Data Protection based on Searchable Encryption and Anonymization Techniques

Matheus M. Silveira[1], **Rafael L. Gomes**[1]

[1]Universidade Estadual do Ceará (UECE), Brasil

matheus.monteiro@aluno.uece.br,rafa.lopes@uece.br

Abstract. *Data leakage compromises companies' confidentiality and directly impacts the existing privacy laws, as well as it is necessary to perform a light integration with the legacy systems, in order not to harm the performance of its services. Within this context, this paper presents an innovative cloud system to protect the private data of existing databases (legacy systems of clients) based on Searchable Symmetric Encryption for Databases (SSE-DB) and Permutation and Proprieties Maintenance Anonymization (PPM-Anon), attaching a security solution to the existing databases (without any change in these legacy systems). Results from real experiments using a real cloud environment suggest that the proposed solution is suitable for protecting the data without harming the performance of the existing services.*

1. Introduction

Data leakage compromises companies's confidentiality and directly impacts the existing privacy laws [Gong et al. 2022]. Nowadays, it is necessary to follow data protection regulations of privacy laws, like General Data Protection Regulation (GDPR) in Europe and Lei Geral de Proteção de Dados (LGPD) in Brazil, to avoid privacy issues and possible fees [Portela et al. 2023]. Additionally, data is considered one of the most important assets of companies and it is crucial to protect this asset [Costa et al. 2021, Moreira et al. 2021]. In this way, micro, small, and large companies, as well as government institutions, need to comply with the points listed by privacy laws, since this may have an impact on business, when dealing with data from their customers and employees, at the time of making the data portability, when cooperating internationally, etc [da Silva et al. 2020]. Thus, it is necessary to deploy security solutions to protect the data efficiently [Gupta et al. 2022].

One existing approach to protect the data is the usage of encryption techniques, that convert the input data (original) into output (encrypted). The conversion is based on a key, where only authorized entities have it and can decrypt the data [Aparajit et al. 2022]. However, data encryption techniques demand processing time, harming the possibility of performing frequent encryption and decryption of great amounts of data. Consequently, it will increase the time to perform a search process and the retrieval of the desired data [Gupta et al. 2022].

Another security approach for this scenario is Anonymization techniques. These techniques aim to make sensitive data transmitted to the Internet non-identifiable, preserving users' privacy [Yuan and Wu 2022]. Anonymization techniques emerge as a crucial approach to meet the aforementioned aspects of privacy laws, as they enable users to be protected in a non-reversible way. Nevertheless, existing anonymization

techniques enable different levels of anonymization, which can change the context of the data, making it impossible to apply intelligent solution (such as Artificial Intelligence) techniques to identify patterns, which can harm the management of the service [Silveira et al. 2023b, Silveira et al. 2023a].

Within this context, this paper presents an innovative cloud system to protect the private data of existing databases (legacy systems of clients) based on two designed techniques: (1) Searchable Symmetric Encryption for Databases (SSE-DB) and (2) Permutation and Proprieties Maintenance Anonymization (PPM-Anon). SSE-DB is an evolution of the original SSE [Li et al. 2019a] that enables SHA256, Encryption partitioning, and Encryption of Multiple Tables in SQL databases. Therefore, SSE-DB overcomes slow processing limitations, enabling higher effectiveness when considering huge databases and dynamic data. In the same way, PPM-Anon is an extension of a technique described in reference [Aleroud et al. 2016] and it generates synthetic data keeping mathematical properties, such as mean and standard deviation by permuting the eigenvectors instead of generating new ones, which avoids precision errors in these measurements.

Initially, the original sensitive data is securely stored in the cloud environment using SSE-DB (allowing the search and retrieval of the original sensitive data from the cloud), and later the original sensitive data is anonymized in the client's database using PPM-Anom (keeping the context of the data and, consequently, maintaining its usability). Thus, the system's goal is to prevent data leakage and privacy breaches, attaching a security solution to the existing databases, i.e., without any change in these legacy systems. The proposed system is part of a Research and Development project with LACNIC[1], which aims to develop solutions for industry to protect sensitive data in the Internet.

