A Solution for Privacy-Preserving and Security in Cloud for Document Oriented Data (By Using NoSQL Database)

Elias Bassa Badacho
(M.Tech in Computer Science and Engineering)
Department of Computer Science (Lecturer)
Wolaita Sodo University, Ethiopia
P.O. Box 138, Wolaita Sodo, Ethiopia

Abstract

Cloud computing delivers massively scalable computing resources as a service with Internet based technologies those can share resources within the cloud users. The cloud offers various types of services that majorly include infrastructure as services, platform as a service, and software as a service and security as a services and deployment model as well. The foremost issues in cloud data security include data security and user privacy, data protection, data availability, data location, and secure transmission. In now day, preserving-privacy of data and user, and manipulating query from big-data is the most challenging problem in the cloud. So many researches were conducted on privacy preserving techniques for sharing data and access control; secure searching on encrypted data and verification of data integrity. This work included preserving-privacy of document oriented data security, user privacy in the three phases those are data security at rest, at process and at transit by using Full Homomorphic encryption and decryption scheme to achieve afore most mentioned goal. This work implemented on document oriented data only by using NoSQL database and the encryption/decryption algorithm such as RSA and Paillier's cryptosystem in Java package with MongoDB, Apache Tomcat Server 9.1, Python, Amazon Web Service mLab for MongoDB as remote server.

Keywords: Privacy-Preserving, NoSQL, MongoDB, Cloud computing, Homomorphic encryption/decryption, public key, private key, RSA Algorithm, Paillier's cryptosystem
DOI: 10.7176/CEIS/11-3-02
Publication date: May 31st 2020

I Introduction

The cloud computing is massively scalable computing resource as the service via internet-based technologies. Resources are shared among a vast number of consumers allowing for a lower cost of IT ownership **[1].** At present, cloud computing is widely discussed in academia and industry, Virtualization, distributed computing technology and so on. This technology integrates the computing, storage, networking to share resources and other computing resources, and leases to users all over the world via the internet. This kind of resource sharing among several systems or networks could reduce the cost of enterprise information processing, organization, collation, analyzing and accelerates the transmission of information of enterprise from sources to end users within a fraction of minutes. This newly emerging technology supports data storage technology which is known as Virtualization Technology. Virtualization technology is the most important component of the cloud computing that is used to virtualize the storage devices and different hardware components of computer systems as well as system software and application software from either user's point of views or service providers. Cloud computing is growing at a very high speed in the today's pacing digital industry around all over the world.

1.1 Characteristics of Cloud Computing

Cloud computing characteristics, NIST [2, 3] has made effort to provide a unified way to define cloud computing and its main functionality. Despite its complexity and heterogeneous nature, NIST has identified five essential characteristics that represent a cloud computing platform:

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Vertical scaling

1.2 Cloud Computing Service and Delivery Models

A. Delivery Models

Cloud computing provider offer their services according to three fundamental models [4], selecting the proper delivery model depends on the customer needs and requirements. The following are the three types of cloud computing delivery models:

- **Software as a Service:** SaaS is the delivery of business applications that are designed for specific purposes, where cloud providers manage the infrastructure and platforms that run the applications. SaaS has two distinct modes. The first one is Simple multitenancy in which every customer has their own separate resources which run on one or more computing platforms. The second on is fine-grain multitenancy in which the customers' resources are also separated but even more effectively. All customers' resources are shared, but data and accessibilities are separated within the application. Here we list some of the common characteristics of SaaS. Google Apps, Cisco WebEx, and Sales Force have some of these common characteristics:

- Provides web access to commercial software.
- A central location is responsible for managing software.
- The users not required to handle software upgrading and patches.

- **Platform as a Service:** In the PaaS model, the cloud provider delivers computing platforms that usually include an operating system, programming language execution environment, database, and a web or application server. Some of the PaaS providers are providing for customers scalable resources, where the underlying computers and storage resources scale automatically according to the applications' needs and demands so the cloud client does not have to allocate resources manually. PaaS providers are usually provide a computing platform that allows the creation of web applications in a quick and efficient manner, which allows customers to avoid the complexity of buying and maintaining the software and infrastructure required for the task. PaaS is similar to SaaS with the exception that rather than delivering software over the web, the platform for the creation of software is delivered over the web. Its basic characteristics usually include the following [5].
- **Infrastructure as a Service:** In an IaaS model, resources such as host's hardware, software, servers, storage, and other infrastructure components can be provided to customers. IaaS Cloud providers provide on-demand and highly scalable resources which make IaaS suitable for workloads that are temporary, experimental, or change unexpectedly. Many companies now a days, prefer to outsource servers, software, data-center, and so on, rather than purchasing the resources and get a fully on-demand service. The most popular IaaS providers are Amazon Web Services, IBM SmartCloud Enterprise, Rackspace Open Cloud, Windows Azure, and Google Compute Engine [6].

2 Literature Review and Related Works

2.1 Privacy preserving and Security in Document Oriented Data

The foremost issues in cloud data security include data and user privacy, data protection, data availability, data location, and secure transmission. Threats, data loss, service disruption, outside malicious attacks, and multi-tenancy issues are the security challenges included in the cloud. Data integrity in the cloud system means preserving the integrity of stored information. The data should not be lost or modified by unauthorized users. Cloud computing providers are trusted to maintain data integrity and accuracy of data. Data confidentiality is also important aspect from user's point of view because they store their private or confidential data in the cloud. Authentication and access control strategies are used to ensure data confidentiality. The data confidentiality could be addressed by increasing the cloud reliability and trustworthiness in Cloud computing. Therefore security, integrity, privacy and confidentiality of the stored data on the cloud should be considered and are important requirements from user's point of view [7].

2.2 Data Privacy

Privacy: Cloud computing possesses privacy concerns because the service providers have access to the data that is stored on their infrastructure. Cloud providers could accidentally or deliberately alter or even delete information. Many cloud providers can share information with third parties if necessary without a warrant. The permission is granted in their privacy policy, which users to be agreed before they start using cloud services. Privacy solutions include policy and legislation as well as end users' choices for how data are stored. Users can encrypt data that are processed or stored within the cloud to prevent unauthorized access [2, 8].

Agencies planning to place personal information in a cloud service should perform a Privacy Impact Assessment (PIA) to ensure that they identify any privacy risks associated with the use of the service together with the controls required to effectively manage them. Cloud services may make it easier for agencies to take advantage of opportunities to share information. For example, sharing personal information with another agency may be achieved by simply creating user accounts with the appropriate permissions within a SaaS solution rather than having to implement a system-to-system interface to exchange information. Although cloud services have the potential to lower the technical barriers to information sharing agencies must ensure that they appropriately manage access to personal information and comply with the requirements of the Privacy Act 1993 [9].

Service providers typically use privacy policies to define how they will collect and use personal information about the users of a service. US service provider's privacy policies usually distinguish between Personally Identifiable Information (PII) and non-personal information. However, it is important to note that both are considered personal information under the Privacy Act 1993. Agencies must carefully review and consider the implications accepting a service provider's privacy [10, 11].

Privacy-preserving concerns arise whenever sensitive data is outsourced to the cloud where the data is processed and stored. By using encryption, the cloud server (i.e. its administrator) is prevented from learning content in the outsourced databases. But how can we also prevent a local administrator from learning the database content. To prevent data content from local and remote users, the system has to restrict user roles at application level as well as server level by providing user authorization and authentication techniques. The fact that users no longer have physical possession of the outsourced data makes it a formidable task to achieve the data confidentiality and integrity. As the data, in most cases encrypted, have to be not only stored, but also processed in clouds, the cryptography-based data confidentiality and integrity protection approaches are not adequate to satisfy the security requirements. Privacy preserving in cloud environments includes two aspects: data processing security and data storage security. Data processing security covers the issues of how to protect user privacy at runtime in a virtualized cloud platform. Data storage security covers the issues of guaranteeing user data privacy when the data is stored in data center.

2.3 Data Security

The meaning of security is plentiful. Security is the combination of confidentiality, the prevention of the unauthorized disclosure of information, integrity, the prevention of the unauthorized amendment or deletion of information, and availability, the prevention of unauthorized withholding of information [12]. Security plays a critical role in the current era of long dreamed vision of computing as a utility. It can be divided into four sub categories: safety mechanisms, cloud server monitoring or tracing, data confidentiality, and avoiding malicious insiders' illegal operations and service hijacking. In short, the foremost issues in cloud data security include data privacy, data protection, data availability, data location, and secure transmission. The security challenges in the cloud include threats, data loss, service disruption, outside malicious attacks, and multitenancy issues [13]. To prevent the tenants' privacy data leakage in the cloud data security has the following techniques;

- **Data Encryption**, it can effectively prevent the leakage of data and user privacy in cloud but the efficiency of the cipher-text processing is low.
- **Anonymity**, it protects the privacy of user and data through generalization. Generalization has acceptable efficiency, but it may result data loss.
- **Data fragmentation**, it separates attributes which would leak tenants' privacy together into distinct table and hides the relationships between tables. It does not loss data and its executing efficiency is better than data encryption, but it is not applicable for the structure of document.

2.4 Data Integrity

Data integrity is one of the most critical elements in any information system. Generally, data integrity means protecting data from unauthorized deletion, modification, or fabrication. Managing entity's admittance and rights to specific enterprise resources ensures that valuable data and services are not abused, misappropriated, or stolen. Data integrity in the cloud system means preserving information integrity. The data should not be lost or modified by unauthorized users. Data integrity is the basis to provide cloud computing service such as SaaS, PaaS, and IaaS [12].

2.5 Data Confidentiality

Data confidentiality is important for users to store their private or confidential data in the cloud. Authentication and access control strategies are used to ensure data confidentiality. The data confidentiality, authentication, and access control issues in cloud computing could be addressed by increasing the cloud reliability and trustworthiness [12]. **Data Protection to Minimize Data Security Risks**

In cloud computing technology, data should be protected from different users, service providers and from so many different types of illegal data accessing actions to preserve privacy and security of data in the cloud by taking the following methods [14].

