# Fuzzy-based Encryption Mechanism for Privacy Preservation and Evaluation of Query Response Time in Cloud

Vishnupriya.R
*Department of Computer Science and Engineering*
*Karpagam Academy of Higher Education*
Coimbatore

Manikodi.K
*Department of Computer Science and Engineering*
*Karpagam Academy of Higher Education*
Coimbatore

Annish Kumar.A
*Department of Computer Science and Engineering*
*Karpagam Academy of Higher Education*
Coimbatore

L.Selvam
*Department of Computer Science and Engineering*
*Karpagam Academy of Higher Education*
Coimbatore
umaselvam_35@yahoo.com

N.Mekala
*Department of Computer Science and Engineering*
*Karpagam Academy of Higher Education*
Coimabatore

2024 International Conference on Expert Clouds and Applications (ICOECA) | 979-8-3503-8579-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICOECA62351.2024.00034

***Abstract*— Data is considered as a resource, where security is a critical factor in the potential growth of data mining. The protection offered by public infrastructure is insufficient to ensure the privacy of personal data. Edge devices are used by analysts to obtain data. Data accessible by unauthorized parties and privacy leaks at the edge layer are prevalent issues found in earlier study. To tackle the problem, this study introduces fuzzy mechanism, which provides differential privacy with Laplace mechanism. This is where operations like query processing and data processing are performed. The data owner encrypts the dataset by using encryption and adds noise after sending it to the data provider and then moves it to the cloud. Based on the uploaded dataset, the data owner employs the encryption technique to generate an index from the recovered key properties. Conversely, the data supplier authenticates the requests and queries made by the data analyst. After that, the edge server's hash tree looks for the matching data, retrieves it from the cloud, and sends it in an encrypted format to the data analyst. A decryption key is given to each verified data analyst to access the query result. MATLAB 2020a is used to perform this task, and the outcomes demonstrate improved throughput, response time and encryption time.**

***Keywords*— Cloud, Privacy, Query, Server, Encryption, Fuzzy**

## I. Introduction

Since most of the well-known companies and organizations have been negatively impacted by regular data breaches, it is more important to secure an individual's privacy in the modern industrial era [1]. Traditional methods of data security, including cryptography, would protect the data. By retaining the individual's personal data, data privacy is ensured via the assured standard method known as Differential Privacy (DP) [2]. These days, information is gathered from several sources. The Internet of Things (IoT) is also a part of it. As a result, the data may be relevant to applications such as smart grid, automobile communication, smart home, and healthcare [3]. Privacy is a must for all real-time applications. Differential attacks are a concern for the personal information gathered from individuals in these applications. Generally, there are two types of DP: local Differential Privacy (LDP) and centralized Differential Privacy (CDP). When using CDP, the collected information is kept on file with a reliable party that perform data processing before giving analysts access to the processed data [4]. A privacy parameter in DP is essential to figure out the security of the data. It is ensured that the qualities of DP will completely secure the personal data. DP is used for the following purposes:

1) By using less complex calculations, it offers data privacy and facilitates the building of blocks of privacy.
2) Analysts must know the details of the database, where the result was uploaded in order to extract the obtained result during post-processing.
3) Effective management is used to balance accuracy and data privacy.

By introducing noise into the uploaded data, DP allows for the privacy of a dataset. Greater noise tends to increase privacy but it also decreases the precision [5]. Thus, in order to strike a balance between precision and privacy, the noise must be added up in the right ratio. DP is the result of combining two machine learning algorithms: Support Vector Machine (SVM) and deep learning. [6]. The most popular approach for adding privacy to a dataset is Laplace random noise. The use of various algorithms has an impact on data privacy. Similarly, the query results ought not to be affected by the inclusion or deletion of a small portion of the dataset [7]. Meanwhile, because of DP's performance, attackers find it more difficult to forecast the sensitive properties included in the dataset. In order to handle every security concern from the query through the database result, the DP process becomes crucial [8]-[10].

979-8-3503-8579-3/24/$31.00 ©2024 IEEE
DOI 10.1109/ICOECA62351.2024.00034

This work presents the efficient addition of noise to differential privacy by using a combination of fuzzy techniques. This approach feeds fuzzy membership functions, which are then combined with fuzzy logic. Since the things near the periphery are not reliable, their only responsibility is to relay requests and search for the results related to specific queries. In order to preserve search security, this work suggests edge, which can search and retrieve relevant data from the cloud, which is encrypted using lightweight encryption technique. As a result, an entirely sophisticated security system is created. The subsequent subsections illustrate the structure of this research report. The following is a summary of the suggested research work's main contribution.

