8-2017

Enforcing database security on cloud using a trusted third party based model

Victor Fuentes Tello
University of Arkansas, Fayetteville

Enforcing Database Security on Cloud using a Trusted Third Party Based Model

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Science

by

Victor Fuentes Tello
Technological University of Panama
Bachelor of Science in Computer Systems Engineering, 2004
Monterrey Institute of Technology and Higher Education
Master in Information Technology Management, 2009
University of Panama
Specialist in Higher Education, 2011

August 2017
University of Arkansas

This thesis is approved for recommendation to the Graduate Council.

_____
Dr. Brajendra Panda
Thesis Director

_____
Dr. Merwin Beavers
Committee Member

_____
Dr. Wing Li
Committee Member

Abstract

Cloud computing offers a considerable number of advantages to clients and organizations that use several capabilities to store sensitive data, interact with applications, or use technology infrastructure to perform daily activities. The development of new models in cloud computing brings with it a series of elements that must be considered by companies, particularly when the sensitive data needs to be protected. There are some concerns related to security that need to be taken into consideration when a service provider manage and store the data in a location outside the company. In this research, a model that uses a trusted third party (TPP) to enforce the database security in the cloud is proposed. The model describes how a client processes a query securely by using encryption mechanisms and partitioning methods. The client establishes the communication with the TPP to retrieve the data from a cloud storage service. The TPP has two primary functions. First, perform a partition process over the data by using an index from one of the attributes in the table. As a result, the TPP sends to the cloud server the records in encrypted format with an index. Second, the TPP analyzes the client query to retrieve a segment of the data from the cloud based on the query conditions. The final result is submitted to the client in which a minimum workload is executed. Some simulations were performed to evaluate the efficiency of the model by using two partition techniques: Histogram based and Mondrian or Bisection Tree based partition. The strategy of the model is to process as much of the work at the TPP site and securely transmit the result. Using encrypted record in the cloud storage service prevents the provider to have any knowledge about the data and enforces the security of the database.

Acknowledgments

First, I would like to thank my thesis advisor, Dr. Brajendra Panda for all his guidance and dedicated work throughout the development of this research. Without his support and understanding in this process, this project would have never been finished. I would also like to thank Dr. Merwin Beavers and Dr. Wing Li, for being part of my committee.

Specials thanks to all the professors and staff in the Computer Science and Computer Engineering Department, and the Sponsored Students Office at the University of Arkansas for all the help during my master program. Also, thanks to the Fulbright LASPAU program and the Institute of International Education for having given me the opportunity to pursue my degree in the United States.

Most importantly, I need to recognize the support of my lovely family, friends, and co-workers for giving me encouragement when I need it most. I will always be grateful to you.

Dedication

To God, for giving me the gift of living.

To my parents, Victor and Elizabeth, for all their teachings and total dedication throughout my life.

To my brother and sister, Hector and Yarelys, for enjoying and living with me the greatest adventures.

To my nephews, Alberto Luis and Alberto Javier, for showing me that in the world there is much hope and joy.

To my family and friends, for sharing with me pleasant experiences.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1. Cloud Computing

Since the emergence of Internet in the world of information and communication technologies, many have been the developments and services that are offered to organizations around the world to increase their profits and global positioning regarding the implementations and techniques these organizations use. One of these services is Cloud computing which is a technology based on communication over the Internet that provides resources to clients by using hardware and software platforms without worrying about technical issues or execution at any time [1]. According to the National Institute of Standards and Technology (NIST), Cloud computing is a model that provides access to some services without using a considerable management effort by taking advantage of the use of computing resources such as communication networks, servers, software, infrastructure, among others [2].

Cloud computing has given to the industry a different approach where companies can access some resources, computers with high storage capabilities, and services that help to perform their operations with fewer costs and economic benefits. Moreover, several companies such as Amazon, Google, and Microsoft have developed Cloud computing systems to offer services to businesses and users around the globe allowing complete access to more computing power and resources [3].

Cloud computing systems have five essential characteristics. On-demand self-service, where the users can access the services and resources without depending on human interaction each time they need. Broad network access, which means that the users access the capabilities using networks through different platforms and devices such as laptops, smartphones, tablets, and workstations. Resource pooling that allows different clients or customers to access the

resources (storage, memory, processing, others) dynamically by using a multi-tenant model depending on the client's demand and without knowing the physical location of the provider. Rapid elasticity, because the system releases its capabilities automatically to scale according to the demand, which gives the idea that the capabilities are unlimited regarding amount and time. Measured service, where the provider and customer can monitor the resource usage to optimize and control the operation at cloud level by applying a level of abstraction according to the service [2].

The service models in cloud computing are classified depending on the resources and services the users can access after a payment agreement with the provider. The service models are categorized into Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) [4]. When an organization uses a Software as a Service model, it allows the users to access the software and its functions whenever they need to use it (on demand) without worrying about the installation, maintenance, and backup services. The users in the organization access the software capabilities by using a web browser [5] [2]. In the Platform as a Service model, the user can develop or program on the cloud by using programming languages, services, libraries and tools that are available in the cloud infrastructure without managing or controlling the infrastructure [2]. When applying the Infrastructure as a Service model, the cloud services allow the customers to use capabilities related to storage, networks, processing, and computational resources that are needed to run required software and application. As in the previous model, users are not allowed to manage the cloud infrastructure but have limited control over several systems such as operating systems and network elements [2].

1.2. Advantages and challenges of Cloud Computing

Cloud computing offers several advantages to users and organizations regarding computational resources, capabilities, and architecture. First of all, the services offered by the providers are accessed on demand, and the users only need to work with the capabilities they need to perform their operations. Another advantage of using cloud computing is the cost for the organization because using cloud services reduces the amount of money invested in technology. Companies can select a plan from the provider at low cost, reduce the IT expenses, and have access to different services to cover its needs [6].

In most of the cases, the cloud infrastructure is in a remote location, and the provider offers an excellent mechanism to update and recover the data. It facilitates the time needed in case of problems to restore the service and operation of the company. Also, the services provided by the cloud are available to be accessed from anywhere at any time [1]. Regarding the storage capacity, cloud computing seems to offer unlimited space to save the data allowing the organization continue using the cloud for storage purpose without worrying about running out of space [6].

Another advantage of cloud computing is the opportunity for innovation in small and medium businesses because the services and tools that the cloud offers can be used to increase their position in the marketplace by developing software and applications that before were very expensive for small organizations to pay. Also, small businesses can scale the services they provide very fast attending their customers' demand due to the ability of software in the cloud. It allows the organization to adapt the software they develop and use with the possible new requirement in the market. Furthermore, cloud computing allows increasing the performance on the clients' computer because the computational task is performed in the cloud environment [7].

Some of the challenges cloud computing faces today are the data security and privacy because cloud computing models are new to several organizations. There are several concerns about how to properly manage applications, communication, and data in different levels. Due to these concerns, there are critical challenges in how to protect the data in a cloud environment, especially when the data is sensitive for organizations, and it is transferred to and from the cloud for regular activities [7].

1.3. Database as a Service Model

Since cloud computing systems offer significant advantages to businesses that implement the models described previously, there is a new paradigm related to the database management systems in cloud computing called Database as a Service (DaaS) or Multi-Tenant Database as a Service (MTDBaaS). This model is associated with the Software as a Service model (SaaS) because it offers database services to the users based on the capabilities that providers can deliver [8].

Formally, Database as a service in cloud computing is defined as a model that enables access to users' databases without having to install, configure, and manage all the software and hardware needed to work with a database management system. In this model, the providers have to guarantee the access and security of the data stored on their servers from possible threats [9]. The DaaS model has several advantages such as the pay-per-use costs which allow organizations to save money on IT resources, especially those related to database management systems. Another benefit is the database platform that can be accessed by the clients using applications or web browser which offer general and specific capabilities to perform the business activities with excellent performance [10].

1.4. Database performance in Cloud

In the last years, several solutions regarding database in cloud computing environment have been proposed due to the adoption of cloud models such as SaaS which allow providers to offer new and compelling solutions to their customers. The services provided not only allow business to run their operations using relational databases but also provide options to work with NoSQL solutions according to the necessities demanded by organizations. However, there are several concerns about data security and database performance while using cloud computing services due to the lack of control of the physical data and the location of the servers. Even though there exist some details regarding security and performance, the solutions presented by cloud service providers allow to protect the data and run a database with good results [11].

The database performance in cloud computing is affected by the characteristics of the platform used between the client and the provider. For example, how the data is stored in servers. If the data is in plaintext or an encrypted format, what type of information is in the cloud (images can consume more time for processing due to the memory needed to read it), among other factors can affect the database performance. Consequently, applications in the cloud are not significantly affected when executing on the web because the infrastructure in the cloud provider is similar to the that needed in an organization, and the differences are the computational power available on the provider site and the interaction between the parties using the provider application [12].

Maintaining database performance in cloud computing environment is very important for the clients, in particular for their satisfaction with the service. The virtualization process in the provider site also affects the database performance, and because of it, the provider needs to manage the usage of their resources in different scenarios. For instance, the disk usage needs to be maintained for proper resource distribution; the applications cannot interfere with each other,

and the workload of the database also needs to be monitored for a proper function. A model for determining the CPU, memory, and the disk is useful when working with different workloads on a single server [12].

1.5. Purpose of this research

This thesis addresses the database security problem in cloud computing environment. Several authors have proposed models to store the data in the cloud using different techniques, but one of the most relevant elements when dealing with a database in the cloud is how to secure the data. The communication between the provider and the client in cloud services models is managed by using a reliable mechanism and applying different techniques to a database to protect the data. One of the mechanisms used to protect the data is to use encryption techniques to change the content to be unreadable to users who lack authorizations. This encryption and decryption process makes the data hard to understand for an intruder. Other mechanisms include partition methods to manage the databases in the cloud.

This document presents a model to increase the security of a database in a cloud environment by applying partition techniques and encryption options and using a trusted third party between the client and the cloud. The trusted third party is used to perform all the operations and to increase the security of the model. A simulation was conducted with different database structures and data records sizes to evaluate the proposed model and obtain results to assess about the efficiency of this design. This document is composed of different sections and includes a brief description of related works, the description of the proposed model, results, and analyses.

2. Background and Related Works

2.1. Introduction

How to protect the data when using cloud computing solutions is one of the main concerns mentioned in different papers and researches about this topic, in particular when dealing with databases due to the importance of the data for the organizations. There are several solutions related to database security on the cloud that need to be taken into consideration when evaluating which method can be applied to protect the data. Some studies propose solutions that address security on databases, but many of them increase the cost of processing when performing transactions. The idea is to use methods that reduce the cost and workload on clients. This section presents relevant information about security in cloud computing, performance in cloud databases, and partition methods as a technique used to protect the data.

2.2. Security in Cloud Computing

Choubey and Nendeo in [4] explain several issues related to security in cloud computing when working with a considerable amount of data between the client and the provider which states the necessity of using proper security mechanisms to protect the data. One of these issues is trust, and it needs to be addressed when companies use a trusted third party with cloud computing solutions. The third party will manage the confidence by using proper techniques such as defining policies that define and govern the relationship between the provider and the client. Also, the third party applies encryption mechanisms in the communication process that help develop a strong sense of trust in the cloud activities. Privacy and confidentiality are also mentioned by the authors in [4] due to the constant access to the services in the cloud. Personal information and credentials need to be protected by secure mechanisms and different locations to increase security in cloud models. Furthermore, the integrity and availability are also issues

presented by the authors as issues to be considered in cloud security. The implemented mechanisms in cloud environments need to guarantee data transfer between the parties in a secure way ensuring integrity and confidentiality all the time.

Several authors have proposed methods to increase the security in communication when using cloud computing services. Hwang et al. in [13] developed a business model for cloud computing to manage the encryption and decryption process as a separate service. The authors propose to use different providers to control the cryptographic operations (Encryption/Decryption as a Service) and to store the data in an encrypted way (and Store as a Service). In this model, the researchers also used a customer relationship management (CRM) as an application system in the cloud to develop the data retrieval and data storage services. In the data retrieval process, after the login process, there is an interaction between the storage cloud service and the CRM cloud service to identify the user's data which is in an encrypted format. After finding the data, the system uses the encryption/decryption service with the user identification as an index to restore the data to the original form and to send it to the user. This process uses a key based cryptographic system to manage the data for each owner, and the CRM system displays the data to the final user.

Yan and Lai in [14] developed a security model for cloud computing based on Diffie-Hellman protocol (Elliptic Curve Diffie-Hellman ECDH) and symmetric bivariate polynomial based secret sharing. The model is called Secure Cloud Computing SCC, and it uses a secret sharing key system with the symmetric property. The authors present two variants of the SCC model. One of them uses a trusted third party mechanism to protect the data, and the other variation does not use it. The model without using a trusted third party is composed of three stages: (1) the key sharing stage, in which the shares are generated with a hash value of the secret

key; (2) mutual authentication stage, where there is a verification of the hash value with using the symmetric property in bivariate polynomial that creates an intermediate key; and (3) the key recovery stage, where the intermediate key is used to recover the key. The difference of this model when using a trusted third party is that the key sharing step is performed in a cloud server using a secure channel.

