


PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION
EDUCATION AND SCIENTIFIC RESEARCH

University of Hamma Lakhdar EL-oued

FACULTY OF EXACT SCIENCES

Computer Science Department

MEMOIRE

FOR THE OBTAINING THE MASTER'S DIPLOMA IN COMPUTER SCIENCE

THEME

Use Of Encryption Algorithms To Secure NOSQL Data In Cloud Databases (The Document Data Model)

By:

Lassoued Rafika

Ahmouda Chayma

Supported on: 28/09/2020

The jury composed of:

Encadreur :	Dr.YAGOUB Med Amine
President :	Dr.Khelaifa Abdennacer
Examiner :	Dr.Ndioui Abd-Elhamid

2019/2020

Gratitude

First, we would like to express our thanks to the supportive supervisor, Yaqoub Mohamed Amin, who has supported us throughout this research project, and we are grateful for his encouragement and insightful advice throughoutto the completion of this work.

We warmly thank the jurors for agreeing to judge our work,than we thank all our teachers in the Department of Computer Science at Hama Lakhdar University.

We would like to thank the people who helped us in conducting this research and everyone who contributed to make this work in good conditions. , And we also thank all who always prayed for us and asked God for our help.

Dedicate

To my parents..

My brothers and my sister ..

family sprouts ..all my family ..

Everyone I love him / her .. near or far

Everyone who helped me ..and all my classmates ..

To every computer science student, researcher, or even an amateur ..

To everyone inspirers throughout the World ..

To everyone who believes in him / her self and believes that success

begins from within..

I dedicate this humble work ..

Rafika

Dedicate

To my parents..

My brothers and my sisters ..

all my family ..

Everyone who helped me ..and all my classmates ..

all my friends

I dedicate this humble work ..

Chayma

Abstract

Cloud computing has become an effective solution for storing data, especially with the increasing need to provide resources for storage and the emergence of what is known as big data,it is the basis of computing engineering for the next generation of information technology (IT). Since cloud storage is a model for storage on the Internet, where the data is stored on many multiple servers distributed on advanced data centers, the untrusted nature of cloud servers remains still to put researchers and leaders of this technology in front of security challenges in order to ensure the confidentiality of the stored data. The main objective of this work is to implement a fully homomorphic encryption technology, this technology allows to provide a fully homomorphic, adaptable and efficient encryption to secure NOSQL data in Cloud databases Mongodb Atlas, this encryption allows the implementation of operations on the encrypted data stored at the cloud level in an accurate and reliable manner.

Keywords— Cloud Computing, Cloud Computing security, fully homomorphic encryption, NoSQL database, Mongodb .

Résumé

Le cloud computing est devenu une solution efficace pour stocker des données, en particulier avec le besoin croissant de fournir des ressources pour le stockage et l'émergence de ce que l'on appelle le big data, le cloud computing est devenu la base de l'ingénierie informatique pour la prochaine génération de technologies de l'information (IT). Puisque le stockage en nuage est un modèle de stockage sur Internet, où les données sont stockées sur de nombreux serveurs multiples répartis sur des centres de données avancés, le caractère peu fiable des serveurs cloud reste encore à mettre les chercheurs et les leaders de cette technologie face aux défis de sécurité afin d'assurer la confidentialité des données stockées. L'objectif principal de ce travail est de mettre en œuvre une technologie de cryptage totalement homogène, cette technologie permet de fournir un cryptage totalement homogène, adaptable et efficace pour sécuriser les données NOSQL dans les bases de données Cloud Mongodb Atlas, et ce cryptage permet également la mise en œuvre d'opérations sur le crypté données stockées au niveau du cloud de manière précise et fiable à partir de là, nous pouvons parler de la fiabilité ou de la confidentialité des données cryptées dans le cloud. D'un autre côté, même si une autre partie non fiable peut accéder aux données, ces données apparaîtront cryptées et illogiques.

Mots clés: Cloud Computing, Sécurité du Cloud Computing, Chiffrement complètement homomorphique, Base de données NoSQL, Mongodb.

ملخص

أصبحت الحوسبة السحابية حلاً فعالاً لتخزين البيانات ، خاصة مع الحاجة المتزايدة لتوفير الموارد للتخزين وظهور ما يعرف بالبيانات الضخمة ، فهي أساس هندسة الحوسبة للجيل القادم من تكنولوجيا المعلومات IT.ونظرا لأن التخزين السحابي هو نموذج للتخزين على الإنترنت ، حيث يتم تخزين البيانات على العديد من الخوادم الموزعة على مراكز البيانات المتقدمة ، فإن الطبيعة غير الموثوق بها للخوادم السحابية لا تزال تضع الباحثين وقادة هذه التقنية في مواجهة تحديات الأمان من أجل ضمان سرية البيانات المخزنة. الهدف الرئيسي من هذا العمل هو تنفيذ تقنية تشفير متجانسة تماما، تتيح هذه التقنية توفير تشفير متجانس تماما وقابل للتكيف وفعال لتأمين بيانات NOSQLفي قواعد البيانات السحابيةMongoDB Atlas، وأيضا تنفيذ العمليات على البيانات المشفرة المخزنة على مستوى السحابة بطريقة دقيقة وموثوقة.

الكلمات المفتاحية ـ الحوسبة السحابية ، أمن الحوسبة السحابية ، تشفير بيانات متماثل تماما، ، قاعدة بيانات NOSQL، Mongodb.

Contents

List of Figures

List of Tables

General Introduction

Over time and in sync with the tremendous progress in information technology and development technology projects and the urgent need to save, analyze and store data in all areas led to the emergence of cloud computing. Cloud computing has become a new over-usable model to host resources and connect them online. As a matter of fact, such approach is not completely new. A few years ago, IBM already offered on-demand computing under the ASP (Application Service Provider) methodology. It represents the fact that an application is submitted in the form of a service. The 1980s were also the start of virtualization technology. All of these concepts and technologies have slowly led to the creation of a new way to present it as a "service" in which a payment is often made according to the consumption. Likewise, as with traditional and public services, it provides high computing power, ample storage space, etc. such as water, electricity, telephone, etc.
Cloud Computing can be defined as a distributed and specialized information technology model that is characterized by being dynamically configurable, participatory, scalable, and is provided upon request via communication networks. It has many advantages such as rapid deployment, payment on the move, lower cost, easy expansion, faster service delivery, and access to the network everywhere and anytime, and more services that will definitely make the use of technology easier, more luxurious and beneficial to the user and given these different characteristics, cloud computing has become an interesting solution for companies and researchers.
With the move to cloud computing, the need to expand data storage has increased. The use of a relational database has become traditional, making the user work on a complex policy in order to distribute the database load across multiple database instances. Often this solution presents a lot of problems and may not be wonderful in the gradual expansion. As an alternative, we went to the cloud-based NoSQL database to overcome some of the shortcomings in the relational SQL databases during the scaling process and deal with big data as it provides many One of the features that allow to provide very effective solutions to many problems related to storing and using data in the cloud is where freedom and dynamism are more in designing databases and uses a variety of data models to access and manage data. These types of databases have been specifically optimized for applications that require a large size for data, low latency, and flexible data models which are achieved by easing some data consistency limitations for other databases.
Cloud Security is the protection of data stored on the Internet via cloud computing platforms from theft, leakage and deletion. Among the ways to provide cloud security are firewalls, penetration testing, obfuscation, and encryption, the overall goal of this theme is to provide a data security solution in the cloud computing environment. This work consists of using encryption algorithms to secure NOSQL data in cloud databases. We have exploited MongoDB Atlas databases which are the core of MongoDB Cloud and is the best way to operate MongoDB. MongoDB document model is the fastest way to innovate, providing flexibility and ease of use for the database.
Our work is organized under the following structure:
Part I:State of the Art
Chapter 1: Cloud computing, in which we explain a set of basic concepts related to cloud computing in terms of composition and characteristics.
Chapter 2: Cloud Storage, Non-Relational Databases, Explanation of MongoDB Database, Highlighting Data Structure and Characteristics.
Chapter 3: Cloud Security, we provide an overview of cloud computing security, Attacks as well as the technologies and algorithms used to ensure cloud security.
Part II:Proposition and Modeling
Chapter 4: In the implementation of our project we suggest a fully homomorphic encryption that maintains the operations the addition, the multiplication and the the order of data. Our encryption system is theoretically safe, in this section the work becomed tangible by clarifying the infrastructure

and the approved encryption algorithm in addition to the approved query examples, and finally the general conclusion.

Part I

State of the Art

Chapter 1

Cloud Computing

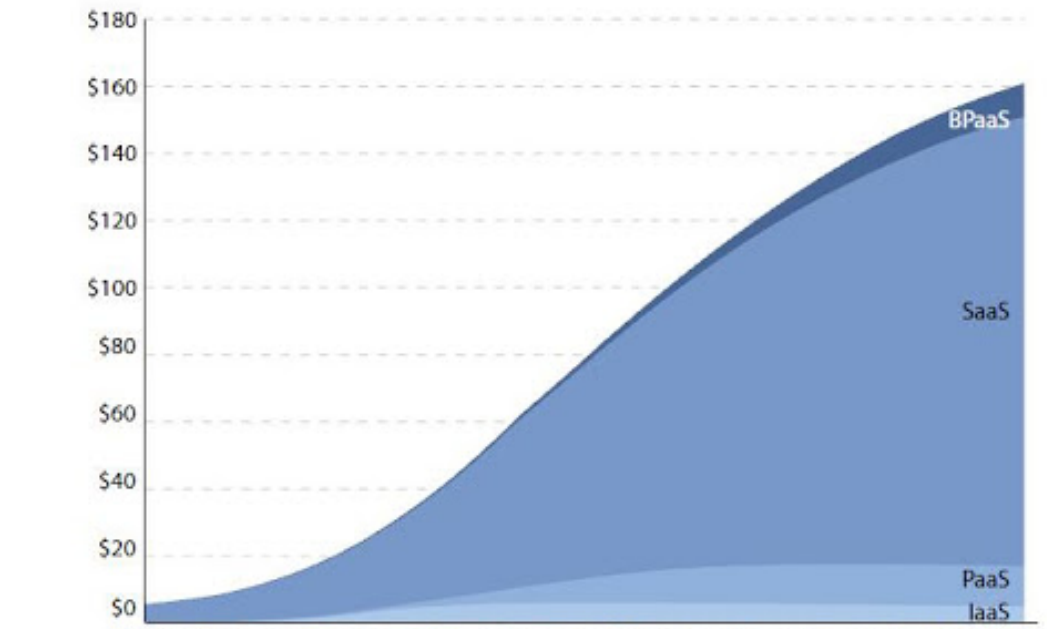
1.1 Introduction

In order to provide maximum support for users and the flexible service, cloud computing emerged as a common solution to provide easy access to external information technology resources that allows file and software sharing and provides high storage space that allows users to use technology with luxury and greater speed.
An increasing number of organizations (such as research centers and institutions) are taking advantage of cloud computing to host their applications. Cloud computing is not an application-oriented but a service- oriented technology. This paper presents the history, concept and many important elements of cloud computing.

1.2 History

We can't say that cloud computing was born at a specific time. A cloud is a network that provides a set of services and information where it is generally considered a collection of services and data consumed.[31]
This notion of consumption was proposed in 1961 during a conference at MIT by John Mc- Carthy who suggested that time sharing could build a bright future in which computing power and even specific applications could be sold as a public service.[31]
This idea was very popular in the 60s but due to the weak hardware, software and network technologies at that time, it disappeared in the mid-70s.
Cloud Computing implements the idea of utility IT of the public service type, proposed by John Mc-Carthy. It can also be compared to the computing cluster in which a group of computers connect to form a single virtual computer allowing high performance computing, but also to grid computing where connected computers that are geographically distributed allow the resolution of a common problem.[31]
At the time, the cost of purchasing and operating IBM mainframes was high and therefore solutions were proposed so that companies could be able to exploit these technologies at a lower cost with the concept of "consumer pay".[31]
ASP, the "application service provider" also had a share in the history of cloud computing. ASP has assigned an application as a service, and this is what we now call SaaS for "software as a service" in the current terminology of cloud computing.[31]
The term Cloud is a simulation with the power grid, in which electricity is produced in large power plants, and then deployed through a network to end users. In the cloud, major power plants are data centers, and the network is most often the Internet and electricity representing IT resources. Cloud computing did not actually appear until 2006 with the advent of Amazon EC2 (Flexible Cloud Computing) .[31]
The real cloud explosion occurred in 2009 with companies like Google (Google App Engine), Microsoft (Microsoft Azure), IBM (IBM Smart Business Service), Sun (Sun Cloud) and Canonical coming to market. Ltd (Ubuntu Enterprise Cloud) .[31]
The cloud computing market reached about $ 5.5 billion in 2008, according to a study conducted by according to a study conducted by Forrester, and it is expected to reach more than 150 billion by 2020, and virtualization has been the cornerstone of the era of cloud computing. In fact, virtualization allows improved management of physical resources to be able to implement many "virtual" systems on a single physical resource and provide an additional layer of hardware abstraction. With the vast and varied

technological advances that have been made over 50 years, we are at the gates to the era of cloud computing, as John McCarthy dreamed in 1961.[31]



	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
BPaaS ($)	0.15	0.23	0.35	0.53	0.80	1.26	1.95	2.93	4.28	6.00	7.66	9.08	10.02
SaaS ($)	5.56	8.09	13.40	21.21	33.09	47.22	63.19	78.43	92.75	105.49	116.39	125.52	132.57
PaaS ($)	0.05	0.12	0.31	0.82	2.08	4.38	7.39	9.80	11.26	11.94	12.15	12.10	11.91