- ✓ The procedure involves uploading the dataset owned by the data owner and responding analyst queries while improving accuracy and protecting privacy.
- ✓ By employing the Laplace approach, the parameter is successfully secured using fuzzy mechanism. To ensure that it works with the datasets of various sizes, the dataset's characteristics and sensitivity are examined.
- ✓ Since public clouds are unreliable, data is uploaded to them after being encrypted using an optimal lightweight algorithm.
- ✓ Untrusted edge devices are configured to search analyst queries to ensure that they are not aware of the data. The search outcomes are downloaded in an encrypted fashion from the cloud.
- ✓ Before the analyst is provided with a decryption key, they must authenticate themselves using the unique credentials.
- ✓ The proposed fuzzy-based encryption mechanism is used to ensure that fewer resources are used and that the system operates more quickly. An efficient way to show the use of an untrusted edge device and the public cloud is to look up the hash values and store the information in a secure manner.

The work is organized as: section 2 gives detailed explanation on prevailing approaches. The methodology is briefed in section 3 with outcomes in section 4. The summary is provided in section 5

## II. Related works

In the conventional paradigm of cloud computing, edge devices often function as consumers. However, with the growing potential of cloud computing and the Internet of Things (IoT), the edge network's data volume is growing quickly [11]. The shift from being consumers of data to providers of it has aided in the expansion of edge network features and services. Cloud and edge computing services are combined by using Location-Based Services (LBS) [12] to guarantee the effectiveness and real-time nature of online commerce. Though these applications provide easy services to consumers, they also encounter several data security issues [13].

In recent years, academia has developed a number of alternatives, such as encryption technologies [14] and dynamic pseudonym schemes [15]. Nevertheless, the majority of these solutions are focused on cloud services [16], and the edge environment is not well researched. Furthermore, the potential of location privacy breaches is increased once more with the emergence of Mobile Crowd-Sensing (MCS) networks.

MCS has been extensively researched and developed recently as a significant sensor network in the Internet of Things. If participants in a group sensing task use their personal smartphones, the job requester's (server's) submitted sensor data may vary [17]. This implies that in order to deduce the location relationship network, by obtaining multi-dimensional participant data, a hostile attacker might be able to influence the relationship between activities and determine the position correlation weight between the individuals. Since the user is nearer the edge server, the attacker can choose the users to include in the sensing work based on the prior knowledge if the job allocation is done via the edge server**.** This allows the attacker to gather pertinent data, create a location-based friendship network, and determine the participants' exact locations [18]. The edge-centric sensing task differs from the server-centric design of the cloud in that it has a smaller geographic scope and makes it simpler to conduct targeted attacks based on the past information. If the attacker creates a location relationship network instead of focusing on a task with a broad geographic scope, the inferred location data will be more accurate.

## III. Methodology

This section presents the suggested fuzzy-based encryption method along with a thorough fix for handling the differential privacy challenges. The proposed methodology is explained in three sub-sections: Query Processing, Data Processing, and System Design.

The proposed system comprises several key entities, each serving a distinct role in facilitating the secure and efficient exchange of data. These entities include the data owner, data provider, edge server, cloud, and data analyst. The definitions and responsibilities of each entity within the system are elaborated as follows:

Data Owner: The data owner is responsible for uploading datasets to the system. The entire dataset, along with the corresponding index that data analysts use to search for information, is uploaded by the data owner. It is presumed that the data owner is a trusted entity in the system.

Data Provider: The data provider is a trusted component of the system tasked with adding noise, encrypting data, hashing index values, and verifying the identity of data analysts.

Edge Server: The edge server acts as an intermediary between the cloud and data provider, facilitating data access for data analyst queries. It retrieves files from the cloud, performs index lookups using hash trees, and forwards queries to the data provider.

Cloud: The cloud serves as the public database in the system, storing the data received from data providers and distributing it to data analysts through edge servers.

Data Analyst: Data analysts, also referred to as data miners, send queries to the edge server. In response, the edge server provides decryption keys and query results to the data analyst.

The proposed system involves two main processes: one for the data owner and another for the data analyst. Initially, the data owner uploads datasets to the data provider, which then encrypts the data using lightweight encryption methods and adds noise using the Laplace technique. The encrypted data is subsequently stored in the public cloud. Meanwhile, index terms provided by the data owner are hashed in the edge server using a hashing function. The data source, serving as the only trusted entity in the system, is responsible for authenticating data analysts and ensuring privacy protection throughout the data exchange process.