In the research presented in [15], Wang and his colleagues propose a combination of random masking with public key based homomorphic authenticator in a model that allows increasing the security in cloud computing storage by using a "privacy-preserving public auditing system." This method introduces a third party auditor (TPA) to verify the data integrity when using a cloud computing system. Once the TPA accesses the data, the method eliminates any knowledge that the auditor can obtain in the verification process. The complete process uses four algorithms in two main stages. The algorithms work with key generation for the users and also for data verification purposes. Two of the algorithms are related to the process in the cloud, in particular, to verify the correctness of the data storage and to audit the process by the third party. The audit process can be simultaneously performed to increase the efficiency.

In paper [16], Rahmani et al. proposed a new model to improve the security in cloud computing. The authors create a model named Encryption as a Service (EaaS) which uses a private cloud service in a third party. One of the advantages of this model is to work with a high number of requests by using a multi-threaded approach. For developing their model, the researchers first implemented a private cloud service to allow users to have more control over the infrastructure and gain control when using a private communication network. The second step was to work with encryption algorithms to create a cryptography library. In this second step, the model applies the message authentication code (MAC) to manage confidentiality and integrity.

The last step was to create a multi-threading model in which the features allow the model to improve the performance of the encryption and decryption process by using different levels and parallel processing.

Another option to increase the security in cloud computing environments is mentioned in the paper [17] presented by Cunsolo and his colleagues. The authors describe a solution based on a combination of symmetric algorithms with high performance and asymmetric algorithms that increase the security. Thus, a good option is to apply a symmetric algorithm to encrypt the data that will be transmitted over the network with an asymmetric algorithm to encrypt the corresponding key. In this case, the user or owner that has the private key can decrypt the symmetric key because the asymmetric algorithm is designed to have the key pair that works with the encryption and decryption functions.

A three-way architecture for securing data in the cloud is presented in [18] by Rewagad and Pawar. For managing the key exchange stage, the authors use the Diffie-Hellman algorithm because it allows the key exchange process to be secure. The authentication process is then performed by applying a digital signatures scheme. The third element in this architecture is the incorporation of the AES (Advanced Encryption Standard) algorithm to perform the encryption and decryption steps. The design uses two different servers in the implementation, one server for the encryption process (called trusted server), and another server to store the data. When a user needs to send a file to the server, the client and the server exchange keys by using the Diffie-Hellman algorithm. The second step is to perform the authentication process by using the digital signature, and finally, the file is encrypted with AES, and it is stored in the server. The same mechanism is used when retrieving a file from the server, and the final result is to obtain the decrypted file.

A cloud storage service architecture is described in paper [19]. In this research, Khanezaei and Hanapi developed a design with a combination of encryption methods (symmetric and asymmetric) for improving security in the cloud. The authentication process between the parties is the first activity in this architecture. Then, the AES technique is used to encrypt the file that the user sends to the cloud. When the user needs to retrieve the file from the cloud, a Cloud Storage System (CSS) is used to deliver the corresponding file after applying the security techniques related to authentication, and encryption/decryption. The contribution of this model is to guarantee a secure communication between the client and the cloud provider.

In their research, Kaur and Singh propose a framework that uses different algorithms such as RSA, AES, DES, and Blowfish to improve security in cloud environments in the encryption/decryption process. This design is called Cipher Cloud Architecture, and it has five layers. In this framework, the user can select the action to be executed such as view file, upload file, download file or delete a file from the cloud. After this operation, the file is sent to the Cipher Cloud in an encrypted format. There is a data access layer that guarantees each user can work with the corresponding file. Another layer named encryption framework allows the users to work with different cryptographic algorithms. This method uses a web connection via HTTPS to establish the connection between the client and the server in a private cloud [20].

2.3. Performance in Cloud Databases

Different factors can affect the database and systems' performance when using cloud computing solutions. In [21], Xiong et al. present a study to identify the resources needed by clients when using cloud computing capabilities by using a smart system to define the proper level of agreement. According to the authors, factors such as shared CPU capacity, memory, client workload, and the number of replicas can affect the system performance in cloud

environments. Taking into consideration the mentioned factors, several tests, and statistical information, the researchers used a machine learning model to learn about the relationship between the system and database performance. This model allows the user to determine a profit margin for the clients based on different resource allocations in the cloud.

A model to manage databases in the cloud is proposed by Zhao et al. in the paper [22] where the databases are placed in a public cloud. This approach uses a virtual machine to evaluate the performance when using several applications such as MySQL and Cloudstone on Amazon EC2. The research team tested the number of database replicas, the geographical location, the number of virtual machines, the amount of workload, and they found that all these factors need to be considered when working with databases in the cloud. However, the researchers conclude that a deep study needs to be developed to compare the performance using different systems and characteristics. Khanghahi and Reza in [23] describe various elements that not only affect the database performance but the use of cloud systems for users and organizations. The architecture of the cloud developed by providers, the capabilities or services they can offer to the clients, and the applications the users can interact with to work in the cloud are factors that can affect the database performance in the cloud.

2.4. Database Partition Methods in Cloud Computing

Taking into consideration the factors that affect security in cloud computing environments, several authors have developed different techniques to work with databases and to protect the data. In previous sections, various studies that use encryption techniques were presented. In [24], the researchers propose a method that allows working with SQL queries over encrypted data in a database executing as much as it is possible in the provider server. The process of decrypting the data takes place at the client site, and the results are presented after

completing the execution of the query. This method uses an equi-width partition technique to manage an index that allows the technique to work with encrypted data. Each relation of the database is stored on the cloud in encrypted form with an index that is assigned after applying the partitioning technique. This index is needed to process the query on the provider site. The paper also describes an algebraic framework to reduce the computation at the client site where the query is divided for this purpose.

Research developed by Omran and Panda in [25] presents three new partition methods when working with databases in the cloud. The central idea of this research is to improve the performance when processing the query and to increase the protection of the data stored in the cloud. The technique processes the SQL queries over encrypted data by using an index that is the result of a partition method. The authors compare four partition methods to determine the efficiency and to compare the times after using each method. Three of the methods were tested by Omran and Panda, and the last one was proposed in [24]. The partition operations take place on the client site, and it begins by deciding the attribute that will be used as an index to divide the data into partitions. By using a partition, each value in an attribute is assigned to a particular range [26].

In [24], the authors used the Equi-width technique to divide the data into several buckets of the same width. This method divides the data into the number of partitions needed by the user. The procedure in this technique is to obtain the difference between the maximum and minimum value of the attributes and then split the result into the number of partitions needed.

3. A Model for Enforcing Database Security on Cloud using a Trusted Third Party

3.1. Introduction

The development of different systems that allow users and organizations to work with their data over the Intenet has increased in the last years. Today it is common that companies adopt cloud computing service models to reduce the cost and investment in technology taking advantage of the providers' storage capacity and other options. One traditional cloud model is Software as a Service (SaaS) where companies can use providers applications to perform their daily operations. Several businesses can also outsource the technology platform, infrastructure, and database services with real benefits for the clients. Even though cloud computing has several advantages, some concerns about data security remain when discussing the challenges of cloud computing services, especially when there are questions about who can see the data. Moreover, researchers have proposed solutions that help companies to protect the data using techniques such as encryption and partitioning. In section 2, several studies on this topic were presented, and the information is used in this research to proposed a model that apply encryption and data partition techniques to manage data in the cloud.

3.2. Using a Trusted Third Party

Cloud service providers offer different capabilities to companies that are interested in using storage, processing, and communication services to reduce costs related to technology infrastructure (hardware and software) with high availability. Different authors have developed techniques using a trusted third party (TPP), in which the TPP has the function of managing the activities and communication between the parties. In this type of solutions, the data is controlled by the TPP, and the users in the companies need to request access to the data and services. In the model developed in this research, a TPP is defined as an external server that performs several

operations to store and retrieve the data from the cloud. The TPP works with a cloud server to increase the security of the data.

3.3. Data Encryption in Cloud

One of the main concerns about using cloud services is how to manage and protect the data that is stored on the servers as detailed in previous sections. Several authors have proposed in their studies to use encryption to protect the data. Omran in [25] suggests encrypting the data before sending the data to the cloud. The studies presented in the literature review for this research recommend using symmetric or asymmetric algorithms to perform the data encryption, where symmetric encryption is more recommended. There is an efficient method for working with an encrypted database on the cloud to overcome the problem of requesting all the encrypted records. By using an index related to the encrypted record, a database management system can send the corresponding records without decrypting the tuples to validate the conditions in a query. This technique helps in reducing the time and the workload to perform the operation with data from the cloud. The indexes can be stored in the cloud with the encrypted records to increase the database performance. As a result, the number of encrypted tuples and time used to decrypt the records will be reduced. The model developed for this research includes the use of encryption and decryption techniques to enforce the security of the data in the cloud.

3.4. Proposed Research Model

The proposed model is composed of three main segments: the client environment, a trusted third party, and a cloud environment. First, the original data will be sent to a trusted third party (TPP) from the client site to store the corporative database in a cloud storage environment.

The organization and the trusted third party (TPP) share a private key to encrypt and decrypt the files they are exchanging. The trusted third party has the function of performing the

partition techniques over the data, encrypt the records, and send the encrypted records to a cloud storage service. Figure 3.1 shows a diagram that illustrates the research model.

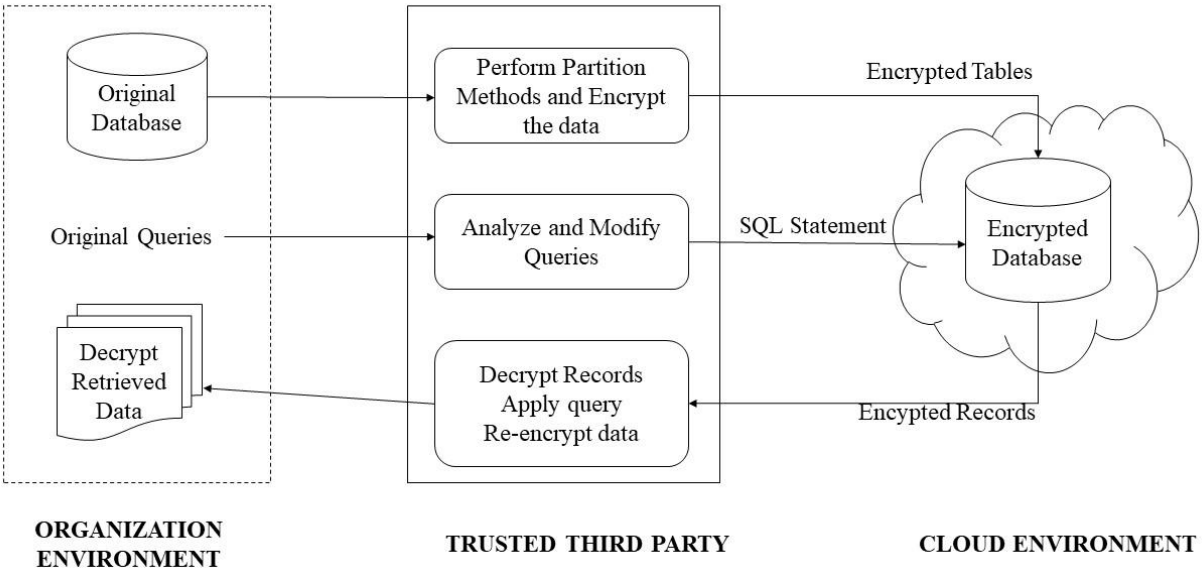

Figure 3.1. Proposed Research Model

Second, the TPP manages the queries from the client once the user in the organization sends a request. For this purpose, the TPP analyses and modifies the query depending on the partitioning technique used to obtain the indexes. Then, a new query is sent to the cloud storage service to retrieve the records that satisfy the condition. Finally, the TPP receives the records, decrypt the encrypted original tuples, apply the query conditions, encrypt the results using the shared key between TPP and the client, and sends the results to the client site. The communication between the TPP and the cloud uses a cryptographic technique to protect the connection between them.

During the initial storage, the TPP divides the data into partitions, and they are sent to the cloud storage service. For each relation (R) with its attributes (a) in the database called R(a_1, a_2, a_3, … ,a_n), there is an encrypted relation (E) in the cloud database named R^E[identifier,(a_1, a_2, a_3,

$\dots, a_n)^E$]. The "identifier" in the encrypted relation corresponds to the index for each tuple in the database. Each attribute in the database is associated with a partition index by using an identification function.

The goal in this model is to reduce the workload on the client site. To enforce the security over the data, the cloud service stores the data in an encrypted way which makes it hard for an intruder to decrypt the information. The TPP executes most of the workload in this model. First, the TPP creates the partition over the data, and then it manages all the request by using the result of the partitioning process. For this model, it is important to notice that the TPP share a different key with each organization. Thus, the TPP could offer the service to multiple organizations.

