MBNA Publishing House Constanta 2021

Proceedings of the International Scientific Conference SEA-CONF

SEA-CONF PAPER • OPEN ACCESS

Cloud Architectures based on Searchable Encryption for Maritime Software Industry

To cite this article: Marius Iulian MIHAILESCU and Stefania Loredana NITA, Proceedings of the International Scientific Conference SEA-CONF 2021, pg.177-188.

Available online at www.anmb.ro

ISSN: 2457-144X; ISSN-L: 2457-144X

doi: 10.21279/2457-144X-21-023

Cloud Architectures based on Searchable Encryption for Maritime Software Industry

Marius Iulian Mihailescu[1] and Stefania Loredana Nita[2]

[1]*Spiru Haret* University,
Scientific Research Center in Mathematics and Computer Science
[2]Instiute for Computers, Integrated Systems Department

m.mihailescu.mi@spiruharet.ro

Abstract. The paper will describe different scenarios on how cloud architectures can be used together with complex encryption techniques (e.g., searchable encryption), by providing a concrete implementation scenario of searchable encryption for maritime software.

1. Introduction

As we have seen, in 2020 many businesses and institutions were facing very powerful cyberattacks on their infrastructures, resulting in corrupted privacy of the data stored in the cloud, creating security breaches and allowing different vulnerable points to be exploited and producing data leakages (e.g., SolarWinds, FireEye). Starting with 2021, we have facing other powerful security attacks on different components of complex systems (e.g., Microsoft Exchange Servers).

The main purpose of the work presented in this paper is to give with a first attempt for the practical implementation of a searchable encryption mechanism applied in a cloud environment for maritime software applications (locally running or in the cloud). With the help of this mechanism, we will have at least an extra level of security in protecting the search process without exposing real data outside through the network's endpoints (shoreside, main cloud, and local cloud from the ships), keeping all the documents and data locally, and using only the search requests sent by the data owner (DO). An interesting approach for cloud architectures based on searchable encryption as a security mechanism can be seen in [10]. In [23], the author is showing and demonstrating a challenging solution for a visualization laboratory for earth sciences based on a multidisciplinary approach. His solution can be adjusted properly also for the maritime industry and data management, opening interesting research directions.

What is *searchable encryption*? Different sources will give multiple definitions, some of them quite fancy. In simple terms, a searchable encryption scheme represents a way to protect the sensitive data of the users, while providing protection for the searching process on the server-side. In Section 2 we will elaborate on how this goal of searchable encryption can be achieved.

In the last few years, a variety of searchable encryption mechanisms have been proposed, and many institutions are making important efforts in adopting searchable encryption techniques into their information security procedures. Using searchable encryption techniques, the data is protected in an unsafe environment, such as third-party servers. Therefore, the cloud servers or data centers can host various types of data encrypted using particular encryption schemes, while the data user is enabled to obtain specific encrypted data or documents that meet pre-defined search rules based on query-keywords.

Searchable encryption can be defined as well, as follows: it is an encryption system that contains well-defined algorithms which enable the search operation to be applied on the encrypted data, eliminating the need of retrieving from the server and then decrypting it to get the desired result. Searchable encryption can include additional algorithms that manage the access of the users to the search ar decryption algorithms based on the characteristics of the user in the system (role and permissions).

Paper Structure. The structure of the paper has five (5) Sections, as follows:

- *Section 1 – Introduction* summarizes the importance of having dedicated and secure cloud architectures, pointing out the main vulnerabilities points and how they can be encountered in real situations.
- *Section 2 – Literature Review* provides the most important cloud architectures that can be used together with complex encryption schemes (e.g., searchable encryption).
- *Section 3 – Preliminaries* presents the notions and definitions that will be used further in our work, as well as an overview of the general searchable encryption scheme.
- *Section 4 – Maritime Software Industry Challenges* will introduce the main challenges for porting the software applications and their resources in a secure cloud computing environment.
- *Section 5 – The proposed scheme* will discuss the cloud architecture(s) that we propose, which can be used together with a searchable encryption mechanism for the maritime software industry.
- *Section 6 – Conclusions* will present our results by pointing out the main difficulties that we have encountered and how we approach them for providing fixing solutions.

