 58 | |

| | | |
|---|---|---|
| ; | 59 | |
| < | 60 | |
| = | 61 | |
| > | 62 | |
| ? | 63 | |
| @ | 64 | |
| [ | 91 | |
| \ | 92 | |
| ] | 93 | |
| ^ | 94 | |
| _ | 95 | |
| ` | 96 | |
| { | 123 | |
| \| | 124 | |
| } | 125 | |
| ~ | 126 | |
| A | 65 | **Uppercase Letters** |
| B | 66 | |
| C | 67 | |
| D | 68 | |
| E | 69 | |
| F | 70 | |
| G | 71 | |
| H | 72 | |
| I | 73 | |
| J | 74 | |
| K | 75 | |

| | | |
|---|---|---|
| L | 76 | |
| M | 77 | |
| N | 78 | |
| O | 79 | |
| P | 80 | |
| Q | 81 | |
| R | 82 | |
| S | 83 | |
| T | 84 | |
| U | 85 | |
| V | 86 | |
| W | 87 | |
| X | 88 | |
| Y | 89 | |
| Z | 90 | |
| a | 97 | **Lowercase letters** |
| b | 98 | |
| c | 99 | |
| d | 100 | |
| e | 101 | |
| f | 102 | |
| g | 103 | |
| h | 104 | |
| i | 105 | |
| j | 106 | |
| k | 107 | |
| l | 108 | |

| | | |
|---|---|---|
| m | 109 | |
| n | 110 | |
| o | 111 | |
| p | 112 | |
| q | 113 | |
| r | 114 | |
| s | 115 | |
| t | 116 | |
| u | 117 | |
| V | 118 | |
| W | 119 | |
| X | 120 | |
| Y | 121 | |
| Z | 122 | |
| 0 | 48 | **Numbers** |
| 1 | 49 | |
| 2 | 50 | |
| 3 | 51 | |
| 4 | 52 | |
| 5 | 53 | |
| 6 | 54 | |
| 7 | 55 | |
| 8 | 56 | |
| 9 | 57 | |

### 3.3 Encryption Process

The steps involved in the encryption process are as follows:

1. Firstly, the text file (data) to be stored in the cloud is loaded into the system.
2. After loading the data, it is represented in its decimal equivalence of the ASCII table.
3. The decimal equivalence of the text data was converted to residues (forward conversion) with the moduli set **{$2^n$, $2^{2n}$-1, $2^{2n}$+1}** so as to generate the first level of encryption for the concerned text file.
4. The next step in the encryption process is application of the Advanced Encryption Standard (AES-256) algorithm on the residues so as to generate second level of encryption for the file.
5. The last step in the encryption process is to store the cipher text generated from the above steps in the cloud database.

#### 3.3.1 Forward Conversion

The forward converter is designed to convert from decimal number system to RNS using the moduli set $\{2^n, 2^{2n}-1, 2^{2n}+1\}$. Given the moduli set$\{2^n, 2^{2n}-1, 2^{2n}+1\}$ $m_1 = 2^n, m_2 = 2^{2n}-1,\ m_3 = 2^{2n}+1$. The decimal value of each character in the text data is converted to residues using the formula in equation (3.1).

$$r_i = |X|_{m_i} \tag{3.1}$$

Where $r_i$ are the residues and $m_i$ are the moduli sets

Moduli set = **{$2^n$, $2^{2n}$-1, $2^{2n}$+1}**

If n=2, moduli set = {4, 15, 17}

Dynamic range M is obtained as a result of the multiplication of $m_1$, $m_2$ and $m_3$ as shown in equation (3.2)

$$\prod_{i=1}^{3} m_i = m_1 * m_2 * m_3 \quad (3.2)$$

The ASCII table in Table 3.1 shows the decimal equivalence of alphabets, numbers and special characters. The forward converter was tested with two decimal numbers from the ASCII table and their results are as follows;

i. a = 97 from the ASCII table, X = 97, $m_1 = 4,\ m_2 = 15, m_3 = 17.\ r_1 = |97|_4 = 1,\ r_2 = |97|_{15} = 7,\ r_3 = |97|_{17} = 12.$ Therefore $r_i$ = {1, 7, 12}. This implies $|97|_{4,15.17} = \{1,7,12\}$

ii. @ = 64 from the ASCII table, X = 64, $m_1 = 4,\ m_2 = 15, m_3 = 17.\ r_1 = |64|_4 = 0,\ r_2 = |64|_{15} = 4,\ r_3 = |64|_{17} = 13.$ Therefore $r_i$ = {0, 4, 13}.

The conversion continues for each character present in the text data.

### 3.3.2 AES Encryption Algorithm

Step 1 Residues from forward conversion

Step 2 Generate Random Key of 32bytes

Step 3 Initialize number of rounds Nr, Nr= 14

Step 4 Initialize counter i (i = 1)

Step 5 If $i \leq Nr-1$ then

$i = i + 1$

Sub bytes

Shift rows

Mix columns

Add round key

Step 6 else

Step 7 Sub bytes

Step 8 Shift row

Step 9 Add round key

Step 10 Cipher text generated

### 3.4 Decryption Process

The decryption process is quite familiar to the process of encryption. All the steps mentioned in the encryption process are to be performed in a reverse chronological order starting from the last step and all the way up to the first step. The steps involved in a decryption process are:

1. The encrypted data file (cipher text) is retrieved, read and loaded into the system
2. Implement the AES-256 algorithm of decryption on the text file so as to generate the first level of decryption for the file.

3. Each of the residues present in the file is converted to its decimal equivalence using Chinese Remainder Theorem algorithm (reverse conversion) with the moduli set **($2^n$, $2^{2n}$-1, $2^{2n}$+1)** so as to generate the second level of decryption.
4. The next step is to convert the decimal equivalence to its corresponding ASCII in order to get the plain text.
5. The plain text generated from decrypting the cipher text file is displayed.

### 3.4.1 AES Decryption Algorithm

Step 1 Cipher text from cloud storage

Step 2 Generate Random key of 32bytes

Step 3 Initialize number of rounds $N_r$, $N_r$= 14

Step 4 Initialize counter i (i = 1)

Step 5 If i ≤ Nr-1 then

i = i + 1

Inverse Shift rows

Inverse Sub bytes

Inverse Mix columns

Inverse Add round key

Step 6 else

Step 7 Inverse Shift rows

Step 8 Inverse Sub bytes

Step 9 Inverse Add round key

Step 10 Residues Cipher text

### 3.4.2 Chinese Remainder Theorem (CRT)

The reverse conversion for the decryption process is designed to convert from RNS to decimal using the moduli set $\{2^n, 2^{2n}-1, 2^{2n}+1\}$. The CRT algorithm is used for the reverse conversion of the residues present in the file to their decimal equivalence. Formula used for the CRT algorithm is shown in equations (3.3) and (3.4).

$$X = \langle \sum_{i=1}^{n} M_i \langle M_i^{-1} \rangle_{m_i} x_i \rangle_M \tag{3.3}$$

$$X = |x_1 M_1 M_1^{-1} + x_2 M_2 M_2^{-1} + x_3 M_3 M_3^{-1}|_M \tag{3.4}$$

Dynamic range M is obtained as a result of the multiplication of $m_1$, $m_2$ and $m_3$ as shown in equation (3.5)

$$\prod_{i=1}^{3} m_i = m_1 * m_2 * m_3 \tag{3.5}$$

Given an RNS number X and the moduli set, the CRT is applied as follows;

Let the RNS number be X = $\{x_1, x_2, x_3\}$ with respect to the moduli set $\{2^n, 2^{2n}-1, 2^{2n}+1\}$ where $m_1 = 2^n$, $m_2 = 2^{2n}-1$, and $m_3 = 2^{2n}+1$.

$M_i^{-1}$ is the multiplicative inverse of $M_i$ with respect to the moduli $m_i$. It is demonstrated as shown in equation (3.6)

$$\left| M_i \times M_i^{-1} \right| = 1 \tag{3.6}$$

The CRT converter was tested with the residue: $x_i = \{1, 7, 12\}$ with respect to the moduli set $\{4, 15, 17\}$.

$$M = 4 \times 15 \times 17 = 1020, \qquad M_1 = \frac{1020}{4} = 255, \quad M_2 = \frac{1020}{15} = 68, \quad M_3 = \frac{1020}{17} = 60,$$

$$M_1^{-1} = 3,$$

$$M_2^{-1} = 2, M_3^{-1} = 2$$

$$X = |1 \times 255 \times 3 + 7 \times 68 \times 2 + 12 \times 60 \times 2|_{1020} = 97$$

The decimal value X can be gotten from the equation above which results to 97.