- *Maintaining data integrity*, it ensures that data at the rest are not subject to unauthorized accessing. Multi-tenancy is the core technological approach for creating efficiencies in the cloud, but the technology, if implemented or maintained improperly, can put a cloud users' data at risk of hijacking, corruption, and unauthorized access. A cloud user should expect contractual provisions obligating a cloud provider to protect its data and the user ultimately may be entitled to some sort of contract remedy if data integrity is not maintained.
- *Accessibility and availability of data*, cloud providers long-term viability will be connected to its ability to provide its customers with almost continual access to their services, data, and resources.
- *Disaster recovery*, cloud providers should control and manages the user's data stored in cloud server and be able to recover lost data from backup.

- *Viability of the cloud providers*, it includes several aspects of the cloud, such as technical capacity of providers, economical capacity, data recovery skills, awareness of their virtual machines, data locations, data segregation, privileging user access, limiting themselves from taking any types of data manipulations, protecting their system from different types of malicious attacks and so on.
- *Outsourcing encrypted data* to cloud providers to preserve privacy of outgoing data on the way to service provider server
- *Searching query in secure manner*, user also has to search their *queries in a secure manner* and according to the privilege given from cloud providers.

2.6 Document Oriented Database

A document-oriented database or document store is designed for storing, retrieving and managing document-oriented information, also known as semi-structured data. Document-oriented databases are one of the main categories of NoSQL databases [15]. It is more and more difficult to deal with big data and high concurrency data for traditional RDBMS in the cloud storage. NoSQL database has been popular in cloud storage, and a documented oriented database also has attracted much attention as a kind of NoSQL database. In today world, more organizations and enterprises deploy their data in a document oriented database.

Now a day, the rapid development of data science beside to IT technology, it classifies data into three main classes such as structured data, semi-structured data, and unstructured data those are more complicated and impracticable in relational database environments. Conventional relational database system use two-dimensional table for data creation with properties like transactions, complex SQL queries, and multi-table related query. However, multi-table queries are not effective for huge data queries. Scalability in relational databases requires powerful servers that are both expensive and difficult to handle. NoSQL provides the flexibility to store entire data in terms of documents Instead of the conventional method of a table row- column. NoSQL is extensively useful when we need to access and analyze huge amounts of unstructured data or data that's stored remotely on multiple virtual servers [16]. There are four different NoSQL databases:

1. **Key-value stores**: key-value store is a system that stores values indexed for retrieval by keys. These systems can hold structured or unstructured data and can easily be distributed to a cluster or a collection of nodes as in Amazon's DynamoDB and Project Voldemort.
2. **Column-oriented databases**: Column-oriented database is a system that stores data in the whole column instead of a row, which minimizes disk access compared to a heavily structured table of columns and rows with uniform sized fields for each record as in HBase and Cassandra.
3. **Document-based stores**: These databases store data and organize them as document collections, instead of structured tables with uniform sized fields for each record. With this database, users can add any number of fields of any length to a document as implemented in Couch DB, Mongo DB.
4. **Graph databases**: These databases use nodes, edges, and properties to represent and store data in the form of graphs. It is possible to represent data as a graph-like structure, which can be easily traversed as in Allegro Graph and Neo4j. NoSQL databases offer several benefits over relational databases, including:

- **Reduced complexity**. The rich feature set and strict ACID properties of relational databases may not be necessary for some data sets.
- **Higher throughput**. Cassandra writes 2,500 times faster into a 50GB database than MySQL [17]. BigTable can process 20 petabytes per day [39].
- **High degree of scalability on commodity hardware**. NoSQL databases do not rely on highly available hardware and are designed to handle failure efficiently. Data can be partitioned across hardware more efficiently than relational database sharding. Hardware nodes can be added and removed relatively easily.
- **More flexible data model**. NoSQL databases are not restricted to the relational data model which can be inefficient for unstructured data sets.

While NoSQL databases address some problems with the relational model, they also present their own set of problems. Most notable is the weaker guarantees offered by NoSQL must balance consistency, availability, and partition tolerance and that strong forms of all three properties cannot be achieved simultaneously [18, 19]. NoSQL databases generally sacrifice consistency for increased availability and partition tolerance. In contrast to ACID properties provided by relational databases, many NoSQL systems claim to provide BASE properties basically available, soft state, eventually consistent. Another weakness of NoSQL databases is the lack of a common interface like Structured Query Language (SQL). SQL simplifies and standardizes database manipulation in relational databases. NoSQL databases each have a unique programming interface that uses a high-level procedural language (e.g., Java) and requires more complex programming than SQL to perform the same task.

3. OUTLINED SOLUTION FOR PROPOSED SYSTEM

For data privacy and user privacy, a solution of data storage, virtualization, and security of virtual machine, data manipulation and query are presented in this project. In this project the main database file is stored in MongoDB with a key/value pair which is a typical NoSQL database structure and is encrypted with built-in AES encryption algorithm. The plain Text is encrypted by using Paillier's encryption scheme which is additive homomorphic cryptosystem. For the purpose of security and privacy of data as well as user during data transmission, the TLS/SSL which is Transport Layer Secure or Secure Socket Layer standard security protocol is used to keep internet connection secure and safeguard for any sensitive that is being sent between two parties by making most trusted end-to-end communication, encrypting data and authenticating data in transmission channels. Here to prepare certificate RSA encryption scheme is used and configured with Tomcat web server and windows server 2012 to make HTTP protocol secure which is HTTPS.

3.1 Security Components of Cloud System and Its Risks

The components of cloud security are cloud service provider, cloud broker, data owner and cloud data consumers.

A. *Security Risks of Cloud Computing*

There are several security risks presented by cloud computing. The following risks are identified by the Cloud Security Alliance (CSA), which is a nonprofit organization established to define parameters for security guidance in cloud computing [20].

- **Data breaches:** Attackers may take advantage of a customer's poorly designed database and might get to every client's data.
- **Data loss:** Several issues may cause data loss such as attackers, careless service providers, or disasters.
- **Account or service traffic hijacking:** Many malicious actions by attackers can be achieved in this area; some examples are as follows such as gain access to customers' credentials, manipulate data, redirect clients to illegitimate sites and make the customer's accounts a new base for launching other subsequent attacks.
- **Insecure interfaces and APIs:** IT administrators depend on interfaces for cloud management and monitoring. APIs are integral to security and availability of general cloud services. Therefore, third parties and organizations on many occasions are known to build on these interfaces and inject advertising services or other software.
- **Denial of service (DoS) attacks:** DoS attacks can cause availability issues to one or more services. DoS attacks can cost service providers customers and can cost customers significant losses.
- **Malicious insiders***:* This can be employees with the cloud service providers or contractors with malicious purposes who may cause damage to both the customers and the cloud service provider.
- **Cloud abuse:** There are many examples of cloud abuse: A customer using the cloud service to break an encryption key too difficult to crack on a standard computer and a customer planning to launch a (DoS) attack, spread malware or any illegal activity.
- **Organizations adopting cloud technologies without understanding the associated risks***:* Before the adoption of cloud services, organizations must understand and identify the cloud computing security risks, and make a rational decision on whether the organization should take advantage of cloud computing technologies or not. This should include how optimal benefits can be obtained by cloud computing, and what security precautions must be taken.

B. Shared technology vulnerabilities

This threat exists at every type of the delivery model for a cloud service provider where a compromised component such as software, a platform, or infrastructure which can affect the whole environment. Cloud computing involves even more security risks and the security risks mentioned above are just the most common. Other types of security risks related to ownership of data, shared access, isolation failure, and virtual exploits also exist [21].

3.2 Data Security

Security can be defined as state of freedom from a danger, risk or attack. Information security can be defined as the task of guarding information which is processed by a server, stored on a storage device, and transmitted over a network like Local Area Network or the public Internet. Information security means protecting information and information systems from unauthorized access, use, disclosure, disruption, modification or destruction [4].

- AAA stands for Authentication, Authorization and Accounting. It is a set of primary concepts that aid in understanding computer and network security as well as access control. These concepts are used daily to protect property, data, and systems from intentional or even unintentional damage. And secondly it is used to support the Confidentiality, Integrity, and Availability (CIA) security concept.
- **Confidentiality:** The term confidentiality means that the data which is confidential should remain confidential. In other words, confidentiality means secret should stay secret.
- **Integrity:** The term integrity means that the data being worked with is the correct data, which is not tampered or altered.
- **Availability:** The term availability means that the data user need should always be available to user.
- **Authentication** provides a way of identifying a user, typically requiring a User-id or Password inputs before

granting a session. Authentication process controls access by requiring valid user credentials.

- **Authorization** is the process that determines whether the user has the authority to carry out a specific task. Authorization controls access to the resources after the user has been authenticated.
- **Accounting** keeps track of the activities the user has performed in the server.

3.3 Data Protection in the Cloud Environment

3.3.1 Data at Rest:

The data at rest refers data that has been saved to database in permanent storages. Generally, data at rest is encrypted by a symmetric key [22]. Disk encryption is a technology which protects information by converting it into unreadable code that cannot be deciphered easily by unauthorized people. Disk encryption uses applications such as disk encryption software or hardware tools to encrypt every bit of data that goes on a disk or disk volume. It is used to prevent unauthorized access to data storage. Database encryption can generally be defined as a process that uses an algorithm to transform data stored in a database into cipher-text that is incomprehensible without first being decrypted. It can therefore be said that the purpose of database encryption is to protect the data stored in a database from being accessed by individuals with potentially malicious intentions. The act of encrypting a database also reduces the incentive for individuals to hack the aforementioned database as meaningless encrypted data is of little to no use for hackers.