2. Preliminaries

Before moving forward with the main contributions of the papers, we will give a short overview of the notions that we are using in this paper (see Table 1) and we present the general structure of a searchable encryption scheme. The first step is to get familiarized with the concepts behind the searchable encryption mechanism before moving forward. The scheme that we have chosen to be presented as an example, has in its structure six algorithms, as we see them listed below:

1. $Gen_{key}(\omega) \rightarrow (pub_{key}, prv_{key})$
   The generation of the pair of keys is done by the Gen_{key} algorithm. The pair is formed as (pub_{key}) and (prv_{key}). The important parameter for this generation process is represented by the security parameter (ω).
2. $E(d_i, pub_{key}) \rightarrow Enc(c_i)$
   Using an encryption function (E), the encryption algorithm takes as input the public key (pub_{key}) and a plain document (d_i) and outputs the encryption $(Enc(c_i))$ of the document parameter.
3. $B_i(d_i, k_w, pub_{key}) \rightarrow j$
   The algorithm for building the index structure (B_i) has many input parameters, as follows: the plain document (d_i), the keyword that describes the current document (k_w), and lastly, the public key (pub_{key}). The output is represented by the index structure j.
4. $T(k_w, prv_{key}) \rightarrow T_{k_w}$
   The algorithm is responsible for generating the trapdoor value that will take two parameters as input, namely the query keyword and the private key (prv_{key}), while the output is one value called trapdoor (T_{k_w}).
5. $S(T_{k_w}, pub_{key}, j) \rightarrow Enc(C)$
   The algorithm (S) that searches through the encrypted documents requires three elements as input: the previously generated trapdoor (T_{k_w}), the public key (pub_{key}), and lastly, the index

structure (j), while it returns the set of encrypted documents $Enc(C) = \{c_{il}, \dots, c_{iw}\}$ that matches the search criteria.

6. $D\big(Enc(C), prv_{key}\big) \rightarrow D(Enc(C))$
   The algorithm for decryption (D) requires the following values as input: the encrypted document $Enc(C)$, and the private key prv_{key}, resulting in the set of plain documents $Set_D = \{D_{j1}, \dots, D_{jw}\} \subset S$.

Table 1. Notations

Acronym	Representation
DO	Data Owner
DU	Data User
GCSI	Global Cloud Server Infrastructure
DC	Document's control
CMDC	Cloud Maritime Data Center
MEI	Maritime Enterprise Infrastructure
MLC	Maritime Local Cloud
IDSFS	IDS/Firewall/Security Gateway
δ	Set of documents
AC	Access Control
d_i	document
pub_{key}	Public key
prv_{key}	Private key
ω	Security parameter
gen_{key}	The generation key algorithm
Enc	Encryption function
c_i	The encrypted document
B_i	Building index
k_w	Keyword
j	The index
T_{k_w}	The trapdoor function generated using the keyword
C	The set of the encrypted documents
D	The algorithm used for decryption
Set_D	The set of the plain documents
K_o	The owner's secret key
K_s	The server's key
PP	The public parameters of the system
R_U	The function for user revoking
P	The policy
S	The ID of the server
$Desc_{doc_s}$	The description of the documents set
PA	Probabilistic algorithm
DA	Deterministic algorithm

3. Literature Review

The current literature review is pointing out some of the main contributions which can be used as a reference guideline for practical implementation, by having a clear delimitation between the theoretical and practical background of searchable encryption.

A normal question that arises is related to the connection between the searchable encryption scheme and the control of the access. In [27], Song *et al.* managed to provide a modern classification for different searchable encryption flavors, and in [35] we have a comprehensive study about a searchable technique with the help of which the searchable encryption and access control can be used in the same system.

Searchable Symmetric Encryption (SSE). The contribution of Curtmola *et al.* [26] is important because it firstly provides access for single users. The work-papers [16, 8, 7, 19] bring arguments as a matter of demonstrating and proving the non-equivalent security mechanism that are listed in different works [13, 9, 12, 11]. In this field were studied multi-user SSE re-encryption starting with [26], while in [12] this is achieved based on re-encryption and broadcast encryption. In [24] the authors proposed a solution based on SSE and oblivious RAM. Other important contributions are focused on different challenges, such as: the trapdoor value is generated from multiple hashing operations [3], dynamic searchable encryption based on leakage-resilient for verifiable memory [4], searchable symmetric encryption designed to have multiple levels of access [5], symmetric searchable encryption based on Boolean mechanisms [6]. Another application of the SSE is designed for private information retrieval [14]. In [25] the authors propose a scheme designed for blind storage, in which the information that is learned by the server or data owner from the search process is very limited.