Results from real experiments using a real cloud environment suggest that the proposed solution is suitable to protect the data through encryption and anonymization of a database, where several scenarios were evaluated (varying the size of the database and the load requested in the encryption, search, and anonymization processes).

The remainder of this paper is organized as follows. Section 2 details the existing solutions for data protection. Section 3 describes the designed system, while Section 4 discusses the experiments performed and the results. Finally, section 5 concludes the paper and presents future work.

2. Related Works

Thabit et al. [Thabit et al. 2021] designed a cryptography technique for cloud computing security using two layers of encryption, ensuring the security of sensitive and confidential data during transport and storage. Similarly, Mann et al. [Mann et al. 2021] present an approach for ensuring data protection in dynamically changing cloud-based systems, which analyzes the configuration of the cloud-based system automatically to detect changes in the threats to data protection or in the availability of data protection mechanisms. However, both solutions do not evaluate the protection of the data in the existing system, neither consider the scenario of usage of protected data.

Wang et al. [Wang et al. 2022] propose a privacy-enhanced retrieval technology

[1] programafrida.net/en/archivos/project/sistema-de-proteccion-de-datos-basado-en-tecnicas-de-anonimizacion-y-que-cumple-con-las-leyes-de-privacidad

for cloud-assisted IoT, which is designed through an implicit index maintained by edge servers and a hierarchical retrieval model that preserves data privacy by hiding the information of data transmission between the cloud and the edge servers. In the same way, Suresha et al. [D and Karibasappa 2021] present a technique to enhance the data protection using key derivation based encryption, aiming to provide confidentiality, authentication and modification for the data stored in cloud. Nevertheless, these proposals do not consider scenarios where it is necessary to search and to get data from the encrypted database, limiting its applicability in existing legacy systems.

Based on the literature review, these existing proposals do not perform suitable data protection when it is necessary to search and retrieve data for users or end systems. Moreover, this work do not consider scenarios where the protected data is used as the input for other solutions, being necessary to anonymize this data to be used, while privacy issues are preserved.

3. Proposal

The behavior of the proposed system can be defined in the following steps: (1) the original sensitive data is securely stored in the cloud environment using SSE-DB; (2) later the original sensitive data is anonymized in the client's database using PPM-Anom (keeping the context of the data and, consequently, maintaining its usability); and, (3) the API receives the requests to search and retrieve the original sensitive data from the cloud.

Thus, the goal of the system is to prevent problems of data leakage and privacy breaches, attaching a security solution to the existing databases, i.e., without any change in these legacy systems. Next, we will detail the SSE-DB and PPM-Anon techniques, in Subsections 3.1 and 3.2, respectively.

3.1. SSE-DB for Data Protection

Our new approach SSE-DB evolved the original SSE in the following aspects: (I) Usage of SHA256 instead of MD5, in order to improve the security level by the generation of a 256-bits output expressed as 64 hexadecimal characters (while MD5 generates 128-bit with 32 hexadecimal characters); (II) Encryption partitioning, split of the data encryption to avoid problems of memory overload; and, (III) Encryption of Multiple Tables in SQL databases, allowing faster encryption and decryption.

A general SSE working model consists of a request of a Trapdoor w made by the client (data owner) to the Cloud service provider (server) that will return a list of the index of the documents that contain w. Since giving a decryption key to every user is not a safe approach, having an efficient search method is essential to keep this model functional.

One of the primary models of SSE-DB is one called Searching on Private-key Encrypted Data. It consists of giving the user that encrypted the data an access key that, after the required data is encrypted and stored in the Cloud, allows him and every user he gives the key to make requests in that database without decrypting it. According to reference [Li et al. 2019a], this scheme is made by the set of three algorithms $SK = (Gen, Enc, Dec)$, the first two are probabilistic algorithms and the last one is a deterministic algorithm. Initially, Gen is used to generate a random secret key K using an arbitrary security parameter as input. The algorithm Enc will use K and a message

m to generate the cryptography c of the message. Finally, Dec uses K and c to do the reverse work and re-generate m, working as a decrypting method.