3.3.2 Data in Process

In the case of data in use, data is more vulnerable to network based attaches and threats that can cause various types of problems and risks up on the system. Here to bring the security in the cloud environment, the data owner as well as cloud providers should control the access as tightly as possible and incorporate some types of authentication to ensure that all stakeholders are using, accessing and manipulating data in the proper manner. The major actions the user has to take is frequently tracking and reporting the relevant information in proactive manner to detect suspicious activity, diagnose the potential threats and disabled accounts and attacked data to improve the security of the system [8].

3.3.3 Data in Transit

The TCP/IP protocol suite does not have any in-built mechanism for the protection of moving data. Protection of Data, when moving in network is crucial in computer networking. IPSec Protocol Suite provides security to the network traffic by ensuring Data Confidentiality, Data Integrity, Sender and Recipient Authentication and Replay Protection [23]. Some network threats which are mitigated by using IPSec are data corruption in traffic, data theft in traffic, passwords and account theft and network based attacks. IPSec Protocol Suite is based on Internet Engineering Task Force (IETF) standards. Since IPSec is an IETF standard, we can have interoperability between different Firewall, Router and Operating System vendors. We can use IPSec to create VPN tunnels between devices made by different vendors like Cisco, Juniper, Microsoft, RedHat, Checkpoint, Palo Alto, ...etc. IPSec (Internet Protocol Security) provides protection to Network Data Traffic (Primary Goals of IPSec) in four different ways listed below.

1) **Confidentiality:** The Data in network traffic must be available only to the intended recipient. In other words, the Data in network traffic must not be available to anyone else other than the intended recipient. IPSec provides Data Confidentiality to Data by encrypting it during its journey.

2) **Integrity:** The Data in network traffic must not be altered while in network. In other words, the Data which is received by the recipient must be exactly same as the Data sent from the Sender. IPSec (Internet Protocol Security) provides Data Integrity by using Hashing Algorithms.

3) **Authentication:** Sender and the Recipient must prove their identity with each other. IPSec provides Authentication services by using Digital Certificates or Pre-Shared keys.

4) **Protection against Re-play Attacks**: Network Re-play attacks also called as man-in-the-middle attack which allows an attacker to spy the network traffic between a sending device and a receiving device. Later, the Re-play attacker uses the information he gained illegally for fake authentication, fake authorization or to duplicate a transaction. IPSec protects against Re-play attack by using sequence of numbers which are built into the IPSec packets. By using this sequence numbers, IPSec can identify the packets which it has already seen [9].

3.4 User Privacy Preserving and Data Security Approaches

There are two approaches those are used to keep security and preserve privacy of data and user of data within the cloud system.

A. End-to-End Encryption (Data-Owner→Internet→Cloud Storage→Data User)

This approach is most preferable and most secured way of cloud data communication and it preserves the privacy of data and user from source to destination of the data because the data is remain encrypted within Internet and cloud storage that prevents unauthorized access of data and even cloud vendors cannot access the data. Here, the data is decrypted only on the user side since the user owns the private key and can decrypt the data and can use data for his/her business purpose. So it is called end-to-end encryption of data as shown Figure 3.1 below.

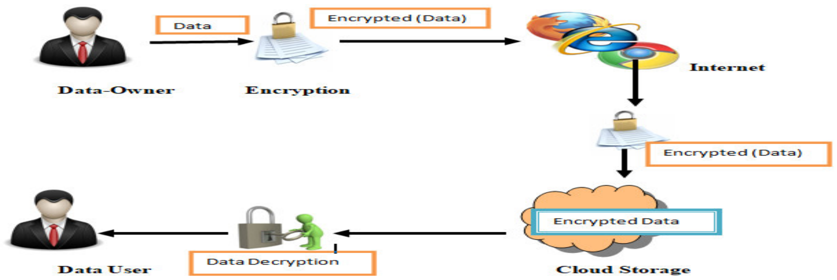


Figure 3.1 End-to-End encryption of data

B. Half-way Encryption (Owner→End of Internet)

This approach also used to encrypt data up to cloud system and decrypts data, then stores the decrypted data in the cloud storage so the cloud vendor and unauthorized user can access and manipulated the data easily because it is non-encrypted. Finally the end users access the decrypted, which is not secured by the cloud system as shown Figure 3.2.

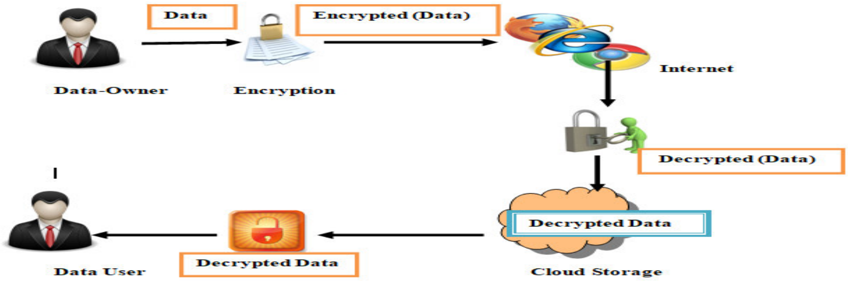


Figure 3.2 Data owner to cloud server data encryption approach

3.5 Proposed System and Its Model

The Figure 3.3 shows the general architecture of the proposed work and its details is explained as follow:

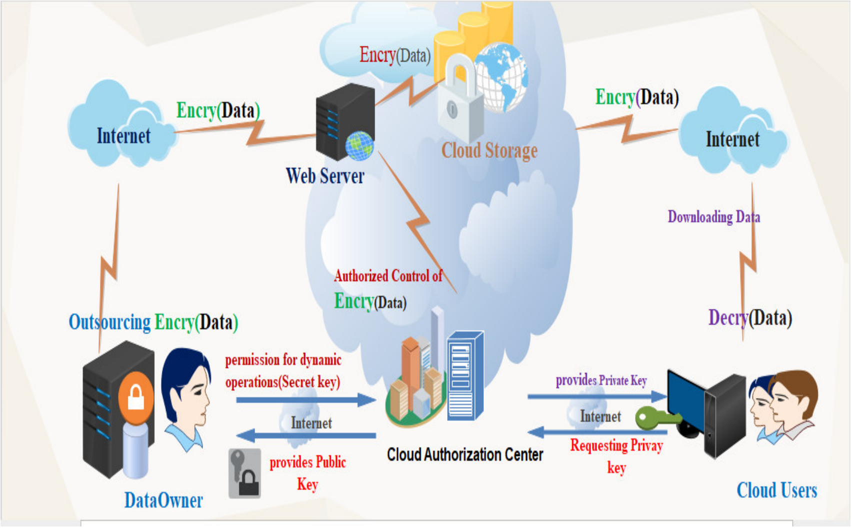


Figure 3.3 Proposed system model

3.6 Contributions of the proposed system and Components

The proposed system has four major components, these are listed as follows, such as data owner, cloud administration, data consumer or customer and cloud authorizer system. This system more deeply explains, investigates and implements the document oriented data in cloud by using MongoDB, security issues of data and user in cloud computing, preserves the privacy of user and data in cloud in the two techniques such Data privacy at storage and data privacy in process /transit, resolves the storage problems in cloud by applying virtualization techniques to create virtualized environment, data encryption and data decryption processes and lastly proffers secure data access techniques and procedures with testing tools.

3.7 Data Security and Homomorphic Cryptosystem

3.7.1 RSA Cryptosystem

This cryptosystem is one of the initial cryptosystem and it remains most employed cryptosystem even today. The system was invented by three scholars Ron Rivest, Adi Shamir, and Len Adleman of MIT in 1977, and hence, it is termed as RSA cryptosystem. We will see two aspects of the RSA cryptosystem, firstly generation of key pair and secondly encryption-decryption algorithms [24, 25].

3.7.2 Generation of RSA Key Pair

Each person or a party who desires to participate in communication using encryption needs to generate a pair of keys, namely public key and private key. The process followed in the generation of keys is described below,

- **Generate the RSA modulus (n):** Select two large primes, p and q. Computing their system modulus N=p*q. For strong unbreakable encryption, let N be a large number, typically a minimum of 512 bits. Note that, **Ø** (N) = (p-1)(q-1)
- **Find Derived Number (e):** selecting at random the encryption key **e where 1<e< Ø (N), gcd(e, Ø(N))=1.** Number e must be greater than 1 and less than (p − 1)(q − 1). There must be no common factor for e and (p − 1)(q − 1) except for 1.
- In other words two numbers e and (p – 1)(q – 1) are co-prime.
- **Form the public key**
- The pair of numbers (N, e) forms the RSA public key and is made public. Interestingly, though n is part of the public key, difficulty in factorizing a large prime number ensures that attacker cannot find in finite time the two primes (p & q) used to obtain n. This is strength of RSA.
- **Generate the private key:** solve following equation to find decryption key d

e*d=1 mod Ø (N) and 0≤d≤N

Private Key d is calculated from p, q, and e. For given N and e, there is unique number d. Number d is the inverse of e mod (p - 1)(q – 1).

This means that **d** is the number less than (p - 1)(q - 1) such that when multiplied by e, it is equal to 1 modulo (p - 1)(q - 1). This relationship is written mathematically as follows,

e*d = 1 mod (p − 1)(q − 1). The Extended Euclidean Algorithm takes p, q, and e as input and gives d as output. And lastly publish their public encryption key: PUBK={e,N} and keep secret private decryption key: PRK= {d, p, q}

Encryption Data: To encrypt a message M the sender, and it obtains **public key** of recipient PUBK={e,N}, lastly, Computes: $C=M^e$ mod N, where 0≤M<N

Decryption: To decrypt the ciphertext C the owner and it uses their private key KR={d, p, q}

Computes: $M=C^d$ mod N, note that the message M must be smaller than the modulus N block if needed [26, 27].