PEKS - Public-key encryption with keyword search. In [2], the authors propose a very unique and interesting searchable encryption scheme. Its main idea is to enables multi-creators for data and one recipient [40]. This approach is very useful in software applications with applicability in fields such as maritime, physics [20-22], or military where the design and the development require to provide access to classified documents just for certain users or groups of users. An important remark here is that many works have been focused on multi-destinations and access control, demonstrating in this way a very high potential for real-world scenarios. Here, there is a separation between the searchable encryption and access control that rely on third parties enabled for filtering the results obtained from the search [15] or designing search queries [7, 17]. Another approach in the literature is using attribute-based encryption with capabilities for keyword search [11, 7].

Using cloud services, the users can work with various types of documents (documents, sheets, slides, etc.), an interesting security aspect is being raised by images and video files. In this situation, if we are dealing with images that represent vital and confidential components, it is important to have a second approach as a security mechanism. For this, in [28] the authors proposed an interesting algorithm for image scrambling using a chaos-based algorithm for generating random permutations. In [29], the authors are using a chaotic map and demonstrate how it can be used in a real-time image encryption scheme.

In [30], the authors are discussing about an e-lottery system based on anonymous signatures. This is a very good example for the case in which if a malicious person wants to have an immediate benefit, he would be able to manipulate the searchable encryption scheme and the security advantage computation, by creating a proper security game between users, data owners and servers. Also, another important aspect that needs to be taken into consideration is represented by the fact that we never should trust the user, as many errors and security breaches can be caused by an untrained user. For this, we have found an interesting solution in [31] in which the authors are discussing about an insider threat detection based on natural language processing and analyzing the personality profiles.

4. Maritime Software Industry Challenges

Due to the complexity of the software applications and cloud computing environments, a set of critical vulnerabilities are being exploited with success every year. According to the OWASP (Open Web Application Security Project) classification, we have taken into consideration a set of five vulnerabilities that are exploited for maritime software applications.

The vulnerabilities are:

1. *Injection flaws.* Maritime software applications represent very complex software systems with a significant amount of configuration files and with a high number of functions and modules.

An injection flaw is taking place when untrusted data is being sent such as a query (see Step 5 – The searching algorithm from Section 2).

2. *Using components that have known vulnerabilities.* A component represents a collection of libraries, frameworks or other modules. Usually, in most of the cases, the components are running under the same privileges as the maritime software application. Once the system has a component that is vulnerable, the attackers will take every advantage, exploit it and causing data loss.
3. *Broken access control.* Most of the maritime software applications have strong authentication mechanisms and cryptographic mechanisms. In the case of maritime software applications is very important to implement proper restrictions for users and to inflict them.
4. *Security misconfiguration.* When developing maritime software applications, security misconfiguration is recommended to be treated as a part of security analysis. As the maritime software industry is very complex and provides a variety of multiple software applications, most of the security misconfigurations are represented by the followings: default configurations that are set in a wrong mode, incomplete configurations, providing open cloud solutions for storing documents and other types of data, HTTP headers and their configuration, explicit error messages that expose sensitive data and information.
5. *Insecure deserialization.* Most of the time, deserialization is creating opportunities for the attackers to inject and execute attacks based on the escalation of the privileges. Deserialization is often creating flaws that are resulted in remote code execution.

Essentially speaking, it is very important to have test cases for developing maritime software applications. Every functionality that is being developed needs to be tested in the early stages of the development process. By doing so, it will make sure that the vulnerabilities will be identified properly and eliminated immediately. In order to accomplish such complex tasks, it is recommended to include in the test cases procedure the static code analysis. To perform static code analysis, it can be used free or paid tools, such as Klocwork[1], Apache Yetus[2] or CodeScene[3]. Static code analysis will give us the verification of the software properties that are used in safety-critical computer systems and to locate the potentially vulnerable code.

5. Proposed Scheme

Due to its characteristics and capabilities, searchable encryption has become an important encryption technique, which lets the data users to generate search queries (based on specific keywords) that are applied directly over the encrypted data. For a proper parametrization and application, understanding who are the participants represents one of the main goals together with the components of a searchable encryption system.

The following participants are involved in the system: the data owners DOs who own the set of documents $\delta = \{\delta_1, \dots, \delta_n\}$, encrypt the documents and sends them to cloud storage; the data users DUs who can initiate the search process with the approval of the DO. After DU initiates the search query, further the DO transmits the query to the cloud storage. The last participant is the cloud server, which stored the encrypted documents, applies the search operation and sends the encrypted result to DU.