Taking into consideration the security concerns, this model allows the organization to work on their activities instead of being worried about security issues because it is managed by a third party. Also, the security concerns in the cloud environment are minimal because of the adoption of encrypted messages in the communication process between the parties. The advantages of the proposed model are summarized as follows:

- The main benefit of this model is that the work performed at the client site is minimum.
- The whole process is divided into two areas: the third party which uses mapping functions to find the data, and the cloud storage service.
- Storing the data in an encrypted way, and using cryptographic methods to manage the communication between the parties, allow to enforce the security over the data.

3.5. Research Model Implementation

The implementation of the proposed model uses the concepts and techniques explained in the previous sections. The trusted third party and the cloud storage service were implemented by using a server for each one. Each server uses Apache and MySQL services to manage the database capabilities. Moreover, the code used in the simulation was developed in Java. The first part of the implementation was to prepare the third party server to perform the partitioning methods and to analyze the queries from the client site. The analysis of each statement is fundamental in the implementation because the result of this analysis allow the TPP to obtain the index used in the partition process. The partition methods selected for this implementation were the Mondrian or Bisection Tree Based Partition as used by Omran and Panda in [25] and [26], and the Histogram Based Partition used in the research conducted by Hacigumus and colleagues in [24]. The second activity was to prepare the server for the storage cloud service to interact with the tables that contain the encrypted records and the index values.

Next step in the implementation was to execute the partition methods to store records in the cloud. In the next section, the time this operation takes for each partition technique is presented. For this activity, the client sends the database file to the TPP to perform the partition method. The index attribute is selected from the database in the TPP to obtain the ranges or segments depending on the method used. For each index, the record related to it is encrypted, and then the complete database is saved in the cloud storage server.

Once the cloud server has the encrypted database, the client can send the queries to retrieve the data. The client established the communication with the TPP in a secure way to submit a query, and after that, the TPP analyzes the query to manage the index by using an index table on this server. As a result, the query is modified, and then it is transferred to the cloud storage service to

retrieve the data. The cloud sends the records that satisfy the condition to the TPP by using a secure channel. The next step is to decrypt each record in the TPP and run the original query to obtain the correct results because the cloud delivers all the records that match the partition ranges. Finally, the TPP sends the results to the client using a cryptosystem to protect the data. The consuming times in the operations executed in this design are obtained for future analysis.

3.6. Partition Algorithms

In this section, the algorithms used to divide the data into partitions are presented.

3.6.1. Mondrian or Bisection Tree Based Partition Algorithm

The first algorithm in this section is the Mondrian or Bisection Tree Based partition used by Omran in [25] and [26]. This algorithm uses the database file that the client sends to the TPP. As part of the parameters, it needs the attribute to be partitioned and the termination condition that is related to the frequency of the values.

Input: Database table

Output: Partition category

1. Initialize partition set = {}

2. Select attribute from the table to be partitioned

3. Put values and its respective frequencies in an array

4. Sort the array (ascending order)

5. Define termination condition

6. Calculate the median of the values to obtain two groups (left and right)

7. If ((frequency in right group <= termination condition) or (one value in right group)

    7.1. Stop dividing

    7.2. If (group)

        7.2.1. Put the minimum value, the maximum value, and the category (sequential or random) in partition set

    7.3. If (One value in group)

        7.3.1. Put the value and the category (sequential or random) in partition set

8. If ((frequency in left group <= termination condition) or (one value in right group)

    8.1. Stop dividing

    8.2. If (group)

        8.2.1. Put the minimum value, the maximum value, and the category (sequential or random) in partition set

    8.3. If (One value in group)

        8.3.1. Put the value and the category (sequential or random) in partition set

9. Else GoTo (6)

10. Return (Partition set)

11. End

3.6.2. Histogram-Based Partition Algorithm

This algorithm divides the data into partitions of the same range, based on the number of partitions defined by the user. This technique was also used by Omran and Panda in [25] to compare their methods with the Histogram-based partition, and it was also used in [24]. As in the previous section, the result is the partitioning categories, and it needs the table containing the attribute that will be used as an index.

Input: Database table

Output: Partition category

1. Initialize partition set = {}

2. Select attribute from the table to be partitioned

3. Define the size of the bucket (part_size)

4. Obtain the minimum and maximum value of the attribute to be partitioned

5. Range = ⌐ ((maximum – minimum) / part_size) ¬

6. Min_Range = minimum, num_bucket = 0

7. Max_Range = Min_Range + Range

8. Put Min_Range, Max_Range, category number (sequential or random) in Partition set

9. Min_Range = Max_Range, num_bucket = num_bucket + 1

10. If (num_bucket < part_size) GoTo 7

11. Return (Partition set)

12. End

3.7. Retrieval Algortithm

Once the data is in the cloud server with a partition index related to it, the data can be retrieved to the TPP. This algorithm details the steps used to retrieve the data from the cloud storage service.

Input: Original query

Output: Data that satisfies the query

1. Read original query

2. Parse the query to obtain the partition index name and the values in the conditions

3. Use the partition set (from partitioning process) stored in TPP to get the partition range and the category number

4. Define a new query using the category

5. Send the query to the cloud storage service (the cloud service runs the query)

6. Read the result from the cloud

7. Decrypt the retrieved records

8. Use the original query with the decrypted records to obtain the final results

9. Return the result

10. End

4. Results and analysis related to the proposed model

4.1. Introduction

In chapter three, two partition methods were explained including several examples to illustrate how those methods work. The proposed model explained in this document uses the Histogram-based partition method proposed in [24], and the Mondrian or Bisection-Tree-based model proposed in [25] and [26] for indexing the data with the encrypted records on a server or Cloud environment by using a trusted third party between the client and the Cloud. In this research, a comparison related to the efficiency of the proposed model and those methods used in [25] are presented to analyze the performance of including a trusted third party between the client and a Cloud server. In this chapter, the performance results of the proposed model are presented and explained to show the models' efficiency.

4.2. Applying partition methods

Two partitioning methods were applied to a relational table called "Employee" to compare the response times in this research. The first part of the simulation was to send the complete data to the cloud server. For this activity, the trusted third party uses an index attribute, and it encrypts each record using a standard algorithm. Figure 4.1 shows the consumption time used to partition and encrypt the employee table having 100,000 records with a record size of 78 bytes. The values in the graph show that the Mondrian or Bisection Tree based method takes more time to complete the process than the Histogram based method. The trusted third party executes all the workload in this operation without the client participation.

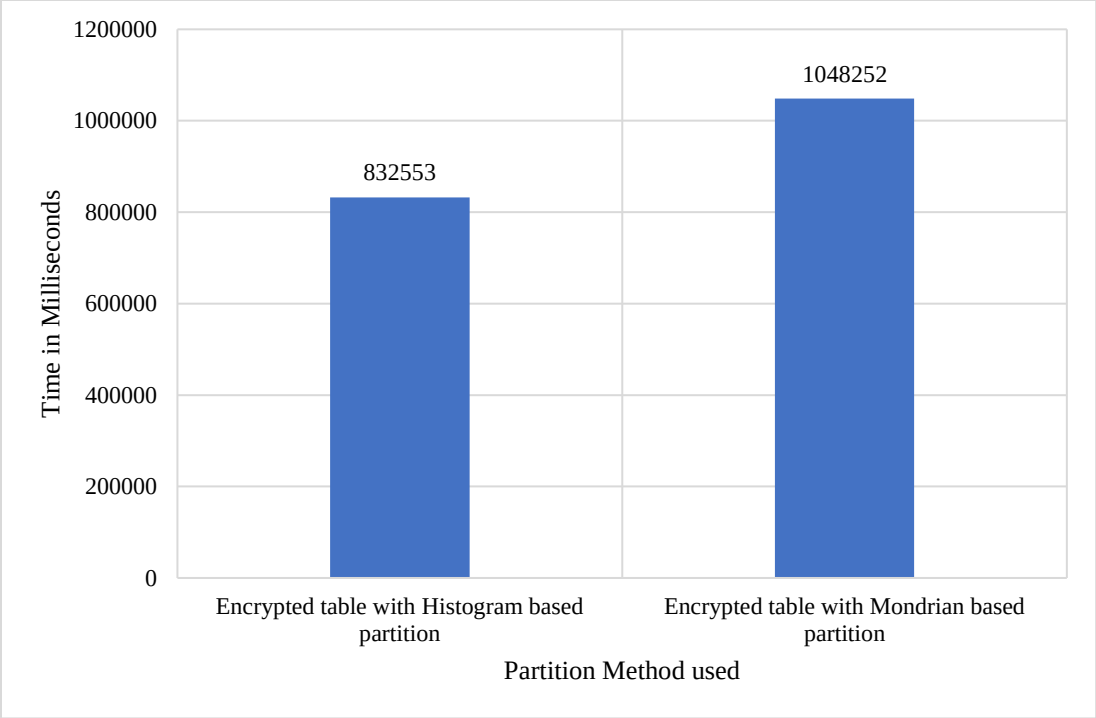

Figure 4.1. Consumption time to encrypt 100,000 records with record size of 78 bytes

4.3. Sending data to the Cloud

To send data to the Cloud environment by using a third party, the data is transmitted to the trusted third party in encrypted format by using a standard algorithm. The communication between the client and the trusted third party is protected by using a public and private key system. Once the data is with the trusted third party, the data is decrypted, and the query is analyzed to define the index that is assigned by performing a partition method. After the last operation, the records are sent to the Cloud server in encrypted form with an index for each record. The indexes are saved in a table in the trusted third party, and they are used to encrypt and decrypt the data from the Cloud. The communication between the trusted third party and the Cloud is protected by using a public and private key cryptosystem.

4.4. Retrieving data from the Cloud

Once the data is in the Cloud, a query is defined in the client site to retrieve the data. The main idea is to reduce the processing time on the client site and to process the operation in the trusted third party. The query from the client is protected by using encryption, and it is sent to the trusted third party. Once in the third party, the query is decrypted and analyzed to obtain the index by using the attribute name from the category list. The result is a new query that uses the index related to the partition to retrieve the data from the Cloud. After this operation, the query is sent to the Cloud to run and retrieve the data. Then, the trusted third party processes the query again to obtain just the results needed by the client. The final results are encrypted and sent to the client site, where the only operation performed on the client is the decryption of the results.

4.5. Applying partition methods using a trusted third party

For measuring the effectiveness, two partition methods were tested in this research: Histogram-based method presented in [24], and Mondrian or Bisection-Tree-based method proposed in [25] and [26]. Each partition method and an encrypted table without partition were tested to evaluate and compare the results. For evaluation purpose, four tables with different structures were created, and the records were stored using MySQL server. The records were processed in a simulated Cloud environment. Each method was tested using a table containing 100,000, 200,000, and 500,000 records respectively. The following sections show the results of different scenarios with and without using a trusted third party.

4.5.1. Data retrieved from a table having 100,000 records with record size 78 bytes

A table called “Employee” was created with six attributes for testing purposes with a record size of 78 bytes, and this table was populated randomly. The employee table structure is described in table 4.1, and it has 100,000 records. The attribute used for indexing using the

partition methods was “Salary”. Also, three different tables were used to evaluate the partition methods applied which include one table for testing the performance without using a partition method.

Attribute	Attribute Type	Size (bytes)
Id	int	5
Name	varchar	25
Address	varchar	30
Salary	int	5
Sex	char	1
Telephone	varchar	12
Total Size		78

Table 4.1. Employee table structure

Table 4.2 describes the conditions used in the where clause to retrieve different percentages of the data for measuring the performance of the partition methods utilized in the research for a table with record size 78 bytes.

Percentage of records retrieved	Where Conditions
5%	Where (salary >= 450 and salary < 3000)
15%	Where (salary >= 100 and salary < 7500)
25%	Where (salary >= 100 and salary < 55000)
50%	Where (salary >= 2000 and salary < 81000)
75%	Where (salary >= 1000 and salary < 95000)

Table 4.2. Conditions used in the where clause to retrieve different percentage of the data

After running the queries that include the where condition described in Table 4.2 using the employee table, the results are summarized in Table 4.3 where the time is measured for each partition method in milliseconds. Table 4.3 also shows the times for the encrypted table without partition and for the Histogram and Mondrian based partition with and without using a trusted third party to compare the times. The results show that the response time when using a trusted third party between the client site and the Cloud is greater than the time without using a third party. It seems to be logical because this operation requires an additional server to complete the

process. The client site performs less job in this method because the trusted third party performs the primary operations.

Percentage of Data Requested	Encrypted table without partition	Encrypted table without partition using a third party (TP)	Mondrian based partition	Histogram based partition	Mondrian based partition using a TP	Histogram based partition using a TP
5%	1134	3085	877	905	2045	2402
15%	1154	3439	971	993	2330	2570
25%	1184	3688	1028	1045	2674	2829
50%	1255	4232	1143	1164	3647	3769
75%	1308	4592	1274	1301	4342	4577

Table 4.3. Response time in milliseconds for each partition method (100,000 records with record size 78 bytes)

Figure 4.2 and figure 4.3 show the behavior for the response time when performing the operation with and without using a trusted third party using the different partition methods. As it is shown in the graph, the time for the encrypted table without partition using a trusted third party is greater than the time obtained without using the third party. Figure 4.3 shows the same behavior for the Mondrian and Histogram based partition methods. When using a trusted third party, the consuming time is higher. When requesting 25% to 75% of the records, the response times seem to be also greater. This behavior could be related to the ranges in the where conditions described in Table 4.2.