3.7.3 Algorithm and Architecture

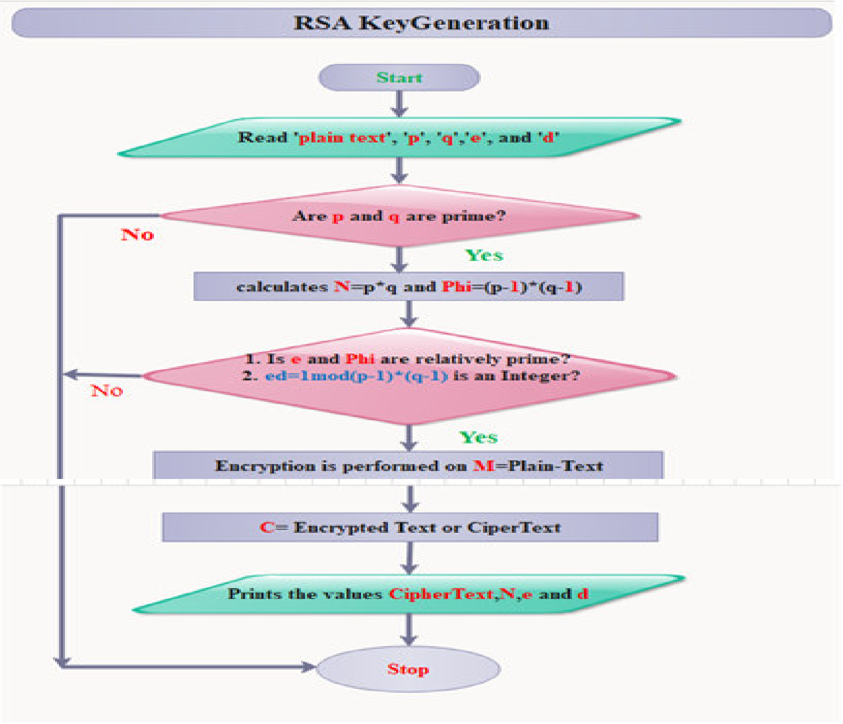

RSA KeyGeneration
Start
Read 'plain text', 'p', 'q','e', and 'd'
Are p and q are prime?
No
Yes
calculates N=p*q and Phi=(p-1)*(q-1)
1. Is e and Phi are relatively prime?
2. ed=1mod(p-1)*(q-1) is an Integer?
No
Yes
Encryption is performed on M=Plain-Text
C= Encrypted Text or CiperText
Prints the values CipherText,N,e and d
Stop

Figure 3.4 RSA Algorithm architecture

Paillier's Additive Homomorphic Cryptosystem and Self-Blinding

Pascal Paillier's introduced his cryptosystem in 1999 published Public-Key Cryptosystem based on Composite Degree Residuosity Classes [28] [29].

a. Definition of Paillier's Cryptosystem

It is one of asymmetric encryption and decryption algorithm which is an additive homomorphic scheme; this means that given only the public key and the encryption of M1 and M2, one can compute the encryption of M1 + M2. Like other the cryptosystem such as RSA and Elgamal, it also follows key generation procedures by taking two prime numbers p and q and chooses one random integer g, that g ε $Z^*_n{}^2$ (i.e., g in invertible modulo n^2) such that n and L(g $^\lambda$ **mod** n^2).

L: $\mathbf{Z^*_n{}^2} \rightarrow \mathbf{Z_n}$, u $\mapsto$ u-1) /n and λ denotes the Carmichael functions λ(p*q) = lcm(p-1,q-1). The public key is the tuple (n, **g**) and the secret or private key are the two prime factors (p, q). So the more details of key generation and algorithms are explained as follows:

b. Paillier's Algorithm and key derivation procedures

To understand how Paillier's Cryptosystem, one should have knowledge on the following basic mathematical concepts and theorems.

1. Greater common divisor (gcd) of two or more non-zero integers is the largest positive integer that divides the numbers without a remainder. The gcd of a and b is written as gcd (a, b) or sometimes simply as (a, b). Two numbers are called coprime or relatively prime if their gcd is equaled to 1.
2. Least common multiple ***(lcm)*** of two or more non-zero integers is the smallest integer that is divisible by every member of a set numbers without a remainder. It is a remarkable fact ***ab = lcm (a, b)*gcd (a, b),*** thus ***lcm (a, b) = (a*b)/gcd(a, b),*** this fact is can easily be seen that ***gcd (a, b)*** is product of the common prime factors of ***a*b***, and the remaining factors would results ***lcm(a, b).***
3. Euler's Totient function (Ø function). The Totient of a positive integer n is defined to be the number of positive integers less than or equal to n that are coprime to n. If n is factorized to distinct prime numbers **p** and **q**, then ***Ø (n) = (p-1)*(q-1).***
4. Carmicheal's function (Lambda function) is given by the least common multiple (lcm) of all the factors of the Totient function ***Ø (n).*** If n can be factorized to prime number **p** and **q**, i.e
λ = lcm (p-1)*(q-1).
5. The modular multiplicative inverse of an integer α modulo m is an integer x such that
$\alpha^{-1} = x$ mod m, this is equivalent with ***αx = 1 mod m***. Notice: the multiplicative inverse of α modulo m exists if and only if α and m are coprime **(gcd (α, m) =1).**
6. Frequently used notations in Paillier cryptosystem,

Z_n – set of integers
$Z_n{}^*$ - set of integers coprime to n, this set consists of Ø (n) numbers of integers
$Z^*_n{}^2$ – set of integers coprime to n^2, this set consists of nØ(n) number of integers.

Key generation

Choose two large prime numbers p and q randomly and independently of each other such that gcd (p*q, (p-1)*(q-1)), this property is assured if both primes are of equal length.

Compute ***n = p*q*** and ***λ = lcm(p-1)*(q-1)*** and Select random integer **g** where **gε** $\mathbf{Z^*_n{}^2}$

Order of g is multiple of n in $\mathbf{Z^*_n{}^2}$**,** $\mathbf{Z^*_n{}^2}$ **is a unit or invertible elements of** $\mathbf{Z^*_n{}^2}$

Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse***: μ =(L(g***$^\lambda$ ***mod n***^{2} ***))***$^{-1}$***mod n***, where function L is defined as *L(x) =* $^{x-1}$*/n* . Note that the notation **a/b** does not denote the modular multiplication of **a** multiple of the modular multiplicative inverse of **b** but rather the quotient of **a** divided by **b**, i.e., the largest integer value ***v>=0*** to satisfy the relation **a>=v*b**. The public (encryption) key is (n, g) and the private (decryption) key is (λ, μ). If using p, q of equivalent length, a simpler variant of the above key generation steps would be to set **g = n +1**, and **λ = Ø(n), μ = Ø(n)**$^{-1}$ **mod n,** where **Ø(n) = (p-1)*(q-1).**

Encryption

Let M be a message to be encrypted where M ε $\mathbf{Z_n}$ **and Select random r where r ε** $\mathbf{Z^*_n}$**, Compute ciphertext as:** $\mathbf{C = g^M * r^n \bmod n^2}$

Decryption

Let C be the ciphertext to decrypt, where Cε $\mathbf{Z^*_n{}^2}$ **and Compute the plaintext message as,**

M = L(C $^\lambda$ **mod n**^{2}**) * μ mod n where as μ = ((p-1) * (q-1))**$^{-1}$ **mod n.**

c. Homomorphic addition of plaintexts

A notable feature of the Paillier's cryptosystem is its homomorphic properties along with its non-deterministic encryption. As the encryption function is additively homomorphic, the following identities can be described:

➢ Multiplying encrypted messages results in the addition of the original plain Texts mod n, that means the product of two ciphertexts will decrypt to the sum of their corresponding plaintexts,

$D(E(M1,r1) * E(M2, r2) \bmod n^2) = M1 + M2 \bmod n$

- The product of a ciphertext with a plaintext raising g will decrypt to the sum of the corresponding plaintexts,

$D(E(M1, r1) * g^{M2} \bmod n^2) = M1 + M2 \bmod n.$

d. Homomorphic multiplication of plaintexts

An encrypted plaintext raised to the power of another plaintext will decrypt to the product of the two plaintexts, the Paillier's cryptosystem takes full advantages of Carmicheal's Theorem, as evident in the following properties. Note: D(u) means decryption u, E(u) means encryption u.

$D(E(M1, r1)^{M2}) \bmod n^2) = M1*M2 \bmod n$ and

$D(E(M2, r2)^{M1}) \bmod n^2) = M1*M2 \bmod n$

Property 1:

Formally: $D[E(M1) *(M2) \bmod n^2] = M1 + M2 \bmod n$,

Proof: let $C_1 = g^{M1} *r_1^n \bmod n^2$,

$C_2 = g^{M2} *r_2^n \bmod n^2$, then

$C_1 * C_2 = (g^{M1} *r_1^n \bmod n^2) *(g^{M2} *r_2^n \bmod n^2) = [g^{M1} * g^{M2} * r_1^n * r_2^n] \bmod n^2 = [g^{M1 + M2} *(r_1 * r_2)^n] \bmod n^2$ **,** here both r1,r2 ε Z^*_n and also r1*r2 εZ^*_n , so this the encryption of a new message, M1 + M2, the random elements r1*r2.

Decrypting the product of C_1 and C_2 begins by raising $(C_1 * C_2)$ to the $\lambda(n)$ which is more explained as follows,

$\mathbf{(C_1 * C_2)^{\lambda(n)} = g^{\lambda(n)(M1 + M2)} *((r_1 * r_2)^{n* \lambda(n)}) = g^{\lambda(n)(M1 + M2)} \bmod n^2}$ by Carmicheal's Theorem. It is still true that $\mathbf{g^{\lambda(n)} = (1 + n)^{\lambda(n)[[g]](1 + n)} \bmod n^2}$. So substituting, we have

$\mathbf{(C_1 * C_2)^{\lambda(n)} = (1 + n)^{\lambda(n)[[g]](1 + n)*(M1 + M2)}= 1 +(M1 + M2)* \lambda(n)[[g]](1 + n) *n \bmod n^2}$, so that

$L((C_1 * C_2)^{\lambda(n)} \bmod n^2) = (M1 + M2) * \lambda(n)*[[g]]*(1 + n) \bmod n$,

μ is still inverse of $\{ \lambda(n)[[g]](1 + n) \}$, so completing decryption process by multiplying by μ yields M1 + M2 mod n.