We have two categories of *participants* in our scheme (see Figure 1), *human participants* and *hardware components* (represented by servers and cloud components). This being said, in Figure 1 we have the following participants together with their roles:

- *Human participants*
    - DO has two roles in the proposed scheme: to encrypt the documents as they are created by DUs, and to process the queries for different documents (documents

[1] Klocwork, https://www.perforce.com/products/klocwork
[2] Apache Yetus, https://yetus.apache.org/
[3] CodeScene, https://codescene.io/

that are already encrypted and stored on the server or newly created documents that will be encrypted and immediately sent for being stored).
    - DUs have the responsibility to create queries for the documents in order to access them.
- *Hardware participants (components)*
    - *IDS/Firewall/Security Gateway (IDSFS)* – has the role of scanning all the requests that are going through DO and coming from the shoreside through the maritime infrastructure. Also, provides a high security mechanism for creating secure communication channels.
    - *Maritime Local Cloud (MLC)* – represents the entire infrastructure of servers and interconnected network devices from the cruise ship/vessel/tanker. It serves as the main point of storing the documents and data that are exchanged between shoreside and ship.
    - *Maritime Enterprise Infrastructure (MEI)* – represents the entire communication and network devices infrastructure that guarantees the satellite communication between the ship and shoreside.
    - *Document's control (DC)* – represents the load balancer used to deal with the requests that are coming back and forward (we are talking about millions of requests that can happen in a couple of minutes, for example in the case of embarkation days for cruise ships).
    - *Global Cloud Servers Infrastructure (GCSI)* – represents the entire infrastructure (hardware and software services) where the data are stored and managed.
    - *Cloud Maritime Data Center (CMDC)* – represents the entire local infrastructure where all the documents and databases are stored. It acts as a backup and disaster recovery solution. The main idea is not to store everything in the cloud without proper measures for backup, security purposes and so on.

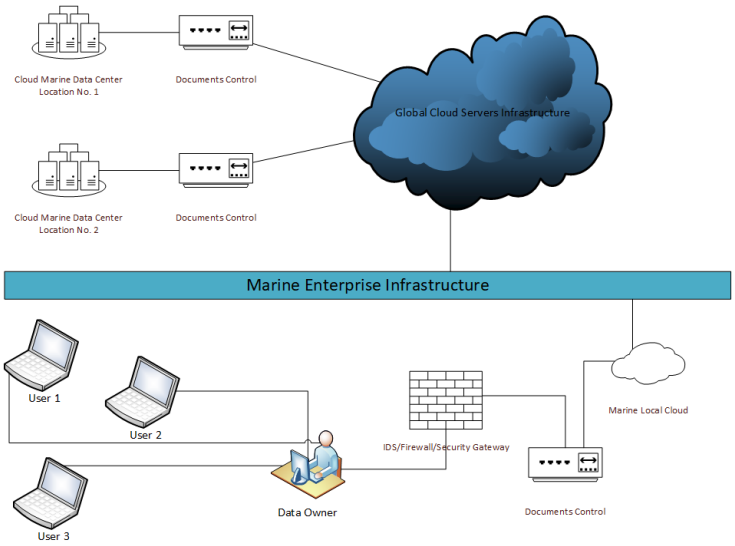

Figure 1. Maritime Enterprise Infrastructure (MEI)

Our scheme (see Figure 1) is based on the following scenario:

- Step 1 – The DO encrypts the documents and sends them to the MLC. Before arriving on the MLC, the data is validated by the following two mechanisms: *IDSFS network component* and *DC component*. The first component is responsible for making sure that the integrity of the documents is untouched (e.g., forged), and the second component's purpose is to ensure that all

the requests from the data owner are executed accordingly by allocating the proper hardware resources, playing the role of a *load balancer.*

- Step 2 – DUs are creating query requests for different documents, having the possibility to access those documents, represented as outsourced data on the servers from the maritime cloud infrastructure. In order to have powerful and optimized search queries, a very interesting solution is proposed in [18];
- Step 3 – The query requests are sent to the DO who validates and approves them. The validation process is represented by checking if the requested documents are already encrypted and stored on the servers from the shoreside data centers.
- Step 4 – The servers from the maritime cloud infrastructure will return the search results by querying the CMDC through the DC workflow, acting as a load balance as well;

Based on the roles of the participants and the proposed scheme in Figure 1, we have designed the following algorithms that can be implemented:

1. $(\boldsymbol{K_O, K_s, PP}) \leftarrow \boldsymbol{gen_{key}}(\boldsymbol{1^{\omega}, P, S})$. The first algorithm gen_{key} represents a PA, his goal is to output the K_o, the K_S of the server, and the PP of the system. The gen_{key} algorithm will generate the values (K_O, K_S and PP) based on the following input parameters: the security parameter (1^{ω}), the policy (P) for the system and lastly, the unique ID of the server (S). The DO triggers this algorithm.
2. $\boldsymbol{I_D} \leftarrow \boldsymbol{B}(\boldsymbol{Desc_{doc_s}, K_O, PP})$. The second algorithm B represents a PA, his goal is to output the index structure I_D, constructed from the following input values: the keywords that describe the set of documents $Desc_{doc_s}$, the private key (K_O) of the owner and the public parameters of the system (PP). The DO triggers this algorithm.
3. $\boldsymbol{K_U} \leftarrow \boldsymbol{F}(\boldsymbol{u, \theta(u), K_O, PP})$. The third algorithm F represents a PA that is used for registering users into the maritime system and for providing access to the proper servers, which outputs the private key K_U for the data user. This value is obtained based on the following input parameters: the unique identity of the user u , the values that describe the level of the access $\theta(u)$, DO's private key K_O, and lastly the public parameters PP of the system. The DO triggers this algorithm.
4. $\boldsymbol{T_{(\theta,\omega(u))}} \leftarrow \boldsymbol{Q}(\boldsymbol{\omega, K_u})$. The fourth algorithm Q represents a DA that returns the token value $T_{(\theta,\omega(u))}$, computed based on the input parameters: the keyword $\omega \in \Delta$ (in which Δ is the dictionary from which the keyword is chosen) and the private key $\boldsymbol{K_u}$ of the data user. DU triggers this algorithm, but firstly it is verified whether DU has a proper clearance or level of access $\omega(u)$. If affirmatively, the tokenization algorithm runs, otherwise DU will receive a proper message.
5. $\boldsymbol{R_{(\theta,\omega(u))}} \leftarrow \boldsymbol{S}(\boldsymbol{T_{(\theta,\omega(u))}, I_D, K_s})$. The fifth algorithm S represents a DA that returns the set of documents $R_{(\theta,\omega(u))}$ that matches the search token or a failure symbol φ if there are not such documents, based on the following inputs: the token $T_{(\theta,\omega(u))}$ resulted from the previous algorithm, the index structure I_D, and the server's key K_S. The server triggers this algorithm. An important characteristic here is the fact that the identifiers $d_j \in D_{(\theta,\omega(u))}$ are incorporated into the outcome, containing ω, with the property that $\omega(d_j) \leq \omega(u)$, where $\omega(u_i)$ is the clearance (access level) for DU.
6. $(\boldsymbol{K_O'}, \boldsymbol{K_S'}) \leftarrow \boldsymbol{R_U}(\boldsymbol{u, K_O, PP})$. The sixth algorithm R_U represents a PA that returns the new keys for the data owner K_O' and the server K_S' after a data user is removed from the system,

based on the input parameters: the identifier u of the user that will be removed, the secret key K_O of the DO, and the public parameters PP. the DO triggers this algorithm.

The correctness of the SE scheme lays in the following fact: for all $k \in \mathbb{N}$, for all K_O, K_S that are generated by $gen_{key}(1^{\omega}, P, S)$, for all I_D that are resulted from $B(Desc_{doc_s}, K_O, PP)$, $\omega \in \Delta$, $u \in U$, and for all K_U that are resulted by $F(u, \theta(u), K_O, PP)$, $S(T_{(\theta,\omega(u))}, I_D, K_s) = R_{(\theta,\omega(u))}$.

6. Conclusions

The current work has examined and demonstrated how a searchable encryption scheme can parametrize properly in order to apply it in a real environment, such as the maritime software industry. Each aspect of research that has been pointed out raised a set of challenges, as follows:

- During the state-of-the-art and finding the proper searchable encryption schemes from theoretical background made us conclude that in practice, the hardware resources that are necessary to be available and properly allocated are modest.
- During the implementation process of our scheme and integrating it with the infrastructure [32-34] has been raising some challenges, in order to obtain maximum reliability [35-38], efficiency and timely execution of the computations and the algorithm implementations [39].
- When we have proposed the access policy for the users of a maritime desktop/laptop station, the challenges raised were in determining exactly what documents each type of user needs to access. The policy has been implemented and we have obtained positive results without experiencing any security issues.
- The proposed theoretical scheme (see Section 5) represents the starting point from a mathematical point of view, by sketching a big picture of the mathematical algorithms with respect to the hardware infrastructure. In Section 4 we have described the entire system for simulation purposes and how the scheme has been implemented for simulation.
- The scheme (see Section 4) has been implemented in a virtualized system (using VMWare[4] and ESXi[5]), by developing a maritime software application that is similar to Napa Logbook[6] for the maritime industry. Our scheme manages to show success, both theoretically and practically.