Figure 1.1: Forecast of the size of the public cloud computing market [31]

1.3 Definition

The definition for the cloud can seem murky, but essentially, it's a term used to describe a global network of servers, each with a unique function. The cloud is not a physical entity, but instead is a vast network of remote servers around the globe which are hooked together and meant to operate as a single ecosystem and to provide dynamically scalable infrastructure.[30] [32]
Computing is a subscription-based service where shared network storage and computer resources can be obtained for many users while maintaining the privacy and transparency of use of each user.[30] [32]
These servers are designed to either store and manage data, run applications or deliver content or a service and you can accessing them online from any Internet capable device the information will be available anywhere you go and anytime you need it.[30] [32]
Cloud computing is compared to traditional concepts of "grid computing", "distributed computing", "utility computing", or "autonomic computing".[30]
Forrester defines cloud computing as:
« A pool of abstracted, highly scalable, and managed compute infrastructure capable of hosting end-customer applications and billed by consumption ».[30]

1.4 Cloud Computing Characteristics

1.4.1 On-Demand Self-Service

A service consumer can make use of the computing capabilities without requiring human interaction with each service's provider.[31]

1.4.2 Broad Network Access

Cloud capabilities are available over the network and accessed through various platforms (e.g, mobile phones, laptops, and tablets) .[31]

1.4.3 Resource Pooling

The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. It is characterized by the location of the independence feature in which the customer has no control or knowledge over the exact location of the provided resources but may be able to specify the location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth and virtual machines.[31]

1.4.4 Rapid Elasticity

Capabilities can be rapidly and elastically provisioned; it can be quickly scaled out, and quickly scaled in. For the user, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.[31]

1.4.5 Measured Service

Cloud systems automatically control and optimize resources use by leveraging a metering capability at some level of appropriate abstraction depending on each service (e.g., storage, processing, bandwidth, and active user accounts) and resources usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service. The advantage here is that you are paying for exactly what you are using.[31]

1.5 Cloud Service Models

1.5.1 Software as A Service (SaaS)

it refers to software in the cloud represents the capability provided to the user to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through an interface such as a web browser (e.g. web-based email like Gmail is a form of SaaS provided by Google.[27]

1.5.2 Platform As A Service (Paas)

The customer has the freedom to build his own applications, which run on the providers infrastructure. To meet manageability and scalability requirements of the applications, PaaS providers offer a predefined combination of OS and application servers, where the consumer does not manage or control the underlying cloud infrastructure, but has control over the deployed applications and possibly application hosting environment configurations.[27]

1.5.3 Infrastructure As A Service (IaaS)

The ability provided to the user is to take advantage of processing, storage, networks, and other essential computing resources in order to provide an alternative for the user to deploy and run arbitrary programs which may include operating systems and applications.[27]

1.6 Deployment Models

1.6.1 Public Cloud

A cloud is called public when services are rendered by third-party providers over a network open for public use, meaning that you share the same hardware, software, and network devices with other clients of the same provider (other companies, for example).[29]

Advantages	Disadvantages
Reduces time in developing, ,testing, and launching new products	Higher security risks due to vulnerabilities as a result from shared resources
Cost effectiveness-there is no need to invest in expensive infrastructures	Network performance can suffer instabilities due to spikes in use
Pay-as-you-go scalability -you only pay for what you use	public clouds are usually less customizable.

Table 1.1: Advantages & Disadvantages of Public Cloud[22]

1.6.2 Private Cloud

There is little to no difference between public and private clouds from the technical point of view, as their designs are very similar, but a private cloud refers to a cloud deployment model operated exclusively for a single organization, whether it is physically located at the company's on-site data center, or is managed and hosted by a third-party provider, or a combination of both .[29]
Because this cloud deployment model is only accessible by a single company, there are less security concerns as all data is protected behind a firewall.[29]

Advantages	Disadvantages
More possibilities for customization of the cloud environment	Accessing data from remote locations can be signficantly more difficult
Higher security and prvacy as resources are not shared with others	High costs for inverting in a private cloud infrastructure
Enchanced reliability and greater control over the server	Operating expenses as the company is responsible for the maintenance

Table 1.2: Advantages & Disadvantages of Private Cloud[22]

1.6.3 Hybrid Cloud

Hybrid Clouds, as their name suggests, are a combination of private and public cloud deployment models that are bound together to provide the benefits of both infrastructures to the company using them and attempt to overcome the limitations of each approach, a hybrid cloud encompasses the best features where offers more flexibility and better control and security for user application data and advantageous pricing. [29]

Advantages	Disadvantages
Flexibility and control -the company can choose to allocate resources depending on each specific case	Requires more maintenance,which can result in higher operating expenses for the company
Cost-effectiveness -as public clouds provide scalability ,you only pay for the extra capacity if you need it	Initial costs for activating both infrastructures can be really high for many organizations
Enhanced organizational agility for developing and testing new applications	Data and application integration can be challenging when building a hybrid cloud

Table 1.3: Advantages & Disadvantages of Hybrid Cloud[22]

1.7 Cloud Computing Benefits

1.7.1 Efficiency / cost reduction

Don't need to spend huge amounts of money on purchasing and maintaining equipment because that the using of cloud infrastructure drastically reduces costs. [9]

1.7.2 Data Security

Cloud offers many advanced security features that guarantee that data is securely stored and handled. [9]

1.7.3 Scalability

If business demands increase, you can easily increase your cloud capacity without having to invest in physical infrastructure. This level of agility can give businesses using cloud computing a real advantage over competitors. [9]

1.7.4 Mobility

Cloud computing allows mobile access to corporate data via smartphones and devices just a couple of clicks. Users can get access to their works anytime, anywhere, via any devices of their choice, as long as you stay connected to the internet.[9]

1.7.5 Disaster Recovery

Cloud-based services provide quick data recovery for all kinds of emergency scenarios from natural disasters to power outages.[9]

1.7.6 Control

Cloud enables your full control over your data. You can easily decide which users have what level of access to what data. Since one version of the document can be worked on by different people, and there's no need to have copies of the same document in circulation.[9]

1.7.7 Competitive Edge

Not every company plans to migrate to the cloud, at least not yet. However, organizations which adopt cloud find that many benefits that cloud offers positively impacts their business. [9]

1.8 Cloud Computing Limitation

1.8.1 Network Connection

If there are problems of network connectivity, accessing the cloud also becomes a problem. Performance of the cloud applications also Depends heavily on network performance at the client's side. [20]

1.8.2 Control Of Data Security

Where exactly is the cloud and is it really secure? These are questions arising for users that have confidential data.[20]
The client's data can be susceptible to hacking or phishing attacks, because In a public cloud the client does not have the control over security of his own data and Since the servers on cloud are interconnected it is easy for malware to spread.[20]

1.8.3 Hidden Cost

Although cloud computing offers cost benefits, it has some hidden or additional costs as well. Clients are charged extra for data transfer or other services.[39]

1.9 Conclusion

The cloud provides many options for a daily computer user, so that it enters the world of cloud computing by providing access via any internet connection. However, with this ease there are also insurmountable flaws. Therefore, we should be aware of the risks of using and storing data in the cloud, Because the cloud is a great target for malicious individuals and hackers. In this chapter, we have explained many aspects of cloud computing.

Chapter 2

Cloud Storage

2.1 Introduction

The cloud has become a very important part of modern technology and with increase of data all over the world, a new concept emerged, which is cloud storage, which allows to provide many solutions and benefits for many problems.
Cloud storage means storing your files, data, and everything you want on the cloud through some sites that allow you to do this. Such sites are connected to giant servers (cloud service providers) allowing each user of these services to store their data according to the policy and privacy of cloud service providers.

2.2 Cloud Storage Types

2.2.1 Public Cloud Storage

Storing data between virtual resource groups known as public clouds. Because there are some risks when systems that store your data are not owned or managed, many organizations use containers to transfer workloads and applications between public cloud environments. Fixed storage solutions (such as Red Hat Gluster Storage) help prevent these containers from failing, leading to a loss of eligible applications for all their data.[1]

2.2.2 Private Cloud Storage

Storing data among the default resource groups known as Private Clouds, which are obtained from custom systems. The companies use platforms like OpenStack® resource pools to private clouds.it is considered more secure than public cloud storage solutions, where servers are shared across many customers. [1]

2.2.3 Hybrid Cloud Storage

It is the storage of data between a group of connected more public or private cloud environments. While the public and private cloud environments that constitute a mixed cloud are separate entities ,data between them is facilitated through a complex network of LANs, WPNs, application programming interfaces (APIs), VPNs or containers. This separate structure allows storing important data in a private cloud, less sensitive data in a public cloud, and transferring data between any environment as desired.[1]

2.3 Cloud Storage Formats

2.3.1 File Storage

"File storage is the dominant technology used on Network-attached storage (NAS) systems and is responsible for organizing data and representing it to users. Its hierarchical structure allows us to navigate data from top to bottom easily, but increases processing time".[1]

2.3.2 Block Storage

"Block storage splits a single storage volume (like a cloud storage node) into individual instances known as blocks. It's a fast, low latency storage system ideal for high performance workloads".[1]

2.3.3 Object Storage

"Object storage involves pairing a piece of data with unique identifiers known as metadata. Since objects are uncompressed and unencrypted, they can be accessed very quickly at huge scale making them ideal for cloud native applications".[1]

2.3.3.1 Relational Database

Relational databases are structured databases, It use Structured Querying Language (SQL), They contain tables with columns and rows. Each row is an entry, and each column sorts a specific type of information, such as a name or address, the structure of a relational database allows you to link information from different tables through the use of foreign keys (or indexes).[35] [2] [3]
And in order for relational databases to be effective, the data needs to be stored in a structured manner, making them a good choice for applications that involve the management of several transactions. Some of the most popular SQL databases include Microsoft Access, MySQL and Oracle.[35] [2] [3]

2.3.3.2 Non-Relational Databases(NoSQL)

Non-relational databases are Non-structured databases, It also called NoSQL databases ("non SQL" or "non relational"), they are far more flexible than relational databases because they contain unstructured data, they provide a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases. Non-relational databases are ideal for big data, Most popular non-relational databases being MongoDB, DocumentDB, Cassandra, Coachbase, HBase, Redis, and Neo4j.[35] [2] [3]
NoSQL Databases are categorized into four types:

2.3.3.2.1 Key-Value Store

Data is stored in pairs, as a sequence of records that are indexed by a key /value. It is designed in such a relevance way to handle lots of data and heavy load.[17] [34]
Examples : Redis, Dynamo.

2.3.3.2.2 Column-Based Store

The data is organized by columns rather than rows, the columns are grouped into groups of related data that are accessed together.[17] [34]
Examples : HBase, Cassandra.

2.3.3.2.3 Graphs Based Store

This type of database allows for storage of entities as well as relationships between those entities, it does away with the common row-column structure.[17] [34]
Examples : Neo4J, Infinite Graph.

2.3.3.2.4 Document Store

Document-oriented storage supports data storage and retrieval as a key value pair, but the value portion is stored as a document. Document stores are often encoded in text formats such as JSON or XML format. The values in the document store follow a predefined hierarchical structure that includes metadata for stored content, the value is understood by the DB and can be queried.[17] [34]
Examples : CouchDB, MongoDB.

2.4 Document Data Model Example:MongoDB

2.4.1 MongoDB History

Eliot Horowitz and Dwight Merriman developed MongoDB in the year 2007, because they experienced some scalability issues with the relational database while developing enterprise web applications at their company DoubleClick. The name of the database was derived from the word humongous.
In 2009, MongoDB was made as an open source project. Many companies started using MongoDB for its amazing features. The New York Times newspaper used MongoDB to build a web-based application to submit the photos. In 2013, the company was officially named to MongoDB Inc.[11]

2.4.2 MongoDB Definition

MongoDB is an open source database management system (DBMS) that uses a document-oriented NoSQL database model which supports various forms of data and used for high volume data storage.
As a NoSQL database, MongoDB shuns the relational database's table-based structure to adapt JSON like documents that have dynamic schemas which it calls BSON.
This makes data integration for certain types of applications faster and easier. MongoDB is built for scalability, high availability and performance from a single server deployment to large and complex multi-site infrastructures.[11] [16]
Organizations use MongoDB:

- Adobe
- LinkedIn
- McAfee
- FourSquare
- eBay
- MetLife
- SAP

2.4.3 MongoDB Platforms

- MongoDB available in community and commercial versions via vendor MongoDB Inc. MongoDB Community Edition is open source release, but MongoDB Enterprise Server provides additional security features, an in-memory storage, administration and authentication features, and monitoring capabilities through Ops Manager.[21]
- MongoDB Compass which is a graphical user interface (GUI) provides users a way to work with document structure, conduct queries, index data and more. To visualize data and create reports using SQL queries, the MongoDB Connector for BI allows users to connect the NoSQL database to their business intelligence tools.[21]
- MongoDB Atlas is a cloud database as a service launched by MongoDB Inc in 2016. it runs on AWS, Microsoft Azure and Google Cloud Platform. More recently, MongoDB released a platform called Stitch for application development on MongoDB Atlas, with plans to extend it to on-premises databases.[21]

2.4.4 MongoDB Features

- Each database contains collections which in turn contains documents.[19]
- Each document can be different with a varying number of fields. The size and content of each document can be different from each other.[19]
- The document structure is more in line with how developers construct their classes and objects in their respective programming languages.[19]

- use NoSQL databases, the rows (or documents as called in MongoDB) doesn't need to have a schema defined beforehand.[19]
- The data model available within MongoDB allows you to represent hierarchical relationships, to store arrays, and other more complex structures more easily.[19]
- Scalability: The MongoDB environments are very scalable.[19]