An important observation about the graphs is that the Mondrian based partition method is more efficient regarding the time when comparing with Histogram based partition method and the encrypted table without partition. Figure 4.4 shows the difference in time between Mondrian and Histogram based partition methods to clarify the information shown in the graph 4.3 which shows an overlapping of the lines.

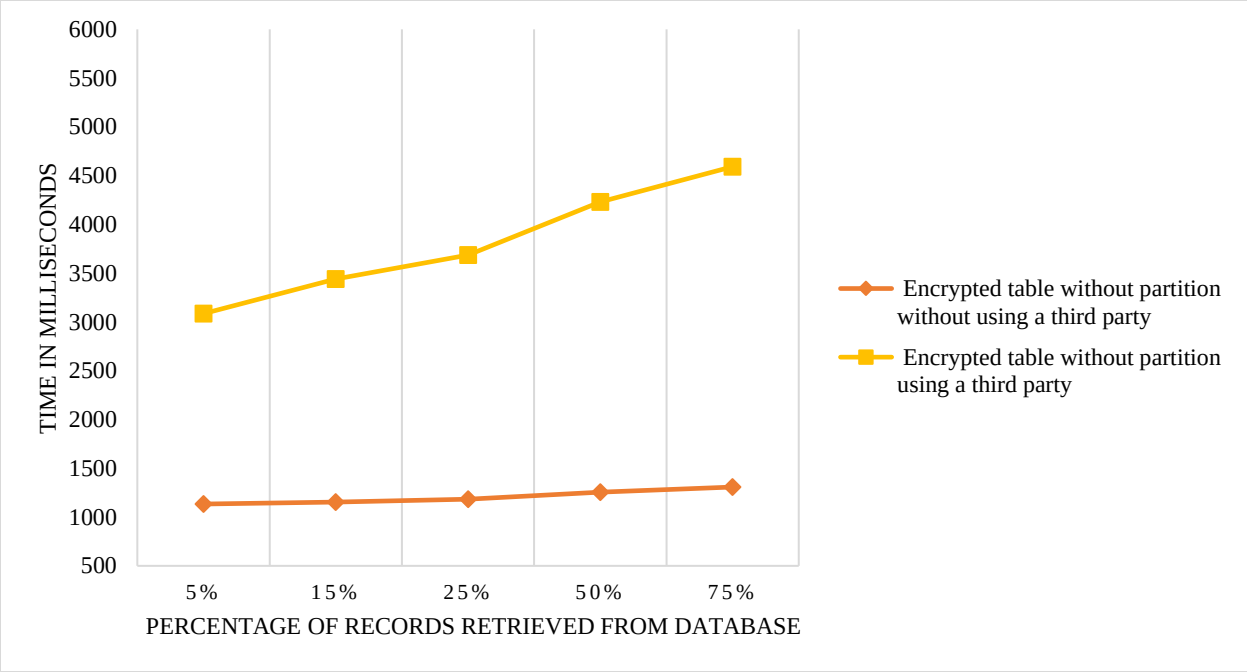

Figure 4.2. Response times for data retrieved from a table having 100,000 records with record size 78 bytes

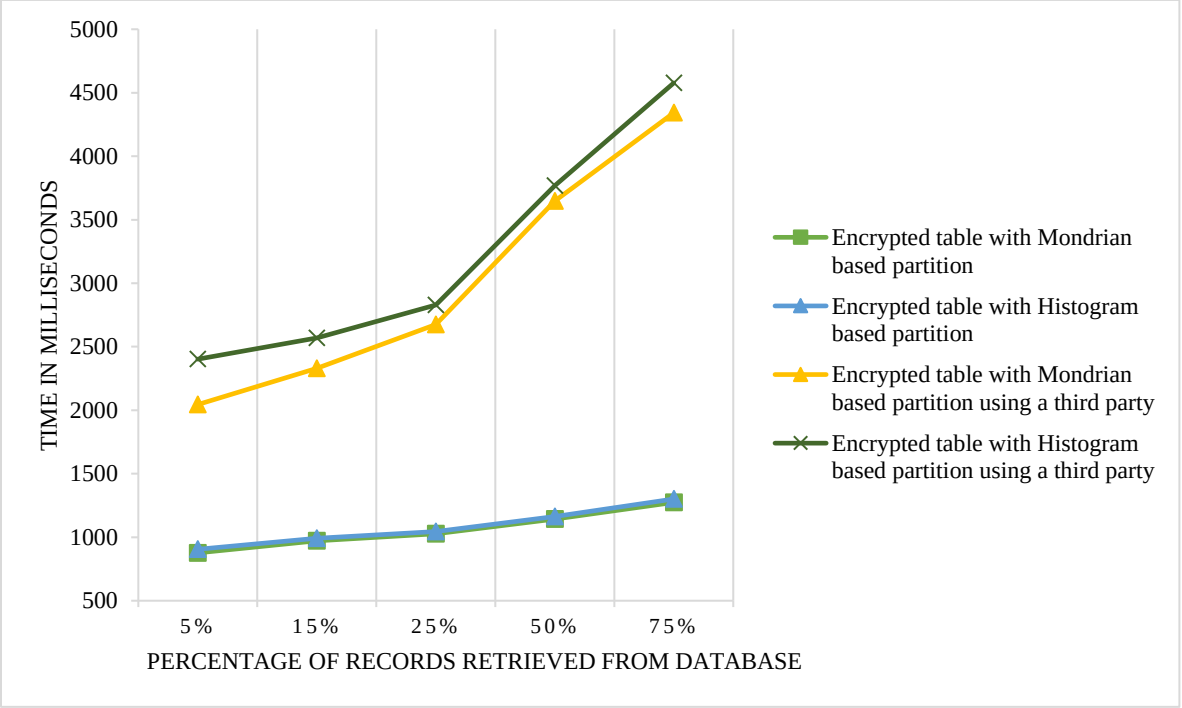

Figure 4.3. Response times for data retrieved from a table having 100,000 records with record size 78 bytes using Mondrian and Histogram based partition

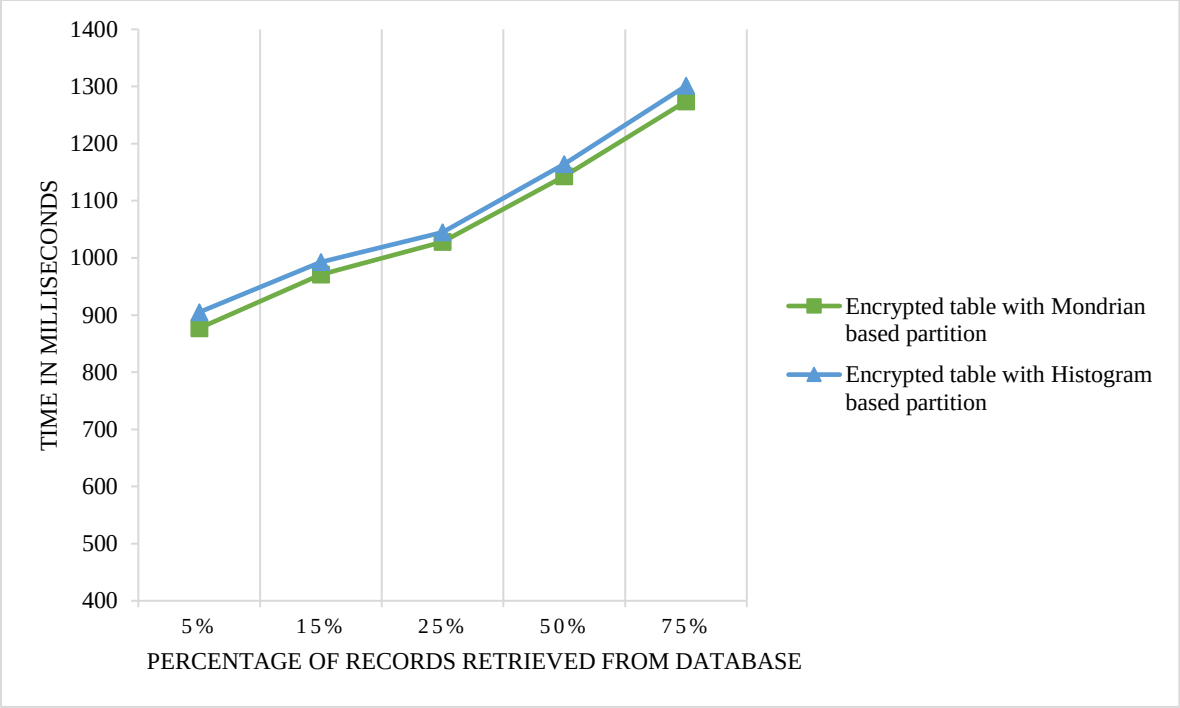

Figure 4.4. Response times for data retrieved from a table having 100,000 records with record size 78 bytes using Mondrian and Histogram based partition

Data Requested	Encrypted table without partition	Encrypted table with Mondrian based partition	Encrypted table with Histogram based partition
5%	100000	5793	9489
15%	100000	15187	21354
25%	100000	26391	29476
50%	100000	50824	54203
75%	100000	76466	89103

Table 4.4. Retrieved records for each partition method (100,000 records with record size 78 bytes)

The number of records retrieved after performing the queries with and without using a trusted third party is presented in Table 4.4. The number of records corresponds to different percentages of data retrieved from the employee table. When using an encrypted table without partition, the number of records retrieved is the total of records in the table (100,000) because the where condition cannot be applied to encrypted records. The number of retrieved records is the same whether a trusted third party is used or not.

Regarding the results, the Mondrian based partition method retrieved fewer records than the Histogram based partition method, which indicates that the first method seems to be more efficient than the second. Figure 4.5 shows the graph to illustrate the number of retrieved records after using the encrypted table without partition and with partition methods tested in this research.

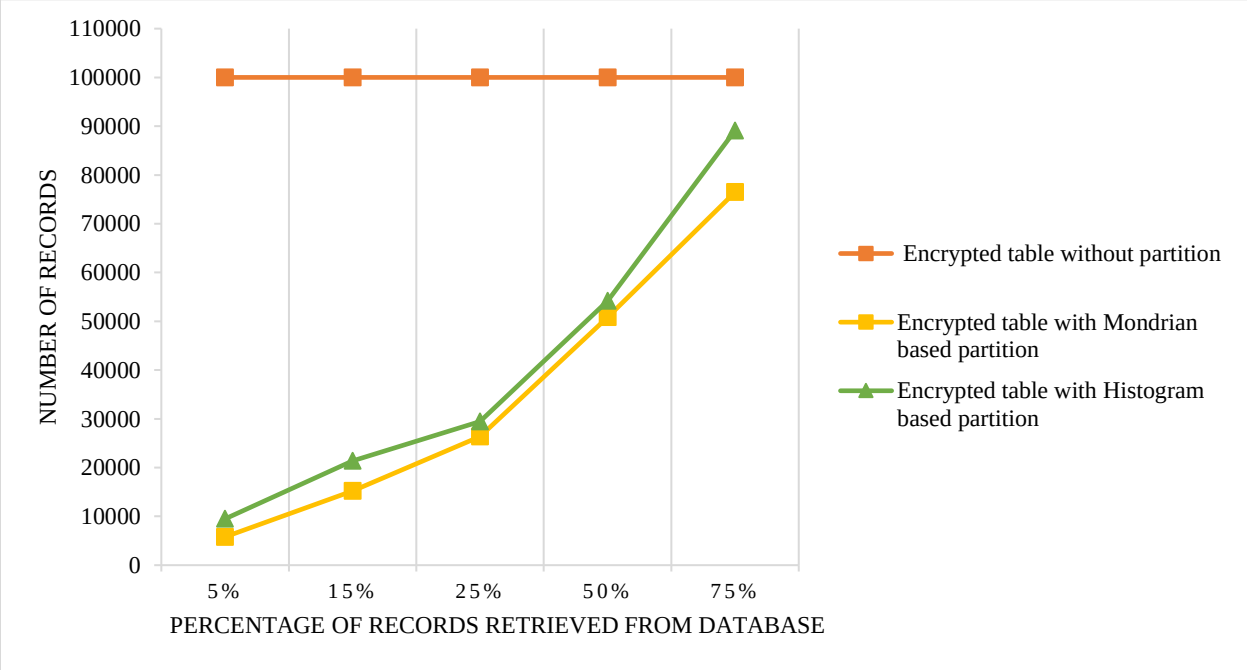

Figure 4.5. Number of records retrieved for each approach (having 100,000 records with record size 78 bytes)

4.5.2. Data retrieved from a table having 100,000 records with record size 178 bytes

A new employee table was created by increasing the record size to evaluate the efficiency of the model with a different table structure. This section presents the results using a table with 100,000 records having record size of 178 bytes. The employee table has seven attributes, and as in the previous case, the table was populated randomly. The employee table structure is described in table 4.5, and the attribute called salary was used as the index to perform the partition for both, Mondrian and Histogram based methods. Also, three tables were used to

evaluate the partition methods which includes one table for testing the performance without using a partition method, and two tables for the tested methods.