Future research directions. As a plan for future research directions, we have proposed a couple of milestones that we want to reach and fulfill, such as:

- We want to extend our scheme and to provide implementations for many as possible software applications (web, desktop) for cloud computing, edge computing, fog computing and with a special focus for Internet-of-Things (IoT) devices.
- We have already started and moving forward with our scheme in different multi-disciplinary fields, such as physics, biology, meteorology and agriculture. In physics, our scheme has to be improved for special and dedicated complex systems, especially if the software applications are built with the goal of collecting the data from the performed experiments, related to tungsten experiments (as we can see in [7] and [8]) or plasma [9]. The experiments are quite complex, the software applications that generate reports and other documents that contain experiment results, and they are made available in cloud infrastructures require special attention from processing on high-performance computation architectures perspective [24] and security perspective by guaranteeing confidentiality, integrity and authentication of each process with respect for data.

[4] VMWare, https://www.vmware.com/

[5] ESXi, https://www.vmware.com/products/esxi-and-esx.html

[6] Napa Logbook, https://www.napa.fi/software-and-services/ship-operations/napa-logbook/

- Collecting and comparing the results and provide better solutions as the technology evolves (security analysis, timely execution, developer's effort etc.) for searchable encryption implementations. Also, big data support will be added and providing analytics.

References

[1] Alderman, James & Martin, Keith & Renwick, Sarah. (2017). Multi-level Access in Searchable Symmetric Encryption. 35-52. DOI: 10.1007/978-3-319-70278-0_3.

[2] Tarik Moataz and Abdullatif Shikfa. 2013. Boolean symmetric searchable encryption. In Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security (ASIA CCS '13). Association for Computing Machinery, New York, NY, USA, 265–27. DOI: 10.1145/2484313.2484347.

[3] David Cash, Paul Grubbs, Jason Perry, and Thomas Ristenpart. 2015. Leakage-Abuse Attacks Against Searchable Encryption. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15). Association for Computing Machinery, New York, NY, USA, 668–679. DOI: 10.1145/2810103.2813700

[4] Shangqi Lai, Sikhar Patranabis, Amin Sakzad, Joseph K. Liu, Debdeep Mukhopadhyay, Ron Steinfeld, Shi-Feng Sun, Dongxi Liu, and Cong Zuo. 2018. Result Pattern Hiding Searchable Encryption for Conjunctive Queries. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18). Association for Computing Machinery, New York, NY, USA, 745–762. DOI: 10.1145/3243734.3243753.

[5] Peter Baumann. 2011. Accelerating computationally intensive queries on massive earth science data: (system demonstration). In Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases (AD'11). Association for Computing Machinery, New York, NY, USA, 31–35. DOI: 10.1145/1966895.1966899.

[6] Paul Brown and Michael Stonebraker. 1995. BigSur: A System For the Management of Earth Science Data. In Proceedings of the 21th International Conference on Very Large Data Bases (VLDB '95). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 720–728.

[7] V. Marascu, C. Stancu, V. Satulu, A. Bonciu, C. Grisolia, G. Dinescu, Material erosion and dust formation during tungsten exposure to Hollow-Cathode and Microjet Discharges, APPLIED SCIENCES-BASEL, Volume: 10, Issue: 19, Article Number: 6870, Published: OCT 2020. DOI: 10.3390/APP10196870.

[8] V. Marascu, A. Lazea-Stoyanova, A. Bonciu, V. Satulu, G. Dinescu, Tungsten particles fabrication by a microjet discharge, MATERIALS RESEARCH EXPRESS, Volume: 7, Issue: 6, Article Number: 066509, Published: JUN 2020. DOI: 10.1088/2053-1591/AB955D.

[9] V. Marascu, A. Lazea-Stoyanova, C. Stancu, G. Dinescu, The influence of plasma operation parameters on synthesis process of copper particles at atmospheric pressure, PLASMA PROCESSES AND POLYMERS, Volume: 15, Issue: 1, Article Number: e1700091, Published: JAN 2018. DOI: 10.1002/PPAP.201700091.

[10] Alejandro Aguilar-Sierra. Visualization laboratory for Earth Sciences: a multidisciplinary visual learning environment. In SIGGRAPH '09: Posters (SIGGRAPH '09). Association for Computing Machinery, New York, NY, USA, Article 96, 1. DOI: 10.1145/1599301.1599397.