### a) Data analysis based on query

The data analyst processes queries by sending them to an online edge server. The data analyst requests authentication using biometric and identification is considered. The fingerprint is the biometric utilized here, and since it is unique to each person, it is regarded as credentials for security. The request for authentication is sent to the data provided by the edge server after reaches it. The data supplier keeps track of identity, password, and biometric information in addition to the licensed data analysts. The authentication is carried out by using this saved data.

The request for authentication represents the credentials for security as $D_{id}, pw, and\ F_p$ for the identification, password, and fingerprint respectively. This is the request R that was sent to the data analyst.

The data analyst is authorized to send the authentication request if $R = 1$ is obtained. This becomes illegitimate if $R = 0$. Hashed queries are only accepted by authorized data analysts for searching. Every security credential is separated by a XOR operator as the edge server cannot be trusted. As a result, the XOR operator secures the data analyst's credentials. Based on the XOR operator's simplicity, this operation requires fewer resources and takes less time. After authentication, the data analyst sends a hashed query to the edge server. The hash query provided examines the tree from the root to the left and right nodes. Using the matching hash values discovered in the search results, the necessary query data is retrieved from the cloud. Cloud data extraction results in cipher text that needs to be decrypted with a decryption key. Once the data analyst is authorized, the data source also provides this key.

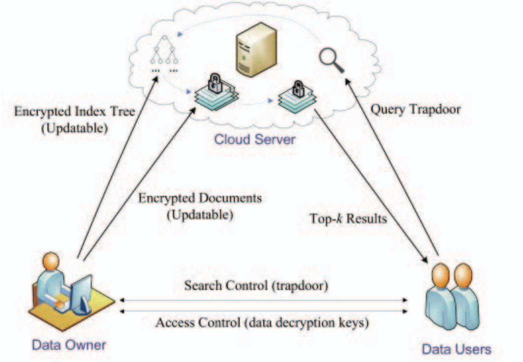

Fig 1 Fuzzy-based data encryption

### b) 3.2. Fuzzy network model

With the integration of fuzzy logic with neural networks, this system is often referred to as a fuzzy system. The architecture of a fuzzy system can be changed to satisfy the framework's specifications. The goal of this research study is to provide differential privacy and hence we have chosen to use fuzzy-based mechanism as an input in the proposed framework. This suggests a loss of accuracy due to a compromise between privacy and accuracy. To handle tabular data, we have used fuzzy rules; one example of this is the adult and heart disease dataset, despite the fact that they are often used for data. Which has spatial elements, such as image and video data**.** A fuzzy system's architecture typically consists of the following components:

- ✓ The input variables' input layer.
- ✓ Fuzzy sets are utilized as the fuzzy connection weights in conjunction with fuzzy rules for training the dataset, which is typically found in the second layer.
- ✓ The variables provided by the third layer of output;
- ✓ As it gets the fuzzy membership functions, the second layer also known as the neural layer depicts the fuzzy rules.

In this case, the clear values of private attributes like the adult dataset's age and wage attributes become fuzzy. Data disturbance caused by this fuzzification protects the privacy of original data. Here, the fuzzy logic is selected for performing data randomization instead of other techniques like k-anonymity, l-diversity, etc., since fuzzy logic can be used to ensure both discrete and numerical (continuous) qualities. Previous research demonstrates that a Fuzzy neural system outperforms a regular network model in terms of accuracy for the given task. According to Reference [12], the accuracy of a handwriting recognition system using a fuzzy-based encryption is higher of about 99% than that of a regular neural network for the training set, which has an accuracy of 97%.

## IV. Results

This study has considered 5000 edges and 1,000 vertices. The weight of each edge is determined randomly. MATLAB 2020 programming languages, Windows 10 operating system, an Intel (R) Core (TM) i5-5350U CPU running at 1.80 GHz, and 8.00 GB of RAM are used in this experiment. To assess the level of privacy protection, entropy is utilized to calculate the location privacy protection for fuzzy-based encryption mechanism. Let $M$ represent the quantity of occurrences that make up an event. Given $A$ and $N$, the total number of occurrences. Here, $M$ is the degree of ambiguity inherent in occurrence $A$ and $N$ is the overall amount of uncertainty. $Log_2 N - \log_2 M$ is the uncertainty $H$. When a system's uncertainty rises, the information entropy also rises. It means that there is better privacy protection, making it harder for an attacker to determine the exact position of the mobile user. In general, the k-anonymous algorithm has an entropy of information of $\log_2 k$. To find the precise location of a user, an attacker must first deduce UserSquare from the SensingGrid and acquire the unique location from UserSquare. Fuzzing sites are located at k different locations to confound the attacker. As a result, in our approach, the information entropy $H = \frac{1}{R} \times \log_2 k$, and the likelihood P of privacy in a location leaking can be characterized as $1/k \times R, (R \leq 1)$. To assess the uncertainties of fuzzy model as a variable, We made use of different fixed and user grids' area ratios. The scheme's information entropy matches that of the standard k anonymous process, which is regarded as the baseline, when the user grid equals the fixed grid. Based on the experimental outcomes shown in Fig [], we may conclude that, for a given $k$ conditions, the attacker's likelihood of determining the perceived user position decreases as entropy increases.