A general SSE encryption scheme algorithms, that are [Li et al. 2019b]: Keygen(s), Trapdoor(MK, w), BuildIndex(R, MK) and Search(T, I).

1. **Keygen(s):** is an algorithm that should run in the client side to generate a master key MK based on a security parameter.
2. **Trapdoor(MK, w):** is an algorithm executed by the client, which takes MK and a keyword w as the input, and outputs the trapdoor T_w of word w.
3. **BuildIndex(R, MK):** is an algorithm that should be run by the client by taking MK and a record R as the input, and outputs the index IR for record R.
4. **Search(T, I):** is an algorithm that should be run by the server by taking a trapdoor T_w and a document's index IR as the input, and outputs 1 if w ϵ R or 0 otherwise.

The generated encrypted data, along with the associated indexes of all the keywords, are in general kept in safety by the server. Thus, using the private-key encrypted data searching method, the server will only be accessed by using the given access key to it.

We use $R = \{R_1, R_2, ..., R_n\}$ to represent a set of n records that will be encrypted so that it is possible to carry out the search process without decrypting the data. The encrypted data is represented by $I = \{I_1, I_2, ..., I_m\}$. Also, $R_{i,j}$ represents the j-th keyword in the i-th record. We use $S = \{S_1, S_2, ..., S_k\}$ to represent the set of k words to be searched in the encrypted table. The set E must be stored in the cloud and in this way operations can be executed in a more efficient way.

In addition to the general scheme, the digest of each row is stored in the database. This information is used to efficiently execute operations like update, delete and insert. The encrypted data with the associated indexes will be stored in the server in SSE-DB. To perform the search process, the client generates the trapdoor T_w for a word w and sends T_w to the server that performs the search process for each record in R.

The encryption scheme uses ciphers and hash functions in the build and search process and also needs access the token to work. The final encrypted data will be kept in the cloud by the server. In this scheme, the server will not be able to delete, modify or share the stored data with others, being just a secure way to store client's sensitive data. The server can contain multiple databases and at each request made by the client, they need to specify which one is being queried.

The build process consists in a data transfer between the client and the cloud server. The client, that has a database with sensitive data, will send it to the cloud to be encrypted and the original data will now be stored in a secure ambient (cloud server). The original client database will be replaced by a copy of the cloud database with encrypted sensitive data that will be returned to the client. This approach guarantees that in case of any security issue in the client side no sensitive data stored in the encrypted database will be leaked because SSE-DB encryption doesn't allow decrypting the stored data if the attacker does not have the original data.

This occurs because the encryption process consists in using Secure Hash Algorithms (SHA) as the hash function and then using Advanced Encryption Standard (AES)

symmetric block cipher as the primary encryption mode. The AES is used with two different keys and two different ciphers to generate the encryption of a single record. The first key used is the keyword of the record and it's used with the master key MK to generate a Trapdoor T_w of the record w with AES. The Trapdoor consists in a function that computes the hash of the keyword and then outputs the result after using a pseudo-random function. Then, this trapdoor is used to generate the final codeword of w with other AES function calls, along with the original ID of the record. Finally, this codeword is stored with a secure index generated based on the number of columns of the database.

For each index I generated from the encryption of a Database B, each record R_i is separately encrypted and stored in a different table. This process is a polynomial function and requires a considerable time to work, but it will only be needed to do this process one single time for each database required by the client. This function also allows multiple table encryption by time.

The search process will be requested by the client when they provide a query to a required table of the encrypted database containing keyword information. After this, the server will receive this query and run the SSE-DB search algorithm to respond with a list of all of the matched identifiers in the required database that contain that keyword. It is the safer way to guarantee that client will receive what they are looking for and still will not compromise the security of the stored sensitive data.

To search in the encrypted database generated by using the build function, the algorithm will re-calculate the same keys used to previously encrypt the data. That means, in the search function we will have the same encryption functions used to build the database to make it possible to re-generate the original cipher for the given keyword. After acquiring this AES cipher, the function will look for matches in all the required tables, saving the indexes of the matches found. Finally, a list with all the matched identifiers will be returned to the client.