Property 2: A full encryption of the second message is, in truth, un-needed in the above property and the r^n calculation in the second encryption can be left out while attaining the same results.

Formally: ***$D(E(M1)* g^{M2} \bmod n^2) = M1 + M2 \bmod n$*** or it will give the next formula after applying random number r,

$D(E(M1, r1) * g^{M2} \bmod n^2) = M1 + M2 \bmod n$

Proof: let $C_1 = g^{M1} *r_1^n \bmod n^2$, here take into account g^{M2},

Their product is,

$C_1* g^{M2} = (g^{M1} *r_1^n) * g^{M2} \bmod n^2 = g^{M1} +^{M2} * r_1^n \bmod n^2$

Notice: The right-hand side is, as far as decryption is concerned, exactly the same as the above right-hand side. There is g raised to the (M1 + M2) power, and an element of Z_n^* raised to n, if this element contains r1, r2, r1*r2, or some other $r_{i,}$ it does not matter, because in the first step of decryption, this element is raised to the $\lambda(n)$*n power, and yields one by Carmicheal's Theorem:

$(C_1* g^{M2})^{\lambda(n)} = g^{\lambda(n)(M1 + M2)} * r_1^{n \lambda(n)} = g^{\lambda(n)*(M1 + M2)} \bmod n^2$

At this point, it should be clear that the right-hand side is exactly identical to the right-hand side of raising $(C_1 * C_2)$ to the $\lambda(n)$ in the above property, and so the rest of the decryption process is identical and so it is omitted. The result of continuing decryption from this point will give the same result as above and the result will be (M1 + M2) mod n, thus making the decryption of the product of an encrypted message with a message essentially encrypted with r = 1 and still it is, M1 + M2.

Corollary to property 2: Self-blinding process

In this technique, one can change the ciphertext without affecting the original value of the plain text and it is used to identify error in between original ciphertext C and self-blinded ciphertext C', so after decryption process both C and C' should result in the same plain text unless there will probably error within C and C' [30].

Formally: ***$D[E(M)*g^{n*x} \bmod n^2] = n*x + M = M \bmod n$***

Proof: This is direct proof from property 2 by letting M1 = M, and M2 = n*x.

Notice: The final calculation in decrypting the message in Paillier's Cryptosystem is done mod n, and the messages must be broken into blocks such that each blocks, M ε Zn. Here one does not take care of the final reduction of ***n*x***, and no need of decrypting M2 because it is work for *n*x* not an element of Zn. The original ciphertext, C, becomes completely different ciphertext,

$C' = C* g^{n*x} \bmod n^2$, but because of n*x = 0 mod n, the original message, M, is still as it is after the decryption of ciphertext C'. Eventually, if C and C' are sent over a channel, where C' is a self- blinded copy of C, then the transmission error can be identified if C and C' do not decrypt to the same plain text message.

Property 3: Raising a cipher text to constant power yields the constant multiple of the original plain text.

- Formally: $D[E(M)^k \bmod n^2] = k*M \bmod n$
- Proof: Let $C = g^{M}*r^n \bmod n^2$, then C= E(M) and

$E(M)^k = C^k = (g^M*r^n)^k = g^{M*k} * r^{n*k} \bmod n^2$ and the decryption process is as follows,

$(C^k)^{\lambda(n)} = (g^{M*k} * r^{n*k})^{\lambda(n)} = g^{M*k*\lambda(n)} * r^{n*k*\lambda(n)} \bmod n^2$

Notice: $r^{n*k*\lambda(n)} = (r^k)^{*n*\lambda(n)} \bmod n^2$, where r^k is element of Z_n^* , so Carmichael Theorem says that raising it to the $\lambda(n)*n$ power makes it congruent to 1 mod n^2

So, $(C^k)^{\lambda(n)} = g^{M*k*\lambda(n)} = (1+n)^{\lambda(n)*[[g]](1+n)*M*k} = 1 + M*K*\lambda(n)*[[g]](1+n)*n \bmod n^2$

Then $L((C^k)^{\lambda(n)} \bmod n^2) = M*k*\lambda(n)*[[g]](1+n) \bmod n$, and

$\mu * L((C^k)^{\lambda(n)} \bmod n^2) = \lambda(n)^{-1}([[g]](1+n))^{-1}*M*k*\lambda(n)*[[g]](1+n) = M*k \bmod n$

More generally, an encrypted plaintext raised to a constant **K** will decrypt to the product of the plaintext and the constant,

❖ $D(E(M1, r1)^K) \bmod n^2) = K*M1 \bmod n$

Property 3: Raising an encrypted message to the power of a second message yields in the multiplication of plain text messages.

Formally: $D[E(M1)^{M2} \bmod n^2] = D[E(M2)^{M1} \bmod n^2] = M1*M2 \bmod n$

Proof: this property follows directly from property 3 by letting M = M1 and k = M2 or by letting M = M2 and k = M1.

However, given the Paillier's encryptions of two messages, there is no known way to compute an encryption of the product of these messages without knowing the private key [31].

e. Key Generation in Paillier's cryptosystem

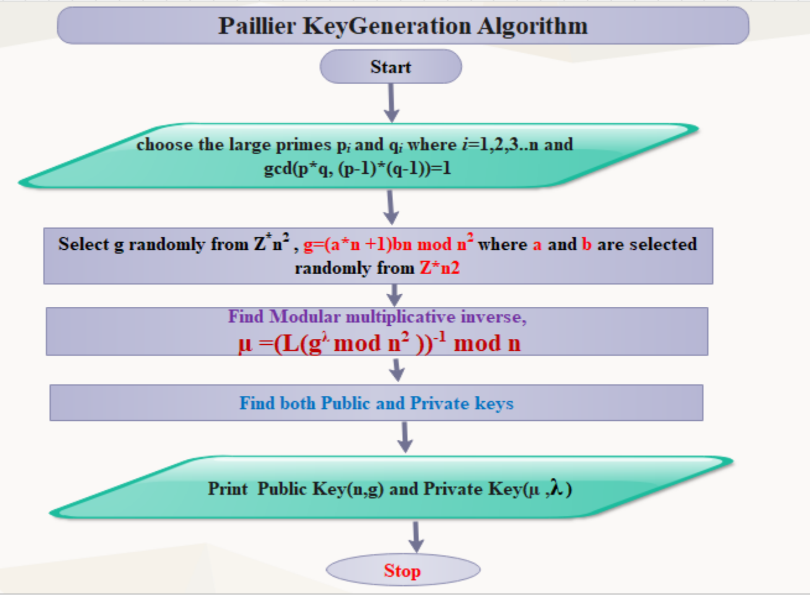

Paillier KeyGeneration Algorithm
Start
choose the large primes p_i and q_i where i=1,2,3..n and gcd(p*q, (p-1)*(q-1))=1
Select g randomly from Z^*n^2 , g=(a*n +1)bn mod n^2 where a and b are selected randomly from Z*n2
Find Modular multiplicative inverse,
$\mu = (L(g^{\lambda} \bmod n^2))^{-1} \bmod n$
Find both Public and Private keys
Print Public Key(n,g) and Private Key(μ ,λ)
Stop

Figure 3.5Paillier's key generation

a) Input: A plain Text message

b) Output: An encrypted message for multiple participants

Step 1: Initialize the variables, $p_1, p_2, p_3, ..., p_n$ and $q_1, q_2. q_3, ..., q_n$, such that the variables should be large primes.

Step 2: Generate the public key for each participants $n_1, n_2, n_3, .., n_k$, such that n = p*q.

Step 3: Choose the semi-random variable 'g'

Step 4: Calculate the encrypted values for each participant by using the given formula.

Equation 1

$C = g^M * r^n \bmod n^2$ ----------------------------Equation (1)

a) Input: An encrypted message divides among many participants C_i

b) Output: The plain Text message among participants

Step 1: Calculates the values of λ by the lcm of p_i and q_i

Step 2: Evaluate the values 'u' by applying the formula

$u_i = g^{\lambda(ni)} * r_i^{ni} \bmod n^2$

Step 3: Calculate the inverse of the L(u_i) which will give the plain text value, by applying the following two formulas;

Equation 2

L(u) = (u-1)/n ---------------------------------------Equation(2)

Equation 3

$L(g^{\lambda(n)} \bmod n^2) = k$ ------------------------------ Equation(3) or

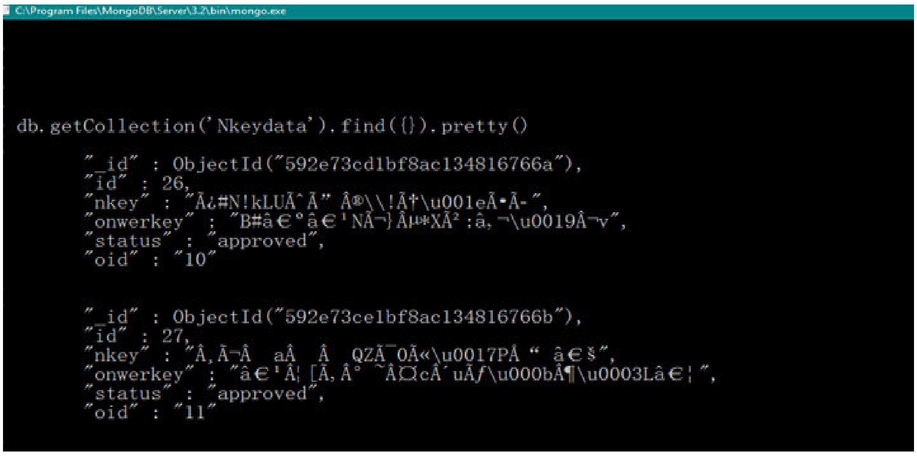
$M = L(\mathbf{L(C^{\lambda} \bmod n^2) * \mu \bmod n})$,

