# Towards a Hybrid Data Partitioning Technique for Secure Data Outsourcing

Sultan Badran
*Faculaty of Graduate Studies*
*Palestine Polytechnic University (PPU)*
Hebron, Palestaine
176030@ppu.edu.ps

Nabil Arman
*Dept. of Computer Science and IT*
*Palestine Polytechnic University (PPU)*
Hebron, Palestaine
narman@ppu.edu

Mousa Farajallah
*Dept. of Computer Engineering*
*Palestine Polytechnic University (PPU)*
Hebron, Palestaine
mousa_math@ppu.edu

***Abstract*— In light of the progress achieved by the technology sector in the areas of internet speed and cloud services development, and in addition to other advantages provided by the cloud such as reliability and easy access from anywhere and anytime, most data owners find an opportunity to take advantage of the cloud to store data. However, data owners find a challenge that was and is still facing them in the field of outsourcing, which is protecting sensitive data from leakage. Researchers found that partitioning data into partitions, based on data sensitivity, can be used to protect data from leakage and to increase performance by storing the partition, which contains sensitive data in an encrypted form. In this paper, we review the methods used in designing partitions and dividing data approaches. A hybrid data partitioning approach is proposed to improve these techniques. We consider the frequency attack types used to guess the sensitive data and the most important properties that must be available in order for the encryption to be strong against frequency attacks.***

***Keywords*— hybrid data partitioning, data outsourcing, data sensitivity, cloud.***

## I. INTRODUCTION

Data outsourcing is vulnerable to different types of attacks. In fact, secure and efficient retrieval of outsourced data is still an open challenge. Data sensitivity is one of the most important security issues that need to be investigated.

In the past, the data owner avoided data outsourcing (sensitive and non-sensitive data), or encrypt all data stored in cloud to protect the sensitive data. However, to improve data security against attacks, the data partitioning techniques are proposed [1] [2] [3]. These techniques divide a relation into a set of relations based on data sensitivity.

One of the major concerns of data owners today is the leakage of data to external parties, mainly the sensitive data. On another hand, the data owners need to outsource the data in a certain situation to benefit from cloud features such as accessing the services or data from anywhere/anytime, the user pay only for used services or data, increases data reliability, supports parallel and distributed computing.

The challenges in security in outsourcing data lead the researcher to find solutions to improve the security. One of these solutions is partitioning data, where the data in a relation is divided into smaller relations depending on selective attributes or certain tuple values such as sensitive data. Then store the new relation in different data centers that may be in different locations.

Generally, to get the partitioning categories there are many different data partitioning approaches such as Data Sensitivity Partition, Frequency of Use Based Partition, and Space-Based Partition.

This research is motivated by the work to develop information systems solutions such as health record systems, which have high sensitive data. In [2] data outsourcing was used in several application contexts. Data partitioning can be used to partition data into sensitive and non-sensitive relations [2].

This paper discusses the different types of partitioning techniques and discuss different approaches that have been proposed by researchers to get the partitioning categories. It highlights how researchers used those partitioning techniques to increase the security and protecting sensitive data. A hybrid data partitioning approach is proposed to improve these techniques in terms of security and performance.

The remaining sections in this paper are organized as follows: Section II covers the literature review. Section III presents the data partitioning types. The data partitioning security are covered in Section IV. In section V we explain the query inference attacks. The hybrid data partitioning technique is presented in section VI. And in section VII, we conclude the paper.

## II. LITERATURE REVIEW

Because of the importance of data security, especially in these days, most companies and organizations are turning to cloud solutions. However, the companies and organizations are concerned about their data, stored in cloud, from leakage and loss of privacy. Therefore, researchers have developed different techniques to improve the security of the database and the performance of the queries. In the literature, the security of sensitive data was achieved using data partitioning techniques.

Several researchers have us encryption to secure data such as Searchable Encryption in [4]. Recently, several data partitioning techniques use encryption for secure data in databases queries and avoid data leakage. Data partitioning was used to improve the data security and the performance in [1], [2], [3], [5], [6] and [7].