[11] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky. Searchable symmetric encryption: improved definitions and efficient constructions. In Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, pages 79–88. ACM, 2006. DOI: 10.1145/1180405.1180417.

[12] D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. In 2000 IEEE Symposium on Security and Privacy, pages 44–55. IEEE, 2000. DOI: 10.1109/SECPRI.2000.848445.

[13] Dorian Gorgan. 2014. Spatial data processing on high performance computation architectures.

In Proceedings of the 7th Euro American Conference on Telematics and Information Systems (EATIS'14). Association for Computing Machinery, New York, NY, USA, Article 1, 1–4. DOI: 10.1145/2590651.2590653.

[14] Sriram Krishnan, Christopher Crosby, Viswanath Nandigam, Minh Phan, Charles Cowart, Chaitanya Baru, and Ramon Arrowsmith. 2011. OpenTopography: a services oriented architecture for community access to LIDAR topography. In Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications (COM.Geo '11). Association for Computing Machinery, New York, NY, USA, Article 7, 1–8. DOI: 10.1145/1999320.1999327.

[15] N. S. Loredana, "On recommendation systems applied in big data," 2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Ploiesti, 2016, pp. 1-6. DOI: 10.1109/ECAI.2016.7861068

[16] Dăscălescu, A.C., Boriga, R.E. A novel fast chaos-based algorithm for generating random permutations with high shift factor suitable for image scrambling. Nonlinear Dyn 74, 307–318 (2013).DOI: 10.1007/s11071-013-0969-6.

[17] Radu Boriga, Ana Cristina Dăscălescu, Adrian-Viorel Diaconu, "A New One-Dimensional Chaotic Map and Its Use in a Novel Real-Time Image Encryption Scheme", Advances in Multimedia, vol. 2014, Article ID 409586, 15 pages, 2014. DOI: 10.1155/2014/409586.

[18] Florin Medeleanu, Ciprian Răcuciu, Madlena Nen, Zieduna Liepe & Narcis Florentin Antonie (2019) Fair e-lottery system proposal based on anonymous signatures, Applied Economics, 51:27, 2921-2933, DOI: 10.1080/00036846.2018.1563671.

[19] S. Eftimie, R. Moinescu and C. Răcuciu, "Insider Threat Detection Using Natural Language Processing and Personality Profiles," 2020 13th International Conference on Communications (COMM), Bucharest, Romania, 2020, pp. 325-330, DOI: 10.1109/COMM48946.2020.9141964.

[20] Opris, V.N. & Opris, M.E. 2016, "EXPERT SYSTEMS RUNNING ACROSS MULTIPLE CLOUDS. A SUSTAINABLE PERSPECTIVE", Scientific Bulletin "Mircea cel Batran" Naval Academy, vol. 19, no. 2, pp. 585, 2016. DOI: 10.21279/1454-864X-16-I2-076.

[21] Opris, V.N. & Racuciu, C. 2015, "THE EXPERT SYSTEMS ANALYSIS USING THE CONCEPT OF BIG DATA AND CLOUD COMPUTING SERVICES", Scientific Bulletin "Mircea cel Batran" Naval Academy, vol. 18, no. 2, pp. 46-50, 2015. DOI: 10.21279/1454-864X-17-I1-0 8.

[22] Ioannis Demertzis, Rajdeep Talapatra, and Charalampos Papamanthou. 2018. Efficient searchable encryption through compression. Proc. VLDB Endow. 11, 11 (July 2018), 1729–1741. DOI: 10.14778/3236187.3236218.

[23] Lei Xu, Xingliang Yuan, Ron Steinfeld, Cong Wang, and Chungen Xu. 2019. Multi-Writer Searchable Encryption: An LWE-based Realization and Implementation. In Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security (Asia CCS '19). Association for Computing Machinery, New York, NY, USA, 122–133. DOI: 10.1145/3321705.3329814

[24] Florian Hahn and Florian Kerschbaum. 2014. Searchable Encryption with Secure and Efficient Updates. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14). Association for Computing Machinery, New York, NY, USA, 310–320. DOI: 10.1145/2660267.2660297.

[25] Johannes Blömer and Nils Löken. 2018. Cloud Architectures for Searchable Encryption. In Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES 2018). Association for Computing Machinery, New York, NY, USA, Article 25, 1–10. DOI: 10.1145/3230833.3230853.