where as $\mathbf{\mu = ((p-1) * (q-1))^{-1} \bmod n}$

4. Implementation of Encryption Algorithms on Document Data

4.1 Security of Data at Rest

Database encryption can generally be defined as a process that uses an algorithm to transform data stored in a database into cipher-text that is incomprehensible without first being decrypted. It can therefore be said that the purpose of database encryption is to protect the data stored in a database from being accessed by individuals with potentially malicious intentions. Here the document encryption is carried out by built-in AES algorithm. The act of encrypting a database also reduces the incentive for individuals to hack the aforementioned database as meaningless encrypted data is of little to no use for hackers [31].



```
C:\Program Files\MongoDB\Server\3.2\bin\mongo.exe

db.getCollection('Nkeydata').find({}).pretty()

        "_id" : ObjectId("592e73cd1bf8ac134816766a"),
        "id" : 26,
        "nkey" : "Ã¿#N!kLUÃˆÃ" Â®\\!Ã†\u001eÃ•Ã-",
        "onwerkey" : "B#â€°â€'NÃ¬}Âµ*XÃ²:â,¬\u0019Ã¬v",
        "status" : "approved",
        "oid" : "10"

        "_id" : ObjectId("592e73ce1bf8ac134816766b"),
        "id" : 27,
        "nkey" : "Â,Ã¬Â  aÂ  Â  QZÃ¯0Ã«\u0017PÅ "  â€š",
        "onwerkey" : "â€'Ã¦[Ã,Â° ˜Ã☐cÂ´uÃƒ\u000bÃ¶\u0003Lâ€¦",
        "status" : "approved",
        "oid" : "11"
```

4.2 Data security in process

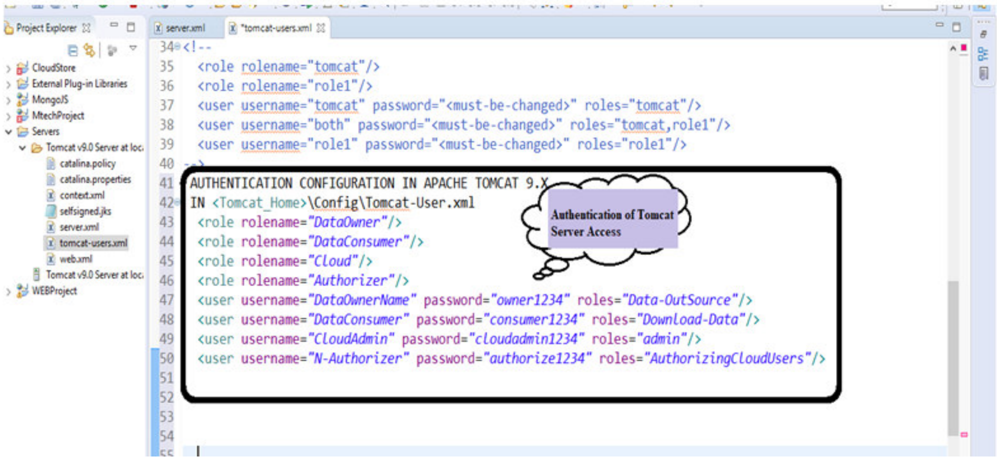
In the case of data in use, data is more vulnerable to network based attaches and threats that can cause various types of problems and risks up on the system. Here to bring the security in the cloud environment, the data owner as well as cloud providers should control the access as tightly as possible and incorporate some types of authentication to ensure that all stakeholders are using, accessing and manipulating data in the proper manner. The major actions the user has to take is frequently tracking and reporting the relevant information in proactive manner to detect suspicious activity, diagnose the potential threats and disabled accounts and attacked data to improve the security of the system [**31**].

4.2.1 Configuring Security Constraint for Applications

- **Authentication**

Configuration of security realm in *tomcat-user.xml* file

Security Realm is context that maintains username, password which could be validated while performing authentication while the user is trying to access the web applications in Apache Tomcat web server.

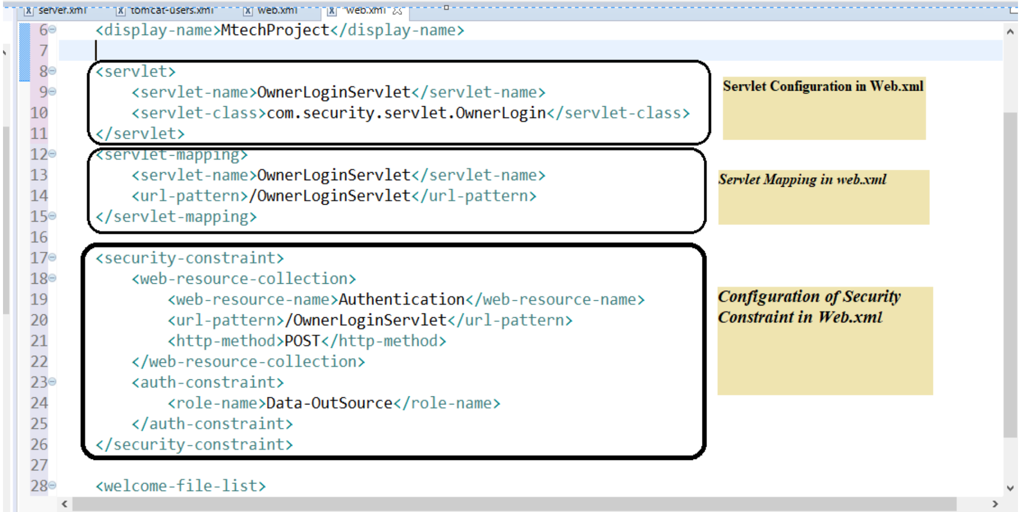


```
<!--
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
  <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
  <user username="role1" password="<must-be-changed>" roles="role1"/>
-->
AUTHENTICATION CONFIGURATION IN APACHE TOMCAT 9.X
IN <Tomcat_Home>\Config\Tomcat-User.xml
  <role rolename="DataOwner"/>
  <role rolename="DataConsumer"/>
  <role rolename="Cloud"/>
  <role rolename="Authorizer"/>
  <user username="DataOwnerName" password="owner1234" roles="Data-OutSource"/>
  <user username="DataConsumer" password="consumer1234" roles="Download-Data"/>
  <user username="CloudAdmin" password="cloudadmin1234" roles="admin"/>
  <user username="N-Authorizer" password="authorize1234" roles="AuthorizingCloudUsers"/>
```

Authentication of Tomcat Server Access

- **Authorization:**

Authorization is used to check the access permissions of the user to access certain resources of the application by assigning access permission to user roles, especially in web applications to limit access to resources.



```xml
    <display-name>MtechProject</display-name>

    <servlet>
        <servlet-name>OwnerLoginServlet</servlet-name>
        <servlet-class>com.security.servlet.OwnerLogin</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>OwnerLoginServlet</servlet-name>
        <url-pattern>/OwnerLoginServlet</url-pattern>
    </servlet-mapping>

    <security-constraint>
        <web-resource-collection>
            <web-resource-name>Authentication</web-resource-name>
            <url-pattern>/OwnerLoginServlet</url-pattern>
            <http-method>POST</http-method>
        </web-resource-collection>
        <auth-constraint>
            <role-name>Data-OutSource</role-name>
        </auth-constraint>
    </security-constraint>

    <welcome-file-list>
```

Servlet Configuration in Web.xml

Servlet Mapping in web.xml

Configuration of Security Constraint in Web.xml

4.3 Data security in Transit

The TCP/IP protocol suite does not have any in-built mechanism for the protection of moving data. Protection of Data, when moving in network is crucial in computer networking. The IPSec (Internet Protocol Security) Protocol Suite is a set of network security protocols, developed to ensure the Confidentiality, Integrity, and Authentication of Data traffic over TCP/IP network. IPSec Protocol Suite provides security to the network traffic by ensuring Data Confidentiality, Data Integrity, Sender and Recipient Authentication and Replay Protection [31]. Some network threats which are mitigated by using IPSec are:

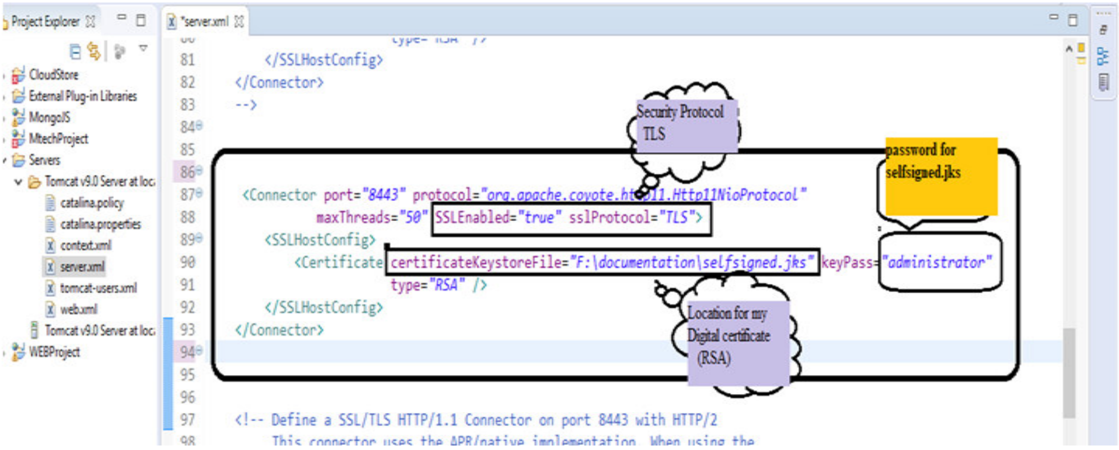
- Data corruption in traffic
- Data theft in traffic
- Passwords and Account theft and
- Network based attack