The authors in [1], [2] and [3] have used data partitioning (vertically and horizontally) based on data sensitivity for classifying tuples into sensitive and non-sensitivity tuples and attributes. The proposed solution assume that the entire database may not contain sensitive tuples, and non-sensitive tuples can be exploited to handle limitations of encryption-based approaches. They proposed a new definition to secure data based on data sensitivity and to improve the security against inference attacks such as frequency-count and workload-skew attacks by proposing an approach called “query binning (QB)” that can process joint queries through the sensitive and non-sensitive tuples. The data partitioning and QB, in addition to enhancing the performance, it improves the security and prevent data leakage. However, the used technique are unable to use more than one value for criteria in queries.

The authors in [5] have proposed a symmetric encryption technique to protect the data privacy during privacy-

preserving data mining (PPDM). The technique is implemented using vertically partitioned relation based on data sensitivity. The technique used different encryption algorithms to encrypt the sensitive attributes in different relations at the same time. The algorithms were applied to three data sets that were prepared as MS Excel files with different sizes. In addition, four symmetric encryption techniques were implemented using sensitive attributes only: AES, DES, Rijndael, and RC2. The privacy results of the proposed techniques were better than using just one encryption algorithm on each partitioned relation.

In [6], the authors presented an algorithm to prevent the leakage of sensitive data or loss of privacy from the relations in a database stored in cloud. They have developed a model with a view to offer secure data management capability in cloud databases. That model used two approaches to partitioning data: the first is using attributes relationship to divide database relations, and the second is using the data sensitivity to divide the relations. They proposed a model to deal with the results of partitioning model, which is used to store partitioned relations in data center. The model can be stored in one cloud data center, which provides good performance and security, or store the partitioned relations in different cloud data centers locations that improve the security. They distribute data into different data centers in different locations to improve the security and do not consider the performance.

In [7], the authors presented Data Partitioning Methods to Process Queries on Encrypted Databases on the Cloud. They explored a technique to improve the query processing performance and at the same time keep the database relations secure on a Cloud by encrypting relations from any leakage or attack. They point to the ability to protect data from any leakage or attack by designing encrypted databases that process the SQL queries on encrypted relations. The main idea in this solution is handling the query on encrypted data stored on cloud without a need to decrypt it. The result of query is decrypted at the client side. In addition, they have developed four different techniques to index and partition the data as follows:

- Frequency of use Based Partition.
- Space Based Partition.
- Mondrian or Bisection Tree Based Partition.
- Histogram Based Partition.

They compared the efficiency of the first three techniques with the Histogram-Based partition. The indexes and partitions are used to process the query and select part of the data from the Cloud. The indexes' data can be stored on the Cloud or on-premises server with the encrypted database relation. This leads to a reduction in the overall processing time, which contains the data transfer time from the cloud to the query requester site as well as the data decoding and processing time for the requester. In addition, these techniques are used to compare encrypted relation and the unencrypted relation. The comparison results contain the running time and retrieved different number of tuples in the relation with different tuple sizes. They found that "the encrypted relation with Frequency-of-Use-Based partition and the encrypted relation with Bisection-Tree-Based partition are the most efficient kinds of partitions". In addition, they explain a particular issue which show how combining Frequency-of-Use-Based and Bisection-Tree-Based to enhance the performance by methods data partitioning.

In addition to Data Ppartitioning techniques, researchers have explored Inference Attacks to understand how adversary attack the encrypted databases and how the leakage of sensitive information occurred [8].

The authors of [8] have studied the inference attacks against encrypted database (EDB) systems based on property-preserving encryption (PPE) scheme which is CryptDB. They presented a series of attacks that discover the plaintext from ciphertext encrypted using deterministic encryption (DTE) and order preserving encryption (OPE) encrypted database attributes. They consider four different well-known attacks such as frequency analysis and sorting. They did the experiments in electronic medical records (EMR) to evaluate these attacks. The EMR data belong to real patients from 200 U.S. hospitals. To perform the attacks experiments, they assume that the adversary has access to EDB in steady state. In addition, the adversary has the access to some auxiliary information about the system and/or the data such as application details, public statistics and prior versions. They simulate the EMR-EDB to make the experiment scenario to try the considered attacks, then they analyze the results. Their experimental results illustrate that the sensitive information can be recovered when the adversary has a background knowledge about EDB data and properties. In numbers, their attacks recovered more than 80% of the patient records out of 95% of the hospitals when using OPE-encryption to encrypt attributes (e.g., age and disease severity), in other hand when using DTE in certain encrypted attributes (e.g., sex, race, and mortality risk). The attacks recovered more than 60% of the patient records out of 60% of the hospitals.