Attribute	Attribute Type	Size (bytes)
Id	int	5
Name	varchar	25
Address	varchar	30
Salary	int	5
Sex	char	1
Telephone	varchar	12
Comments	varchar	100
Total Size		178

Table 4.5. Employee table structure for 178 bytes record size

The information in Table 4.6 presents the conditions used in the where clauses to retrieve the data from the employee table to measure the performance of the partition methods when using a record size of 178 bytes for this research. After running the queries using the where conditions described in Table 4.6, the results are presented in Table 4.7 where the time is measured for each partition method in milliseconds. The information in Table 4.7 includes the times for the encrypted table without partition and for the Histogram and Mondrian based partition with and without implementing a trusted third party between the client and the Cloud to compare the times.

Percentage of records retrieved	Where Conditions
5%	Where (salary >= 100 and salary < 4000)
15%	Where (salary >= 1500 and salary < 9500)
25%	Where (salary >= 1000 and salary < 60000)
50%	Where (salary >= 10000 and salary < 85000)
75%	Where (salary >= 100 and salary < 95000)

Table 4.6. Conditions used in the where clause to retrieve different percentage of the data with a record size of 178 bytes

Percentage of Data Requested	Encrypted table without partition	Encrypted table without partition using a third party (TP)	Mondrian based partition	Histogram based partition	Mondrian based partition using a TP	Histogram based partition using a TP
5%	1210	1622	955	982	1233	1313
15%	1254	1691	1031	1059	1335	1408
25%	1391	1871	1180	1217	1580	1627
50%	1468	1979	1364	1412	1708	1938
75%	1611	2134	1517	1584	1963	2079

Table 4.7. Response time in milliseconds for each partition method (100,000 records with record size 178 bytes)

The results in Table 4.7 show that using partition methods is more efficient in terms of response time when compared with the times when using an encrypted table without partition. Also, the results show that the response time when used a trusted third party between the client site and the Cloud is greater than the time without using a third party as the results obtained in the previous section.

Figure 4.6 shows a graph of the response times with and without using a trusted third party between the client site and the Cloud server when querying an encrypted table without partition. In this graph, as the percentage of data requested from the server grows, the time it takes to complete the operation grows proportionately. Figure 4.7 shows the resulting graph about the response times when using the Mondrian and Histogram based partition methods. The times when using a trusted third party is greater for both approaches than that without using the third party. In term of performance, the results indicate that the Mondrian based method is more efficient than the Histogram based approach because it consumes less time than the second method. When the trusted this party is not used in the simulation, the difference in times is slightly different when using the partition methods, and the Mondrian based method is more efficient.

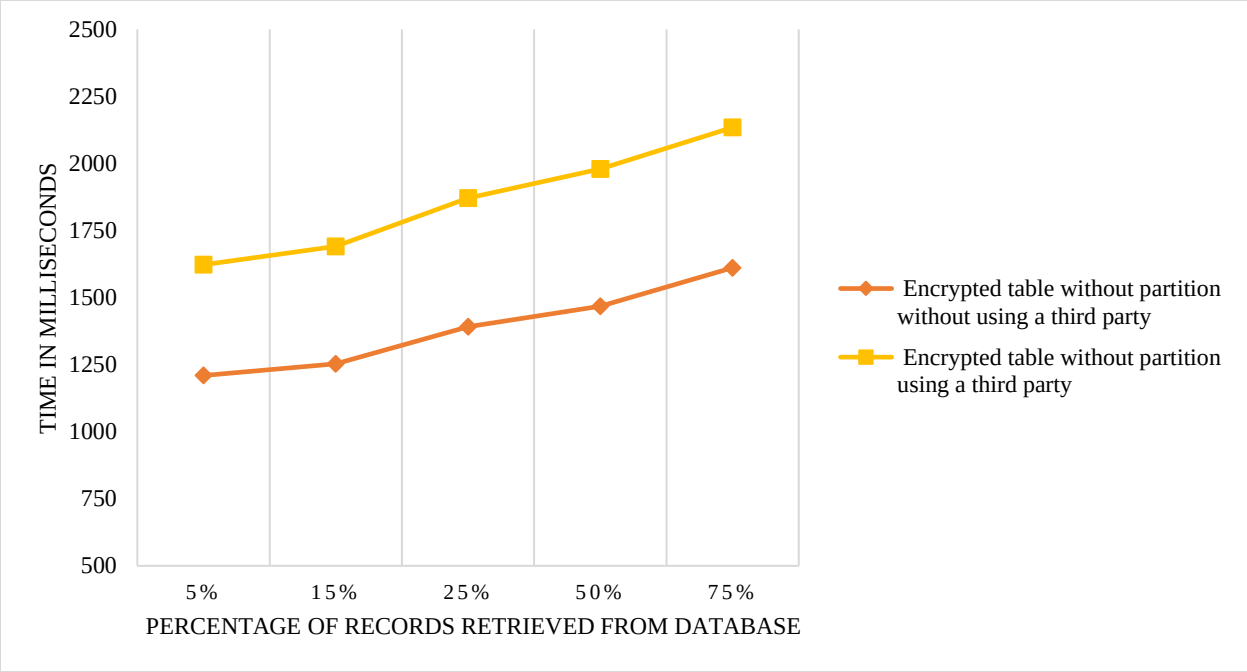

Figure 4.6. Response times for data retrieved from a table having 100,000 records with record size 178 bytes

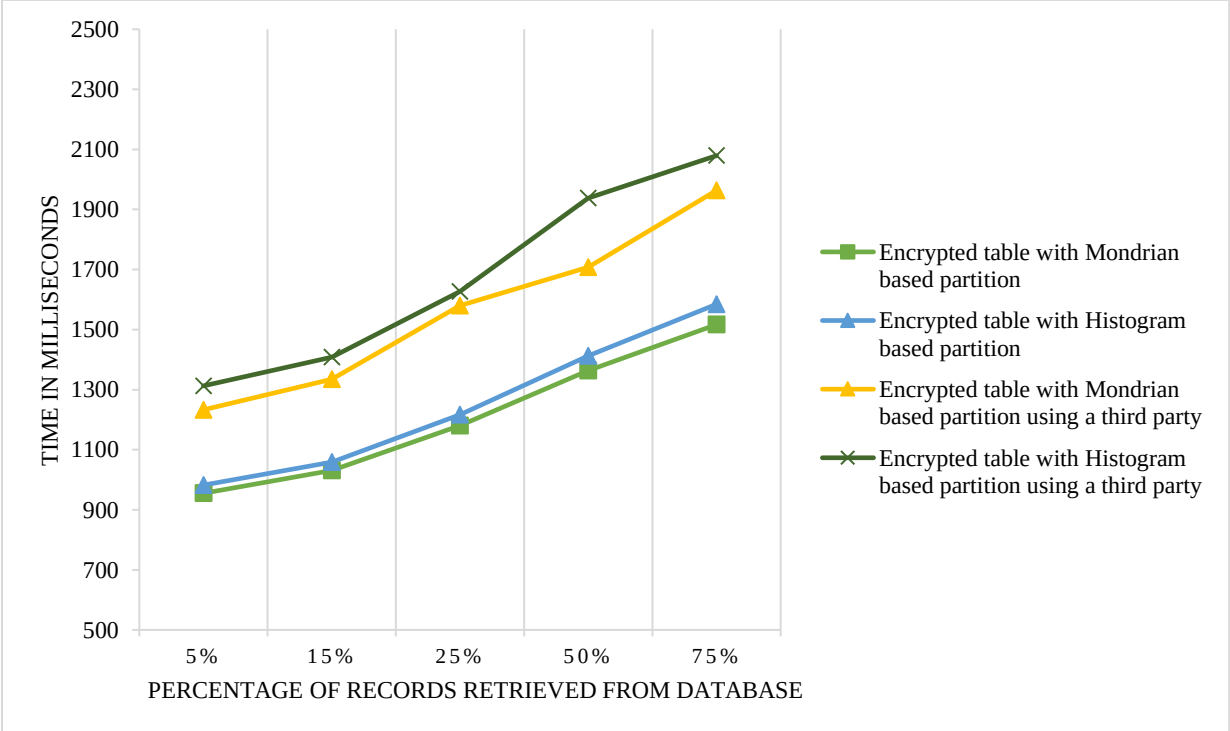

Figure 4.7. Response times for data retrieved from a table having 100,000 records with record size 178 bytes using Mondrian and Histogram based partition

The number of records retrieved after completing the operations with the employee table and record size 178 bytes is summarized in Table 4.8, and it represents the records obtained when requesting different percentages of data from the employee table. The number of records retrieved without using a partition method is the total of records in the table (100,000) because the where condition cannot be applied to encrypted records and the operation returns the complete table.

Figure 4.8 shows a graph representing the number of records retrieved by using the queries to get the data from the employee table. The number of records is the same in both cases, with and without using a trusted third party. The number of records when using the Histogram based partition method is greater than that when using the Mondrian based partition method.

Data Requested	Encrypted table without partition	Encrypted table with Mondrian based partition	Encrypted table with Histogram based partition
5%	100000	5631	8761
15%	100000	15436	20266
25%	100000	26761	29477
50%	100000	51326	54683
75%	100000	76852	86269

Table 4.8. Retrieved records for each partition method (100,000 records with record size 178 bytes)

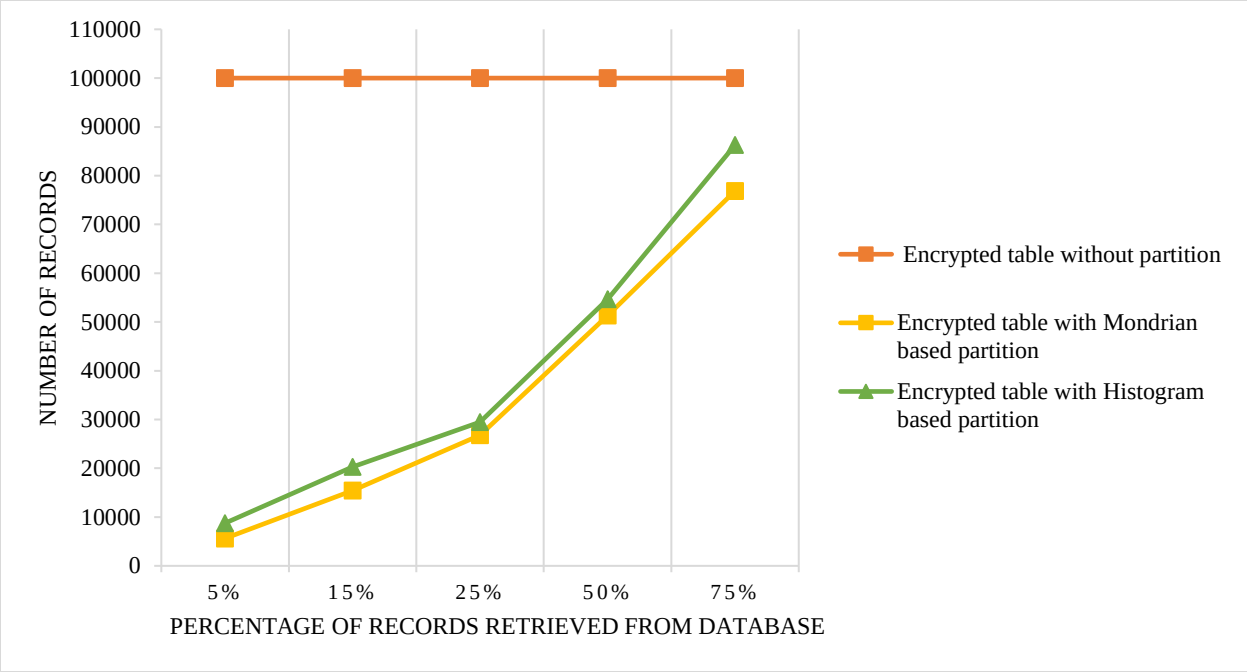

Figure 4.8. Number of records retrieved for each approach (having 100,000 records with record size 178 bytes)

4.5.3. Data retrieved from a table having 200,000 records with record size 78 bytes

This section presents the results when using the employee table with 200,000 records and record size 78 bytes. The table structure is the same as the structure described in Table 4.1 that has six attributes, and uses salary as the index attribute when performing the partitions. As in the previous sections, three tables were used to evaluate the partition methods which includes one table for testing the performance without using a partition method, and two tables for Mondrian and Histogram based partition methods.

After running the queries using the where conditions are shown in Table 4.2 to obtain different percentages of data from the employee table, the results are presented in Table 4.9 where the times are measured for each partition method in milliseconds. Table 4.9 is composed of different columns that indicate the times for the encrypted table without partition when using and without using a trusted third party. Also, the times for the Histogram and Mondrian based

partition with and without implementing a trusted third party between the client and the Cloud are included to compare the times among them.