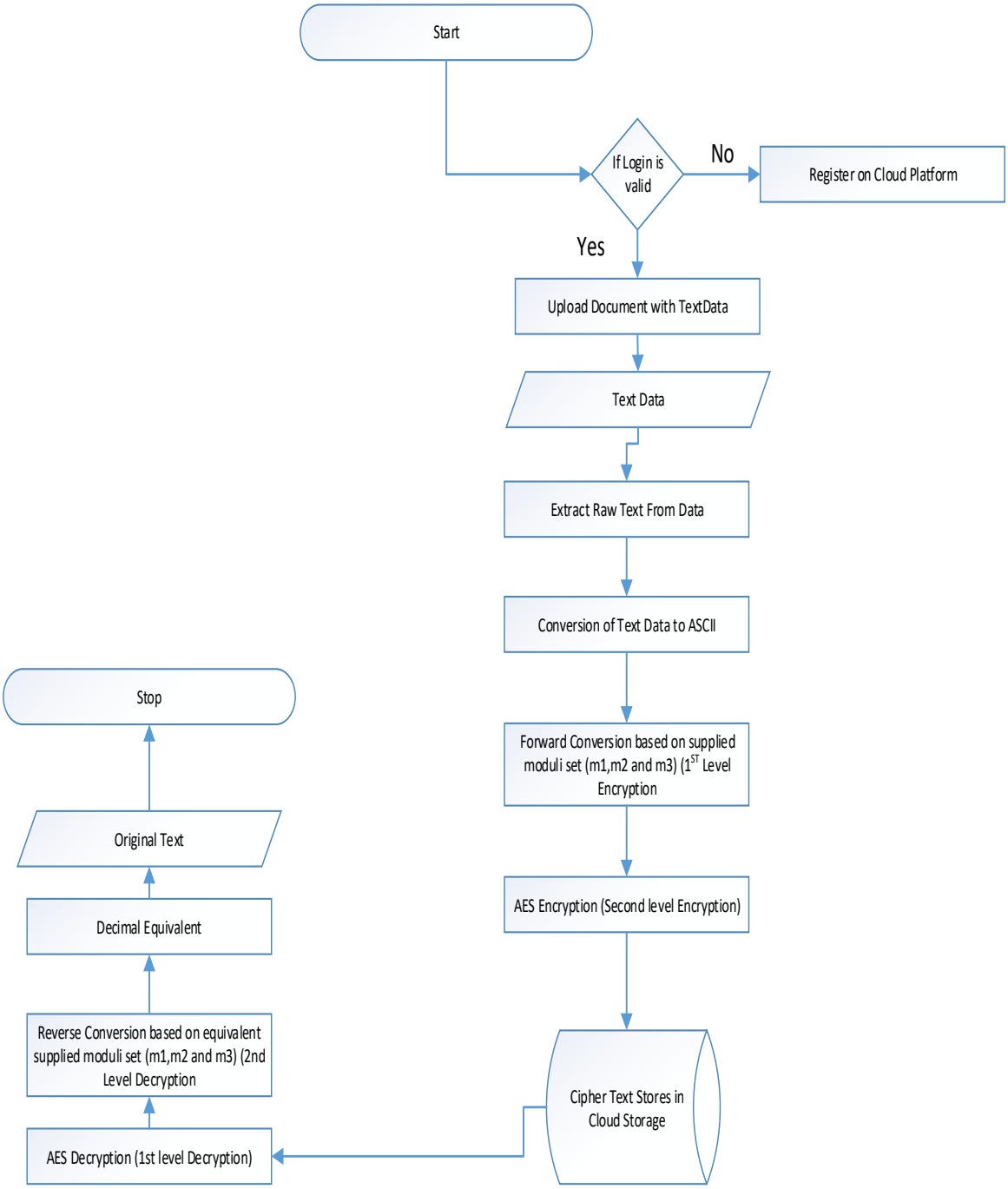


***Figure 3.2: Flowchart of encryption and decryption process***

### 3.5 System Implementation

The developed system (algorithm and graphical user interface) was implemented using PHP and JavaScript programming language. MySQL was used for the database management system. JavaScript was used for the mathematical operations and functions involved. The implementation was also done by the installation of Heroku. Heroku makes use of command line interface (CLI) to manage applications, stipulation add-ons, etc. The system was deployed on Heroku cloud platform so as to demonstrate the cloud implementation. It is deployed by using a PHP application.

The Uniform Resource Locator (URL) of the developed and deployed data security system in the cloud is https://data-security.herokuapp.com

Figure 3.3 shows the screenshot of the main interface of the data security system.

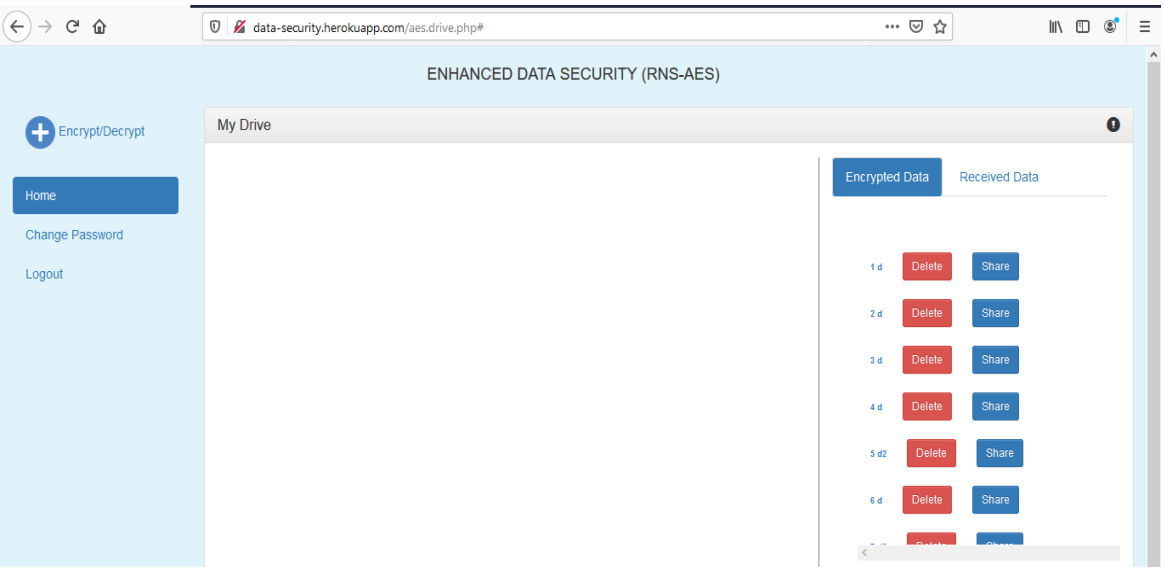


***Figure 3.3: Main Interface of the developed system***

### 3.6 Performance Evaluation Metrics for the System

The developed system was evaluated using the following metrics.

- Encryption time: The encryption time is termed as the time it takes an encryption algorithm to generate cipher text from the given plain text.
- Decryption time: The decryption time is termed as the time it takes the decryption algorithm to generate plaintext from the given cipher text.

  The encryption and decryption time are based on the file size.

  Encryption time or Decryption time = $\sum_{i=1}^{N_b} t(i)$

  Where $N_b$ = Number of blocks, $t_i$ = time to encrypt each block

- Throughput of Encryption: The throughput is termed as the quantity of the particular data that is encrypted in a given amount of time. It is the ratio of the size of plaintext encrypted in kilobyte to the encryption time in seconds. The encryption time is used to calculate the throughput of encryption using equation (3.7).

$$Encryption\ Throughput = \frac{size\ of\ text}{time\ required\ for\ encryption} \tag{3.7}$$

$$\sum_{i=1}^{N_b} Mi/ti$$

  Where $M_i$ = input message block, $t_i$ = time to encrypt each block

- Security Level: Security Level is a measure of the strength that a cryptographic primitive such as a cipher or hash function achieves. Security level is usually expressed in "bits", where n-bit security means that the attacker would have to

perform $2^n$ operations or trials (n = key bit length) to break it. Attacks should be less in a multilevel encryption when compared to single level encryption.

# CHAPTER FOUR
# RESULTS AND DISCUSSION

## 4.1 Experimental Setup

In developing an enhanced data security in cloud storage, a novel cryptographic approach known as RNS-AES was implemented using PHP and JavaScript programming language for this research. In the experiments, the developed system encrypts different file sizes of text data (2kb, 4kb, 6kb, 8kb and 10kb). A comparative result with two encryption approaches which are AES and RNS-AES was generated and reported. The performance metrics used to compare both approaches are encryption time, encryption throughput, decryption time and security level. Other metrics used to compare the developed approach with existing algorithms and multilevel approaches are key length, cryptographic strength and possibility of attacks. The developed system is ran and tested on a Laptop with the following specification: Intel (R) Pentium (R) CPU P6200 @2.13Ghz 2GB RAM and 64-bit Operating System, in which the performance data is collected.

## 4.2 Text Conversion to ASCII Code Result

The text file to be encrypted is chosen and loaded into the system. The result generated when each of the character present in the text data are converted to the corresponding decimal equivalence is shown in Figure 4.1