## III. Data Partitioning

Nowadays, the size of data grows dramatically in many large-scale solutions, so the data is divided into partitions that can be managed and accessed separately [9]. Partitioning can be used for many purposes: it can be used to improve scalability, improve security, and optimize performance [2] [9] [10]. The cheap data storage is usually used to partition and archive the older data [9].

However, the partitioning techniques must be chosen carefully to maximize the benefits and to minimize adverse effects [9].

### *A. Why Using Data Partitioning?*

- Improve scalability. The single database system has a limitation when considering the used hardware resources. On the other hand, using the data partitioning techniques lead to optimal resources use, because duplication of hardware resources, the single database is divided and distributed over more than one data center/location.
- Improve performance. Using data partitioning makes the selection of query transactions time smaller. Data partitioning can make the query processing more efficient. Query transactions that affect more than one partition are executed in parallel.
- Improve security. Data partitioning is used to increase the security such as divide the data into different databases or locations based on sensitivity and apply different security policy to the sensitive data.

- Improve availability. Partitioning data across multiple data centers avoids a single point of failure. If one center failed, only the data in that center is unavailable. Transactions on other data centers can operate.

### B. *How to design data partitions?*

There are three typical partitioning techniques for data: horizontal partitioning, vertical partitioning, and functional partitioning [2] [9] [10]. The selected technique of partitioning depends on two factors: the first is the purpose of partitioning, and the second is what the owner wants to improve in the system (security, performance, or both).

- Horizontal partitioning technique: in this technique, the original relation and all partitions have the same relation schema. Each partition contains a specific subset of the tuples [2] [9]. Fig. 1 shows horizontal partitioning. In this example, Staff member's data is divided into two relations based on the department value. Each relation holds the tuples for a contiguous range of shard values (IT with HR in one relation and second relation include marketing department tuples), organized alphabetically by ID attribute.

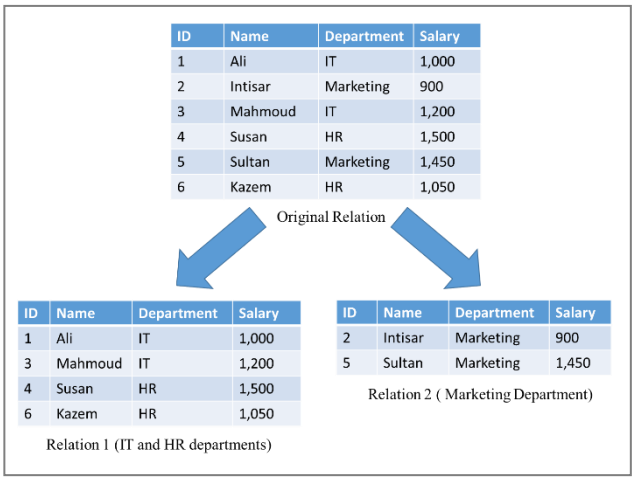


Fig. 1. Horizontally partitioning data

- Vertical partitioning technique: in this technique, each partition holds a subset of relation attributes. The attributes are divided according to their pattern of use or property. For example, Sensitive attributes are placed in a relation and non-sensitive attributes in another relation [2] [5]. Fig. 2 shows an example of vertical partitioning. In this example, different attributes are stored in different relations. One relation holds tuples that is sensitive data, including ID, and salary. Another relation holds non-sensitive data (ID, Name and Department) [9]. In the above example the ID belong to the divided relations to rejoin the relations to return back to original relation.

- Functional partitioning: in this technique, data is divided to different relations according to context of system. For example, in an HR system the Staff member's data is stored in one relation and payroll data in another. Fig. 3 shows how the data is partitioned into two relations, where each partition is stored in different locations.