MongoDB Example: A document modeled in MongoDB.[19]

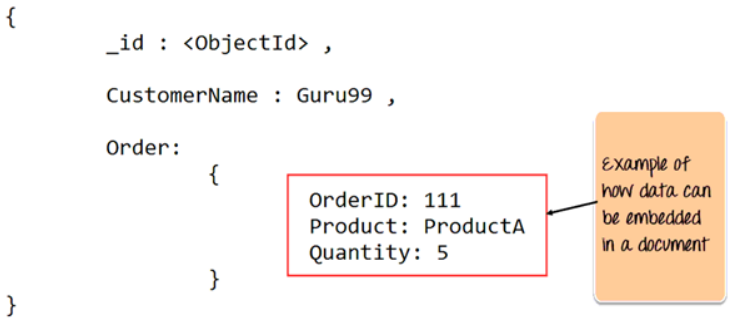


Figure 2.1: A Document Modeled In MongoDB [19]

2.4.5 Key Components of MongoDB Architecture

Below are a few of the common terms used in MongoDB:

- **The _id field :** is like the document's primary key. If you create a new document without an _id field, MongoDB will automatically create the field (Mongo DB will add a 24-digit unique identifier to each document in the collection).[19]

_id	CustomerID	CustomerName	OrderID
563479cc8a8a4246bd27d784	11	Guru99	111

Table 2.1: Key Components of MongoDB[19]

- **Collection :** This is a grouping of MongoDB documents is collection that equivalent of a table which is created in any other RDMS such as Oracle or MS SQL.[19]
- **Cursor :**Clients can iterate through a cursor to retrieve results by using a pointer (Cursor) to the result set of a query.[19]
- **Database :**This is a container for collections, a MongoDB server can store multiple databases.[19]
- **Document :** A record in a MongoDB collection is basically called a document. The document, in turn, will consist of field name and values.[19]
- **Field :** A name value pair in a document. A document has zero or more fields. Fields are analogous to columns in relational databases.[19]

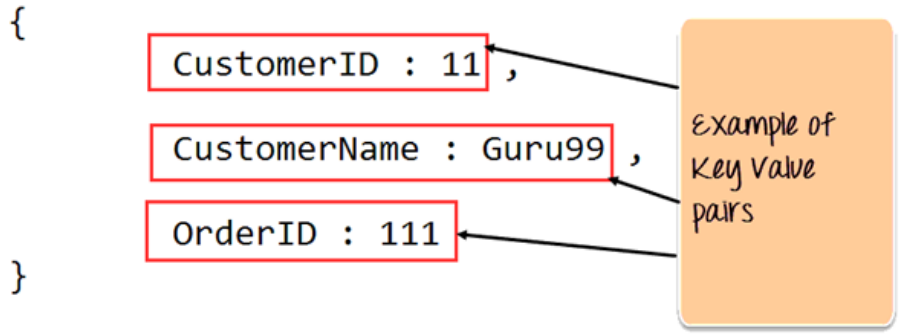


Figure 2.2: Example of key value pairs[19]

- **JSON :** This is known as JavaScript Object Notation. This is a human readable, plain text format for expressing structured data. JSON is currently supported in many programming languages.[19]

2.4.5.1 MongoDB Uses :

Below are the few of the reasons as to why one should start using MongoDB :

1. **Document Oriented :** It stores the data in documents. This makes MongoDB very flexible and adaptable to real business world situation and requirements.[19]
2. **Ad Hoc Queries :** MongoDB supports search by field, regular expression searches, and range queries.[19]
3. **Indexing :** Indexes can be created to improve the performance of searches within MongoDB.[15]
4. **Replication :**It supports Master Slave replication.[4]
5. **Multiple Servers :** Database can run over multiple servers[4]
6. **Auto Sharding :** This process distributes data across multiple physical partitions called shards. Due to sharding, MongoDB has an automatic load balancing feature.[4]
7. **Aggregation :**MapReduce can be applied to enable batch processing of data as well as perform aggregation operations.[15]
8. **Failure Handling :**In MongoDB, it's easy to cope with cases of failures. Due to the existence of Huge numbers of replicas give out increased protection and data availability.[4]
9. **GridFS :** Any sizes of files can be stored. GridFS feature divides files into smaller parts and stores them as separate documents.[4]
10. **Document Oriented Storage :** It uses BSON format which is a JSON like format.[4]
11. **Schema-less Database:** It is a schema-less database written in C++ .[4]
12. **Procedures :** MongoDB JavaScript works well as the database uses the language instead of procedures.[4]

2.4.6 Data Modelling in MongoDB

As we have seen from the Introduction section, the data in MongoDB has a flexible schema. This sort of flexibility is what makes MongoDB so powerful. [19]
Some considerations while designing Schema in MongoDB:

- Design your schema according to user requirements. [14]
- Combine objects into one document if you will use them together. [14]
- Separate them (but make sure there should not be need of joins.[14]

- Duplicate the data (but limited) because disk space is cheap as compare to compute time.[14]
- Optimize your schema for most frequent use cases.
- Doing complex aggregation in the schema.

keep these questions in mind when modeling data in Mongo:

- What are the needs of the application?.[19]
- What are data retrieval patterns? [19]
- Are frequent insert's updates and removals happening in the database?[19]

Data Model Design:

2.4.6.1 Embedded Data Models

When using MongoDB, we can embed related data in a single structure or document. Generally, these schemas are known as "denormalized" models, consider the following diagram [1]:

Figure 2.3: Embedded Data Models [1]

2.4.6.2 Normalized Data Models

Normalized data models describe relationships using references between documents .[1]

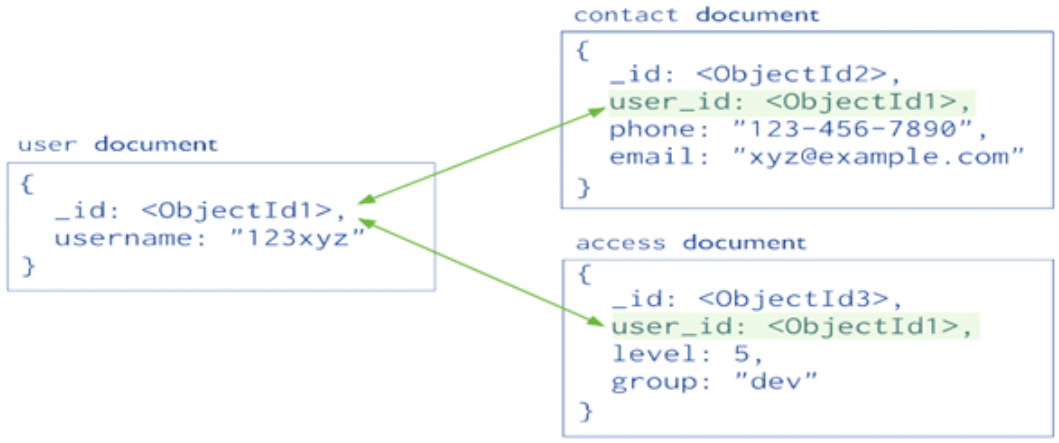


Figure 2.4: Normalized Data Models [1]

2.5 Conclusion

MongoDB is a relative newcomer in the database arena, and is the most popular among the NoSQL databases. It is a great tool for building data warehouses.
NoSQL offer a very good alternative to RDBMS in terms of data management. One of the big advantages of a document database is scalability through the use of embedding data. Another thing to keep in mind is how to best model your one to many document relationships that can be done with either referenced or embedded data. In this chapter, we have focused on explaining non-relational databases MongDB.

Chapter 3

Cloud Computing security

3.1 Introduction

Security is a major issue in modern digital technologies. Cloud Computing transfers programs and data to large external data centers located in supplier locations, where data management and control cannot be completely reliable, meaning there are many new threats and challenges related to security and reliability. Cloud Security is the top concerns among users because cloud computing is an invisible system for the user, it is understood that customers desperately want their information to be fully protected and the services provided stable.[35]
Cloud security is a set of control-based safeguards and technology protection designed to protect resources stored online from leakage, theft, or data loss. Cloud security provides multiple levels of controls within the network infrastructure in order to provide protection.[35]

3.2 Cloud Computing Security Requirements

There are several requirements that must be met, according to the International Organization for Standardization ISO and this will be highlighted in this section. [23]

3.2.1 Confidentiality

Confidentiality has a primary and significant role in the cloud, especially in controlling the data of organizations in distributed databases. As it is necessary to ensure that the data of the users in the cloud cannot be accessed by another third party that is not allowed, this is very necessary especially when using the public cloud. [38]
There are many factors that threaten confidentiality in the cloud, such as: data breach due to the persistence of this data. The continuation of the data or the so-called data retention is the ability to retrieve this data after it has been deleted or nominally deleted, which constitutes a threat to the confidentiality of the data.[43]

3.2.2 Integrity

Integrity is one of the three main aspects of information security, that is, it is not permissible to modify, operate or delete data except for authorized entities and this is called data integrity (information content) and integrity of origin (data source, often called authentication) i.e. protection.[43] [28]
Integrity is evidence of the Cloud Provider's ability to ensure reliable and correct operation of a Cloud system, this is in order to comply with its legal obligations of service level agreements and technical standards that represent ACID (Atomic, properties, consistency, insulation, and durability). Suppliers have directional obligations Their clients in case of breaches of their accounts.[23]

3.2.3 Availability

Availability is the ability to use the information or resource required. Will it be available and available at the required time when someone wants to use it and access it? Availability in the cloud is intended to

ensure that cloud users can use cloud systems (including these applications and infrastructure.) anytime, anywhere.[28]
There are many threats available, including equipment failures, natural disasters, and denial of service attacks (DDoS Distributed Denial of service). It is difficult to detect such threats, which may lead to a refusal to serve.[23]

3.2.4 Identification And Authentication

Users within the cloud must create authentication information by setting access priorities. In order to validate users within the cloud and protect their files and data.[42]. Authentication can also be defined as a procedure consisting of ascertaining the identity of an entity (person, computer, etc.) in order for this entity to arrive (systems, networks, applications, etc.) it verifies the validity of the entity that was previously defined.[28]

3.2.5 Authorization

An authorization is the granting of rights or privileges to a person or user, such as writing, reading, modifying, deleting, or executing resources and being accompanied by their rights, by the system administrator in Cloud private .[23]

3.2.6 Trust

Trust in information technology is the focus on designing tools that operate according to a specific model of trust in order to assist users in various tasks. Trust was used in the cloud to convince monitors that the system is correct and safe.
Through an effective security policy, trust between the user and the cloud has been guaranteed, as this policy addresses restrictions on jobs and restrictions on access to people's data, etc..[23]

3.2.7 Audit And Compliance

In cloud systems, auditing is the monitoring and tracking of records on an online interface to display a record called the audit log in real time for the administrator's actions on the system and it also includes checking authorization and authentication records to verify assurance of compliance with pre-defined safety policies and standards. Detecting any suspicious behavior or threat. The rules related to safety are determined by examining the registry, which allows suspicious actions to be extracted with ease.[23]

3.2.8 Non-Repudiation

Non- Repudiation in Cloud Computing is a transaction when a data owner sends a request to the cloud provider to download data in a way that neither party can refuse this transaction. In other words, denying the original proves that the data has been sent and that denying access proves that it was received.[26]

3.3 Different Types Of Attacks In Cloud Computing

Security in Cloud Computing is the main concern in order to provide more comfort and confidence to customers inside the cloud. There are a lot of studies that have classified security threats and breaches in the cloud, according to the service deployment model, which is one of the many aspects that must be considered in a comprehensive survey of the security of the cloud .
There are several types of serious attacks that affect the availability, confidentiality, and integrity of cloud resources and services on a large scale in cloud computing systems due to the characteristics of cloud systems. These threats are listed below:

3.3.1 Basic Security

Web 2.0, a major technology that allows the program to be used as a service (such as SaaS). This technology relieves users of the tasks of maintaining and installing programs. Security is more important than ever for such an environment due to the increased number of users using Web 2.0.[23]

3.3.1.1 SQL Injection Attack

SQL injection attacks, a process in which a malicious code is added to the standard SQL code, that is, attackers obtain unauthorized access to the database and access to sensitive information. This is due to the attacker's site being viewed as the original user of the site's misunderstanding of the data the hacker entered and allows him to access the SQL server where he can make various changes. There are various techniques to limit these attacks, such as: Avoid using SQL code generation dynamically in the code etc.[23]

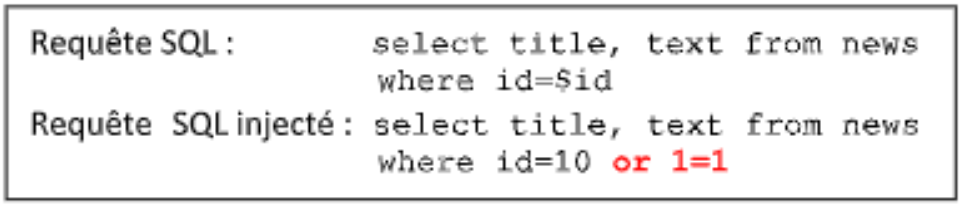


Figure 3.1: Example of an SQL injection attack [23]

3.3.1.2 XSS Attack

Cross-site scripting (XSS) is very common since the creation of the Web 2.0 and is a type of security breach of a website as it infers malicious text programs into web content. Dynamic web sites suffer more than static sites from security threats because of their dynamism in providing services to users. XSS has many capabilities where an attacker can use all the languages the browser supports (JavaScript, PHP, HTML5 ...). [23]
Several solutions were proposed against these attacks such as: Various technologies such as filtering active content, Web application vulnerabilities detection technology etc.. [23]

