electronics

Article

A Novel Approach to Reduce Breaches of Aircraft Communication Data

Shahzaib Tahir [1,*,†], Muhammad Arslan Shahbaz [1,†], Hasan Tahir [2,†], Muhammad Awais [1,†], Fawad Khan [1,†], Ruhma Tahir [3,†], Saqib Saeed [4,†] and Abdullah M. Almuhaideb [5,†]

[1] Department of Information Security, College of Signals, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[2] Department of Computing, School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[3] British Telecommunications Plc, Martlesham Heath, Martlesham, Ipswich IP5 3RE, UK
[4] Saudi Aramco Cybersecurity Chair, Department of Computer Information Systems, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam 34221, Saudi Arabia
[5] Saudi Aramco Cybersecurity Chair, Department of Networks and Communications, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam 34221, Saudi Arabia
* Correspondence: shahzaib.tahir@mcs.edu.pk
† These authors contributed equally to this work.

Abstract: Aircraft are complex systems that rely heavily on monitoring and real-time communications with the base station. During aviation and flight operations, diverse data are gathered from different sources, including the Cockpit Voice Recorder (CVR), Flight Data Recorder (FDR), logbook, passenger data, passenger manifest etc. Given the high sensitivity of flight data, it is an attractive target for adversaries which could result in operational, financial and safety related incidents. Communications between aircraft pilots and air traffic controllers are all unencrypted. The data, mainly audio communication files, are placed openly within data centers on the ground stations which could lead to a serious compromise in security and privacy. One may rely on the cloud owing to its on-demand features but to thwart possible attacks, the data need to be encrypted first, giving rise to the issue of conducting search over encrypted data. This research presents a novel approach for data security in aviation industry by introducing a semantic-based searchable encryption scheme over the cloud. The designed system has proven to be extraordinarily effective for semantic-based searchable encryption at the word and the text level. The rigorous security and complexity analysis shows that the proposed solution provides a high level of security and efficiency and can be effectively deployed in the aviation sector. The designed scheme is tested through a real-world aviation dataset collected to demonstrate the significance of this research. The proof of concept proves to be secure, privacy-preserving and lightweight while resisting distinguishability attacks.

Keywords: searchable encryption; semantic search; aviation communication; inverted index; trapdoor; ground stations

Citation: Tahir, S.; Shahbz, M.A.; Tahir, H.; Awais, M.; Khan, F.; Tahir, R.; Saeed, S.; Almuhaideb, A.M. A Novel Approach to Reduce Breaches of Aircraft Communication Data. *Electronics* **2023**, *12*, 172. https://doi.org/10.3390/electronics12010172

Academic Editors: Andrei Kelarev, Dimitra I. Kaklamani and Flavio Canavero

Received: 23 November 2022
Revised: 17 December 2022
Accepted: 26 December 2022
Published: 30 December 2022

1. Introduction

In modern wireless communications many technologies are integrated to enable air traffic communication systems. The transition towards seamless connectivity, digitization and integration of modern wireless technology systems in civil aviation can result in cybersecurity attacks. Existing aircraft communication data are insecure as security is not an integral part of aviation systems, which poses a severe threat to the data.

Many vulnerabilities have been found in air traffic communication due to the existence of insecure channels, between an aircraft and ground control systems. While research is ongoing to secure communications in this important area, it is worth noting that risks

associated with data at rest in base station remain unaddressed. The audio files containing communications between the Aircraft Pilot (AP) and Ground Controller (GC) are kept unencrypted, leading to a severe cybersecurity risk for Civil Aviation. This design weakness needs to be addressed because security and privacy can be exploited. It has been observed in recent incidents that Civil Aviation is prone to several cybersecurity attacks [1]; in August 2008, a malware, trojan22, blocked the computer system of flight No. 5022 of a Spanish aircraft. This exploitation through trojan made the aircraft unable to receive and generate alert messages, which could have caused the airplane's collision and led to the loss of lives of 154 passengers. Similarly, in 2009, 48,000 personnel files of the Federal Aviation Administration (FAA) were accessed by an unknown adversary. In 2011, there was another interruption in computer services, which resulted in significant delays faced by travellers. The same incident happened in 2013 at Istanbul Ataturk and Sabiha Gokcen airports, in which passport control systems were compromised, resulting in the delays of several flights. In 2014, flight MH370, a Boeing 777 Malaysian airplane, disappeared from the radar. A conspiracy theory behind this incident is that the plane was hacked remotely. Similarly, in 2014, Iranian hackers claimed that they had attacked the computer systems of 16 countries, including Pakistan, Saudi Arabia, South Korea, and the United States.