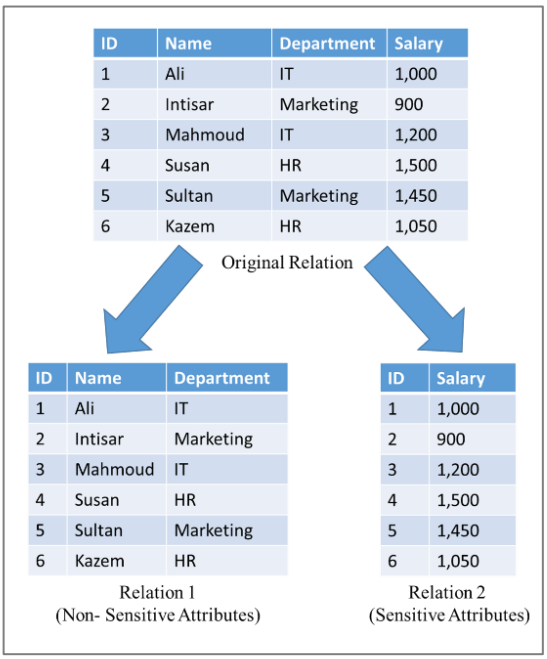


Fig. 2. Vertically partitioning data

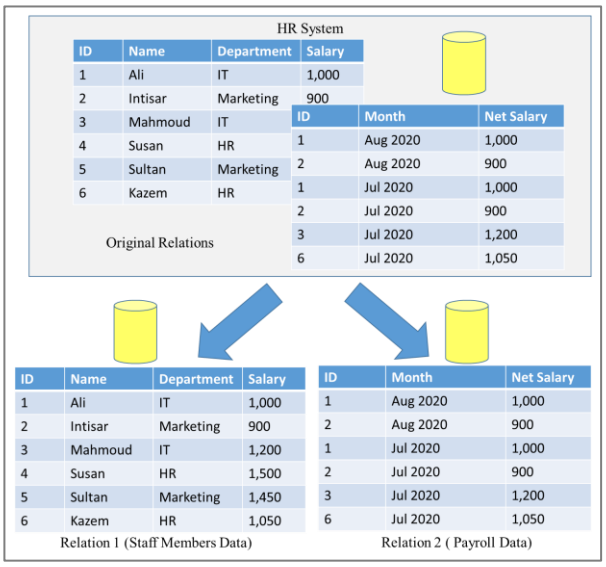


Fig. 3. Functionally partitioning data

These techniques can be combined, and it is recommended that consider them all when you design a partitioning scheme. For example, you might divide data into horizontal or functional partitioning and then use vertical partitioning to further subdivided of data in each horizontal or functional partitioning.

### C. *Type of Partitioning*

There are many approaches to perform data partitioning:

- Data Sensitivity Partition

  In this approach, the partitioned data is divide into a set of relations based on the data sensitivity. For example, if a relation includes a sensitive attribute such as PIN code to access account profile, a new relation is created for the PIN codes attribute and another relation for the rest of the attributes. Keeping a link between tuples in the new relations created is needed to perform a joining operation to retrieve the original relation [2] [3] [6].

- Frequency of use Based Partition

  In this partitioning approach, the transaction log file, and specifically the WHERE clause conditions in each SQL SELECT statement is monitored, then a statistical matrix for selected attributes used in the WHERE clause conditions is produced. This matrix includes the frequency of use of the attribute values in the WHERE clause such as (WHERE SALARY Between 1000 and 2000). After the matrix is created, some cleaning to exclude the smaller frequency counts is performed, then a partition is created for each frequency in the final matrix [7].

- Space Based Partition

  This partitioning approach starts by creating a statistical table that includes the frequency for each value in a specific attribute, and then continues with dividing data according to bucket size. The partitioned relations created are based on comparing the value frequency against the size of the bucket. For example, if the frequency of first value in the statistical table equals the bucket size, a new partition is created and it contains all tuples that belong to the first value. If the sum of frequency of the second and third values are equal, a new partition is created and it contains all tuples that belong to the first and third values. Generally, the new partitions can contain tuples from one or more values, and tuples may belong to values distributed to one or more partitions based on the frequency size and bucket size [7].

- Mondrian or Bisection Tree Based Partition

  In this approach, an attribute is selected to be used to partition data, then the tuples are ordered by the selected attribute. After that, the median is calculated and the two relations are created. The first relation includes the tuples in the right median and the second relation includes the tuples in the left median. The approach is repeated for each new relation until the partition satisfies certain conditions [7].

- Histogram Based Partition

  This approach is used to display statistical information. An Equi-width technique is one of its types. This technique divides the values into stack of equal width. Each stack presents a new relation. This method is simply subtracting the minimum value from the maximum value for the attribute and is used to divide the results by number of stacks [7].

## IV. Data Partitioning Security

The data security level should provide non-link-ability and Ciphertext indistinguishability to protect data [2] [11]. Fig. 4 shows the two properties that needed to be involved in the encryption algorithms:

- Non-link-ability: the adversary does not learn the relationship between any encrypted and plaintext value.
- Ciphertext indistinguishability: the adversary does not learn any relationship between encrypted values.