3.3.1.3 Attack Of The Interceptor

It is the attacker's interception of communications between the two parties. Encryption techniques have been developed to provide protection from these attacks such as: Dsniff, Cain, Ettercap, Wsniff, Airjack, etc.
In order to ensure proper implementation of Cloud Computing, security is required at its various levels such as: server access security, internet access security, database access security, data and software security, etc.[23]

3.3.2 Network-Level Security

To ensure network security of all kinds (shared and non-shared networks, public or private networks, small or large networks) from the security threats you face, you must consider the confidentiality and integrity of the network, control access, and maintain appropriate security against threats from third parties. Network attacks include the following:

3.3.2.1 DNS Attack

Domain Name Server (DNS) conveys the domain name to the IP address of multiple domains and the difficulty to remember. The goal of attacks against it is to use the weaknesses of this protocol to direct

users to pirated sites. These attacks are also divided into two parts: DNS ID spoofing and DNS cache poisoning.[23]
The attack is to retrieve the ID according to the DNS request, and then send harmful responses to the victim of this attack by the DNS server, meaning that the victim will temporarily use the hacker's IP address without her knowledge as shown in the following diagram. This is due to the fact that the DNS servers contain a cache that allows keeping the connection Between the domain name and its IP address.[13]

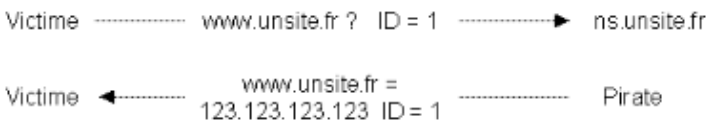


Figure 3.2: DNS ID Spoofing Attack[13]

There are measures that may reduce the effects of DNS threats, such as: Domain Name System Security (DNSSEC) extensions, but sometimes they are not sufficient.[23]

3.3.2.2 Port Scanning

Port scanning is a technology that computer system administrators use to control the security of servers on their networks, in return, which attackers use to try to find vulnerabilities in computer systems, that allow them to discover exploitable access ports such as IP address, MAC address, router, filter portal, firewall rules, and so on. There are many port scanning technologies like TCP, UDP, SYN, FIN, ACK etc. These attacks can be avoided and countered using IDS intrusion detection system or firewall.[23] [28]

3.3.2.3 SNIFFER Attack

Wiretapping or sniffing is used to spy on network traffic by applications capable of capturing packets traveling over the network such as packet sniffer (Sniffer). In the event that data transmitted through this block is not encrypted it is easy to read. Hence all messages are read on the network. There are applications Harmful detection based on ARP (Address Resolution Protocol) and RTT (round trip time). Therefore, data must be encrypted during communication between the two parties as a countermeasure for exploration in order to protect user data.[23] [28]

3.3.2.4 Problem Of Reused IP Addres

A re-used IP address can pose a number of problems by leaving a specific user a network. His previous address is given to another user. This command may harm the security of users, because it is mean that the user data may become available to another user.[23]

3.3.2.5 BGP Diversion

BGP (Border Gateway Protocol) is a directive for Internet traffic that any network on the Internet relies on BGP to access other networks. Hence, BGP hijacking is a type of attack that threatens network security. Two types of these BGP attacks have been reached so far: The first type is advertising The wrong system independent IP addresses (AS) allowing attackers to access IP addresses that they do not belong to, whereas the second type when ASN injects a path that interferes with an existing advertisement, Here the priority path is taken. [23]

3.3.3 Security At The Application Level

Security at the application level is the use of software and hardware resources to ensure application security so that attackers cannot control or modify applications. Threats on the application occur dy-

namically and are adaptable to security controls. Several technologies have been achieved to deal with security issues such as developing an ASIC that is task-oriented.
Among the threats that arise from the unauthorized use of applications: XSS attacks, cookie poisoning, hidden field manipulations, SQL injection attacks, denial of service attacks, correction options and correction background, CAPTCHA test break, etc. [23]

3.3.3.1 Security Problems Linked To The Hypervisor

Cloud Computing relies heavily on the concept of virtualization. This allows many operating systems to run in one virtual machine, so security concerns for new operating systems must be taken into account. Here, the guest system may try to execute some malicious code on the host system that has become inactive which may lose its full control.[23]

3.3.3.2 Denial Of Service Attack

DoS attack makes access to services unavailable for authorized users, so the service becomes unavailable to the authorized user. An example of this is when trying to access a site we cannot access due to the large number of requests Access to it means that the number of access requests exceeded the server capacity.
DoS attack relies on some techniques in its attacks such as filling storage disk space with massive files, or sending a message that resets a mask the target host subnet. There are also defense methods against this attack by using an Intrusion Detection System (IDS) that alerts the system in the event of an attack.[23]

3.3.3.3 Attack By Distributed Denial Of Service

DDoS it is an advanced version of DoS, an attack that aims to silence a machine by blocking it from traffic. The cloud in general and SaaS programs in particular are among the most important targets of this attack and it represents a major threat that is difficult to avoid.
In DDoS attackers have the ability to control the flow of information by allowing certain information to be available at certain times. Any information available for public use is under the control of the attacker. For example, sending UDP packets to the target device in a large number, as the device after that will not be able to respond to any other request.[23]

3.3.3.4 Poisoning By Cookies

A cookie is a text file, which consists of user identification information and is stored by a website (browser). It is security sensitive data that no one should obtain.
The attack on a cookie is aimed at stealing the identity information of a specific user and the ability to change or modify this data in order to allow unauthorized access to a webpage or application. There are several defects that allow the success of this attack such as: theft by smelling or in the middle Cookies pass through HTTP requests. This attack can be avoided by encrypting cookie data.[23]

Figure 3.3: Cookie poisoning [23]

3.3.3.5 Hidden Field Manipulation Attack

Backend processing is mainly intended for e-commerce sites. When an attacker accesses a webpage, some fields are hidden and the information related to the page is hidden.
The moderator modifies a portion of the HTTP request (URL, request string, addresses, cookies, form fields, and hidden fields). The user makes a selection on an HTML page that usually stores this selection as field values and sends it to the application as an HTTP request (GET or POST) An attacker checks the HTML code and changes the hidden field values to modify requests sent to the server.[23]

3.3.3.6 Backdoor And Debugging Options

Backdoor is an option that developers activate before publishing a website that allows them to make changes to code development and implementation on the website. Sometimes these options are ignored, which causes attacks that would not have occurred if the developers had paid attention to these options in order to avoid security problems and breaches.[23]

3.3.3.7 CAPTCHA Breaking

CAPTCHA is a test in which a website user is forced to decode an image or distorted sound in order to protect this site from attack. It was developed to prevent robots or robot operations from using resources. CAPTCHA has been broken by spammers.[23]

3.3.4 Physical Security

Physical security aims to protect the security of the cloud significantly. Data centers are subject to many threats, even from natural hazards. The status of data centers in the room must be considered as a protection system to prepare a multi-level defense.
Physical security components must be supported and equipped with the necessary procedures such as authentication. And alert the security guards when unsuccessful attempts occur. Maintain logins for specific periods.[23]

3.4 Data Security Techniques in The Cloud

There are several techniques for securing data in the cloud. These technologies can be classified according to several factors as shown in the following figure:

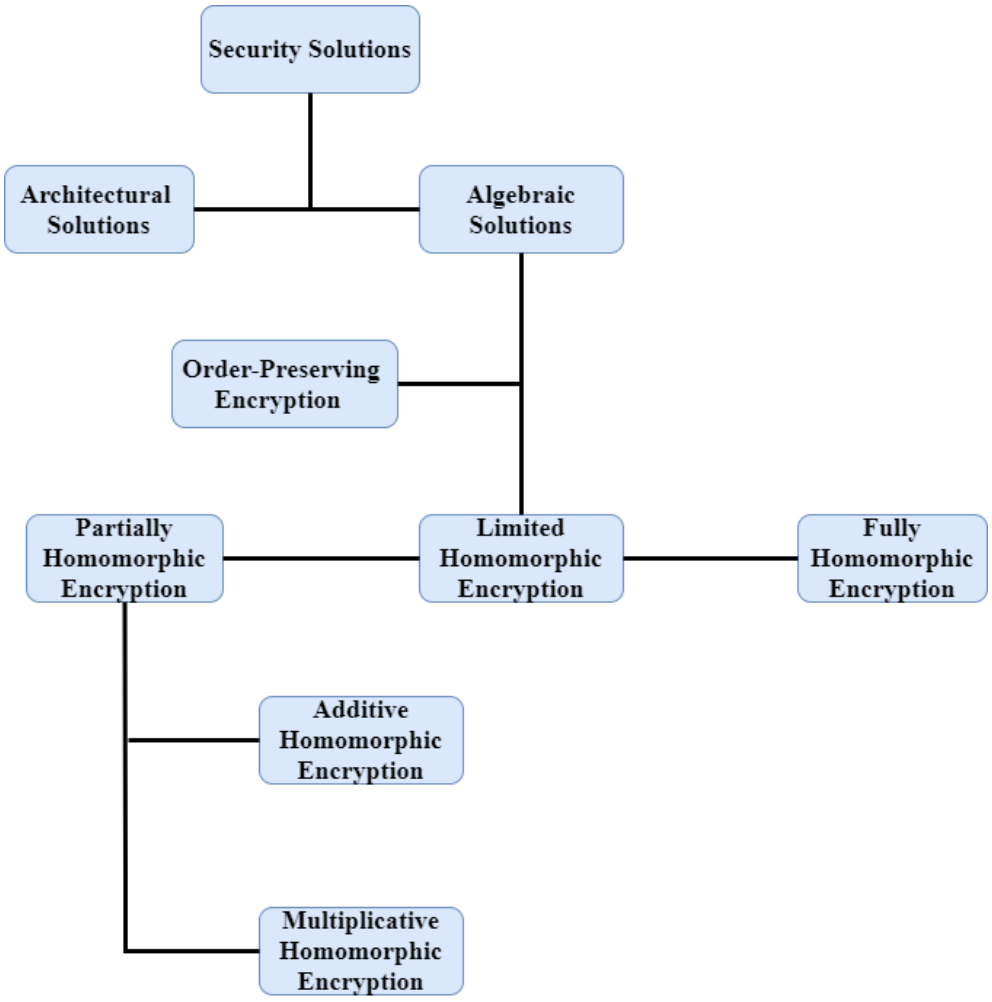


Figure 3.4: Security Solutions

3.4.1 Architectural solutions

Multiple cloud architectures allow maintaining security and mitigating security problems in the cloud.These architectures are suitable for Cloud platforms and generally do not introduce an intermediate proxy or broker server between the client and the cloud provider. The use of multi-cloud mechanism can reveal various theories in order to target different aspects of the security of confidentiality, integrity, consistency and consistency of data stored in different clouds. This class uses the principle of systematically distributing user data across multiple cloud providers.[23]

3.4.2 Algebraic Solutions

In order to ensure the confidentiality of data, the researchers are restricted to providing other data protection techniques. The best-known idea is "Homomorphic", a feature that allows calculations to be performed in the cloud without accessing data or results.[23]

3.4.2.1 Traditional techniques

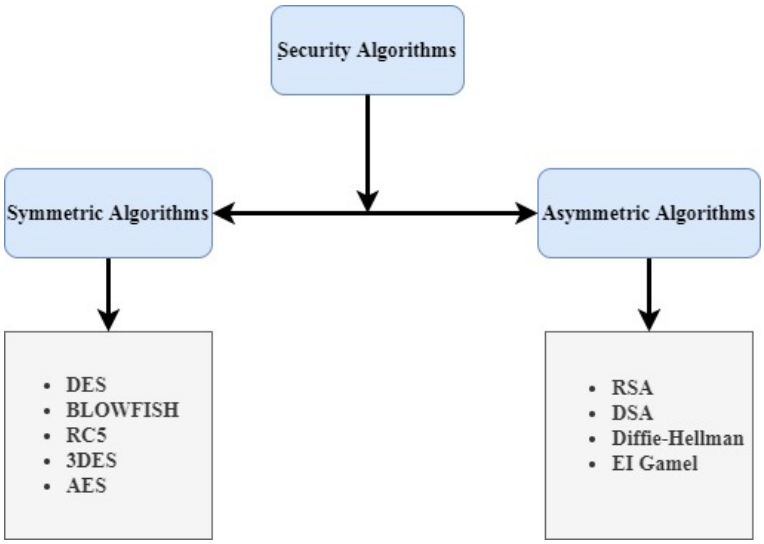


Figure 3.5: Traditional techniques

3.4.2.1.1 Symmetric encryption algorithms

DES Data encryption standard, it was the first coding standard recommended by the National Institute of Standards and Technology 1977 (NIST).[40]

BLOWFISH Developed by Bruce Schneier in 1993. It is one of the most common generic algorithms. Various experiments and research analysis demonstrated the superiority of the blowfish algorithm over other algorithms in terms of processing time and the non-superiority of any attack on it.[40]

RC5 RC5 is a symmetric key block encryption algorithm designed by Ron Rivest in 1994, the use of this algorithm demonstrates that it is safe and has a slow speed.[40]

3DES Triple DES (3DES or TDES), it is a symmetric-key block cipher, which applies the DES cipher algorithm three times to each data block. It appeared as a reinforcement for DES in 1998.[40]

AES The Advanced Encryption Standard (AES), is the new encoding standard recommended by Institute of Standards and Technology (NIST) in 2001 to replace DES. There is only one attack against him known as Brute Force Attack, as the attacker tries to test all character combinations to unlock the encryption .[40]