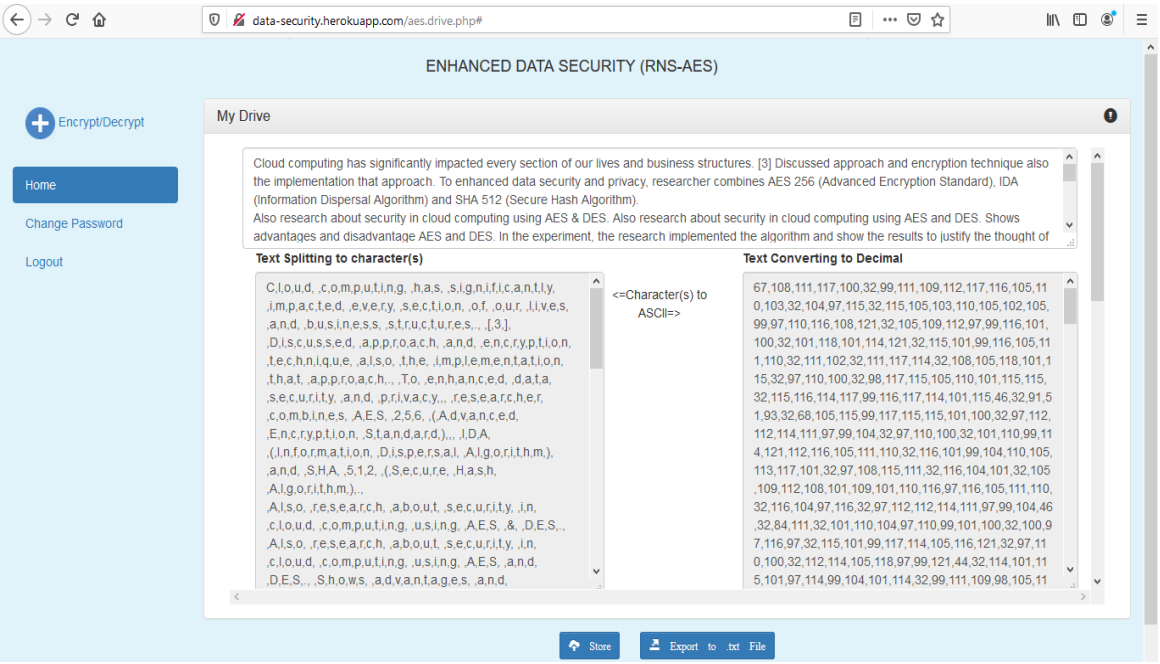


***Figure 4.1: ASCII Character Encoding***

## 4.3 Encryption of Plain Text File

The encryption of the plain text file was done using a multilevel approach tagged RNS-AES. The results generated in the process of encrypting a text file are shown in Figure 4.2, Figure 4.3 and Figure 4.4. Figure 4.2 shows the main page of encryption of different text file sizes. AES password (AES key) and moduli set are needed for the encryption. Figure 4.3 shows the residues of each of the decimal value present in the text file as the first level of encryption. In Figure 4.4, the AES-256 encryption of the residues is presented as the second level of encryption which generates the cipher text.

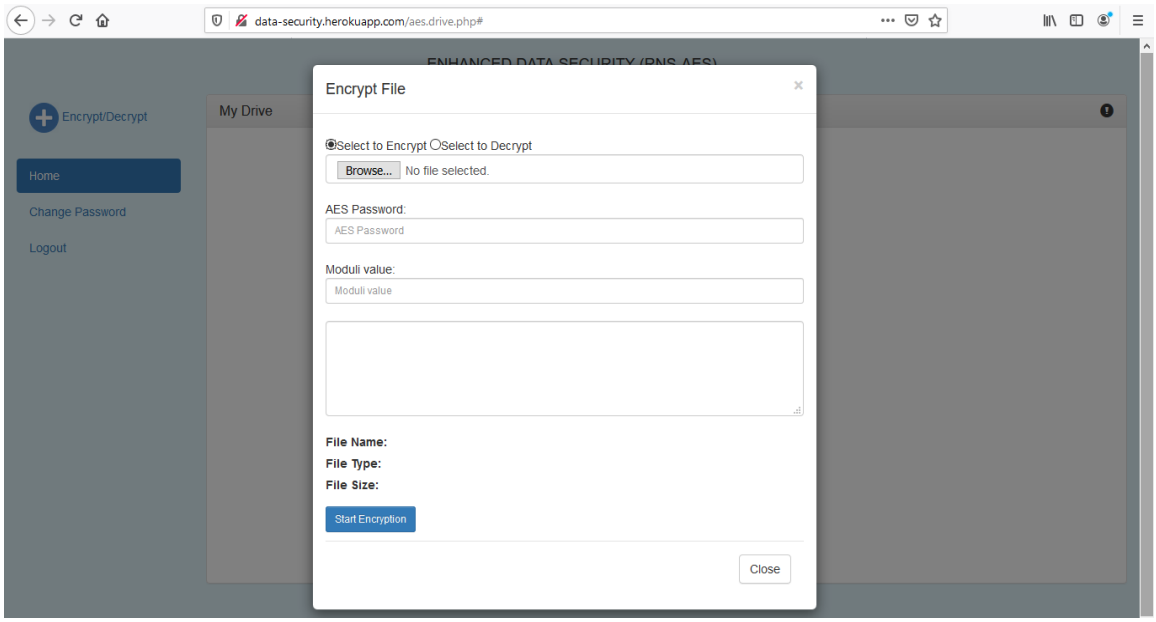


*Figure 4.2: Encryption Page*

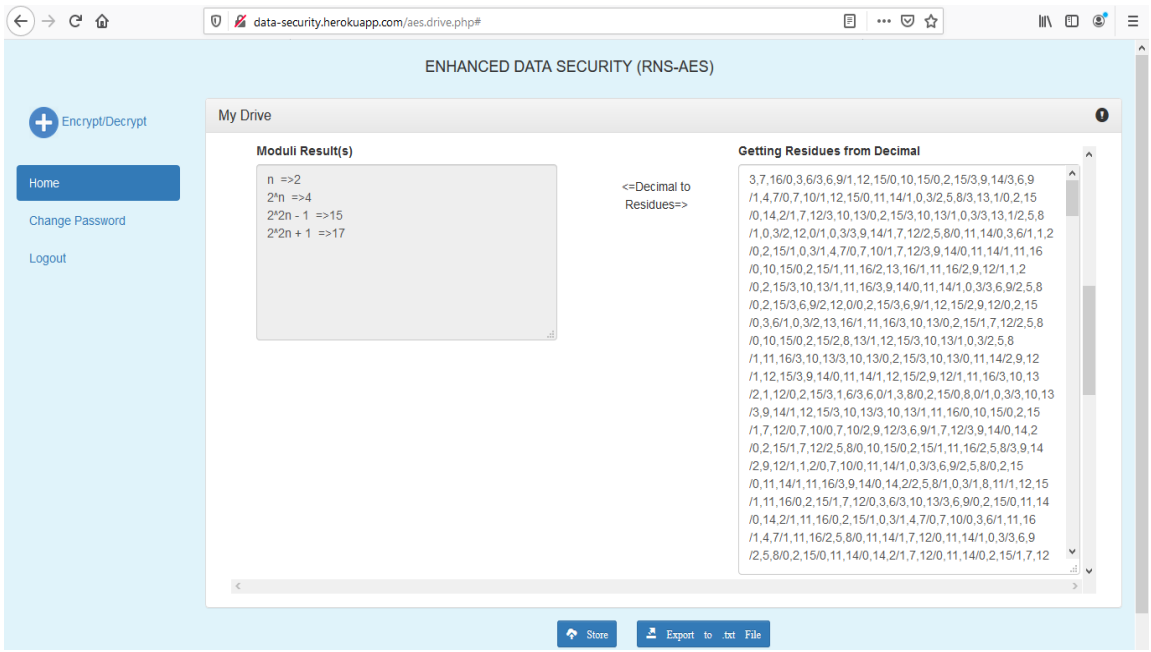


*Figure 4.3: Forward Conversion (Residues)*

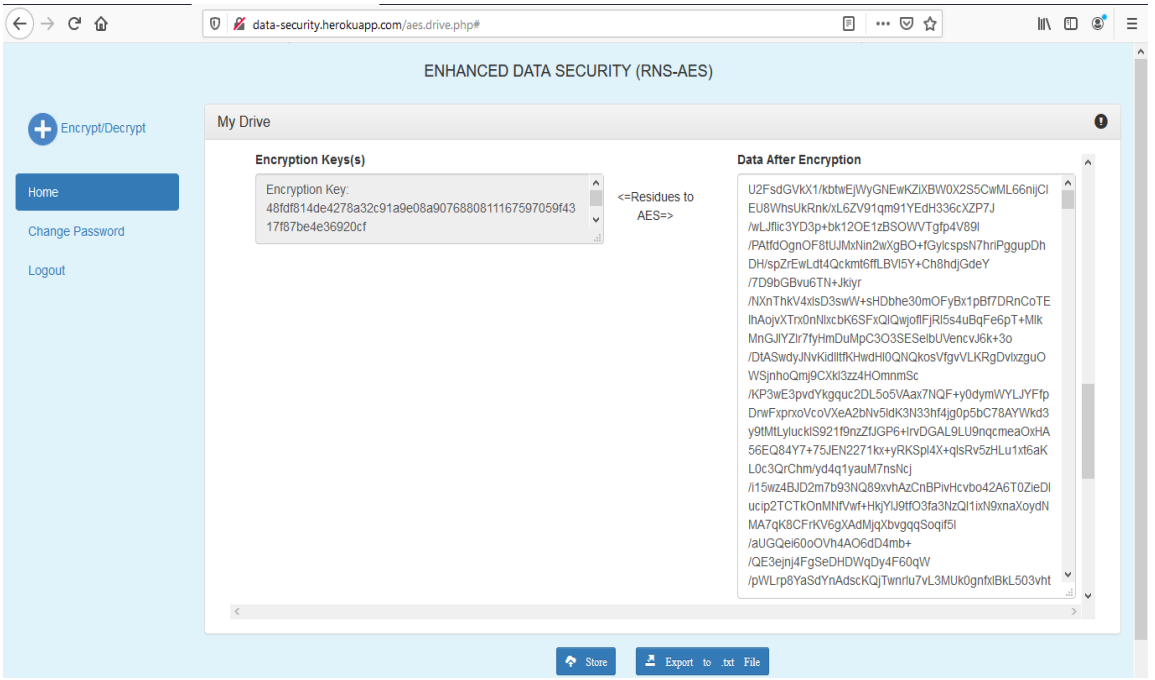


***Figure 4.4: AES-256 Encryption on Residues (Cipher Text)***

## 4.4 Decryption of Cipher Text File

The decryption of the cipher text file generated from the encryption process is done in a reverse chronological order. The inverse of AES-256 algorithm is done on the cipher text file for the first level of decryption and CRT algorithm is also implemented for the second level of decryption. Similar to the encryption process, the AES password and moduli set are also needed for the decryption process, otherwise the cipher text will not be decrypted to plain text. Figure 4.5 shows the implementation of AES-256 on the cipher text file and Figure 4.6 shows the decimal equivalence generated using the CRT algorithm.