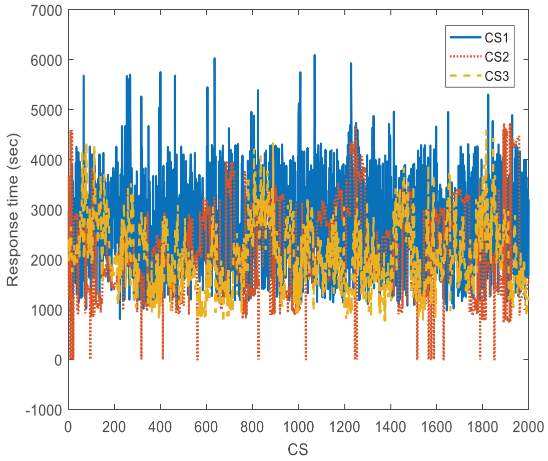

Fig 2 Response time of cloud servers

It can show that the fuzzy-based model has more expressiveness in this situation while maintaining the privacy protection. Furthermore, the information entropy also shows a monotonically increasing trend as the anonymous level $k$ increases while $R$ is fixed, suggesting that location information leaks are difficult to execute. Overall, the anonymized algorithm-driven on user grids contributes well to the privacy feature of the proposed model. Control privacy is measured by the DP's privacy parameter with $\varepsilon$ specification. The definition of DP makes it clear that the query privacy approach offers greater security with the smaller $\varepsilon$. The cacophony enhanced the question results follow a Laplace distribution, or what is referred to as the Laplacian mechanism.

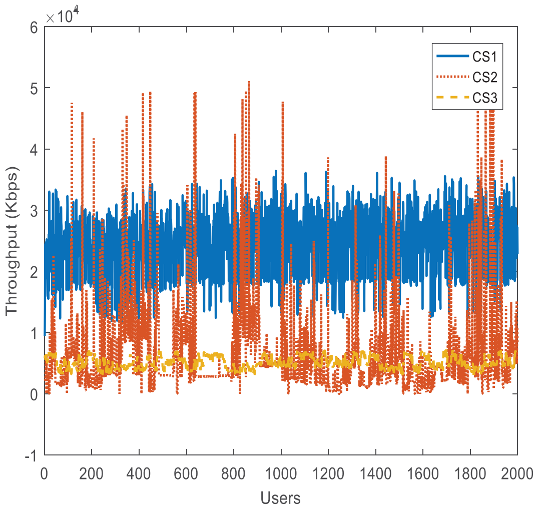

Fig 3 Throughput of cloud servers

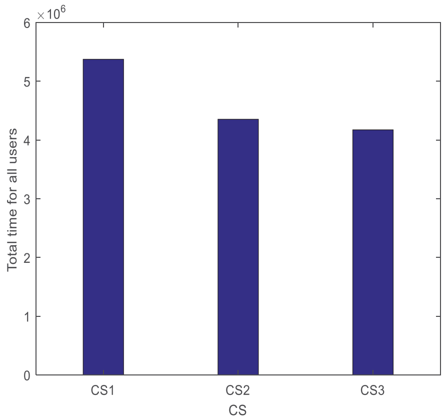

Fig 4 Response time after data encryption by the cloud servers

When determining the global sensitivity, a smaller ε corresponds to a larger b, indicating a stronger ability to withstand attacks, greater added noise, and comparatively less data availability. Global sensitivity and ε are related to the scale parameter $b = \Delta f / \varepsilon$. Thus, the purpose of this experiment is to investigate how various ε values affect the accessibility of data.