4.3.1 SSL/TLS Configuration in Tomcat server with Digital Certificate in Tomcat web server

Digital Certificate

```
þíþí       	 privacy  ]š”;   0,ü0+* 	 ,è‰³IXóÁÂÍ‘)|aº6‡cÊ8|ròPpÔ|6üýƒnbl(óID[ãXø¾dé…#ÆãUÉ1LÉðœ¦
:#8¯>ðûo¤!¾dM   É+6ttýüÇ½ª$K”22Gíö„mo[ãìÁ¢eQ0Ì“üÞ«MƒÃØÑ+pÇ¶h€S,EŸ–% Çä¢ÈHrºÄþÞÜ‡Ð°¡jkfÄÍT’½E‘Aþß§„
Äë‰Âµ?“\aùÂÃav;‡`c{Ò·-˜|ÆÚWsÐNi]’…io>ºÎ        ‡iù6x–~"úl•,.Õ€µS¸‰Œ’Q♠ë#`“ÖÝ<8!$¦ƒe7›eZŸm!€a•¾ff"AD¯
«¢€¾‰R9€ò¯8<ßmÔAUÂäÐn!½áá QÂ6>âÎÊQ}fº§Ûo´Ó½b”HexÌêu4£FdäöfÎVÆä¦0-wñL¶ºZá÷» ùhx¦2Ú,,™r—♠Ò¨Á¦,ƒîzrqx&¬
<ÔIš±µÎ" ÆBøÀxMÛÈ"×@„b{Üåp{ã<N¥aûºXÉôÌþ08¾¨8IŠ§6SíGsÕªÚnféÂQ¹~ìõßªì8ÿ.‡¤C*!ä“ž±)©öª¿9ƒH«Ú¶ø<5Ðß¢Ž@=¯
/57&ß~ê¤m:&S{ù²ó¬ãî¦ýy|Øh¹IkÊïÃ-ÉH³Ö*ÌŸ3Ëºµ+ƒDnº"1/Ó‘=à–¨^TóÎø+·jèá)pÊÅëÅS[ÉáC)Vi{™iÖé{) H†3å’äÏª,ó
 òcBA cs)Ð‘Ép`KÛVübAwêØN{®Ô¡¦Á½bÐ3V@Ö-ŸÙ~d™PGµ–C†‡ò=+w³PF›oü=-ÌŸõ<Í 8;¤õƒ DÉÑý*ç]ò«2=ª,<…3©@çZæÊ¡UA;
„çÙeÎj©%Æ{Ò…B:Þ«\‘ZÂÚmÛ2[x7ù ÏÅüqƒe^ÇNå™MeÙrŠ§Kx‘zcb¡j“2²xòØ„š8˜Ãÿ¦’T}©º5Î›♠Ñ‡Éii¡}¤ºêq v¾”}£ªvõ¹à?
Äâu+9ÿImäÅ,å"F³cš€A¦Ø¢M2"#š½kìÙa^ËõÚbÏÕžÅmÞíÞ2¢¿[ÞKµ’ˆ·2§ê@³!’ÿêr¡Hm·®Ê†FZÝ%áÑÔº·¼+íh=ÖK¥ˆ}˜å)a–ÁM‰
çGì1êiÃ‘~{9ÌÃsÙõgu«9Aƒ ŠÅžæu˜æŽL [å”Xœ§*÷‘ëàÊT´yN[Áï^YÐ½\úèðN¸ ædN_šWxÚx6VoÕÿ“ö.e\›Úoãº Ó¿ì=fÛÒÏøóì½ ?
d&àé›ºJ+d±RO,éã³%ƒsqð¾Aƒ)õu     Äÿý8Ðg0ÿsER“³îQÕ·ö`¡q<tIÇíy     X9CØn:Ncÿ9 Â¬²S&ØÔ¡uÂ…zÊñ¦Ø¥xRWþr       >
GâXÛxor|bæA0HsæçSÜë™ŽÓ,JÚ—µeeì£Ök¼6!y eÿz·~ó9Ê÷n½i—Ê:›_ÀÌø0zdÏyyRgLQƒ5£í=OÒOÎÛ'¶     X.509  Û0,×0, ¿  
        ^0     *†H†÷ 0›10     U IN10U     Telangana1#0!UHyderabad, Telangana state1 0UEngineering
 College,CSE10UOsmania University10UElias Bassa0170731211354Z180127211354Z0›10 U IN10U
Telangana1#0!UHyderabad, Telangana state1 0UEngineering College,CSE10UOsmania University10U
Elias Bassa0,"0        *†H†÷ ,  0,  , ¶Š Ø*—ñR§UHdPu%Þü.øý¸Ë.QJ¦%rÒJ‘—˜–ê‹ÊÚ(‡LÃ_h‘ÐžðBÏþ1(‰hÈº
º§÷úŽ@î^¼ÉÅ•’ÓQCs#pFÓÛÛžžº‰¸ðým45gp’E)à ÷M¶"<Žbm£(Š?9¯¦aš9f¸ h/Áµ¦Ðu T´J"b·î[ë“W“D ÊÄ¶ý£èN}r®4jþ‹V;1NÊttg
¢Y÷¿jÅág´poiã†u R¦Î–5~Òk©íI}ÃöÂ–Ê$2§FvWÔÒc´–Ùê*®ÛFdu™ÙP±y¿PÏ}Î‘³  £!00UÎø¬Qs†ÁŽ$mç—♠Îõ-Ï.0        *†H†
÷ ,  WÖ^á?¢þ€ýÕJ£´¿ÔÄ*õ<ã'‰éÔžF¿@Á<ëØâz!ÎåÕæ ♠ñû›6ûÎå˜u    ®'õŽømÐ6Þçèµ ÇZÕ^è"Ðc¦Y¥4-È±"Gâ«,qý~^ ÖëH»Qy
|Fè»P®`|XYu     ÐÂýq]–|U÷;8‰ñS¨Ž|½ÀŠÛô±¸ñ=W{/¡ÐêC0^šÚ5~—Ö]²œ(ëb‡ÈðkÙoÐ¸ºú1³Tx«p02ð„cn“¶–P·+ÕýÐuÛ½Ê—ì"Õ
Œ±QYÎ÷trçM)ÃHx-ÚZ¿`›rìTM(aCmáA¹¸¢Tµá¬1«ÍŒÀÖ²¡ï¤éÐ
```

Configuration of Digital certificate



```xml
        </SSLHostConfig>
    </Connector>
    -->


    <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
               maxThreads="50" SSLEnabled="true" sslProtocol="TLS">
        <SSLHostConfig>
            <Certificate certificateKeystoreFile="F:\documentation\selfsigned.jks" keyPass="administrator"
                         type="RSA" />
        </SSLHostConfig>
    </Connector>



    <!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443 with HTTP/2
         This connector uses the APR/native implementation. When using the
```

Security Protocol TLS

password for selfsigned.jks

Location for my Digital certificate (RSA)

5 Results and Discussion

5.1 Experimental Design and Setup

These all experiments are conducted on document oriented data in NoSQL database system on Windows 10. The project work implemented on the Full Homomorphic (Pailliar's cryptosystem and RSA) encryption and decryption scheme to preserve privacy and security of the unstructured document data specially JSON data in the MongoDB 3.2.0 Server. All these experiments of solutions are carried out on Java programming language with JDK 1.8, Python 3.2.0 and the web server with Apache Tomcat 9.0. The prototype system ran on Laptop computer with Intel CORE i3 2.0 GHz processor, 1TB Hard Disk and 4GB Memory and lastly this project is implemented on Amazon Web Service mLab for MongoDB as cloud data storage.

5.2 Experiment Results and Performance of Algorithms

These experiments have been done to test the privacy preserving and security of input data in this project work solution. The experiment was done to verify the security level the encryption/decryption processes, public and private key generation process, time complexity and space complexity.

5.3 Security Performance of RSA Cryptosystem

This project work randomly Sampled data to Generate Public and Private Key in RSA: here the size of the data is randomly selected to generate both the public keys (e, N) and private (p, q, d) keys in RSA cryptosystem.

5.4 Performance of Data Encryption in RSA Algorithm

As shown **on** Figure 5.1, the values of N is rapidly increases with big integer values intervals while the values of **e** is increases steadily for a given small size plain text data and finally output which is cipher text also increases rapidly as the value of N increases. Generally, as analyzed in this paper work below the size of output increases, the data encryption time increases and more memory place to cipher text is required as the size of key increases.

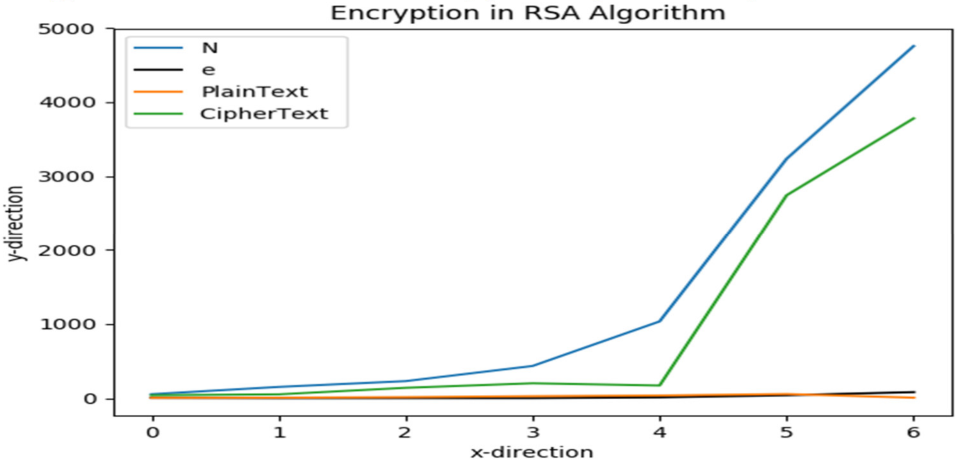


Figure 5.1 Encryption of data in RSA

5.5 Performance of Data Decryption in RSA Algorithm

As shown on Figure 5.2, decryption process in RSA is inverse of encryption process in RSA, here the decrypting data takes less time, and plain text takes less memory place even if the value of all private key increases or regardless of decryption key sizes.