Each of the decimal present in the file is mapped to its equivalent character in order to get the plain text as shown in Figure 4.7.

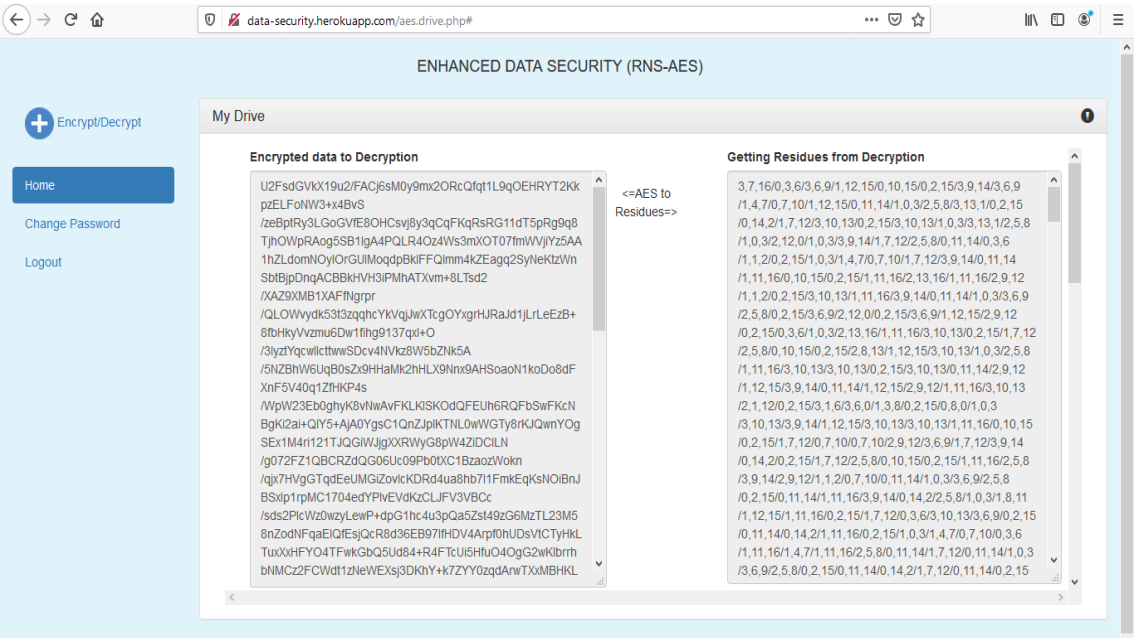


***Figure 4.5: AES-256 Decryption on Cipher Text File***

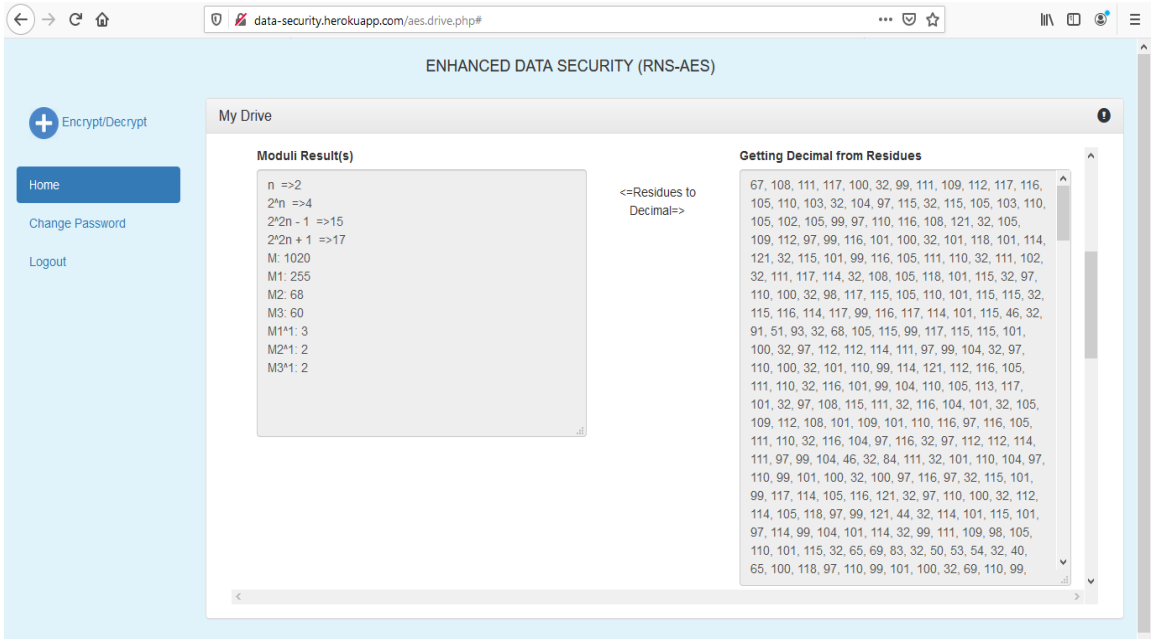


*Figure 4.6: Reverse Conversion (Chinese Remainder Theorem)*

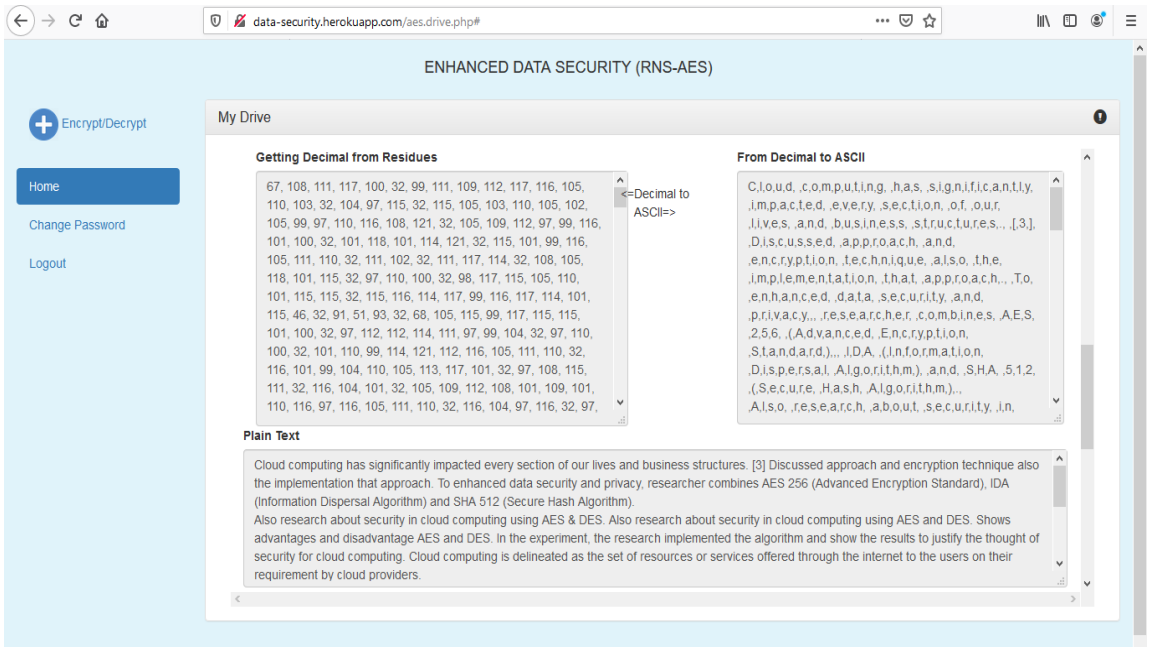


*Figure 4.7: Decimal to ASCII Characters (Plain Text)*

## 4.5 System Performance Evaluation

### 4.5.1 Encryption Time, Encryption Throughput and Decryption Time

Table 4.1 displays the results of the encryption time, encryption throughput and decryption time for the RNS-AES approach compared with varying input file sizes (2kb, 4kb, 6kb, 8kb and 10kb).

***Table 4.1: Encryption Time, Encryption Throughput and Decryption Time for RNS-AES***

| **File size(Kb)** | **Encryption Time (sec)** | **Encryption Throughput (Kb/sec)** | **Decryption Time (sec)** |
|---|---|---|---|
| 2 | 0.510 | 3.9216 | 2.807 |
| 4 | 0.876 | 4.5662 | 10.192 |
| 6 | 1.277 | 4.6985 | 23.349 |
| 8 | 1.687 | 4.7421 | 31.324 |
| 10 | 1.954 | 5.1177 | 37.831 |

Figure 4.8 shows the graphical representation of the encryption time for different text files. It shows that with increasing input file sizes, the encryption time also increases. X-axis represents the file sizes and Y-axis represents the encryption time of the files. Also, shown in Figure 4.9 is the graphical representation of the throughput of the encryption, that is, the speed at which the plain text files are encrypted with respect to time. It can be deduced from the figure that with increasing file size, there is a slight increase in speed (kb/sec) although the differences were small. Figure 4.10 shows the graphical representation of the

decryption time for different text file sizes. It indicates that as the file size increases, the decryption time also increases. Similarly the X-axis represents the file sizes and Y-axis represents the decryption time.