3.4.2.1.2 Asymmetric encryption algorithms

RSA This is an internet coding and authentication system that uses an algorithm developed in 1977 by Ron Rivest, Adi Shamir, and Leonard Adlemann. The RSA algorithm is the most used encoder because its speed. Until now it is the only algorithm used to generate private and public keys and encryption.[40]

DSA The Digital Signature Algorithm (DSA) is a Federal Information Processing Standard for digital signatures. It was proposed by the National Institute of Standards and Technology (NIST) 1991 for use in the Digital Signature Standard (DSS) and was certified as FIPS 186 in 1993.[40]

Diffie-Hellman Key Exchange (DH): Diffie-Hellman Key Exchange considered the oldest process in the field of encryption, it is a specific method for exchanging encryption keys over a public communications channel. Keys are not actually exchanged but they are jointly derived, this method allows them to know each other in advance to create a shared secret key via an insecure communications channel, which can then be used in subsequent communications using symmetric key encryption.[40]

EL Gamel It was described and developed by Tahir Al-Jamal in 1984. it is an asymmetric key cipher algorithm for public key cipher based on Diffie - Hellman key exchange. Sentence encryption is used in the free program GNU Privacy Guard.The Digital Signature Algorithm (DSA) is a variant of the ElGamal signature scheme that which should not be confused with ElGamal encryption.[40]

3.4.2.2 Moderne techniques

3.4.2.2.1 Partially Homomorphic Encryption Partially symmetric encryption (PHE) allows various operations to be performed on encrypted data such as multiplication or addition but not together. Many advanced encrypting systems can be classified into two parts. The first section is homogeneous add-on encrypting schemes that allow for limited or unlimited additions to encrypted data. As for the second section, homogeneous multiplication encrypting schemes allow for limited or unlimited multiplication on the encrypted data .[23][44]

3.4.2.2.2 Somewhat Homomorphic Encryption Somewhat Homomorphic Encryption(Limited symmetric encryption) allows many operations of additions and multiples but with a limited number and method. such as it is allowed to add multiple times while multiplication is possible only once (one multiplication allowed and an unlimited number of additions to encoded data). [23][44]

3.4.2.2.3 Fully Homomorphic Encryption Fully homomorphic encryption (FHE) is a new security concept, it can calculate any type of function on encrypted data, which can be run on encrypted inputs to produce an encryption of the result.
Since the users of this type encryption never need decrypted its inputs, it can be run by an untrusted party without revealing its inputs.[23]

3.4.2.2.4 Ordre-Preserving Encryption This encryption ensures to save the order between the data elements based on their encrypted values and that is without disclosing the data, by using a different way like establishing indexes on the encrypted data that can be used for request order queries. There are many ways to create this feature (linear and non-linear functions for data indexing).[23]

3.5 Conclusion

Despite the rapid development of cloud computing applications in the recent period, security risks remain a major and targeted challenge.
In this chapter, we introduced the concept of IT security in distributed systems and generally in cloud computing, then we continued to describe various attacks and encryption algorithms in the cloud environment.

Part II

Proposition and Modeling

Chapter 4

Proposed Approach

4.1 Introduction

The Internet is now indispensable to providing and maintaining data as cloud computing provides an effective solution, but the untrusted nature of the servers poses a barrier to this technology.
There is a real need to find an effective encrypting algorithm that performs calculations on encrypted data without the need for decryption. In this paper, we suggest a fully homomorphic encryption . Our encryption system is theoretically safe.
Here we use the non-relational database system NoSQL that MongoDB Atlas delivers the world's leading database for modern applications as a fully automated cloud service.

4.2 System Architecture

This section provides a framework that explains the basics and structure of the proposed encryption system targeting the NoSQL database to secure data in the cloud. In the first experiment, we worked to provide a secure query for office applications, while the work could be expanded to include secure query processing for web applications, mobile phones, and other applications that require data storage and use. The encryption system includes a ciphering plan based on a trusted proxy. The proxy ensures a secure interaction between user's applications and cloud NoSQL database server.
Several important goals were taken into consideration during the design of this system, as follows:

- The supports users' access to an encrypted NoSQL database.
- The cloud provider cannot access the data.
- Data integrity and preservation.
- Hide the complexity of security mechanisms from end users.
- Transparent access to the database.
- The homomorphic.
- The Order preserving.

The fully homomorphic encryption (FHE) system is based on the general principles of NoSQL database products. JSON format, which is a dominant format in NoSQL databases, can be read by human and machine, and will be worked on MongoDB Atlas databases (Cloud Service).
Figure 4.1 shows a schematic representation of the proposed FHE System Architecture.

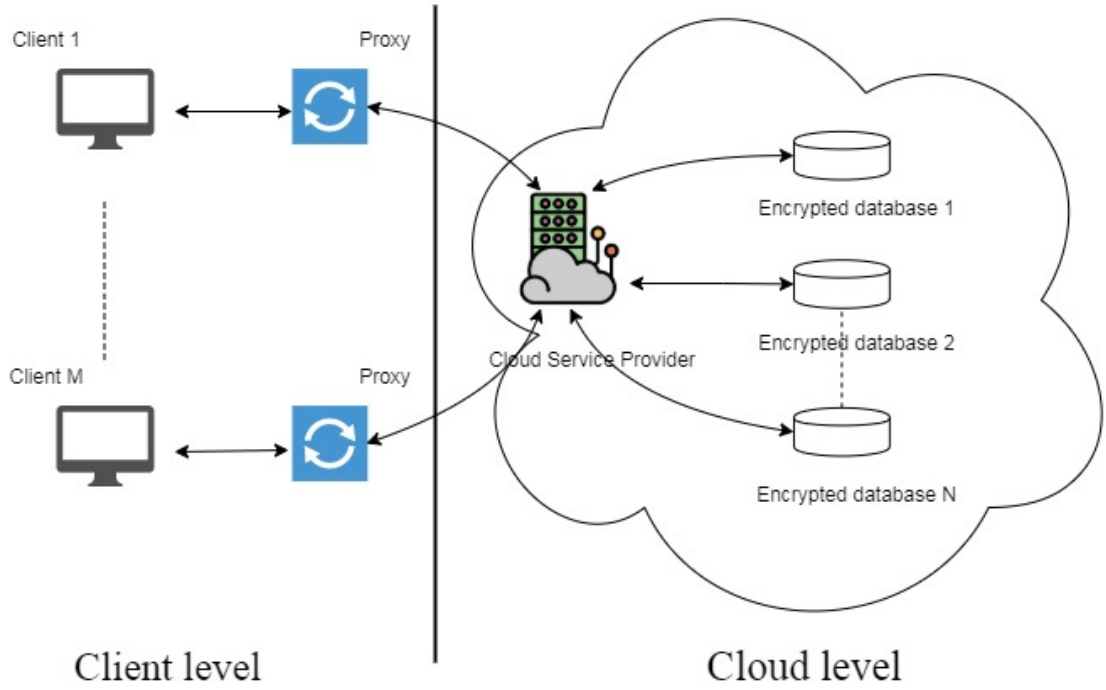


Figure 4.1: System Architecture

The FHE system has two layers. The first is an untrusted public cloud database service provider layer. The second layer is a client who stores data on the cloud. The system design depends on the proxy, that provides client access to the remote cloud server running the NoSQL database processing system. In this system, query processing involves three stages:

1. Client-side query is served the JSON format executed by the client software (client's application).
2. Trusted proxy converts it into an encrypted query that can be executed directly in the cloud and decrypts it before forwarding the result to the client's applications.
3. Server-side query processing by NoSQL database server.

4.3 Preliminary And Formal Model

Homogeneity is defined as follows:
We have two groups (M,$\triangle$) *and* $(C, \circ)$.
M *indicate* (*respectively* e_C) *the neutral element of* M (*respectively de* C).
An application $f : M \to C$ *is a group homomorphic if* :

* If x and y are two elements of M, then $f(\mathrm{x} \triangle y) = f(x) \circ f(y)$.
$* f(e_M) = e_C$.

Property :
Let f: M $\to C$ *be a homomorphic of groups.*
So :
1.$f(1_M) = 1_C$.
2.*For any element* x *of* $M, f(x^{-1}) = f(x)^{-1}$.
3.*For any non* $-$ *zero integer* $n, f(x^n) = f(x)^n$.
4.*For any non* $-$ *zero integer* $n, f(x^{-n}) = f(x)^{-n}$.

A fully homomorphic encryption system is a system in which any function on encrypted data can be evaluated. And since we can express any function as polynomial, where polynomial consists of a series of additions and multiples, once the encryption system allows the evaluation of a random number of additions and multiples to the encoded data here the encryption system is fully homomorphic.

Formally, If elements c_1 *and* c_2 *are from group* C, *and we have* m_1 *and* m_2 *respectively from group* M.*An encryption function* $f : M \to C$ *is a homomorphic s'there are two operations* $\triangle$ *and* $\circ$ *such as* :

$$f^1(c_1 \Delta c_2) = f^1(c_1) \circ f^1(c_2) = m_1 \circ m_2 \tag{4.1}$$

Commonly, $\triangle$ *is a standard addition or multiplier ,but this is not always the case.We need to use the standard computation to gain more efficiency that allows for algebraic symmetric algebraic operation.In addition, the following axioms must be fulfilled* :
Closing : *For each* m_1, $m_2 \in M$ *the result of the operation* $m_1 \circ m_2 \in M$.
Element of identity: *There is an element* $e \in M$, *such that for any element* $m \in M$, *the equality* $m \circ e = e \circ m = m$ *is true.Such e is a unique element , so we call e the identity element.*
Inverse element: *For each* $m_1 \in M$, *there is an element* $m_2 \in M$ *such that* : $m_1 \circ m_2 = m_1 \circ m_2 = e$ *where e is the neutral element.*

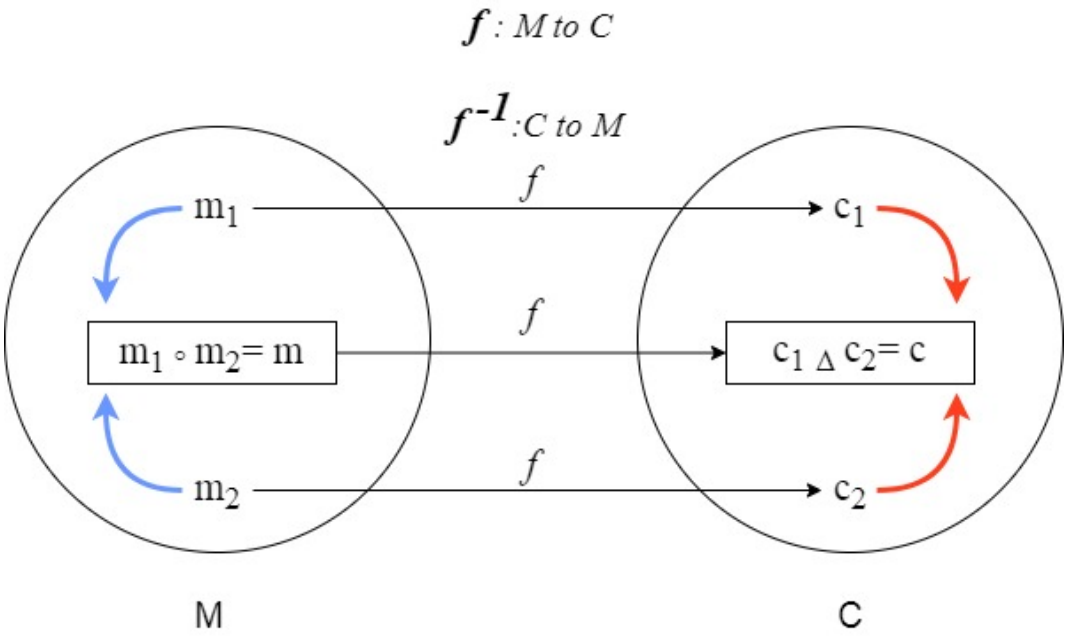


Figure 4.2: Homomorphic Encryption

Figure 4.2 shows the group homomorphic of group f: M $\to$ *C such that* $(M; C; K; f; f^{-1})$ *the parameters of the proposed encrypting technique.*

Where M and C are the messages used and their encrypted texts, K is the encryption key, f is the encryption algorithm and f^{-1} *represents the decryption algorithm used. Symmetric encrypting assures us that if the plain text forms a group* $(M; \triangle)$ *and the encrypted text forms a group* $(C; \circ)$, *then the encrypting algorithm must be a map from group M to group C, this means* $f_k : M \to C$, *where* $k \in K$ *is a secret key that checks the following equality* :

$$f_k(m_1 \circ m_2) = f_k(m_1) \triangle f_k(m_2) \tag{4.2}$$

4.3.1 Encryption And Decryption Scheme

We present mathematical definitions and important concepts required in the proposed approaches. In this work, we deduce the encrypting system from the homomorphic encrypting functions proposed by James Dyer et al. In [25], we define the FHE function as follows in equation 4.3:

$$f : c_i = (m_i + rand_i * p) \ mod \ p^2 + m_i * k^2 \tag{4.3}$$

where

$$rand_i = (m_i * k) mod \ p^2, p < k \tag{4.4}$$

$$f^{-1} : m_i = (c_i \ mod \ k^2) mod \ p \tag{4.5}$$

Configuration and conditions: For this purpose , the proposed encrypting system requires the following:

1.k and p are large secret prime numbers where p < k, where (k, p) are known only to the user;

2. For any given slope and m_j, *the following condition must be satisfied* :

$$\forall \ m_i \forall \ m_j, \ m_i + m_j < p \ and \ m_i * m_j < \ p$$

We encrypt and decrypt using k units. The additional symmetric shape is achieved by applying $f(m_1 + m_2) = f(m_1) + f(m_2)$, *and multiplication symmetry with* $f(m_1 \times m_2) = f(m_1) \times f(m_2)$.