Percentage of Data Requested	Encrypted table without partition	Encrypted table without partition using a third party (TP)	Mondrian based partition	Histogram based partition	Mondrian based partition using a TP	Histogram based partition using a TP
5%	1326	1804	1086	1164	1450	1591
15%	1383	1879	1185	1248	1629	1731
25%	1451	2044	1312	1300	1853	1778
50%	1506	2016	1450	1410	1985	1937
75%	1559	2166	1516	1548	2037	2120

Table 4.9. Response time in milliseconds for each partition method (200,000 records with record size 78 bytes)

As it can be seen from the Table 4.9, the times when using the partition methods are less than the time when using just the encrypted table without partition. It is the same behavior for all the cases when introducing a trusted third party between the client site and the Cloud. Thus, the information in Table 4.9 indicates that using the Mondrian and Histogram partition methods is more efficient regarding response time. As in the previous sections, the times for using a trusted third party is greater that the cases when the third party is not used.

Figure 4.9 presents a graph for the response time with and without using a trusted third party between the client site and the Cloud server when using an encrypted table without partition. In this graph, when the percentage of data requested from the server grows, the time it takes to complete the whole operation for the same proportion grows proportionately.

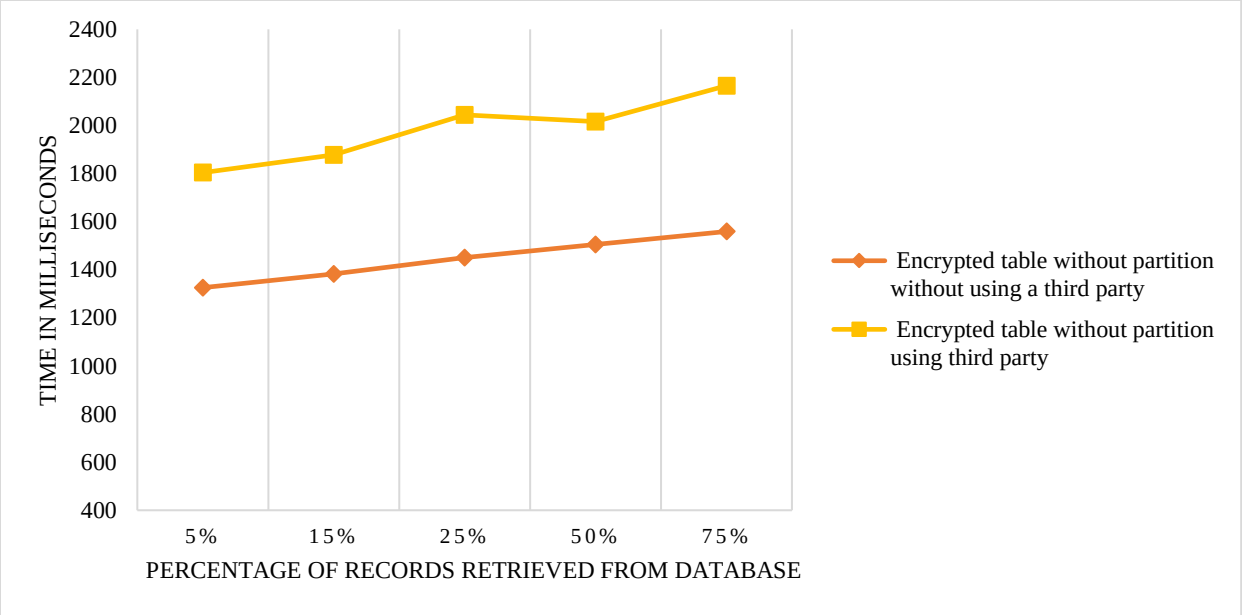

Figure 4.9. Response times for data retrieved from a table having 200,000 records with record size 78 bytes

Figure 4.10 shows the percentages of records retrieved from the employee table in the database, and the times it takes to perform each partition method with and without using a trusted third party. As it can be seen from the graph, the times when using a trusted third party are greater that the times when a third party is not introduced in the simulation. In this case, the trusted third party performs the complete operation before sending the final results to the client in an encrypted way. Thus, the client reduces the workload that is performed on its side while leaving the main processing job to the third party.

An unusual behavior in the graph in figure 4.10 occurs when retrieving 25% and 50% of the records from the employee table. In these cases, the Histogram based partition method is more efficient regarding time than the Mondrian based partition method. A possible cause for this behavior could be the number of records that belong to the partitions.

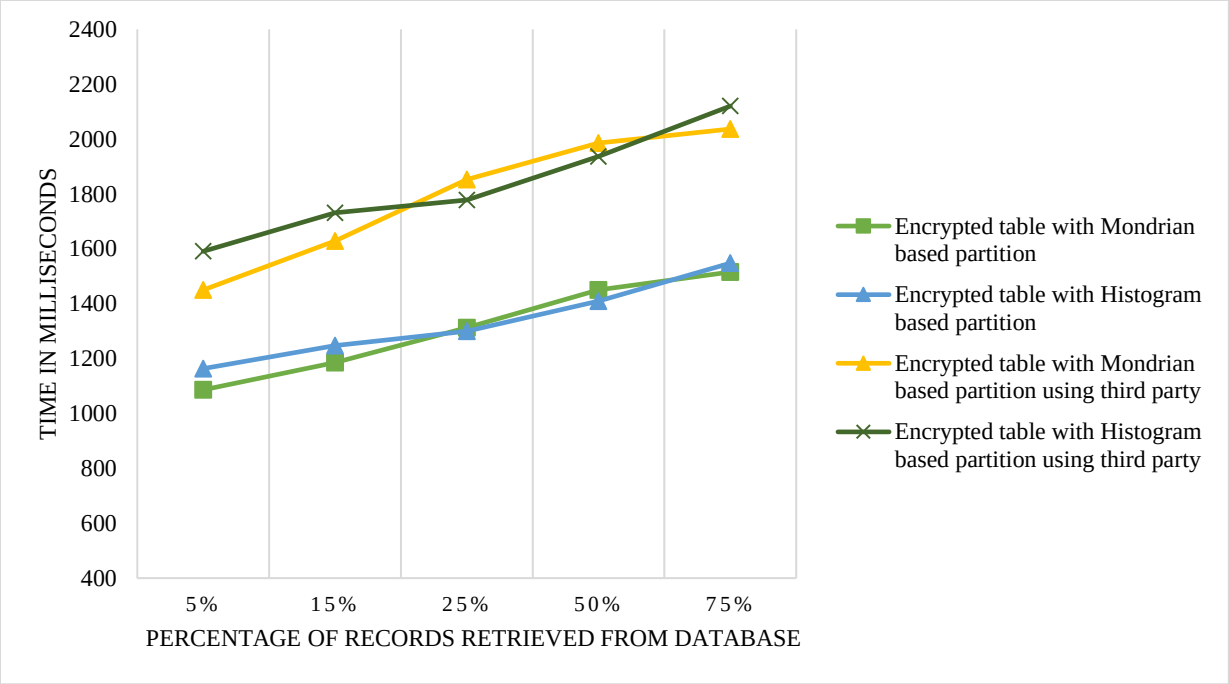

Figure 4.10. Response times for data retrieved from a table having 200,000 records with record size 78 bytes using Mondrian and Histogram based partition

Data Requested	Encrypted table without partition	Encrypted table with Mondrian based partition	Encrypted table with Histogram based partition
5%	200000	10427	20938
15%	200000	31021	35424
25%	200000	62194	60475
50%	200000	112835	115067
75%	200000	150793	162849

Table 4.10. Retrieved records for each partition method (200,000 records with record size 78 bytes)

The number of retrieved records using a table with 200,000 records and record size 78 bytes are presented in Table 4.10, and it can be seen that in the majority of the cases, the Mondrian based partition method seems to be more efficient. This table also shows the difference regarding some records between the partition methods when retrieving 25% and 50% of the data. Figure 4.11 also shows a graph that represents the number of retrieved records for different percentages of data requested.

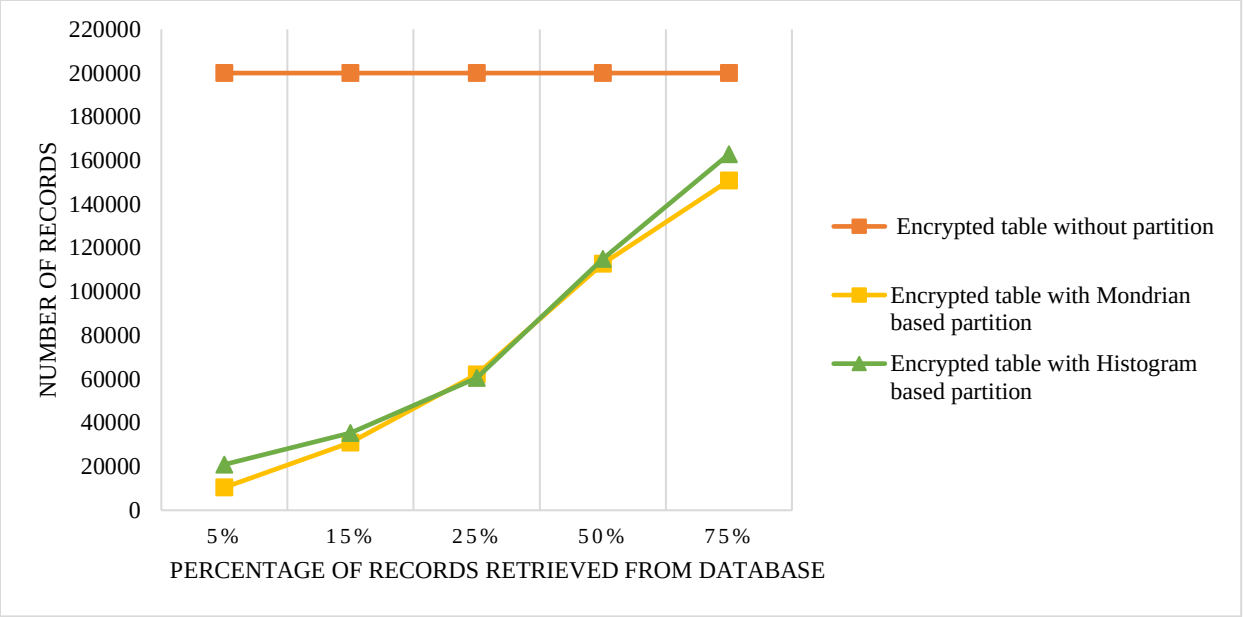

Figure 4.11. Number of records retrieved for each approach (200,000 records with record size 78 bytes)

4.5.4. Data retrieved from a table having 200,000 records with record size 178 bytes

The same simulation was performed to obtain the results by using 200,000 records with record size 178 bytes. The results are listed in Table 4.11 where the time is measured for each partition method in milliseconds, and it summarized the times for each approach.

Percentage of Data Requested	Encrypted table without partition	Encrypted table without partition using a third party (TP)	Mondrian based partition	Histogram based partition	Mondrian based partition using a TP	Histogram based partition using a TP
5%	1476	1896	1321	1378	1757	1937
15%	1545	2070	1519	1537	2034	2159
25%	1613	2139	1562	1587	2092	2286
50%	1720	2273	1672	1694	2219	2350
75%	1840	2431	1776	1812	2287	2555

Table 4.11. Response time in milliseconds for each partition method (200,000 records with record size 178 bytes)

According to the results in Table 4.11, using a trusted third party is less efficient with respect to response time than the cases when a third party is not implemented, and it is because the model uses a server between the client and the Cloud to perform the operations. These results

show the same behavior when using an encrypted table without partition (with a third party) and encrypted table with partitions (using the third party).

Figure 4.12 presents a graph for the response time with and without using a trusted third party between the client site and the Cloud server when using an encrypted table without partition. In this graph, as the percentage of data requested from the server increases, the time it takes to complete the operation grows accordingly.