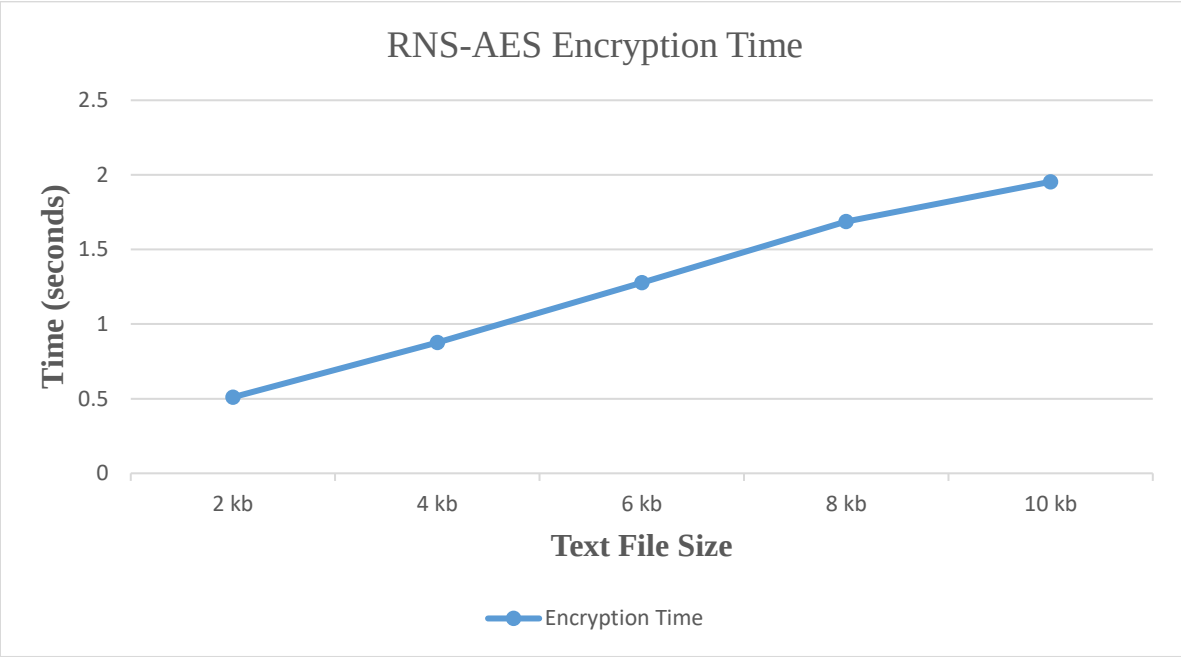


***Figure 4.8: Encryption Time for RNS-AES***

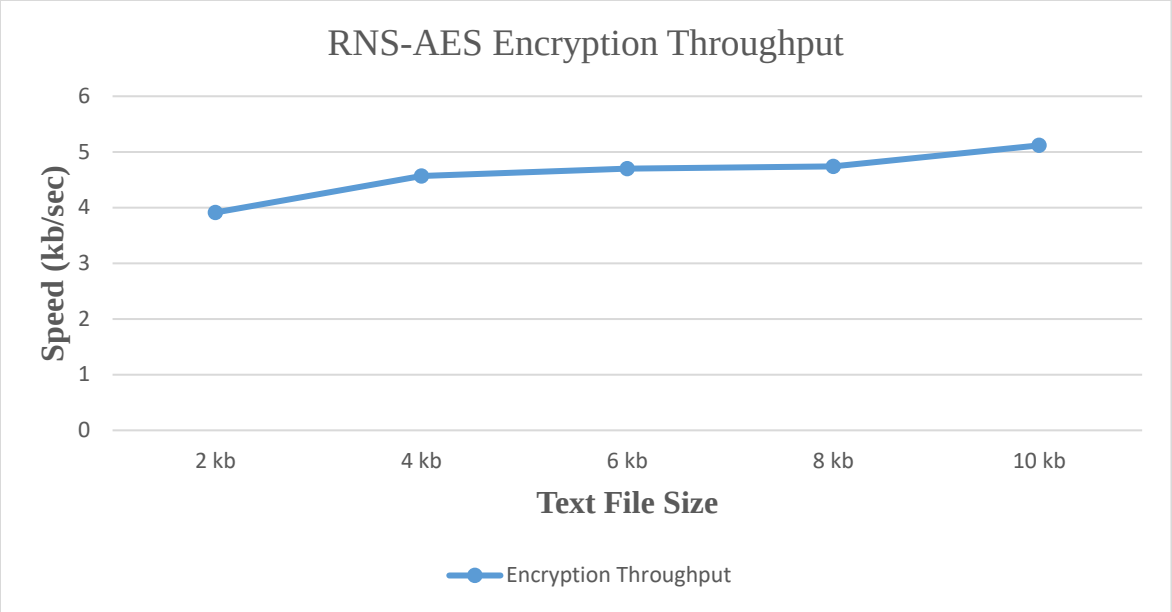


***Figure 4.9: Encryption Throughput for RNS-AES***

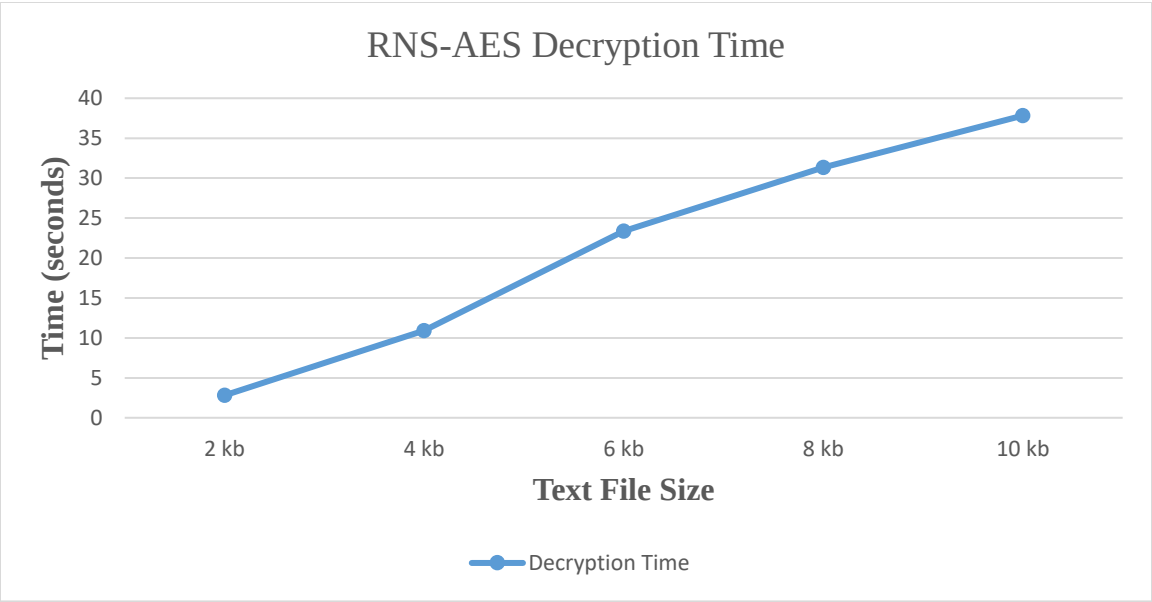


***Figure 4.10: Decryption Time for RNS-AES***

### 4.5.2 Security Level and Key Length

The key length is the size of the key used in the encryption and decryption algorithm. According to several literatures reviewed, the larger the key size or length, the stronger the encryption and the harder it is for attackers to break. The AES algorithm uses 128, 192 and 256 key length but AES-256 was chosen for the purpose of this research so as to make the encryption and decryption stronger. The RNS-AES approach uses a combination of moduli set and 256 bit keys for RNS and AES algorithm respectively. Thus having a stronger key length than other cryptographic algorithms. With a key length of 256 bits of the AES used, it requires $2^n$ (i.e. $2^{256}$) possible trials to perform attack on the algorithm which results to billions of permutations and combinations. Therefore the combination of RNS and AES-256 gives a high level of security than ordinary AES or other encryption algorithms.

In the data security system developed, if an unauthorized person gets to know the AES key used for encryption, he/she will still not be able to decrypt the cipher text without knowing the moduli set used for conversion. This demonstrates the level of security of the data stored in the cloud storage as shown in Figure 4.11.