Key generation: k and p are two large prime randomly .

Encryption: to encrypt a given message $m \in M$, *the cloud user applies the function* $f \circ f$ *equation* 4.3 *to obtain c.*

Decryption: to decrypt an encrypted message $c \in C$, *the cloud user applies the function* f^{-1} *of equation* 4.5 *to obtain m, where k is secret.*

Algorithm 0: Our Shema

1 Calculate: static k $\in$ *a large prime number*;
2 Calculate: static p $\in$ *a large prime number*;
Encryption Enc(m)
Input : m
1 Calculate: $rand_i = (m_i * k) \ mod \ p^2$
2 Calculate: $c = (m_i + rand_i * p) \ mod \ p^2 + m_i * \ k^2$
Output : c
Decryption Dec(c)
Input : c
1 Calculate: $m = (c \bmod k^2) \ mod \ p$
Output :m

4.3.1.1 Fully Homomorphic Encryption

Below this title, we discuss the homomorphic concept of the encryption technology that we have proposed, as we will explain its characteristics and advantages according to the requirements of cloud computing, this approach provided the advantage of fully homomorphic in addition to maintaining order of the data during encryption without resorting to other methods or using an index to index data, and this feature is not present in previous encrypting techniques. Where the cloud user computes the keys k and p. After that we have f is a function of M in C, and m_1, m_2 *are two elements of* M, *if the sum of* $m_1 + m_2$ *and it belongs to* M, *when applying* f *we get the element* $f(m_1+m_2)$ *of* C. *On other hand, after applying the function* f *we get the elements* $f(m_1) + f(m_2)$; *The sum of these new elements in* C *is* $f(m_1) + f(m_2)$.
We convert the encrypted relation from :

$$C=(m_i + rand_i * p) mod(p^2) + m_i * k^2$$

To

$$\text{C=m}_i + £ * p + m_i * k^2$$

Which

$$\text{m}_i + £ * p < p^2$$

And from it the evidence will be as follows :
Evidence :

$$\text{f(m}_1) + f(m_2) = m_1 + £_1 * p + m_1 * k^2 + m_2 + £_2 * p + m_2 * k^2 = (m_1 + m_2) + (£_1 + £_2)p + (m_1 + m_2) * k^2$$

So

$$\text{f-1}(f(m_1) + f(m_2)) = m_1 + m_2$$

Evidence :

$$\text{f(m}_1) * f(m_2) = (m_1 + £_1 * p + m_1 * k^2) * (m_2 + £_2 * p + m_2 * k^2) =$$
$$(\ \text{m}_1 * m_2) + (m_1 * £_2 * p) + (m_1 * (m_2 * k^2)) + (£_1 * p * m_2) + (£_1 * p * £_2 * p) + (£_1 * p * (m_2 * k^2)) +$$
$$((m_1 * k^2) * m_2) + ((m_1 * k^2) * £_2 * p) + (m_1 * k^2) * (m_2 * k^2) =$$
$$(m_1 * m_2) + ((m_1 * £_2) + (£_1 * m_2) + (£_1 * £_2))p + (m_1 * m_2 + £_1 * p * m_2 + m_1 * m_2) + (m_1 * £_2 * p + m_1 * m_2 * k^2)k^2$$

So

$$\text{f-1}(f(m_1) * f(m_2)) = m_1 * m_2$$

4.3.1.2 Preservation Of Order

The goal of maintaining the order is to maintain the complete sequence of the original elements. The proposed encrypting technique maintains the order of the data after it is encrypted as if it were not encrypted on cloud databases. A simple expression of the form $x \times a^2 + b$ is used to provide such an order-preserving scheme. In order to hide the value of the used integers, the coefficient a is kept secret (known only to the cloud user) and the value of $x \times a^2$ is hidden in b using the addition operator. The obtained expressions should respect the indexing order. The order-preserving function is defined as follows: $c_i = m_i * k^2 + R_i$. By definition, $R_i = (m_i + rand_i * p)\ mod\ (p^2)$ is enclosed to $0 \leq R_i < p^2 < k^2$ which ensures the expression is strictly increasing. Therefore, $\forall(m_1,\ m_2)$, if $m_1 < m_2$ then $k^2 \times m_1 + R_1 < k^2 \times m_2 + R_2$. Hence, the basic expression respects the order of the used $m \in M$. This integrated indexing mechanism prevents attackers from breaking the indices without knowing $rand_i$ of any given m_i. We note that obtaining the security benefits of modular offset with order-preserving aim crucially relies on the fact that we only allow the adversary cipher-text to be attacked.

4.3.1.3 Grammar File (Encryption/Decryption)

In our approach, we used the JavaCC Analyzer Builder for the encryption and decryption process. After specifying the grammar rules in the. jj file in addition to specifying the glossary and grammar description of the data that we will encrypt or decrypt in the JJ file and convert it into a Java program that can identify the matches of the rules.
We run javacc on a .jj file. We will get seven Java files as output, including an official grammar analyzer written in EBNF encoding. The encrypting and decrypting process is mainly dependent on the .JJ file that contains the required rules for encrypting and decrypting. Figure 4.3 illustrates the encryption and decryption algorithm in a .jj file :

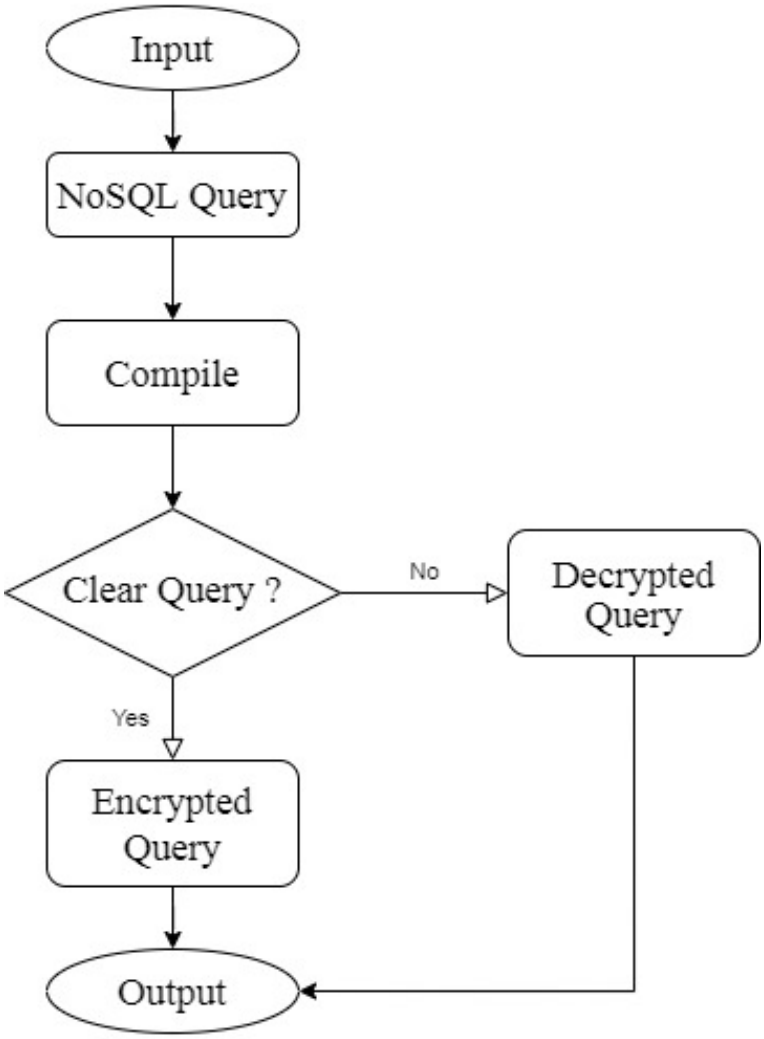


Figure 4.3: Algorithm Encrypte and Decrypte in JavaCC

4.4 Implementation Of The Solution

4.4.1 Development Tools

4.4.1.1 Apache NetBeans

We have chosen an Apache NetBeans environment as the environment for development & JAVA as a programming language.
Apache NetBeans is an open source development environment, tools platform, and framework that enables Java programmers to create desktop, mobile and web applications. The project was originally developed as part of a student project in 1996, later bought by Sun Microsystems in 2000, and over time it became part of Oracle after it acquired Sun Microsystems in 2010. NetBeans was introduced to the Apache Incubator in October 2016.

4.4.1.2 Java

JAVA is a simple object-oriented language which widely used and it is the most popular programming language for Android smartphone applications and is also among the most favored for the development of edge devices and the internet of things. it can be used under Windows, under Linux, and under other

Systems.[41]

4.4.1.3 JavaCC

Java Compiler Compiler (JavaCC) is the most popular parser generator for use with Java applications.[12][8][37]
A parser generator is a tool that reads a grammar specification and converts it to a Java program that can recognize matches to the grammar.[12][8][37]
You specify a language's lexical and syntactic description in a JJ file, then run javacc on the JJ file. You will get seven java files as output, including a lexer and a parser.[12][8][37]
JavaCC is similar to yacc in that it generates a parser from a formal grammar written in EBNF notation.[12][8][37]
Compilers have to perform three major tasks when presented with a program text (source code) .[12][8][37]:

1. Lexical analysis
2. Syntactic analysis
3. Code generation or execution

The bulk of the compiler's work centers around steps 1 and 2, which involve understanding the program source code and ensuring its syntactical correctness. We call that process parsing, which is the parser's responsibility.[12][8][37]
The parser is the second part of the front end of a compiler. The parser analyzes the input program. and determines whether or not it constitutes a legal sentence, or sentences, in the source language. The input to the parser is a stream of tokens produced by the scanner. For a valid input program, the parser produces a parse tree or some other intermediate representation of the input program.[12][8][37]
We can use the javacc as the parser generator to generate the parser for our compiler. In a jjfile we need to provide a context-free grammar for javacc to create the parser.[12][8][37]

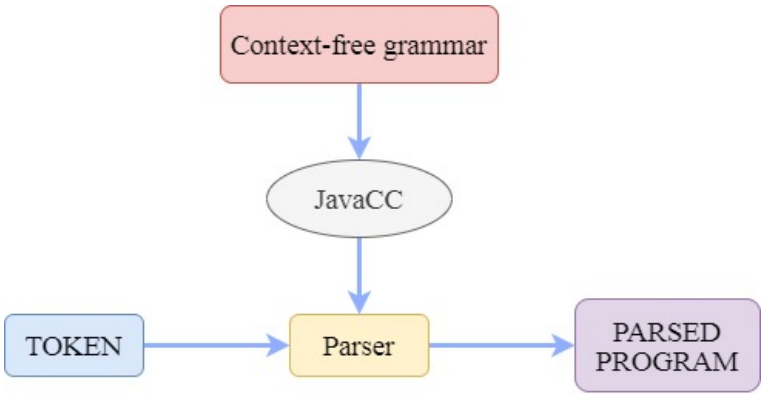


Figure 4.4: Java Compiler Compiler

4.4.1.4 MongoDB Atlas

MongoDB Atlas is a global cloud database service for modern applications. It is published by MongoDB and completely managed via AWS, Azure or Google (As per user choice) the best in class automation and proven practices guarantee availability, scalability, and compliance with the most demanding data security and privacy standards.
We will be working on a traditional NoSQL database. One of the most important benefits of cloud databases is that it can be accessed from anywhere any time by using the internet, and it also is scalable, and designed to achieve reliability and performance.[5][6]

4.4.2 Case Study

We describe our encryption system. For this, we present a simple scenario in which we assume that the cloud database is only accessible by the client. Our goal is to highlight the main stages of treatment:

- The client connects to the cloud-level database through the Proxy.
- The client creates, inserts, and interrogates his data.
- The proxy encrypts/decrypts the data and sends it to the cloud.

Here we have an example of some employees' listings as shown in the table below. In general, the NoSQL database consists of many documents in JSON-formatted. In this example, we will add documents for the employees' information. It contains three information, i.e. three fields: identity (ID), name (Name) and salary (Salary). The identification number represents the employee identification number, the name represents the employee's name, and the salary is his salary.

{ID:"415",Name:" Amine ", Salary: 100000}
{ID:"110",Name:" Ali ", Salary: 50000}
{ID:"187",Name:"Youssef ", Salary: 40000}

Original collection: Employee			**Encrypted collection: col_164...**		
ID	Name	Salary	rw_1$0$2 ...	rw_5$4$2 ...	rw_8$6$3 ...
415	Amine	10000	15742...	"5$1$8$..."	14027...
110	Ali	50000	96430...	"3$2$9$..."	93036...
187	Youssef	40000	05408...	"4$7$0$..."	31542...