Even with the need to rebuild the cipher and search record by record in the encrypted DB table, this is a very fast process and can be easily used in real-life applications, which is the focus of this work. A single client can make multiple requests to the server and still receive the answers in an acceptable time. This will be proved later in the Experiments section.

3.2. Anonymization Process

The PPM-Anon proposed in this paper modifies an existing [Aleroud et al. 2016] condensation-based anonymization method that generates a synthetic dataset through the use of information from the original data. The idea to generate a synthetic dataset is to shift the data to another space creating components, like the performed process in the Principal Component Analysis (PCA) to reduce the dimensionality of data. However, in this context, we are interested in preserving as much information as we can about the data. Therefore, when the data is being shifted to another space all the eigenvectors are used.

The process mentioned in the previous paragraph can be reversed, which means that we can shift back the data to the original space. The original method [Aleroud et al. 2016] does the reverse process using randomly generated eigenvectors that have statistical measurements such as mean and standard deviation equal to the measurements of the original eigenvectors. In this way, the data shifted back to the original space

is the synthetic data that share mathematical properties with the original data. The process of generating new eigenvectors could create precision errors in math proprieties such as mean and standard deviation during its generation. PPM-Anon emerges to mitigate these precision errors through the permutation of the axis of each eigenvector instead of generating new ones. The permutation will not change statistical measurements such as mean, standard deviation and others.

The aforementioned information allows us to introduce the PPM-Anon, the Algorithm 1. In the used notation $D = \{D_1, D_2, ..., D_n\}$ represents the original data and $A = \{A_1, A_2, ..., A_n\}$ is the resulting dataset after applying the anonymization method. Next, in Algorithm 1, it is the proposed PPM-Anon, which generates random permutations of each eigenvector to avoid precision errors and still preserve mathematical data proprieties.

Algoritmo 1 PPM-Anon

Entrada Dataset D
Saída Anonymized Dataset A
1: D' ← centeredData(D)
2: CM ← covarianceMatrix(D')
3: E ← eingvaluesAndEignvectors(CM)
4: P ← PCA(D', E)
5: E ← randomShuffle(E)
6: A ← reversePCA(P, E)
7: **Retorna** A

In line number 1 the data is centered (by the function $centeredData$), so we remove the average from each line. In line number 2, the covariance matrix D of the data is calculated (by the function $covarianceMatrix$) and in line number 3 its eingvelues/eignvectors are calculated and are used to perform Principal Component Analysis (PCA) to shift the data to a new space creating components. In line 5, a random permutation of the eigenvectors is generated (by the function $randomShuffle$), and the idea is to use this permutation to shift the data to the original space (by the function $reversePCA$).

The idea to use a permutation of eigenvectors is that it will only change number's positions, which means that measurements like mean, standard deviation, median, mode, variance, and other statistics measurements will still be the same. Thus, this approach will keep the meaning of the data after the anonymization process is performed.

3.3. PPM-Anon combined with Clustering

A K-Anonimity model creates groups of records that are k-indistinguishable, which means that it is not possible to identify individuals in the dataset with probability greater than 1/k. Therefore, the idea here is to combine PPM-Anon with clustering algorithms to achieve something similar to a K-Anonimity model. In the proposed anonymization strategy it is necessary to follow a few steps that can be seen in Figure 1.

- Processing Data: The first step is responsible for making the necessary changes to the dataset, for example, removing null fields, applying some algorithm on symbolic data to change from string to integer, etc.