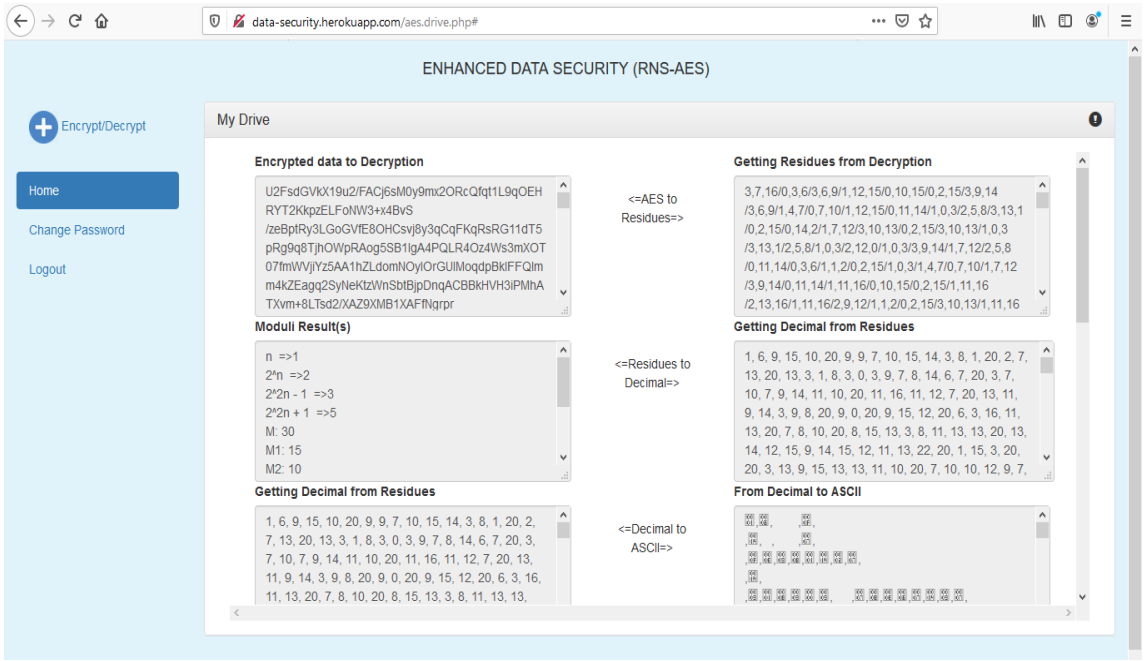


***Figure 4.11: Unauthorized Access Demonstration***

### 4.5.3 RNS-AES Comparison with AES

The developed system with RNS-AES cryptographic approach is compared with AES-256 encryption algorithm. Table 4.2 shows the results of the encryption time, decryption time and encryption throughput for both approaches along with varying input file sizes. Although RNS-AES approach took a little bit of time during encryption than AES approach only. This is because RNS-AES involves some processes (ASCII Conversion, RNS and AES-256) during encryption and decryption. AES alone has just a process during encryption and decryption, that is, the algorithm works only on the text file directly without some processes or conversions. The conversions or series of processes

involved in RNS-AES approach is to achieve a high level of security in storing data securely in cloud. Their encryption time difference (RNS-AES and AES) is not that much which shows that RNS-AES hybridized approach is also better for use.

Figure 4.12 shows that with increasing text file sizes, the encryption time also increases for both approaches. Figure 4.13 shows the trend of the decryption time for the approaches. It shows that AES decryption time is lesser than the encryption time but RNS-AES had a longer decryption time. The X-axis in the Figures 4.12 and 4.13 represents the file sizes and Y-axis represents the time used for encryption and decryption time.

*Table 4.2: Encryption Time, Encryption Throughput and Decryption Time for RNS-AES and AES*

| Approach | File Size (kb) | Encryption Time (sec) | Encryption Throughput (kb/sec) | Decryption Time (sec) |
|---|---|---|---|---|
| **AES-256** | 2 | 0.008 | 250 | 0.007 |
| | 4 | 0.009 | 444.444 | 0.008 |
| | 6 | 0.011 | 545.454 | 0.009 |
| | 8 | 0.012 | 666.667 | 0.011 |
| | 10 | 0.016 | 625 | 0.012 |
| | | | | |
| **RNS-AES** | 2 | 0.510 | 3.922 | 2.915 |
| | 4 | 0.876 | 4.566 | 10.192 |
| | 6 | 1.277 | 4.699 | 23.349 |
| | 8 | 1.687 | 4.742 | 31.324 |
| | 10 | 1.954 | 5.118 | 37.831 |

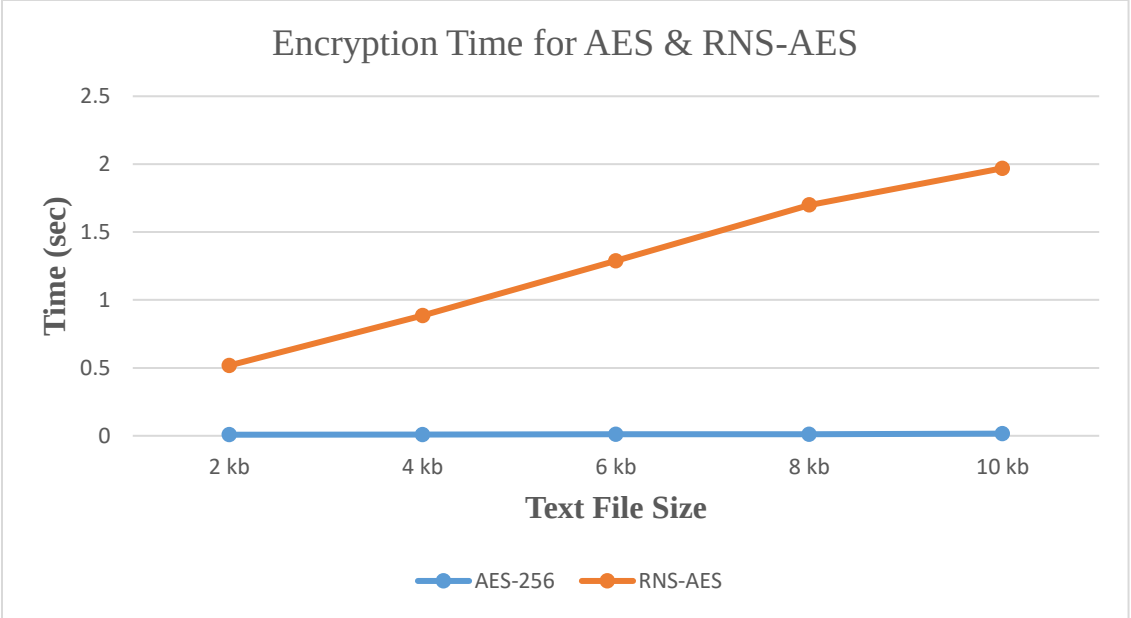


***Figure 4.12: Encryption Time for AES & RNS-AES***

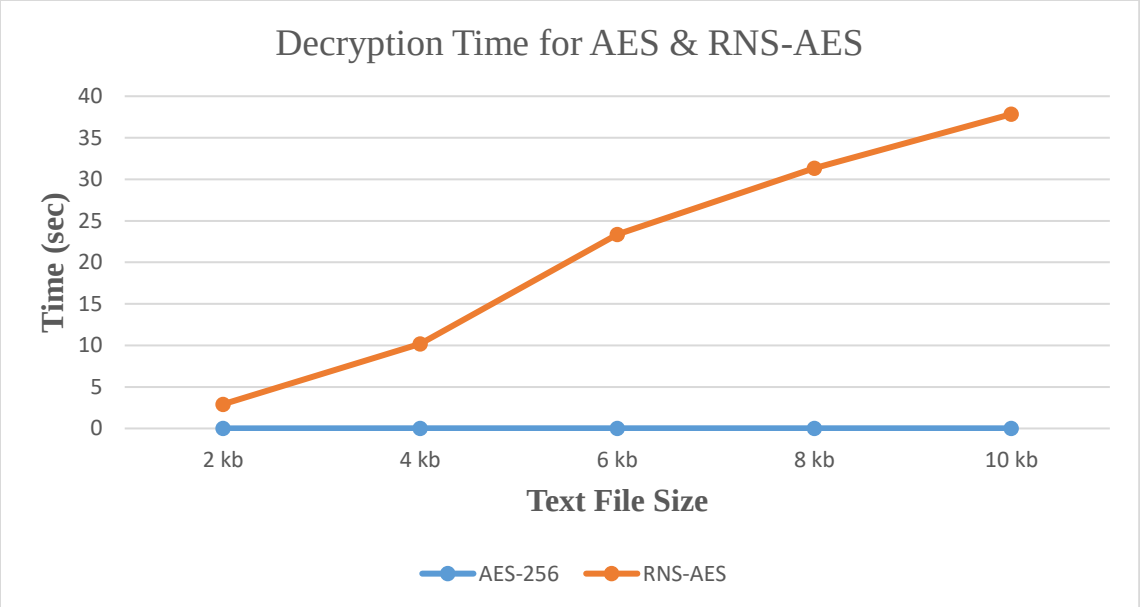


***Figure 4.13: Decryption Time for AES & RNS-AES***

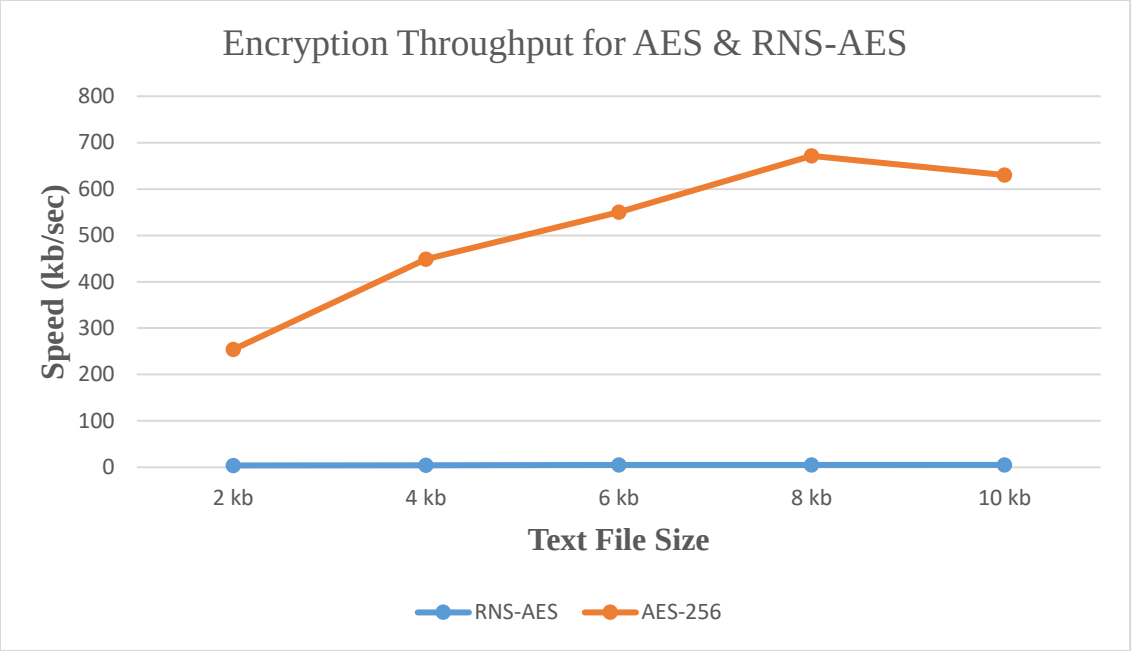


***Figure 4.14: Encryption Throughput for AES & RNS-AES***

### 4.5.4 Comparison with Existing Algorithms

The multilevel approach (RNS-AES) used in this research is compared with existing algorithms as shown in Table 4.3.