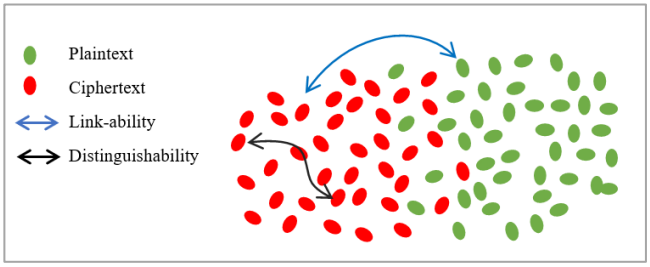


Fig. 4. Data Partitioning Security

## V. Query Inference Attacks

The adversary uses query Inference attacks to recover information about the data or queries by linking encrypted information with openly-available information [8]. The most well-known examples of an inference attack:

- Frequency attack, i.e., an adversary can conclude how many tuples have an similar value [8].
- The workload-skew attack, i.e., an adversary, having the knowledge of frequent selection queries by watching many queries, can guess which encrypted tuples potentially meet the frequent section in selection queries [2].

## VI. Hybrid Data Partitioning Technique

In our research, a hybrid data partitioning approach is proposed to improve the used techniques of data partitioning in terms of security and performance. We use both horizontal and vertical partitioning design techniques. We use the relations depending on attributes' relationship model to partition data. This model partitions data based on data sensitivity. We use the approach of tuples link to improve the security and to make an ambiguous environment for the adversary. The approach of tuples link works to link two sensitive tuples with two non-sensitive tuples, and a query processing retrieves the four tuples even if only one tuple meets the WHERE clause in the SQL SELECT statement. The proposed approach has the benefits of saving time and resources needed for cryptographic operations on non-sensitive data. In addition to that, duo to no sensitive data in plaintext format and metadata belonging to sensitive data stored in cloud, our approach improve protection of the sensitive data. Finally, using the tuples link approach the sensitive data are protected against inference attacks

Fig. 5 illustrates the context of our proposed approach that considers two entities: the first entity is the trusted database that contains the whole data in plaintext format. The second entity is the untrusted database (public cloud) which contains the partitioned relations. There is a trusted connection between the two databases. The first thing that should be done before partitioning data is to decide which attribute should not be subject for partitioning process. For example, there is no need to store passwords attributes in cloud, so no need to use these attributes in the partitioning process. Then the second step using the hybrid data partitioning approach starts with selecting those attributes that are subject for vertical partitioning if the whole attributes' values are sensitive and store them in encrypted format as shown in Fig. 5 relation 1. The remaining attributes are subject to horizontal partitioning. The non-sensitive tuples are stored in plaintext format as

shown in Fig. 5 relation 2, and sensitive tuples are stored in encrypted format as shown in Fig. 5 relation 3. The trusted database, as mentioned above, contains the whole data. In addition, the trusted DB receives select query requests from untrusted DB and executes it. The returned data is only the primary keys of tuples that meet the WHERE clause condition. The results return back to untrusted DB. In the untrusted DB, a joining query between partitioned relations and the returned results from trusted DB using primary keys is executed.

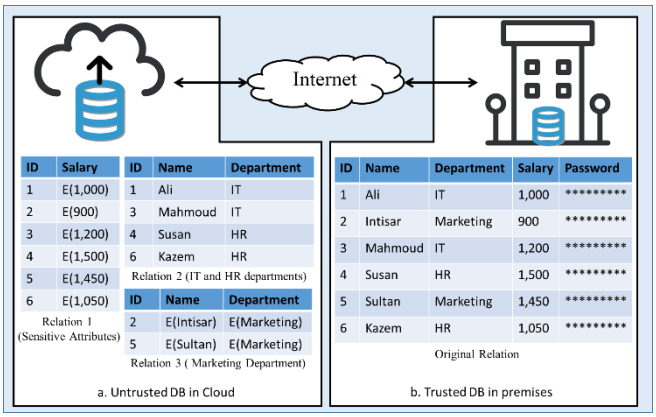

Fig. 5 The context of the proposed approach

## VII. CONCLUSION

The data leakage is the most challenge for data owners when they decide to store their data in outsourcing location. In this paper, we present that the data partitioning as a solution to protect data from leakage.