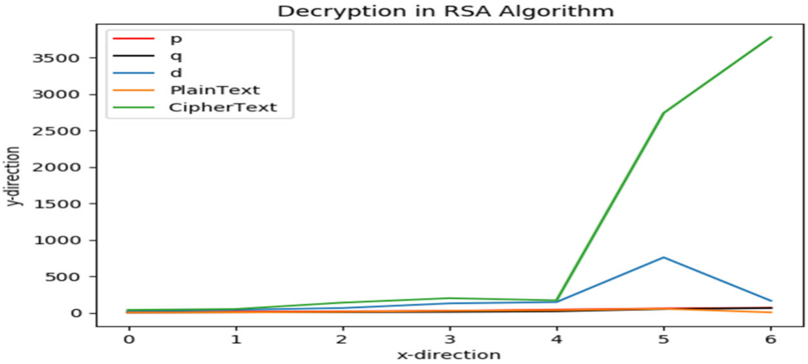


Figure 5.3 Decryption of data in RSA

5.6 Security Analysis of Paillier's Cryptosystem

Paillier's cryptosystem is a homomorphic cryptosystem in which this paper work performed arithmetic operations on the cipher text. In this case we can take two numbers (A and B) and use a public key to make them into ciphertext values. Again we can add the cipher values to get sum of ciphertexts and when we decrypt the sum of ciphertexts, it yield the sum of the two plaintexts value. This also works for multiplication. In real applications, the modulus N and decryption key d are both hundreds of digits long and the smaller the encryption or decryption keys, the less time to perform encryption and decryption operation.

5.7 Performance of Data Encryption in Paillier's algorithm

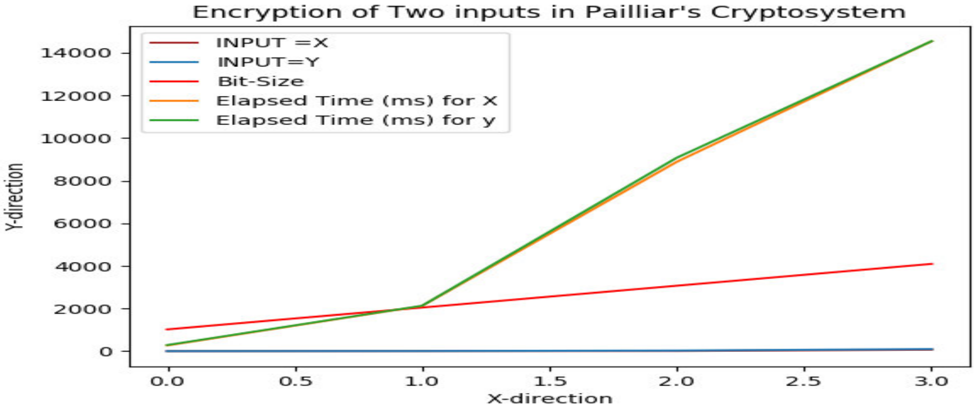


Figure 5.3 Data Encryption in Paillier's algorithm

5.8 Performance of Decryption of Data in Paillier's cryptosystem

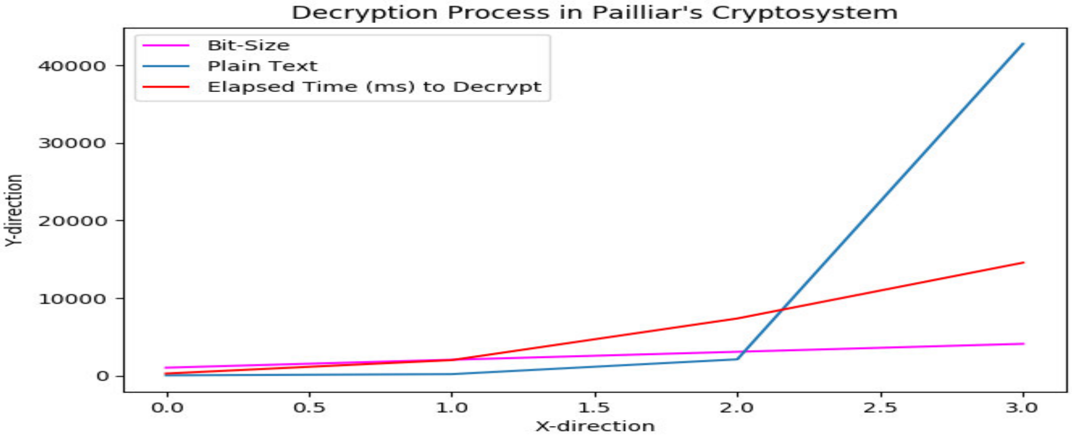


Figure 5.4 Data Decryption performance in Paillier.

6. Conclusion and Future Works

This paper work most deeply investigated the areas of cloud computing, its components, architectures, functions and security requirements as well its techniques how to implement and secure the cloud data, its delivery models, services and IT industries those are providing cloud services. Next to this, the concentration of this paper work is privacy preserving and security issue regarding user privacy, data privacy within cloud computing environments. Here to ensure the privacy and security of data as well user privacy, this paper work put into practice, the End -to-End data encryption approach by applying homomorphic encryption scheme for encryption and decryption process and RSA digital certificate to create the TLS/SSL certificate to make secure and encrypted communication between client machine and web server by providing user privacy via authentication and authorization of users to access resources within server in areas of public key infrastructure. Finally, this paper work implemented unstructured document data in the cloud by using MongoDB as local database server and Amazon Web Service mLab for MongoDB is Database as a Service in the cloud as a data storage. And these all experiments were conducted to proffer the security and privacy issues as proposed. To extend and improve this work further, some other security techniques and procedures such as obfuscation, self-blinding, signature blinding, randomization and anonymization will be implemented.

7. REFERENCES

[1] Dr.L.Arockiam, and S. Monikandan, "Efficient cloud storage confidentiality to ensure data security" 2014 International Conference on Computer Communication and Informatics (ICCCI-2014), Jan 03-05, 2014, Coimbatore, INDIA.

[2] Virtualization: Current Benefits and future potential technology concepts and business considerations, January 2011.

[3]The NIST Definition of Cloud Computing Peter Mell Timothy Grance, september2011.

[4]"https://social.technet.microsoft.com/wiki/contents/articles/30889.cloud-computing-and-security-challenges.aspx".

[5] "http://journals.sagepub.com/doi/full/10.1155/2014/190903".

[6]"http://www.networkcomputing.com/storage/data-protection-cloud-basics/798260399".

[7] Swapnali More, and Sangita Chaudhari. "third party public auditing scheme for cloud storage".7th International Conference on Communication. Computing and Virtualization 2016, Available on "http://www.sciencedirect.com".

[8] "Cloud security and privacy", by Tim Mather, Subra Kumaraswamy and Shahed Latif.

[9]"PrivacyEnhancingAggregationTechniquesforSmartGridCommunications", by Rongxing Lu.

[10] All-of-Government Cloud Computing: Information Security and Privacy Considerations April 2014.

[11] Privacy in Cloud Computing, ITU-T Technology Watch Report, March 2012.

[12] "Data Security and Privacy in Cloud Computing", Yunchuan Sun, Junsheng Zhang, Yongping Xiong and Guangyu Zhu, Published 16 July 2014.

[13] M.Nabeel and E.Bertino, "Privacy preserving delegated access control in the storage as a service model", in Information Reuse and Integration (IRI), 2012 IEEE 13th International Conference on 2012, pp. 645-652.

[14] John Wiley and Sons, cloud computing principles and paradigms Book, 2015.

[15] "https://en.wikipedia.org/wiki/Document-oriented_database".

[16] R. Cattell, "Scalable SQL and NoSQL data stores," ACM SIGMOD Record, vol. 39, no. 4, pp.12–27, 2011.

[17] A. Lakshman and P. Malik, "Cassandra: Structured storage system on a P2P network," in Proceedings of the 28th ACM Symposium on Principles of Distributed Computing, 2009, p. 5.

[18] E. A. Brewer, "Towards robust distributed systems," presented at the Principles of Distributed Computing, Portland, OR, July, 2000.

[19] Website: "www.tutorialspoint.com/MongoDB/".

[20] Analysis of RSA algorithm using pu programming Sonam Mahajan and Maninder Singh.

[21] Implementing secure RSA cryptosystem by using your own cryptographic JCE provider by Kefa Rabah.

[23] "NIST cloud reference architecture, US, department of commerce" or online:" https://bigdatawg.nist.gov/_uploadfiles/M0007_v1_3376532289.pdf".

[24] S. De Capitani di Vimercati, S. Jaljodia, S. Paraboschi, G. Pelosi, and P. Samarati, "Encryption-Based Policy Enforcement for Cloud Storage", in Distributed Computing System workshops (ICDCSW), 2010 IEEE 30th international conference on, 2010, pp. 42-51.

[25] A.Zych, M. Petkovic and W.Jonker, " Efficient key management for cryptographically enforced access control", Computer Standards and amp interfaces, vol.30,pp.410-417,2008.

[26] P. Paillier. Public-key cryptosystems based on Composite Degree Residuosity classes. Advances in Cryptology Eurocrypt, 1592:223{238, 1999}.

[27] Supervisor Univ.Prof. Dipl.-Ing. Dr.techn. Michael Drmota by Sigrun Goluch Taborstra_e 71/3 1020 Wien "The development of homomorphic cryptography From RSA to Gentry's privacy homomorphism", Vienna University of Technology and "https://en.wikipedia.org/wiki/Paillier_cryptosystem".

[28] Homomorphic Tallying with Paillier Cryptosystem, E-voting Seminar by Sansar Choinyambuu-MSE student,12/06/2009.

[29] Online:"https://www.w3resource.com/mongodb/mongodb-java-connection.php".

[30] website "http://www.networkcomputing.com/storage/data-protection-cloud-basics/798260399".

[31]Website"http://collaborate.nist.gov/twiki/cloudcomputing/bin/view/CloudComputing/CloudSecurity".