*Table 4.3: Summary of the Comparison with Existing Algorithms*

| **PARAMETERS** | **DES** | **BLOWFISH** | **RC5** | **RNS-AES** |
|---|---|---|---|---|
| Execution time | Slow | Fast | Slow | Moderate |
| Key length | 56 | 32-448 | 0-2040 | Moduli set+256 |
| Cryptographic Strength | Low | Moderate | High | High |
| Security Level | Low | Adequate | High | High |
| Possibility of Attacks | High | Moderate | Low | Low |

### 4.5.5 Comparison with Existing Multilevel Approaches

The multilevel approach (RNS-AES) used in this research is compared with other existing multilevel approaches as shown in Table 4.4.

***Table 4.4: Summary of the Comparison with Existing Multilevel Approaches***

| **PARAMETERS** | **ELGAMAL + HECC (Devi & Ganesan, 2019)** | **PROBABILISTIC + HOMOMORPHIC (Jayapandian & Zubair, 2017)** | **AES+RSA (Kartit et al., 2016)** | **RNS-AES (Developed Approach)** |
|---|---|---|---|---|
| Encryption and Decryption Time | Slow | Moderate | Moderate | Fast |
| Key Length | 256 | 512 | 1024 | Moduli set+256 |
| Cryptographic Strength | High | High | Moderate | High |
| Security Level | High | High | Moderate | High |
| Possibility of Attacks | Moderate | Moderate | Moderate | Low |

# CHAPTER FIVE

# SUMMARY AND CONCLUSION

## 5.1 Summary

In this research, secure storage of data in cloud is offered based on multilevel encryption of Residue Number System (RNS) and Advanced Encryption Standard (AES-256) algorithm. The use of multilevel encryption enhances data security as the security of sensitive data from unauthorized access is an essential requirement of cloud environment. The enhanced data security was implemented using PHP and JavaScript programming language, then deployed on Heroku cloud platform. When security level in cloud becomes high then confidentiality, integrity and privacy will be more convenient for the users and service providers. The RNS-AES performance was analyzed in comparison with AES algorithm and some existing multilevel algorithms based on some parameters.

The key length (moduli set+256) of the developed approach is better than other algorithms (e.g. AES) compared which guarantees a higher security for the system. AES algorithm is susceptible to brute force attack, that is, unauthorized user can guess the key used for encryption and decryption. In the developed technique, if an attacker tries to guess the key used for AES encryption, he/she will still find it difficult to have access to the original data. This is the importance or strength of RNS as an additional level of security. It prevents anyone with just the public key from being able to decrypt the data. So the combination of RNS and AES provides more security to the data at rest (cloud storage) and

data in transit (movable data). The complexity of the developed approach will further make any attack more difficult to achieve which makes the security attacks low.

The execution times for encryption and decryption is a little bit more than the conventional AES algorithm, though their encryption time difference is not that much. This is because the developed approach tagged RNS-AES involved series of conversions (ASCII, RNS and AES-256) in order to enhance the security of data stored in the cloud. Likewise, the encryption throughput showed that the developed approach is slower than AES but with increasing file size, there is a slight increase in speed (kb/sec) although the differences were small. The encryption time of RNS-AES is faster than the decryption time. Also, the developed system encrypts faster than other existing multilevel approaches when compared. It shows that the developed approach presents enhanced performance result by consuming an average less encryption time.

The increased security strength is the major focus and advantage in this research, therefore throughput and execution time come as a tradeoff.

## 5.2 Conclusion

This research presents an enhanced data security in cloud storage by using RNS and AES-256. The data security was implemented and deployed on Heroku cloud platform. The enhanced approach tagged RNS-AES is not easy for hackers, intruders or unauthorized users to gain access to the users' data stored on the cloud. If by mistake, a hacker tries to guess the secret key used for encryption and decryption, reverse conversion with the correct

moduli set is still needed to be performed in order to get the actual data. Also, due to the mathematical complexity involved, the third party unauthorized access is not going to be very easy. The cipher text generated from the developed approach will not be easy for attackers to decrypt, it involves series of processes: ASCII conversion, moduli set, value of n for moduli set, CRT converter and AES-256 implementation.

Results indicate that the developed data security is more efficient with regards to security level and key length than the existing methods. The performance evaluation also shows that RNS and AES-256 cryptography can be used for data security.

### 5.3 Major Contributions

This research has contributed to the body of knowledge by enhancing data security for cloud storage systems with the use of RNS and AES-256. It ensures the security and privacy of text data files with stronger key length (moduli set+256) and cryptographic strength which gives a high level of security. Therefore, even if the cloud server is attacked by the intruders then the original data or the vital information remains secured as the unauthorized user needs moduli set and AES-256 key bits to retrieve the original data.

### 5.4 Future work

It is recommended that other algorithms can be used alongside RNS so as to increase the security of sensitive data stored in the cloud. Algorithms like RSA, Hyper Elliptical Curve Cryptography (HECC) can be considered so as to increase the key length.

Also strong cryptographic algorithms can also be hybridized for security and performance analysis in terms of speed and throughput. A high processing capability computer system can be considered for encryption and decryption so as to reduce the execution time. Furthermore, the work can be extended to various sensitive and valuable data like videos, images and sounds that will be stored in the cloud storage.

## References


Akashdeep, B., GVB, S., Vinay, A., & Hanumat, S. (2016). Security Algorithms for Cloud Computing . *International Conference on Computational Modeling and Security (CMS 2016)* , 535-542.

Alajmi, Ali, S. S., Adzhar, K., & Mohammed, A.-s. (2018). Cloud Computing Delivery and Delivery Models: Opportunity and Challenges. 5-10.

Albugmi, A., Walters, R. J., Alassafi, M., & Wills, G. (2016). Data Security in Cloud Computing . *Fifth International Conference on Future Generation Communication Technologies (FGCT2016)*, 55-59.

Al-assam, H., Hassan, W., & Zeadally, S. (2019). Automated Biometric Authenetication with Cloud Computing. *Springer Nature Switzerland AG 2019*, 455-475.

Amir, S. M., & Leonel, S. (2017). Introduction to Residue Number System. *Springer International Publishing AG 2017*, 3-17.

Ananthalakshmi, & Rajagopalan. (2019). VLSI implementation of residue number system based efficient digital signal processor architecture for wireless sensor nodes. *Int. j. inf. tecnol.*, 1-12

Anu, Shree, D., & Ahlawat, S. (2017). A Review on Cryptography, Attacks and Cyber Security . *International Journal of Advanced Research in Computer Science, 8*(5), 239-242.

Aremu, I. A., & Gbolagade, K. A. (2017). An Overview of Residue Number System. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 1618-1623.

Arockiam, & Monikandan. (2013). Data Security and Privacy in Cloud Storage using Hybrid Symmetric Encryption Algorithm. *International Journal of Advanced Research in Computer and Communication Engineering, 2*(8), 2-7.

*ASCII Codes|Overview of all characters on the ASCII table*. (n.d.). Retrieved from IONOS Digitalguide: https://www.ionos.com/digitalguide/server/know-how/ascii-codes-overview-of-all-characters-on-the-ascii-table/

*ASCII-American Standard Code for Information Interchange*. (2019). Retrieved from https://www.ionos.com/digitalguide/server/know-how/ascii-codes-overview-of-all-characters-on-the-ascii-table/

Belguith, S., Abderrazak, J., & Rabah Attia. (2015). Enhancing Data Security in Cloud Computing Using a Lightweight Cryptographic Algorithm. *ICAS 2015 : The Eleventh International Conference on Autonomic and Autonomous Systems*, 2-7.

Bih-Hwang, L., Ervin, K. D., & Muhammad, F. W. (2018). Data Security in Cloud Computing Using AES under HEROKU Cloud. *The 27th Wireless and Optical Communications Conference (WOCC2018)*, 3-5.

Chinthagunta, M., & Vidyamadhuri. (2017). Cloud Computing Models : A Survey. *Advances in Computational Sciences and Technology*, *10*(5), 747-761. http://www.ripublication.com.

Devi, & Ganesan. (2019). Environmental Benefits of Enhanced Hecc- Elgamal Cryptosystem for Security in Cloud Data Storage Using Soft Computing Techniques. *Foundation Environmental Protection & Research-FEPR*, 665-677.

Ewang, A. K. (2019). Overview of cloud computing. Akwa Ibom State University.

Feng , Z., Chao, L., & Chun, F. L. (2014). A cloud computing security solution based on fully homomorphic encryption. February 16-19, ICACT2014, 485-488.