Table 4.1: Original collection and encrypted collection

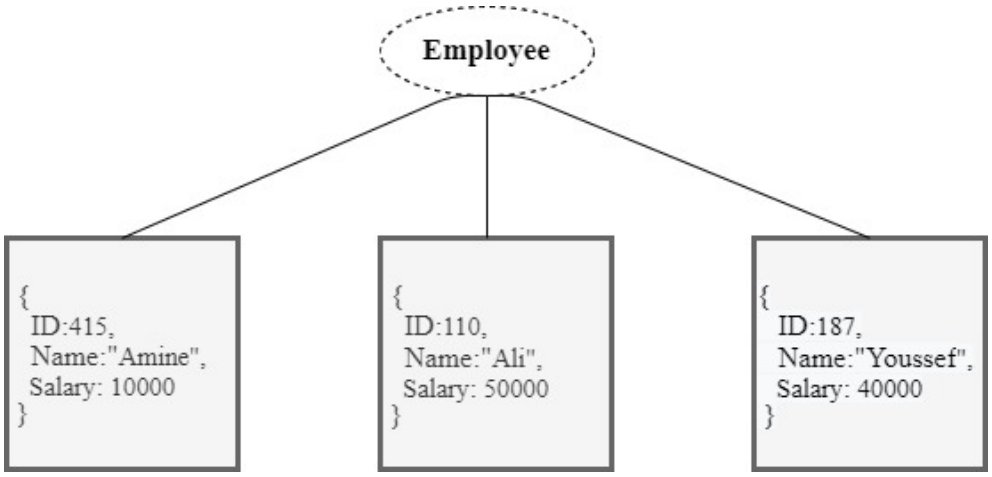


Figure 4.5: Original collection structure

Figure 4.5, The diagram above shows an original collection structure containing the name of all attributes in the form in which they are stored in the Mongodb database that supports and based on JSON.

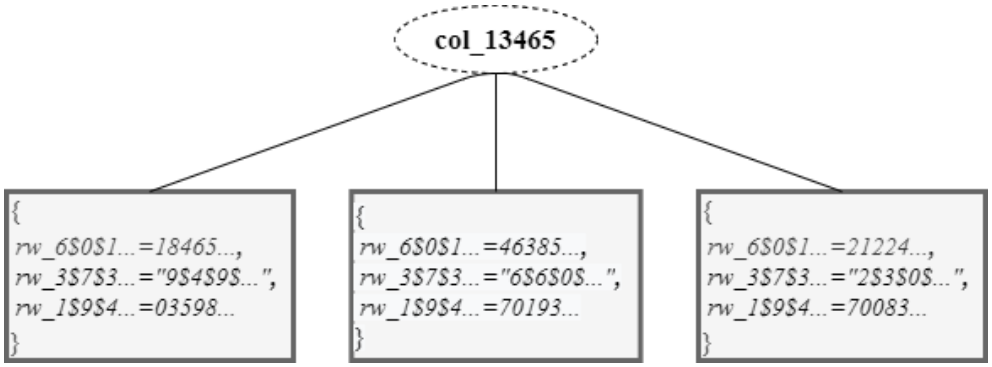


Figure 4.6: Encrypted collection structure

Figure 4.8. The diagram above shows a encrypted collection structure containing the name of all attributes in the form in which they are stored in the Mongodb database that supports and based on JSON.

4.4.2.1 Generation Of Querys

The proxy receives the original request and encrypts it according to the required function and structure in order to obtain a new encrypted new query in order for it to be implemented in the cloud database, and vice versa in the event that the proxy receives an encrypted query. Below we explain the basic queries in the database management process.

4.4.2.1.1 Insertion Query

When receiving the query by the proxy, it encrypts it using the encryption algorithm and after that it executes the encrypted query in the database to inserts a document encrypted into a collection. The form of the query is of this type as follows:

db.Collection_Name.insert({key : value, key : value ,..... })

4.4.2.1.2 Update Query

This query modifies an existing document or documents in a group. Depending on the update parameter, you can either modify specific fields of an existing document or documents, or replace an existing document entirely. The db.collection.update () method updates the document or documents that match the query criteria. The form of the query is of this type as follows :

db.Collection_Name.update({key : value},{$set:{key : value, key : value ,..... }})

4.4.2.1.3 Delete Query

Remove all documents that match the query criteria. The form of the query is of this type as follows:

db.Collection_Name.deleteMany({key : value, key : value ,..... })

4.4.2.1.4 Sum Query

The process of collection inside the NoSQL databases is done using aggregate data, after that calculate the aggregate values for the data in a collection or a view. In the query used in our proposal we use method of aggregate: $group. It means groups input documents by the specified _id expression and for each distinct grouping, outputs a document. Where the _id field of each output document contains the unique group by value. The output documents can also contain computed fields that hold the values of some accumulator expression or accumulator operator and we chose the sum process as an example of accumulator operator within databases, here the sum returns a sum of numerical values. and Ignores non-numeric values. The form of the query is of this type as follows:

db. Collection_Name.aggregate([{$group:{_id:value,key:{$sum:"$value "}}}])

4.4.2.1.5 Sort Query

In No databases, finding or returning documents inside NoSQL databases is done using the find () method, this method returns all documents from a collection and returns all fields for the documents that match the query criteria.
In order to obtain an arrangement for the data in the database, we applied the sort () to the index before retrieving any documents from the database, in order to determine the order in which the query displays the matching documents.The form of the query is of this type as follows :

db. Collection_Name.find().sort({value:1})

- **Ascending / Descending sort**

In the sort parameter, the field or fields for the sort, either the value 1 or the value -1 is specified in order to specify the sort type ascending or descending, respectively and the operation returns documents in the collection sorted in order by field name specified.
In the following table Table 4.2, the sample queries are presented with the encrypted version corresponding to each query.

Q	Original Query	Encrypted Query
1	db.employee.insert({id:"135",name:"amine", salary:30000})	db.col_2187...insert({rw_5$6$2..:1542.., rw_3$1$2$.. :"8$2$4$...",rw_5$7$1..:7185..})
2	db.employee.update({id:"135",$set: { id:"135", name:"amine", salary:30000}})	db.col_2187. . . update({rw_5$6$2..:1542..}, {$set:{$set: { rw_5$6$2..:1542..,rw_3$1$2$..: "8$2$4$...", rw_5$7$1..:7185..}})
3	db.employee.deleteMany({id:"135", name:" amine",salary:30000}	db.col_2187. . . deleteMany({rw_5$6$2..: 1542.., rw_3$1$2$.. :"8$2$4$...",rw_5$7$1..: 7185..})
4	db.employee.aggregate([{$group: { _id :null,result:{$sum:"$salary"}}}])	db.col_2187. . . aggregate([{$group:_id: null,result:{$sum:"$rw_5$7$1..:"}}}]
5	db. employee.find().sort({salary:1})	db.col_2187. . . find().sort({rw_5$7$1..:1})

Table 4.2: Original Query VS Encrypted Query

4.4.3 Comparison And Analysis

For our experiments, we first created a sample database and then set the query encrypting time for the following actions: insert, update, delete, sort (ascending / descending) and aggregate.

1. **Insert :**
   db.employee.insert({Id:123,Name:"Ahmed",Salary:20000})
2. **Update:**
   db.employee.update({Id:123},{ $set:{Id:123,Name:"Ahmed",Salary:20000}})
3. **Delete :**
   db.employee.deleteMany({Id:123,Name:"Ahmed",Salary:20000})
4. **Sort:**
   db.employee.find().sort({Id:1})
5. **Aggregate:**
   db.employee.aggregate([{$group:{_id:null,result:{$sum:"$Salary"}}}])

4.4.3.1 Experiment environment

- **The operating system (Windows) :** The WINDOWS 10 environment was chosen as the experiment environment for our software.

- **Characteristics:** Windows edition: Windows 10 Pro N.
- **System:** Processor Intel (R) Core (TM) i5-3210M CPU @ 2.50 GHz 2.50 GHz
- **Installed memory (RAM):** 8.00 GB

In this set of experiments, when our approach to NoSQL queries is implemented, we assess the common encrypting costs represented by encrypting time and decrypting time. Costs depend on two main factors:

- The type of query (including data size).
- key size (encryption key).

Table 4.3 below shows the results in three columns are data size and encrypted time and decrypted time ($\simeq$):

Query	**Key size (bit)**	**Encrypted data time (ms)**	**Decrypted data time (ms)**
Insert	128	57.66	4.66
Update		63.66	4.66
Delete		53.00	4.33
Sort		40.33	2.66
Aggregate(SUM)		42.66	3.33
Insert	256	63.66	6.00
Update		65.00	8.00
Delete		65.66	6.33
Sort		46.00	3.00
Aggregate(SUM)		43.33	3.00
Insert	512	65.33	9.00
Update		66.33	9.66
Delete		63.66	9.00
Sort		47.00	4.66
Aggregate(SUM)		42.66	4.66
Insert	1024	71.66	13.00
Update		80.66	16.00
Delete		72.66	14.66
Sort		53.00	7.33
Aggregate(SUM)		47.33	8.00
Insert	2048	84.66	23.33
Update		91.00	24.33
Delete		81.66	22.66
Sort		60.00	12.00
Aggregate(SUM)		51.33	10.66

Table 4.3: Encrypting time & Decrypting time (ms)

Experimental results of performance appear the encrypting time and decrypting time and displaying them in Table 4.3 after taking an average of 3 experiments per process, given that the elapsed time during the work of the processor during the implementation may be affected by several factors related to the work environment (of course the effect is very small).
The results show that the encrypting time does not exceed 0.09 seconds in all operations, and the decrypting time does not exceed 0.02 seconds for all operations. In general, the decrypting time is less than the encrypting time, and we can also notice a positive relation between the encryption keys and the encryption time and the decryption time for the operations described. Also, the time difference also

varies with the type of query. The main problem, perhaps in this approach, is the great amplification that this algorithm does to the size of data, this matter in practice poses an obstacle only to storage resources or non-cloud databases.
This problem was solved here because storage was at the level of the cloud that provides sufficient storage space. These results are very important because they confirm that the proposed algorithm is a valid and practical solution to ensure data security at different security levels depending on the size of the encryption keys at the level of cloud database services.

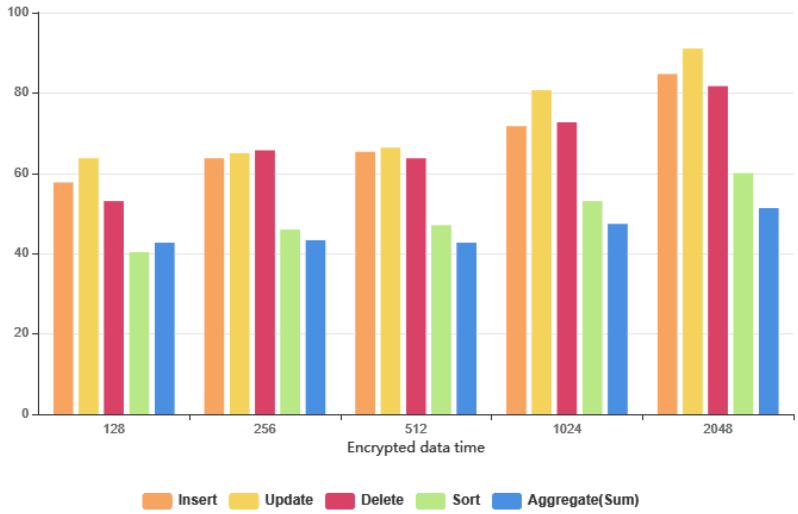


Figure 4.7: Encrypted data time(ms)

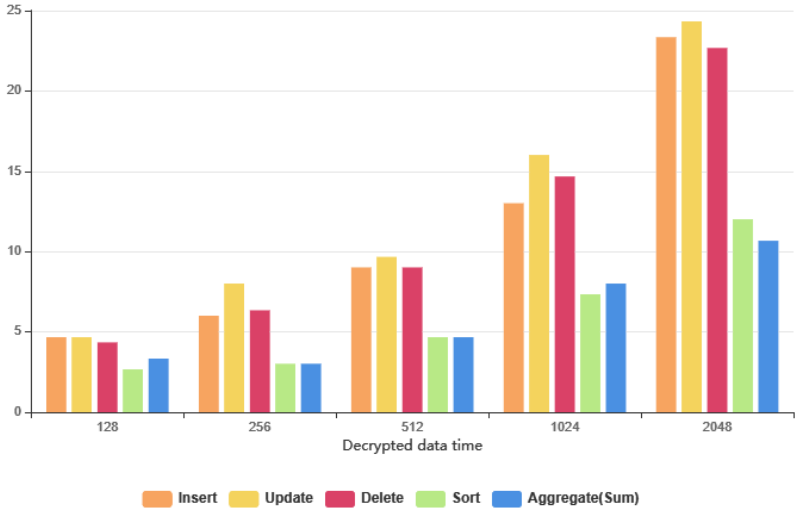


Figure 4.8: Decrypted data time(ms)

4.4.3.2 Problems

Mongodb Atlas (Use as a free service) :

- It does not support many data types.

- Not supported BigInteger or BigDecimal.
- It imposes limits on the size of the data.

4.4.3.3 Solutions

Many of these problems can be solved when using paid services, as this allows spaces to be customized according to user demand.

4.5 Interface And Examples

To facilitate the use of our model, it is important that it has a graphical interface.

- **User Interface**
  User performs operations (Insert, Update, Delete and Sort ID (Order) and It shows the total salaries)to encrypted data at the cloud level.