## V. Conclusion

This research study has focused on query processing, or safe data analyst access to the data, and data processing, or safe uploading of data by the data owner. The following parties were involved in the design of this system: the edge server, cloud, data owner, data provider, and data analyst. In this approach, the data source, which is regarded as a trusted entity performs Differential Privacy (DP). To protect the privacy of personal data, the Laplace mechanism is used in conjunction with the fuzzy-based encryption to introduce noise into the dataset. The sensitivity property is used to determine the parameter. Before the data is transferred to the public cloud, it is encrypted using the portable Piccolo algorithm. Since the public cloud cannot be trusted, data is encrypted before being stored. Using the proposed technique for searching, the hashed key phrases are provided. Before granting the data analyst access to decrypt the data, the security credentials are verified. After testing, the system design is considered as more efficient than earlier research regarding processing time, accuracy, and scalability in the field. It is intended that the fuzzy model will be improved in the future by adding more trustworthy entities in order to address single point failures. By adding Differential Privacy (DP) for privacy protection, hybrid machine learning algorithms can further expand the work.

## References

[1] Xia, J. Zhang, T. Q. S. Quek, S. Jin, and H. Zhu, "Energy-efficient task scheduling and resource allocation in downlink C-RAN," in 2018 IEEE Wirel. Commun. Netw. Conf., Apr 2018, pp. 1–6.

[2] Liu, T. Han, N. Ansari, and G. Wu, "On designing energy-efficient heterogeneous cloud radio access networks," IEEE Trans. Green Commun. Netw., pp. 1–13, 2018.

[3] Al-Dhabi, F. Paraiso, N. Djarallah, and P. Merle, "Elasticity in cloud computing: State of the art and research challenges," IEEE Trans. Serv. Comput., vol. 11, no. 2, pp. 430–447, Mar 2018.

[4] Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: A survey," Futur. Gener. Comput. Syst., vol. 56, pp. 684–700, Mar 2016.

[5] Sharafeddine, K. Jahed, O. Farhat, and Z. Day, "Failure recovery in wireless content distribution networks with device-to-device cooperation," Comput. Networks, vol. 128, pp. 108–122, Dec 2017.

[6] Su, Q. Xu, J. Luo, H. Pu, Y. Peng, and R. Lu, "A secure content caching scheme for disaster backup in fog computing enabled mobile social networks," IEEE Trans. Ind. Informatics, pp. 1–11, 2018.

[7] Yadav, O. A. Dobre, and N. Ansari, "Energy and traffic aware fullduplex communications for 5G systems," IEEE Access, vol. 5, pp. 11 278– 11 290, 2017.

[8] Fu, X. Sun, Q. Liu, L. Zhou, and J. Shu, ``Achieving ef_cient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing," IEICE Trans. Commun., vol. E98-B, no. 1, pp. 190_200, 2015.

[9] Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, ``Fuzzy keyword search over encrypted data in cloud computing," in Proc. IEEE INFO- COM, San Diego, CA, USA, Mar. 2010, pp. 1_5.

[10] Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, and A. V. Vasilakos, "Security and privacy for storage and computation in cloud computing," Information Sciences, vol. 258, pp. 371–386, 2014.

[11] Fu Z, Ren K, Shu J, et al. Enabling personalized search over encrypted outsourced data with efficiency improvement[J]. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(9): 2546-2559.

[12] Zheng Z, Zhou T C, Lyu M R, et al. Component ranking for fault-tolerant cloud applications [J]. IEEE Transactions on Services Computing, 2012, 5(4): 540-550.

[13] Chen W, Lee Y C, Fekete A, et al. Adaptive multiple-workflow scheduling with task rearrangement [J]. The Journal of Supercomputing, 2015, 71(4): 1297-1317.

[14] Jing W, Liu Y. Multiple DAGs reliability model and fault-tolerant scheduling algorithm in cloud computing system[J]. Computer Modeling and New Technologies, 2014, 18(8): 22-30

[15] Patra PK, Singh H, Singh R, et al. Replication and Re-submission Based Adaptive Decision for Fault Tolerance in Real-Time Cloud Computing: A New Approach[J]. International Journal of Service Science, Management, Engineering, and Technology, 2016, 7(2): 46-60.

[16] Duan, C. Chen, G. Min, and Y. Wu, "Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems," Future Generation Computer Systems, vol. 74, pp. 142–150, 2017.

[17] Ghribi, M. Hadji, and D. Zeghlache, "Energy efficient VM scheduling for cloud data centers: Exact allocation and migration algorithms," 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, 2013.

[18] Nazir, K. Qureshi, and P. Manuel, "Adaptive check pointing strategy to tolerate faults in economy based grid," Journal of Supercomputing, vol. 50, no. 1, pp. 1–18, 2009.