[26] Cédric Van Rompay, Refik Molva, and Melek Önen. 2018. Secure and Scalable Multi-User Searchable Encryption. In Proceedings of the 6th International Workshop on Security in Cloud Computing (SCC '18). Association for Computing Machinery, New York, NY, USA,

15–25. DOI: 10.1145/3201595.3201597.
[27] Shi-Feng Sun, Xingliang Yuan, Joseph K. Liu, Ron Steinfeld, Amin Sakzad, Viet Vo, and Surya Nepal. 2018. Practical Backward-Secure Searchable Encryption from Symmetric Puncturable Encryption. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18). Association for Computing Machinery, New York, NY, USA, 763–780. DOI: 10.1145/3243734.3243782.
[28] L. A. Dumitru, S. Eftimie, M. I. Mihailescu, S. L. Nita, V. Opris and C. Racuciu, "A novel architecture for authenticating scalable resources in hybrid cloud," 2016 International Conference on Communications (COMM), Bucharest, Romania, 2016, pp. 251-254. DOI: 10.1109/ICComm.2016.7528254.
[29] Albeanu G., Madsen H., Popențiu-Vlădicescu F. (2020) Computational Intelligence Approaches for Software Quality Improvement. In: Pham H. (eds) Reliability and Statistical Computing. Springer Series in Reliability Engineering. Springer, Cham. DOI: 10.1007/978-3-030-43412-0_18.
[30] F. Popențiu-Vlădicescu, G. Albeanu and H. Madesn, "Reliability of Modern Engineering Systems - Towards a Safer World," 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2019, pp. 1-5. DOI: 10.1109/ICOMET.2019.8673474.
[31] Popentiu-Vladicescu, F., Albeanu, G., & Madsen, H. (2019). Improving software quality by new computational intelligence approaches. In Proceedings of 25th ISSAT International Conference on Reliability & Quality in Design (pp. 152-156). [RQD25-152]
[32] F. Popentiu-Vladicescu and G. Albeanu, "Software reliability in the fog computing," 2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT), Karachi, 2017, pp. 1-4. DOI: 10.1109/ICIEECT.2017.7916578.
[33] Mihailescu, Marius Iulian and Stefania Loredana Nita. "CSAP: Cyber Security Asynchronous Programming With C++20 and C# 8 for Internet of Things and Embedded Software Systems." Examining the Impact of Deep Learning and IoT on Multi-Industry Applications, edited by Roshani Raut and Albena Dimitrova Mihovska, IGI Global, 2021, pp. 249-269. DOI: 10.4018/978-1-7998-7511-6.ch014.
[34] Tao Feng and Weiyou He. 2018. Research on Privacy Preserving of Searchable Encryption. In Proceedings of the 2018 2nd High Performance Computing and Cluster Technologies Conference (HPCCT 2018). Association for Computing Machinery, New York, NY, USA, 58–68. DOI: 10.1145/3234664.3234665.
[35] Nils Löken. 2017. Searchable Encryption with Access Control. In Proceedings of the 12th International Conference on Availability, Reliability, and Security (ARES '17). Association for Computing Machinery, New York, NY, USA, Article 24, 1–6. DOI: 10.1145/3098954.3098987.
[36] Shangping Wang, Xiaoxue Zhang, Yaling Zhang. 2016. Efficiently Multi-User Searchable Encryption Scheme with Attribute Revocation and Grant for Cloud Storage. PloS one, 11(11), e0167157. DOI: 10.1371/journal.pone.0167157
[37] James Alderman, Keith M. Martin, and Sarah Louise Renwick. 2017. Multi-level Access in Searchable Symmetric Encryption. IACR Cryptology ePrint Archive (2017), 211.
[38] Christoph Bosch, Pieter H. Hartel, Willem Jonker, and Andreas Peter. 2014. A Survey of Provably Secure Searchable Encryption. ACM Comput. Surv. 47, 2 (2014), 18:1--18:51. DOI: 10.1145/2636328
[39] Hirano, Takato & Kawai, Yutaka & Koseki, Yoshihiro. (2018). Efficient Trapdoor Generation from Multiple Hashing in Searchable Symmetric Encryption: 14th International Conference, ISPEC 2018, Tokyo, Japan, September 25-27, 2018, Proceedings. DOI: 10.1007/978-3-319-99807-7_10.
[40] Guan, Wenhao & Wang, Yunling & Wang, Jianfeng & Fu, Xiaotong. (2018). Verifiable memory leakage-resilient dynamic searchable encryption. Journal of High-Speed Networks.

24. 201-217. 10.3233/JHS-180591. DOI: 10.3233/JHS-180591