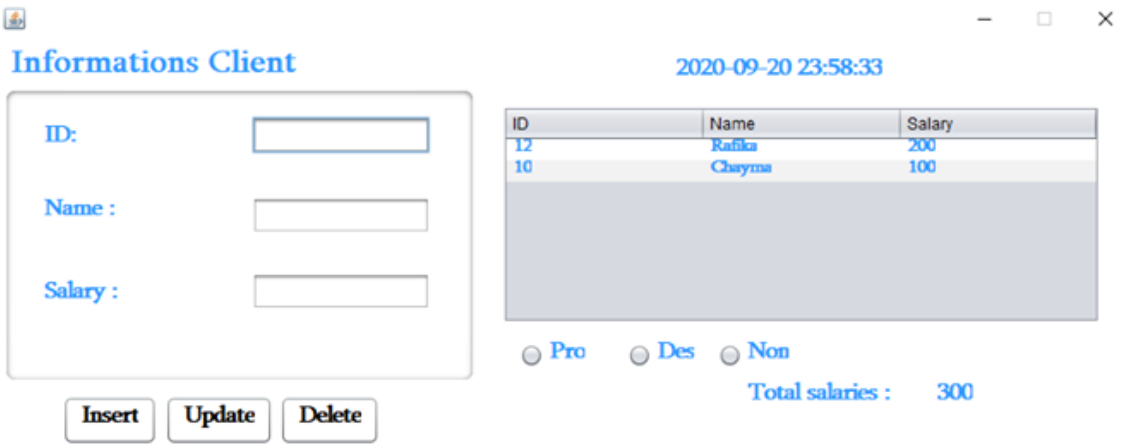


Figure 4.9: User Interface

- **Proxy Interface**
  It displays the clear and encrypted querys for each process used by the user.

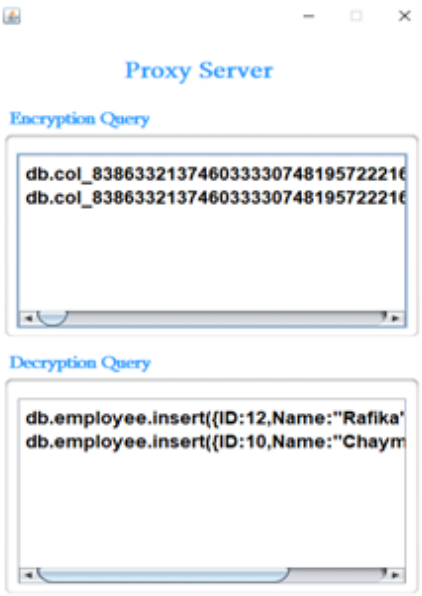


Figure 4.10: Proxy Interface

- **Encrypted Data Located In The Cloud**
  Data at the cloud level is completely encrypted.

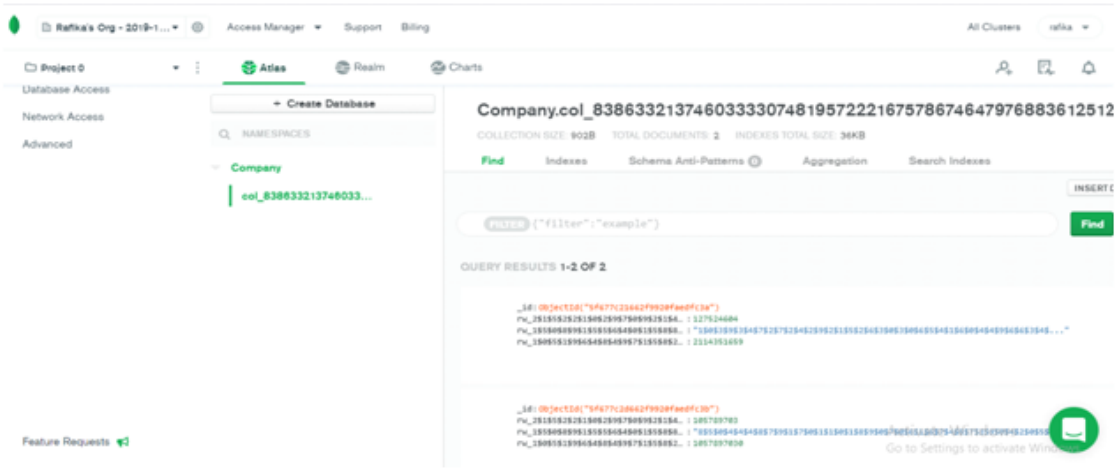


Figure 4.11: Encrypted Data Located In The Cloud

4.6 Conclusion

Data security is a critical problem in a cloud computing environment; in this study, we suggest a way to protect stored cloud data. We worked on fully homomorphic encoding and practical encoding and is easy to use because it depends on linear and standard expressions. We have adapted FHE to the NOSQL database.
Moreover, operations on encrypted data and recovery of desired results after decryption are allowed, as well as maintaining the arrangement after the encryption, and this matter is not found in other previous techniques and this adds a very large value to the proposed technique.

General Conclusion

With the enormous technological advancement, cloud computing with non-relational databases NoSQL, provided ideal solutions to meet the challenges resulting from the use of data and big data. among the problems of cloud computing that remain despite the development in this technology are the difficulties and security challenges facing the cloud computing service provider on the one hand and facing the user on the other side.

In this dissertation, we have proposed a method to generate fully identical encryption on data for storage in non-relational databases NoSQL on Mongodb cloud Atlas. A scenario was presented that assumed that the cloud database would be accessed by users through a proxy server. The efforts made aim to meet the needs of cloud users by providing technology of security through the use of an effective encryption algorithm. This theme addresses this challenge of cloud computing.

In the general introduction and the first two chapters, we focus on the concept of cloud computing and non-relational databases NoSQL Mongodb. In Chapter Three we discuss security issues, privacy and security threats.

In the last chapter (Chapter 4) we introduce an approach based on the data encryption and storage system in the cloud. We take into consideration the client level and the encrypted data is processed at the cloud level.

In conclusion, this dissertation allowed us to examine a wide range of concepts, models, and technologies in the areas of data security and the cloud. Our goal was to achieve a security technology that allows users to ensure the confidentiality of their data at the cloud level and to provide technology that allows to solve the security problems of data stored in the cloud, with an emphasis on remote data handling. We have also shown that the proposed work joins a rich and encouraging research topic.

This theme forms the basis of the work through which we can initiate a new research work in order to improve the submitted work. Finally, we confirm that data security in the cloud is still fraught with challenges and great importance given that the use of the cloud has become a necessity in all areas where there are still many research problems that must be identified and investigated in the issue of insecurity of non-relational databases at the cloud level.

Bibliography

[1] <https://docs.mongodb.com/manual/core/data-model////>. [Online; accessed 13-12-2019].

[2] <https://medium.com/@zhenwu93/relational-vs-non-relational-databases-8336870da8bc>. [Online; accessed 2020].

[3] <https://www.pluralsight.com/blog/software-development/relational-non-relational-databases>. [Online; accessed 2020].

[4] <https://intellipaat.com/blog/whatmongodb/#_MongoDb////>. [Online; accessed 11-12-2019].

[5] <https://www.mongodb.com/cloud/atlas////>. [Online; accessed 2020].

[6] <https://www.mongodb.com/cloud/atlas/faq////>. [Online; accessed 2020].

[7] The apache software foundation announces apache® netbeans™ as a top-level project. <https://www.globenewswire.com/news-release/2019/04/24/1808620/0/en/The-Apache-Software-Foundation-Announces-Apache-NetBeans-as-a-Top-Level-Project.html////>. [Online; accessed 2020].

[8] Build your own languages with javacc. <https://www.javaworld.com/article/2076269/build-your-own-////>. [Online; accessed 2020].

[9] Cloud computing benefits: 7 key advantages for your business. <https://www.globaldots.com/blog/cloud-computing-benefits//>. [Online; accessed 16-12-2019].

[10] Cloud security/cloud computing security. <https://www.beyondtrust.com/resources/glossary/cloud-security-cloud-computing-security//>. [Online; accessed 17-12-2019].

[11] Introduction to mongodb. <https://www.studytonight.com/mongodb/introduction-to-mongodb///>. [Online; accessed 18-12-2019].

[12] Javacc parser. <http://web.cs.wpi.edu/~kal/courses/compilers/JAVACC/JavaccPaser.htm////>. [Online; accessed 2020].

[13] Le dns spoofing : explications. <https://www.securiteinfo.com/attaques/hacking/dnsspoofing.shtml////>. [Online; accessed 2020].

[14] Mongodb - data modelling. <https://www.tutorialspoint.com/mongodb/mongodb_data_modeling.htm////>. [Online; accessed 13-12-2019].

[15] Mongodb : An introduction. <https://www.geeksforgeeks.org/mongodb-an-introduction////>. [Online; accessed 19-12-2019].

[16] Mongodb and confluent platform. <https://www.confluent.io/partner/mongodb////>. [Online; accessed 18-12-2019].

[17] Nosql tutorial: Learn nosql features, types, what is, advantages. <https://www.guru99.com/nosql-tutorial.html////>. [Online; accessed 13-6-2020].

[18] What is cloud storage? <https://www.redhat.com/en/topics/data-storage/what-is-cloud-storage//>. [Online; accessed 10-12-2019].

[19] What is mongodb? introduction, architecture, features example. <https://www.guru99.com/what-is-mongodb.html////>. [Online; accessed 18-12-2019].

[20] Advantages and disadvantages of cloud computing. <http://www.ibs.com.cy/en/blog/advantages-and-disadvantages-of-cloud-computing//>, 2010. [Online; accessed 14-12-2019].

[21] Mongodb. <https://searchdatamanagement.techtarget.com/definition/MongoDB////>, 2018. [Online; accessed 18-12-2019].

[22] Aws vs azure vs google – detailed cloud comparison. <https://intellipaat.com/blog/aws-vs-azure-vs-google-cloud//>, 2019. [Online; accessed 10-12-2019].

[23] YAGOUB Mohammed Amine. *Une approche basée agent pour la sécurité dans le CloudComputing*. Pour l'obtention de grade de DOCTEUR ENSCIENCES Option :Informatique, 2019.

[24] J. Brodkin. Loss of customer data spurs closure of online storage service 'the linkup'. <http://www.networkworld.com/article/2274737/data-center/loss-of-customer-data-spurs-closure-of-online-stohtml////>. [Online; accessed 2020].

[25] D. M. et X. J D. J. *Practical homomorphic encryption over the integers,*. ArXiv Preprint ArXiv:1702.07588, 2017.

[26] M. KROTSIANI et G. SPANOUDAKIS. *Continuous certification of non-repudiation in cloud storage services*. In : Trust, Security and Privacy in Computing and Communications (TrustCom), 2014 IEEE 13th International Conference on. IEEE, 2014.

[27] Ahmed Mohamed Gamaleldin. *An Introduction to Cloud Computing Concepts*. Senior RD Engineer-SECC, 2013.

[28] Fellah Hadjer. *CLOUD COMPUTING ET SECURITE :Une architecture organique pour la sûreté de fonctionnement des processus métiers*. MEMOIRE POUR L'OBTENTION DU DIPLOME DE MAGISTER EN INFORMATIQUE OPTION : Intelligence Artificielle et Imagerie, 2014.

[29] MOHAMAD HAMZE. *Autonomie, sécurité et QoS de bout en bout dans un environnement de Cloud Computing*. Thèse de Doctorat, Unité de Recherche :Le2i - Laboratoire Electronique, Informatique et Image, l'Université de Bourgogne , Spécialité : Informatique., 2015.

[30] Torry harris. *CLOUD COMPUTING – An Overview*. Whitepaper, Torry Harris Bussiness Solutins, 2010.

[31] Zaïd KARTIT. *Contribution à la sécurité du Cloud Computing : Application des algorithmes de chiffrement pour sécuriser les données dans le Cloud Storage*. Faculté des Sciences, 4 Avenue Ibn Battouta B.P. 1014 RP, Rabat – Maroc, 2016.

[32] Grace Lewis. *Basics About Cloud Computing*. Carnegie Mellon ,Software Engeneering Institute, 2010.

[33] Yujian Du Ling Qian, Zhiguo Luo and Leitao Guo. Cloud computing: An overview. <https://www.researchgate.net/publication/221276709_Cloud_Computing_An_Overview/>, 2009. [Online; accessed 14-12-2019].

[34] Kurt Marko. Compare nosql database types in the cloud. <https://searchcloudcomputing.techtarget.com/tip/Compare-NoSQL-database-types-in-the-cloud////>. [Online; accessed 13-6-2020].

[35] Takuya Shuzuki Masayuki Okuhara, Tetsuo Shiozaki. *Security Architectures for Cloud Computing*.

[36] Takuya Shuzuki Masayuki Okuhara, Tetsuo Shiozaki. *Security Architectures for Cloud Computing*. fujitsu sci.Tech.J.,Vol 46, 2010.

[37] Sojan P R. An introduction to javacc. <https://www.codeproject.com/Articles/35748/An-Introduction-to-JavaCC////>. [Online; accessed 2020].

[38] M. JAKOBSSON E. Shi J. Staddon R. Masuoka et J. Molina R. CHOW, P. GOLLE. *Controlling data in the cloud: outsourcing computation without outsourcing control*. In : Proceedings of the 2009 ACM workshop on Cloud computing security. ACM, 2009.

[39] Muhammad Raza. What are the hidden costs of cloud adoption? <https://www.bmc.com/blogs/cloud-adoption-costs///>, 2018. [Online; accessed 14-12-2019].

[40] S. Kinger R.Kaur. *Analysis of Security Algorithms in Cloud Computing.* vol. 3, 2014.

[41] Margaret Rouse. Definition java. <https://www.theserverside.com/definition/Java////>. [Online; accessed 2020].

[42] R. SHIREY. *Internet security glossary.* version 2, 2007.

[43] H. TIANFIELD. *Security issues in cloud computing.* In : Systems, Man, and Cybernetics (SMC) 2012 IEEE International Conference on. IEEE, 2012.

[44] N. P. S. I. B. N. et H. J. R V. P. P., B. P. S. *Survey of various homomorphic encryption algorithms and schemes.* International Journal of Computer Applications, vol. 91, n %18, 2014.