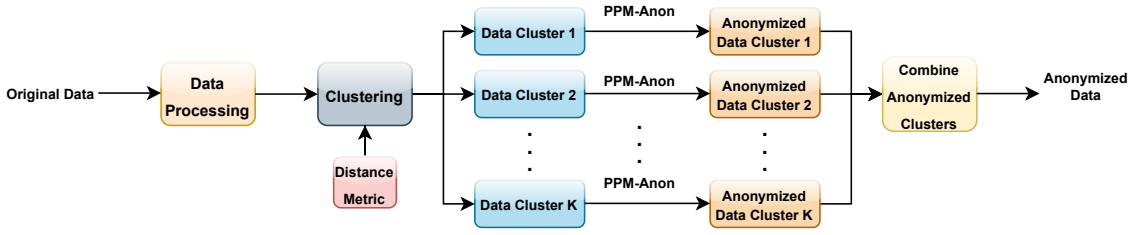

Figure 1. Idea of the anonymization strategy using clustering and PPM-Anon

- Clustering: The second step, which is Clustering, is the step implemented to group records to be anonymized separately. Thus, it is necessary to choose a clustering algorithm and also it is possible to create a customized distance metric for a specific scenario, for the experiments presented in this work, the default Euclidean distance is the used metric. After the clusters are created, the PPM-Anon algorithm is used to anonymize each one of them.
- Combine: The third step, which is the combination of anonymized clusters, is performed in order to keep other information about the clusters in the original order. For example, if each record has a label before the anonymization process, it will also have the same label after the anonymization is performed.

4. Experiments

This section presents the experiments performed to evaluate the performance of the proposed system for data protection, which is available in the repository of the project[2]. To carry out the experiments, a realistic scenario was defined with database information and a real cloud environment, enabling a suitable evaluation of the system and its impact.

4.1. Settings Description

As a cloud environment, we used an Elastic Cloud Server (ECS) in Huawei Cloud[3] with the following configuration: 12 vCPUS, 16GB of Memory RAM, and SSD Disk of 40GB. Regarding the database to be protected, it was deployed a PostgreSQL database and the Python Faker Package[4] to generate the data to fill this database. This data is generated by accessing properties named after the type of data in the generator initialized. Thus, by controlling the database size we can create several different situations and see in which cases the system is still efficient [Portela et al. 2024].

During the experiments, the searching time of SSE-DB and the processing time of PPM-Anom are considered evaluation metrics, since they are the major impact of the system for clients to be protected in a realistic scenario. In both experiments, we varied the size of the database.

4.2. Results

This subsection discusses the results of the experimental performed, where Figures 2, 3 and 4 illustrate the searching time of the SSE-DB for data protection and the processing time of the PPM-Anom for anonymization of a dataset of sensitive data.