We review some researcher's work in the partitioning data; we discuss some of the benefits of partitioning data. In addition, we discuss some of data partitioning techniques for design partitions and methods used for partition data. In other hand, we mentioned what are the frequency attack types and most important properties that must be available in order for the encryption to be strong against frequency analysis attacks.

After make the literature review for the researcher's work in this paper. We find that the data partitioning is suitable solution for protect the outsourced data from leakage.

## REFERENCES


[1] S. Mehrotra, Y. O. Kerim and S. Shantanu, "Exploiting Data Sensitivity on Partitioned Data," From Database to Cyber Security, pp. 274-299, 2018.

[2] S. Mehrotra, S. Sharma, J. D. Ullman and A. Mishra, "Partitioned Data Security on Outsourced Sensitive and Non-Sensitive Data," 2019 IEEE 35th International Conference on Data Engineering (ICDE), pp. 650-661, 2019.

[3] S. MEHROTRA, S. SHARMA, J. D. ULLMAN, D. GHOSH and P. GUPTA, "PANDA: Partitioned Data Security on Outsourced Sensitive and Non-sensitive Data," ACM Transactions on Management Information Systems, 05 2020.

[4] M. A. Abdelraheem, T. Andersson, C. Gehrmann and C. Glackin, "Practical Attacks on Relational Databases Protected via Searchable Encryption," International Conference on Information Security, pp. 171-191, 9 September 2018.

[5] D. Vashi, H. B. Bhadka, K. Patel and S. Garg, "Implementation of Attribute Based Symmetric Encryption through Vertically Partitioned Data in PPDM," International Journal of Engineering and Advanced Technology (IJEAT), vol. 9, no. 1, pp. 868-874, 2019.

[6] O. M. Ben Omran and B. Panda, "A Data Partition Based Model to Enforce Security in Cloud Database," Journal of Internet Technology and Secured Transaction, vol. 3, no. 3, pp. 311-319, 09 2014.

[7] O. Omran and B. Panda, "Data Partitioning Methods to Process Queries on Encrypted Databases on the Cloud," University of Arkansas, Fayetteville, 2016.

[8] M. Naveed, S. Kamara and C. V. Wright, "Inference attacks on property-preserving encrypted databases," in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Machinery, New York, NY, USA, 2015.

[9] "Horizontal, vertical, and functional data partitioning," Microsoft, 11 04 2018. [Online]. Available: https://docs.microsoft.com/en-us/azure/architecture/best-practices/data-partitioning. [Accessed 20 02 2020].

[10] J. P. Meeta and P. B. Mansi, "Overview of Horizontal Partitioning and Vertical Partitioning," in National Conference on "Computer Science & Security" (COCSS-2013), SVIT, Vasad, 2013.

[11] W. L. e. al., "Towards Secure and Efficient Equality Conjunction Search over Outsourced Databases," in IEEE Transactions on Cloud Computing, Early Access, 2020.

[12] T. Peng, C. Xiang, S. Sen, C. Rui and S. Huaxi, "Differentially Private Vertically Partitioned Data Publishing," IEEE Transactions on Dependable and Secure Computing, 2019.

[13] M. A. Panhwar, S. A. Khuhro, G. Panhwar and K. A. Memon, "SACA: A Study of Symmetric and Asymmetric Cryptographic Algorithms," INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND NETWORK SECURITY, vol. 19, no. 1, pp. 48-55, 2019.

[14] B. Maram, J. M. Gnanasekar, G. Manogaran and M. Balaanand, "Intelligent security algorithm for UNICODE data privacy and security in IOT," Service Oriented Computing and Applications, 2018.

[15] J. M. Gnanasekar, "UNICODE Text Security Using Dynamic and Key-Dependent 16X16 S-box," Service Oriented Computing and Applications, 2016.

[16] P. Dhulavvagol, V. Bhajantri and S. Totad, "Performance Analysis of Distributed Processing System using Shard Selection Techniques on Elasticsearch," Procedia Computer Science, vol. 167, pp. 1626-1635, 2020.

[17] V. Ciriani, S. Vimercati, S. Foresti, S. Jajodia, S. Paraboschi and P. Samarati, "Fragmentation Design for Efficient Query Execution over Sensitive Distributed Databases," 2009 29th IEEE International Conference on Distributed Computing Systems, pp. 32-39, 2009.