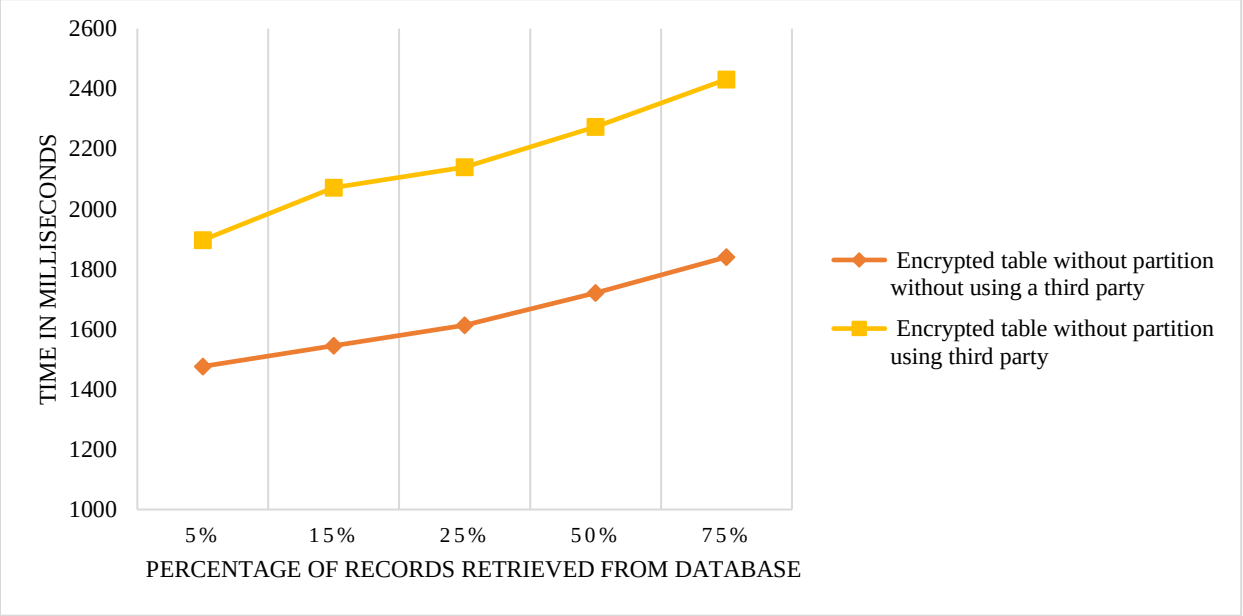

Figure 4.12. Response times for data retrieved from a table having 200,000 records with record size 178 bytes

The results related to the response times for a table having 200,000 records with record size 178 bytes are shown in figure 4.13. As presented in the graph, with and without using a trusted third party, the Mondrian based partition method is more efficient taking into consideration the time required to complete each operation. Using a trusted third party also takes more time to complete the operations as described in previous sections.

When the simulation is performed without a third party, the times are slightly different, and it shows the same behavior, with the Mondrian based partition method being more efficient.

Once the trusted third party is used in the simulation, the separation between Mondrian and Histogram based partition methods is more evident as it shows in figure 4.13. In this last scenario, the Mondrian based partition method continues to be more efficient with respect to time when the results are compared to that of the Histogram based partition method.

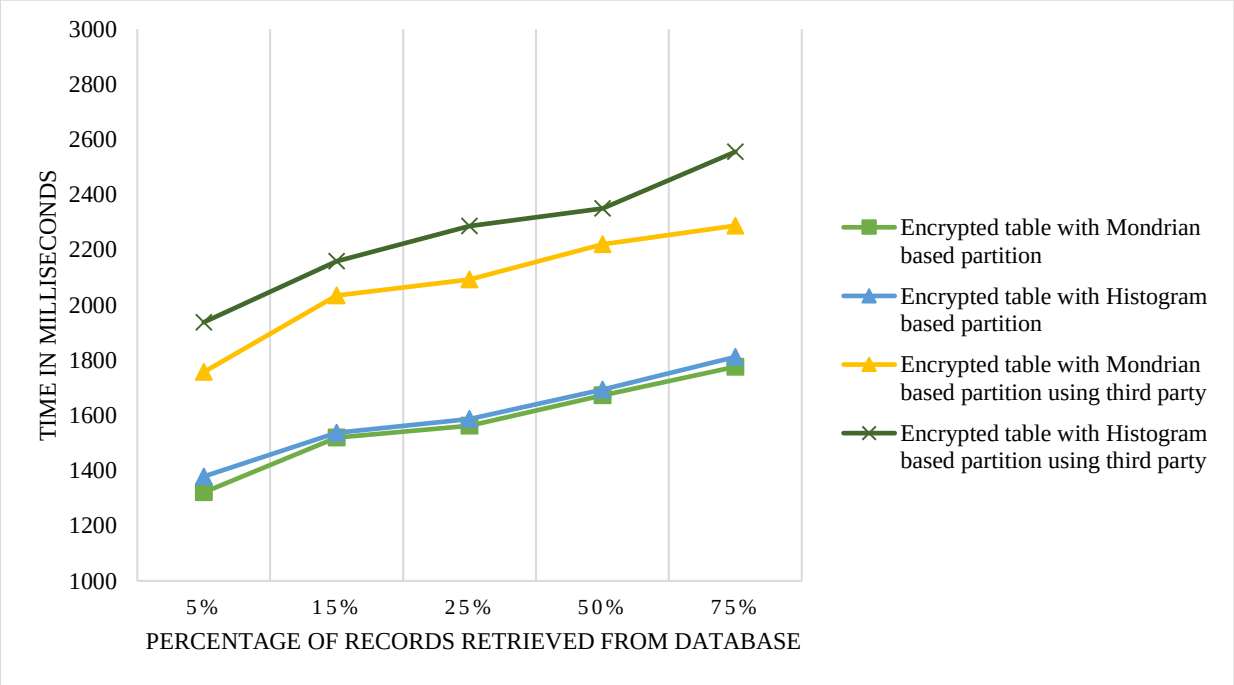

Figure 4.13. Response times for data retrieved from a table having 200,000 records with record size 178 bytes using Mondrian and Histogram based partition

4.5.5. Data retrieved from a table having 500,000 records with record size 78 bytes

The results presented in this part correspond to the tests performed using a table with 500,000 records having record size of 78 bytes. The results in regard to response time appear in Table 4.12, and the table gives the response time for the encrypted table without partition and for the Histogram and Mondrian based partition with and without implementing a trusted third party between the client and the Cloud.

Percentage of Data Requested	Encrypted table without partition	Encrypted table without partition using a third party (TP)	Mondrian based partition	Histogram based partition	Mondrian based partition using a TP	Histogram based partition using a TP
5%	2262	3085	1507	1763	2045	2402
15%	2599	3439	1678	1787	2330	2570
25%	2691	3688	1989	2064	2674	2829
50%	3098	4232	2567	2662	3647	3769
75%	3231	4592	3222	3355	4342	4577

Table 4.12. Response time in milliseconds for each partition method (500,000 records with record size 78 bytes)

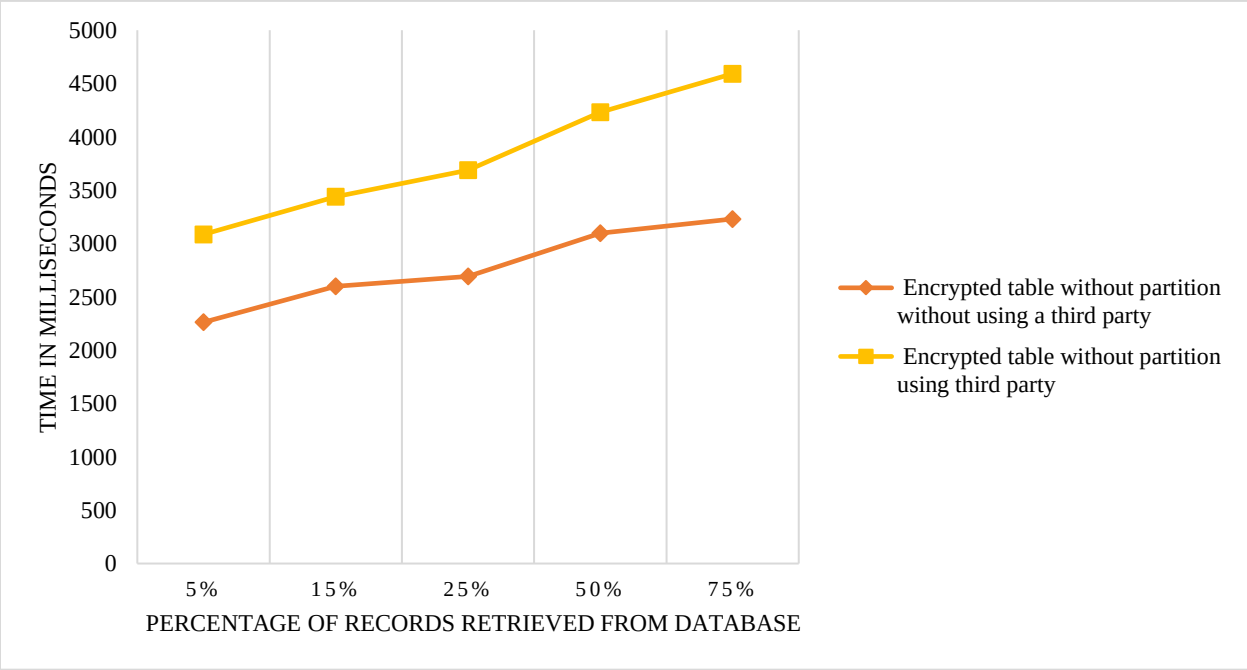

Figure 4.14. Response times for data retrieved from a table having 500,000 records with record size 78 bytes

The results in Table 4.12 are the source data to create the figures 4.14 and 4.15. The graph indicates that the Histogram based partition method is less efficient than the Mondrian based method as is shown in figure 4.14. The results also show the same behavior where the Mondrian based partition method retrieve less number of records taking into consideration the number of records retrieved after each operation.

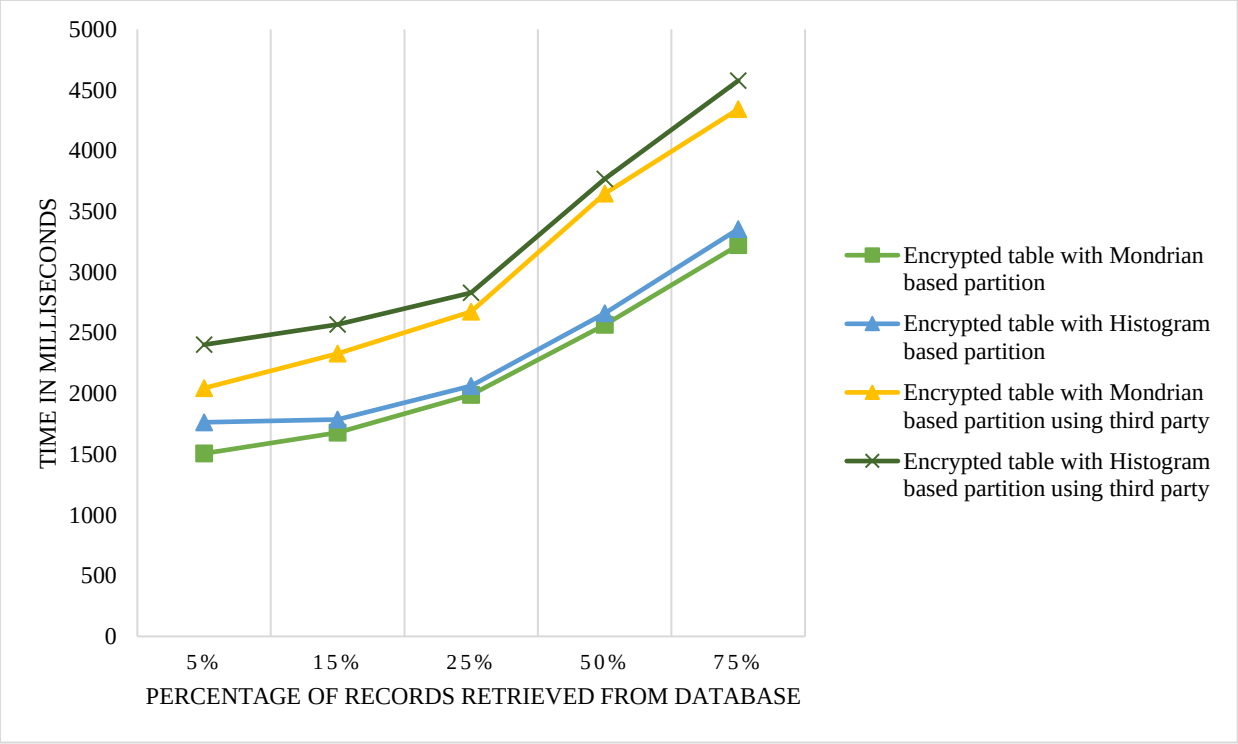

Figure 4.15. Response times for data retrieved from a table having 500,000 records with record size 78 bytes using Mondrian and Histogram based partition

4.5.6. Data retrieved from a table having 500,000 records with record size 178 bytes

To continue evaluating the results of the proposed model in this research, a simulation using a table having 500,000 records with record size 178 bytes was performed, and the results appear in Table 4.13. Figures 4.16 and 4.17 present the final graphs representing the results with respect to response time. Without using a trusted third party, the difference in time indicates that the Mondrian based method takes less time to process the query. When using a third party, the difference in response times with partition methods is greater than the cases when a third party is not used. As in previous cases, the Mondrian partition method is also more efficient regarding the response time. This behavior in the graphs is similar as in the previous sections where when the percentage of required data increases, the time that is taken to complete the operation also increases.