[2]github.com/FRIDA-LACNIC-UECE
[3]huaweicloud.com
[4]pypi.org/project/Faker/

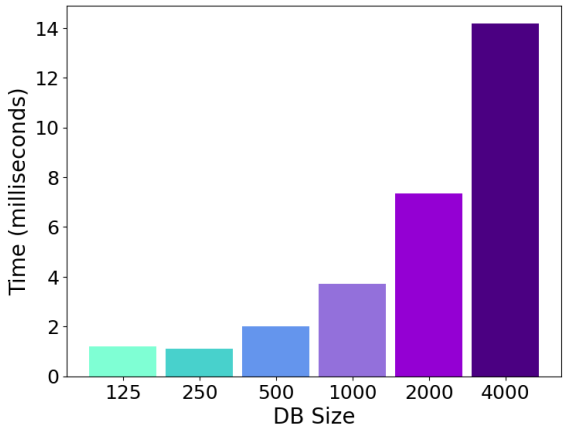

Figure 2. Processing time to anonymize a dataset

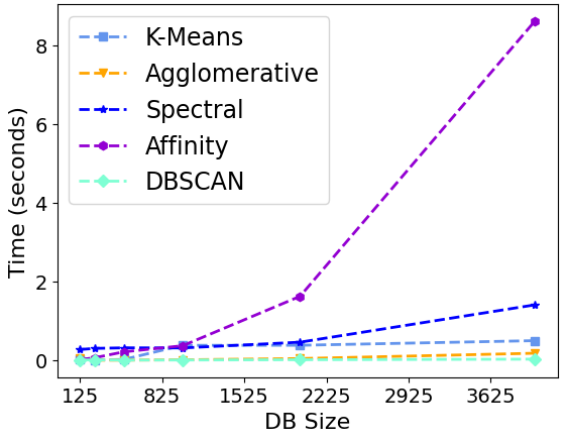

Figure 3. Processing time for a 3-anonymity model

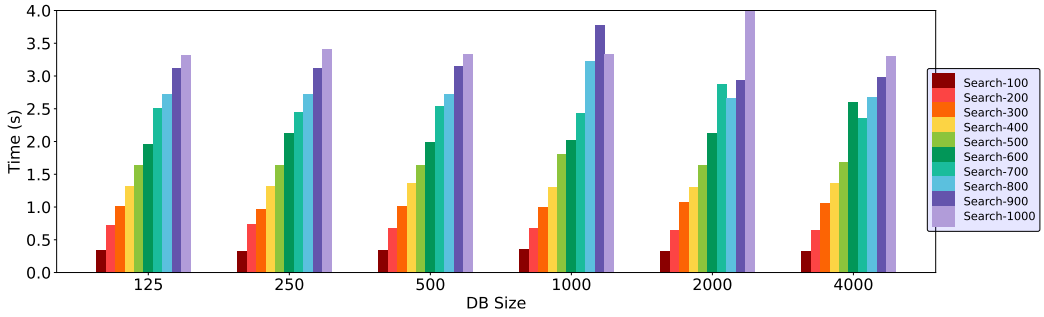

Figure 4. Searching Time of SSE-DB

Regarding the results of PPM-Anom, shown in Figure 2, the processing time to perform the anonymization grows according to the size of the data, since PPM-Anom performs several high computational cost functions, such as calculation of covariance matrix, eigenvalues, eigenvectors, PCA and random permutation. Despite the exponential behavior, the processing time is low when the context of the system is considered. For example, in the higher database size case, the processing time is 14 milliseconds, which is much smaller than a usual Round-trip time in the Internet [Sengupta et al. 2022, Flinta et al. 2020], avoiding a considerable negative impact on the existing communication behavior.

In Figure 3, the processing time for a 3-Anonymity model is presented using different clustering algorithms. The results demonstrate that clustering data into smaller groups results in a significantly reduced processing time for anonymization. Additionally, the choice of clustering algorithm influences the overall performance of the anonymization process. DBSCAN outperformed other algorithms, while Affinity and Spectral clustering exhibited the poorest performance.

As noted in Figure 4, the behavior of the search time is linear for most requests performed by the client, where the differences in searching times in the experiments were small (about 3 seconds as long as the maximum number of searches) considering the size of the databases analyzed. It is possible to note that a bigger amount of requests result in a varying behavior of searching time since the load is high when compared to the size

of the database. This point is illustrated in the case of 600 requests when the database is bigger than 2000. Additionally, it is important to note that the average search time of unique data is around 3 milliseconds, i.e., when the legacy system needs specific data, it can be searched and retrieved without an impact in the system performance when an end-to-end communication over the Internet is considered

5. Conclusion and Future Work

This paper presented a system to protect sensitive data in the existing systems, preventing problems of data leakage and privacy breaches, without any change in it. The proposed system is based on encryption, called SSE-DB, and anonymization, named PPM-Anon, techniques. Thus, it protects the existing systems while enabling the search and retrieval of encrypted data, as well as the availability of anonymized data that can be used as input for other solutions. Results from real experiments using a real cloud environment suggested that the proposed solution is suitable to protect the data without harming the performance of the existing services.

As future work, we intend to evaluate other searchable encryption approaches and other anonymization techniques, expanding the pool of security solutions that can be deployed by the system and, consequently, improving the security level of the companies.

References

Aleroud, A., Chen, Z., and Karabatis, G. (2016). Network trace anonymization using a prefix-preserving condensation-based technique (short paper). In *OTM Confederated International Conferences On the Move to Meaningful Internet Systems*, pages 934–942. Springer.

Aparajit, S., Shah, R., Chopdekar, R., and Patil, R. (2022). Data protection: The cloud security perspective. In *2022 3rd International Conference for Emerging Technology (INCET)*, pages 1–5.