Gbolagade, K. A., & Cotofana, S. D. (2013). An Efficient RNS to Binary Converter Using the Moduli Set {2n+1, 2n, 2n-1}. *Institute of Electrical and Electronics Engineer*, 1-6.

Gowthami, S., & Kousalya. (2017). A Comparative Analysis of Security Algorithms Using Cryptographic Techniques in Cloud Computing. *International Journal of Computer Science and Information Technologies*, *8*(2) 306-310.

Izhar, S., Kaushal, A., Fatima, R., & Qadeer, M. (2017). Enhancement in Data Security using Cryptography and Compression. *7th International Conference on Communication Systems and Network Technologies* , 212-215.

Jayapandian, & Rahman, Z. (2017). Secure and efficient online data storage and sharing over cloud environment using probabilistic with homomorphic encryption. *Springer Science+Business Media New York,* February 2017, 1-13. Doi: 10.1007/s10586-017-0809-4

Kanagavalli, & Vagdevi. (2014). A survey of Homomorphic Encryption Schemes in Cloud Data Storage. *International Journal of Recent Development in Engineering and Technology* , *3*(1), 1-5.

Kartit, Z., marraki, M., Azougaghe, A., & Mustapha, H. (2016). Applying Encryption Algorithm for Data Security in Cloud Storage. January 2016, 6-15. Doi: 10.1007/978-981-287-990-5_12

Karun, H., & Uma, S. (2015). Data Security in Cloud Computing using Encryption and Steganography. *International Journal of Computer Science and Mobile Computing, 4*(5), *May 2015*, 786-791.

Kazim, M., & Zhu, S. Y. (2015). A survey on top security threats in cloud computing. *(IJACSA) International Journal of Advanced Computer Science and Applications, 6*(3), 109-113.

Khan, S., Shekhar, C., & Khare, M. D. (2018). Implementing Cryptographic Method for Ensuring Data Security In Cloud Computing Based On Hybrid Cloud . *International Journal of Scientific Research in Science and Technology (www.ijsrst.com), 4*(8), 338-343.

Khandelwal, M. K., & Saini, H. C. (2019). Review on Security Challenges of Cloud Computing. *International Conference on Advancements in Computing & Management (ICACM-2019)* , April 2019, 1031-1037.

Kumar, R., Raj, H., & Jelciana. (2018). Exploring Data Security Issues and Solutions in Cloud Computing . *International Conference on Smart Computing and Communications (ICSCC)*, 692-693.

Kumar, S., Jatinder, P. S., & Mamta. (2015). Authentication and Encryption in Cloud Computing . *International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, May 2015, 230-233.

Kumar, S., Shekhar, J., & Singh, J. P. (2018). Data Security and Encryption Technique for Cloud Storage. *Springer Nature Singapore Pte Ltd. 2018, Advances in Intelligent Sysytems and Computing 729*, 193-199.

Mohd, D., & Prachi, S. (2018). Review Study of Cloud Computing – Benefits, Risk, Challenges and Security. *ICONIC RESEARCH AND ENGINEERING JOURNALS, 1*(9), 139-141.

Narayana, Sailesh, K., & Jayashree. (2017). A Review on Different types of Deployment Models in Cloud Computing. *International Journal of Innovative Research in Computer, 5*(2), 1475-1481. Doi: 10.15680/IJIRCCE.2017. 0502029

Nishit, M., Tarun, S. K., Varun, S., & Vrince, V. (2018). Secure Framework for Data Security. *Springer Nature Singapore Pte Ltd.* , 62-63. https://doi.org/10.1007/978-981-10-5687-1_6

Papri, G., Vishal, T., & Pravin, B. (2017). Data Security and Privacy in Cloud Computing Using Different Encryption Algorithms. *International Journal of Advanced Research in Computer Science and Software Engineering*, *7*(5), 469-471.

Rady, M., Abdelkader, T., & Ismail, R. (2019). Integrity and Confidentiality in Cloud Outsourced Data. *Ain Shams Engineering Journal*, 275-280. Doi: 10.1016/j.asej.2019.03.002

Rex, C., & Britto, K. R. (2015). Cloud Computing Data Security Issues, Challenges, Architecture and Methods- A Survey. *International Research Journal of Engineering and Technology (IRJET)*, *2*(4), 848-857.

Seth, B., Dalal, S., & Kumar, R. (2019). Hybrid Homomorphic Encryption Scheme for Secure Cloud Data Storage. *Springer Nature Switzerland AG, Recent Advances in Computational Intelligence*, 71-92. Doi: 10.1007/978-3-030-12500-4_5

Sharma, Y., Gupta, H., & Khatri, S. K. (2019). A security model for the enhancement of data privacy in cloud computing. *978-1-5386-9346-9/19/$31.00 ©2019 IEEE*, 1-5.

Shereek, B. M., Muda, Z., & Yasin, S. (2014). Improve Cloud Computing Security using RSA Encryption With Fermat's Little Theorem. *International organization of Scientific Research*, *4*(2), 1-8.

Sirisha, L., & Basha, M. (2018). Analysis of Security Algorithms for Data Sharing in Cloud Computing. *International Journal of Advanced Research in Computer and Communication Engineering*, 46-51.

Sultan, M., & Yasen, K. (2018). Homomorphic Encryption Implementation to ensure Data Security in Cloud Computing. *Journal of Theoretical and Applied Information Technology*, 1826-1836.

Tinankoria, D., & Babak, R. B. (2017). Cloud Computing: A review of the Concepts and Deployment Models. *I.J. Information Technology and Computer Science*, *9*(6), 50-58. Doi: 10.5815/ijitcs.2017.06.07

*Understanding AES 256 Encryption*. (2019). Retrieved from https://www.solarwindsmsp.com/blog/aes-256-encryption-algorithm

*What is Cloud Storage?* (n.d.). Retrieved from Techopedia: https://www.techopedia.com/definition/26535/cloud-storage

Younes, D. (2013). *Residue number system based building blocks for applications in digital signal.* Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Microelectronics.

Zhang, Y., Juels, A., Reiter, M., & Ristenpart, T. (2012). Cross-VM Side Channels and Their Use to Extract Private Keys. *proceedings of the 2012 ACM conference on Computer and communications security*, 305-316.

**Appendix**

```
var contentShow
function conVert2ASCII(t0, password){
    contentShow=document.getElementById("contentShow");
    $('#messageEn').html(contentShow);
    var stRAScii = "";
    var text = $('#messageEn').text().toString();
    var arr = text.split('').toString();
    $('#messageSpliting').append(arr);
    for (var i = 0; i < text.length; i++) {
        if(i === (text.length - 1)){
            stRAScii += text[i].charCodeAt(0);
        }else{
        stRAScii += text[i].charCodeAt(0)+',';
    }
  }
    $('#messageASCII').html(stRAScii);
    $('#alphaToAsii').show(1000);
        getResiduces(t0, password);
}
function getResiduces(t0, password){
    var n = $('#nPassword').val();
            var n1, n2, n3;
            n1 = Math.pow(2, n);
            n2 = (Math.pow(2, 2 * n) - 1);
```

```
            n3 = (Math.pow(2, 2*n) + 1);
    $('#moduliResult').append('n  =>'+n+'\n');
    $('#moduliResult').append('2^n  =>'+n1+'\n');
    $('#moduliResult').append('2^2n - 1  =>'+n2+'\n');
    $('#moduliResult').append('2^2n + 1  =>'+n3);
    var text = $('#messageASCII').val().toString().split(',');
    var residual = "";
  for (var i = 0; i < text.length; i++) {
      if(i === (text.length - 1)){
          residual += (parseInt(text[i])%n1).toString()+',';
          residual += (parseInt(text[i])%n2).toString()+',';
          residual += (parseInt(text[i])%n3).toString();
      }else{
          residual += (parseInt(text[i])%n1).toString()+',';
          residual += (parseInt(text[i])%n2).toString()+',';
          residual += (parseInt(text[i])%n3).toString()+'/';
      }
  }
document.getElementById('gettingResidualData').innerHTML = residual.toString();
    $('#decimalResidual').show(1000);
encryptUsingAES(t0, password);
}
function encryptUsingAES(t0, password){
    var data = $('#gettingResidualData').val();
        var encrypted = CryptoJS.AES.encrypt(data, password);
```

```
        $('#keyResult').append('Encryption Key: '+encrypted.key+'\n');

        $('#keyResult').append('Encrypted Ciphertext: '+encrypted.ciphertext  );

        $('#gettingEncryption').append(encrypted.toString());

                $('#uploadedMetrics').append('File Name:
'+$('#Fname').text().toString().split('.txt')[0]+'\n');

            $('#uploadedMetrics').append('File Type: '+ $('#Ftype').text()+'\n');

            $('#uploadedMetrics').append('File Size: '+$('#Fsize').text()+'\n');

            $('#uploadedMetrics').append('Time Started: '+t0+'\n');

            $('#afterAES').append('File Name:
'+$('#Fname').text().toString().split('.txt')[0]+'\n');

            $('#afterAES').append('File Type: UTF-8'+'\n');

            $('#afterAES').append('File Size:
'+bytesToSize(byteLength(encrypted.toString()))+'\n');

            var t1 = performance.now();

            $('#afterAES').append('Time Ended: '+t1+'\n');

           $('#afterAES').append('Over all Time: '+ ((t1 - t0)/1000) + " seconds."+'\n');

          $('#afterAES').append('Throughput: '+(parseInt($('#Fsize').text())/  ((t1 -
t0)/1000))+'\n');

            $('#encryptedAES').show(1000);

            $('#encryptedDetail').show(1100);

            $('#submitSave').show(1500);

      $('#exportSave').show(1500);

}
```