Percentage of Data Requested	Encrypted table without partition	Encrypted table without partition using a third party (TP)	Mondrian based partition	Histogram based partition	Mondrian based partition using a TP	Histogram based partition using a TP
5%	2337	3102	1581	1716	2085	2405
15%	2680	3527	1759	1883	2316	2713
25%	2793	3693	2098	2333	2779	3299
50%	3189	4219	2485	2598	3308	3739
75%	3305	4410	3099	3193	4114	4524

Table 4.13. Response time in milliseconds for each partition method (500,000 records with record size 178 bytes)

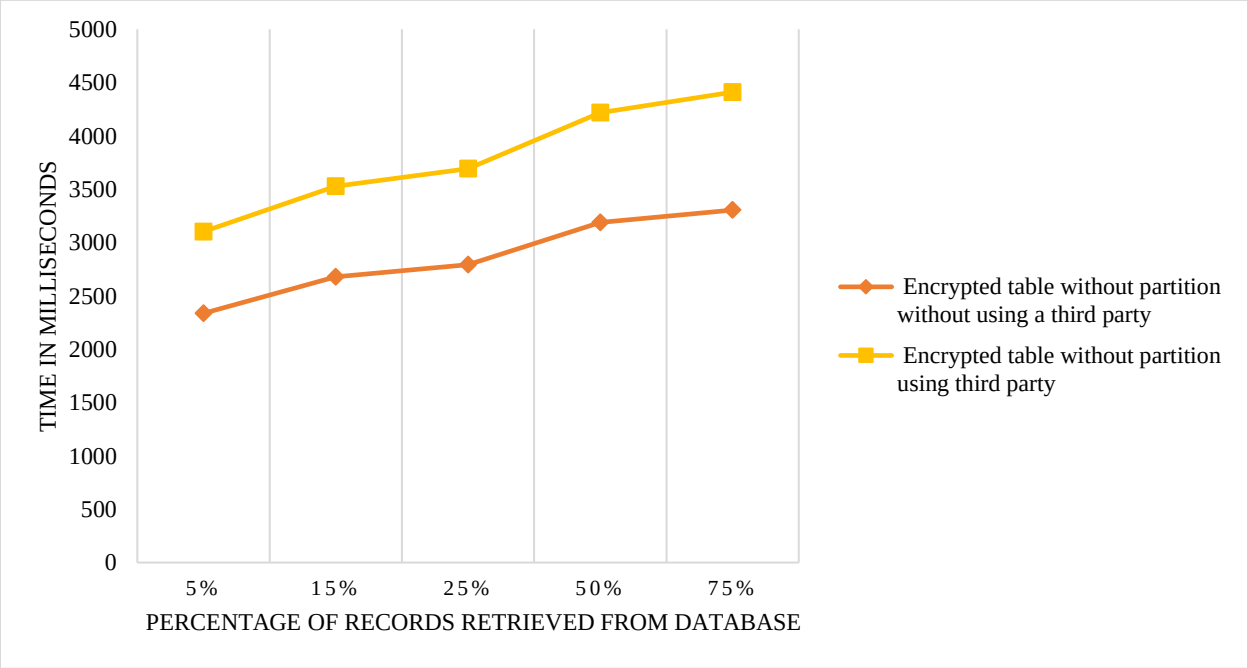

Figure 4.16. Response times for data retrieved from a table having 500,000 records with record size 178

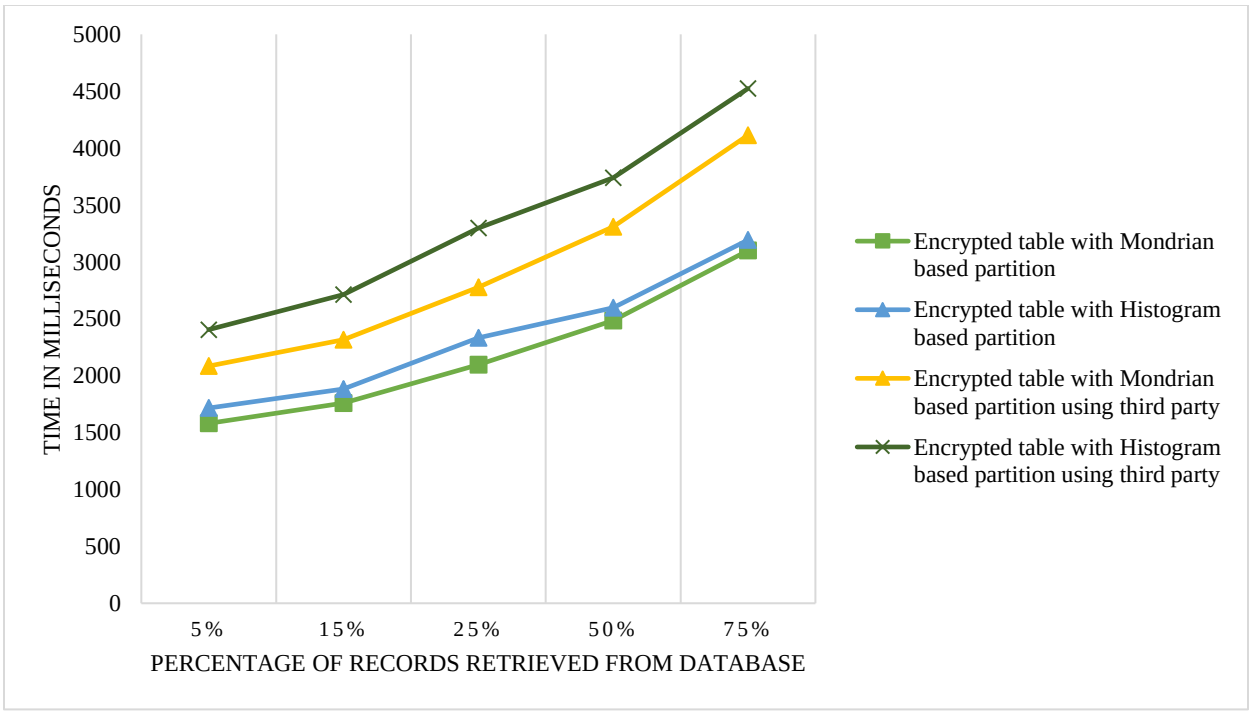

Figure 4.17. Response times for data retrieved from a table having 500,000 records with record size 178 bytes using Mondrian and Histogram based partition

5. Conclusions

In this research, a model that allows enforcing database security in the cloud by using a trusted third party has been proposed. The idea of this design is to reduce the workload at the client site while the trusted third party executes the majority of the activities. This model takes into consideration several advantages of cloud computing services such as storage capabilities and processing in a different location. The communication between the parties is protected by using a cryptographic system. Moreover, by using encryption techniques over the data, the information is secure in the cloud.

Two partitioning methods from the literature review were used to work with an index that allows the model to increase the performance. An attribute from the original table is selected as an index to perform the partition over the data. Each record is encrypted before sending it to the cloud. The combination of the encrypted record and the index increases the security in the cloud because the data stored in the server is hard to understand if it is obtained by an intruder. The index associated with the records in the cloud allows the system to reduce the computing and processing time when a query is run. The trusted third party will send to the client site only the results that satisfy the original query. With this model, the client reduces the time that can be used in performing other activities inside the organization.

The results from the simulations help to conclude that the Mondrian or bisection tree-based partition is more efficient than the Histogram based partition regarding the response time in the retrieval process. Furthermore, the results of this research are similar to the results of the study conducted by Omran who also concluded that the Mondrian technique is more efficient than the Histogram.

The contribution of this investigation is an alternative model to store the databases in with security in the cloud. Even though the complete operation takes more time to be completed, the client site reduces the workload considerably. As future work, this model could be tested using different partitioning methods to compare the results from this research. Also, a web system can be developed to create an application that uses the proposed design with the servers and services placed in a different geographical location.

6. References

[1] S. Khedkar and A. D. Gawande, "Data partitioning technique to improve cloud data storage security," *International Journal of Computer Science and Information Technologies,* vol. 5, no. 3, pp. 3347-3350, 2014.

[2] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology - NIST, Gaithersburg, MD , 2011.

[3] M. Zhou, R. Zhang, W. Xie, W. Qian and A. Zhou, "Security and privacy in Cloud computing: A survey," in *Sixth International Conference on Semantics Knowledge and Grid (SKG)*, Beijing, China, 2010.

[4] S. D. Choubey and M. K. Namdeo, "Study of data security and privacy preserving solutions in cloud computing," in *International Conference on Green Computing and Internet of Things (ICGCIoT)*, Noida, India, 2015.

[5] M. U. Bokhari, Q. M. Shallal and Y. K. Tamandani, "Security and privacy issues in cloud computing," in *3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, 2016 .

[6] A. U and V. S, "A short review on data security and privacy issues in cloud computing," in *IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)*, Bangalore, India, 2016.

[7] M.-G. Avram, "Advantages and challenges of adopting cloud computing from an enterprise perspective," *Procedia Technology* , vol. 12, no. 2014, pp. 529-534 , 2014.

[8] Y. Luo, S. Zhou and J. Guan, "LAYER: A cost-efficient mechanism to support multi-tenant database as aservice in cloud," *The Journal of Systems and Software,* vol. 101, no. 2015, pp. 86-96, 2015.

[9] K. Munir, "Security model for cloud database as a service (DBaaS)," in *International Conference on Cloud Technologies and Applications (CloudTech)*, Marrakech, Morocco, 2015.

[10] C. Curino, E. P. C. Jones, R. A. Popa, N. Malviya, E. Wu, S. Madden, H. Balakrishnan and N. Zeldovich, "Relational Cloud: A Database-as-a-Service for the cloud," in *5th Biennial Conference on Innovative Data Systems Research, CIDR* , Asilomar, California, 2011.

[11] V. Sidorov and W. K. Ng, "Transparent Data Encryption for Data-in-Use and Data-at-Rest in a Cloud-Based Database-as-a-Service Solution," in *IEEE World Congress on Services (SERVICES)*, New York, 2015.

[12] M. G. Xavier, K. J. Matteussi, F. Lorenzo and C. A. F. De Rose, "Understanding performance interference in multi-tenant cloud databases and web applications," in *IEEE International Conference on Big Data*, Washington, DC, 2016.

[13] J.-J. Hwang, H.-K. Chuang, Y.-C. Hsu and C.-H. Wu, "A business model for cloud computing based on a separate encryption and decryption service," in *International Conference on Information Science and Applications (ICISA)*, Jeju Island, South Korea, 2011.

[14] C.-N. Yang and J.-B. Lai, "Protecting data privacy and security for cloud computing based on secret sharing," in *International Symposium on Biometrics and Security Technologies (ISBAST)*, Chengdu, China, 2013.

[15] C. Wang, Q. Wang, K. Ren and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing".

[16] H. Rahmani, E. Sundararajan, . Z. Ali and A. M. Zin, "Encryption as a Service (EaaS) as a Solution for Cryptography in Cloud," *Procedia Technology*, vol. 11, no. 2013, pp. 1202-1210, 2013.

[17] . V. D. Cunsolo, S. Distefano, A. Puliafito and M. Scarpa, "Achieving information security in network computing systems," in *Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, DASC '09.* , Chengdu, China, 2009.

[18] P. Rewagad and Y. Pawar, "Use of Digital Signature with Diffie Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud Computing," in *International Conference onCommunication Systems and Network Technologies (CSNT)*, Gwalior, India, 2013.

[19] N. Khanezaei and Z. M. Hanapi, "A framework based on RSA and AES encryption algorithms for cloud computing services," in *IEEE Conference onSystems, Process and Control (ICSPC)*, Kuala Lumpur, Malaysia, 2014.

[20] M. Kaur and R. Singh, "Implementing Encryption Algorithms to Enhance Data Security of Cloud in Cloud Computing," *International Journal of Computer Applications,* vol. 70, no. 18, pp. 16-21, 2013.

[21] P. Xiong, Y. Chi, S. Zhu, H. J. Moon, C. Pu and H. Hacigumus, "Intelligent management of virtualized resources for database systems in cloud environment," in *IEEE 27th International Conference on Data Engineering (ICDE)*, Hannover, Germany, 2011.

[22] L. Zhao, S. Sakr, A. Fekete, H. Wada and A. Liu, "Application-Managed Database Replication on Virtualized Cloud Environments," in *IEEE 28th International Conference on Data Engineering Workshops (ICDEW)*, Arlington, Virginia, 2012.

[23] N. Khanghahi and R. Ravanmehr, "Cloud Computing Performance Evaluation: Issues and Challenges," *International Journal on Cloud Computing: Services and Architecture (IJCCSA),* vol. 3, no. 5, pp. 29-41, 2013.

[24] H. Hacigumus, B. Iyer, C. Li and S. Mehrotra, "Executing SQL over encrypted data in the database-service-provider model," in *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data (SIGMOD '02)*, Madison, Wisconsin, 2002.

[25] O. B. Omran and B. Panda, "Facilitating Secure Query Processing on Encrypted Databases on the Cloud," in *Proceedings of the IEEE International Conference on Smart Cloud 2016 (SmartCloud 2016)*, New York, 2016.

[26] O. B. Omran and B. Panda, "Efficiently Managing Encrypted Data in Cloud Databases," in *Proceedings of the 2nd IEEE International Conference on Cyber Security and Cloud Computing (CSCloud 2015)*, New York, USA, 2015.