Costa, W. L., Portela, A. L., and Gomes, R. L. (2021). Features-aware ddos detection in heterogeneous smart environments based on fog and cloud computing. *International Journal of Communication Networks and Information Security*, 13(3):491–498.

D, S. and Karibasappa, K. (2021). Enhancing data protection in cloud computing using key derivation based on cryptographic technique. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 291–299.

da Silva, G., Oliveira, D., Gomes, R. L., Bittencourt, L. F., and Madeira, E. R. M. (2020). Reliable network slices based on elastic network resource demand. In *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9.

Flinta, C., Yan, W., and Johnsson, A. (2020). Predicting round-trip time distributions in iot systems using histogram estimators. In *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9.

Gong, X., Chen, Y., Wang, Q., Wang, M., and Li, S. (2022). Private data inference attacks against cloud: Model, technologies, and research directions. *IEEE Communications Magazine*, 60(9):46–52.

Gupta, I., Singh, A. K., Lee, C.-N., and Buyya, R. (2022). Secure data storage and sharing techniques for data protection in cloud environments: A systematic review, analysis, and future directions. *IEEE Access*, 10:71247–71277.

Li, J., Huang, Y., Wei, Y., Lv, S., Liu, Z., Dong, C., and Lou, W. (2019a). Searchable symmetric encryption with forward search privacy. *IEEE Transactions on Dependable and Secure Computing*, 18(1):460–474.

Li, J., Niu, X., and Sun, J. S. (2019b). A practical searchable symmetric encryption scheme for smart grid data. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE.

Mann, Z. , Kunz, F., Laufer, J., Bellendorf, J., Metzger, A., and Pohl, K. (2021). Radar: Data protection in cloud-based computer systems at run time. *IEEE Access*, 9:70816–70842.

Moreira, D. A., Marques, H. P., Costa, W. L., Celestino, J., Gomes, R. L., and Nogueira, M. (2021). Anomaly detection in smart environments using ai over fog and cloud computing. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–2. IEEE.

Portela, A. L., Menezes, R. A., Costa, W. L., Silveira, M. M., Bittecnourt, L. F., and Gomes, R. L. (2023). Detection of iot devices and network anomalies based on anonymized network traffic. In *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pages 1–6.

Portela, A. L. C., Ribeiro, S. E. S. B., Menezes, R. A., de Araujo, T., and Gomes, R. L. (2024). T-for: An adaptable forecasting model for throughput performance. *IEEE Transactions on Network and Service Management*, pages 1–1.

Sengupta, S., Kim, H., and Rexford, J. (2022). Continuous in-network round-trip time monitoring. In *Proceedings of the ACM SIGCOMM 2022 Conference*, SIGCOMM '22, page 473–485, New York, NY, USA. Association for Computing Machinery.

Silveira, M. M., Portela, A. L., Menezes, R. A., Souza, M. S., Silva, D. S., Mesquita, M. C., and Gomes, R. L. (2023a). Data protection based on searchable encryption and anonymization techniques. In *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pages 1–5.

Silveira, M. M., Silva, D. S., Rodriguez, S. J. R., and Gomes, R. L. (2023b). Searchable symmetric encryption for private data protection in cloud environments. In *Proceedings of the 11th Latin-American Symposium on Dependable Computing*, LADC '22, page 95–98, New York, NY, USA. Association for Computing Machinery.

Thabit, F., Alhomdy, S., and Jagtap, S. (2021). A new data security algorithm for the cloud computing based on genetics techniques and logical-mathematical functions. *International Journal of Intelligent Networks*, 2:18–33.

Wang, T., Yang, Q., Shen, X., Gadekallu, T. R., Wang, W., and Dev, K. (2022). A privacy-enhanced retrieval technology for the cloud-assisted internet of things. *IEEE Transactions on Industrial Informatics*, 18(7):4981–4989.

Yuan, S. and Wu, X. (2022). Trustworthy anomaly detection: A survey.