It can be observed that several attacks are possible purely due to the exploitation of data stored on computer systems holding essential Aviation Data (AD). These exploits lead to a high priority compelling case to rethink data storage for the aviation industry. Starting with data at rest, a possible solution is to store the data in the cloud. The infrastructure offers ubiquity and on-demand availability of resources owing to which it can house large amounts of sensitive and nonsensitive data. The cloud is a favoured platform due to its inherent features of elasticity and scalability. However, the need to prevent sensitive data from being accessed by unauthorized users still remains, which is addressed through the encryption of data. The cloud service provider is considered a semi-trusted entity, i.e. honest but curious about the stored data. Due to encryption the data are secure from both insiders and outsiders of the cloud. Since the data are encrypted, the cloud operators cannot extract any information from the sensitive data. Thus there is need for a scheme that facilitates searching over the encrypted documents stored on the cloud [2]. Furthermore, by using the conventional encryption mechanisms that rely of deterministic operations, it may still reveal the search patterns to the cloud. This promotes the need to propose searchable encryption techniques for the aviation sector that resist distinguishability attacks (keyword-trapdoor and trapdoor-index).

This forms the motivation of this research. Thus the focus of this study is to propose a novel approach that secures aviation data while having minimum impact on its usability. This is achieved through presenting a novel semantic-based searchable encryption mechanism that resists distinguishability attacks and preserves search pattern leakages.

Semantic-based searchable encryption can address this issue by extracting files that contain the most semantically related keywords [3] out of the encrypted data stored in the cloud. Semantic-based searchable encryption has been demonstrated to be exceptionally useful at the word and the content level, both in data recovery and in semantic memory research [4,5]. Thus, by discovering the similarity of documents with one another and their effect on ordinary aeronautics correspondence, semantic-based searchable encryption can thwart cyberattacks.

This detailed study makes the following contributions:

- As a novel contribution, we introduce a semantic-based searchable encryption scheme for the aviation sector, where colossal metadata exists, and no security and data searching mechanism exists. Semantic search on encrypted data enhances the efficiency and security of the data and makes the retrieval process more manageable.
- Our system will remove traditional keyword searching and provide a new way of searching based on the roots of the words. A probabilistic trapdoor scheme is presented to remove the possible threats and attacks aimed at exploiting the trapdoor linkability. A probabilistic trapdoor is a randomized search query generated in such a way that,

if the same keywords or roots are searched repeatedly, a new search query will be generated every time. Hence the proposed scheme will provide keyword-trapdoor and trapdoor-index indistinguishability to eliminate the search pattern leakages while enabling semantic search.

- A proof of concept prototype has been implemented and thoroughly tested over an actual aviation dataset collected for this research.

The paper is structured as follows: in Section 2 a summary of wireless technologies in air traffic communication between aircraft, ground stations, and satellites has been presented. This discussion leads to the system model. Section 3 discusses the level of research already carried out. Section 4 describes the preliminary concepts. Section 5 discusses the proposed scheme and correctness and soundness requirements. Section 6 present the security definitions for our proposed scheme, followed by Section 7, which formally presents the proposed scheme. Section 8 presents the computation analysis and performance evaluation. Section 9 concludes the paper by presenting the future work.

2. Wireless Technologies in Aviation

This section summarizes the wireless technologies operating in air traffic communication containing aircraft, ground stations, and satellites. Figure 1 presents an aircraft communication model. The arrows specify the route of the communication between various entities. State-of-the-art wireless communication technology is being used in the aviation industry, with particular focus on the collaboration between types of machinery and their communication with different aircraft components.

An attempt has been made to incorporate all used technologies when focusing on system design. According to their uses, there are two phases: air traffic control and detail sharing services. Air Traffic Control (ATC) standards allow conversation between controllers and aircraft pilots. This includes Primary Surveillance Radar (PSR), Automatic Dependent Surveillance-Broadcast (ADS-B), Secondary Surveillance Radar Mode A/C/S (SSR), VHF, Multilateration (MLAT), and Controller Pilot Data Link Communication (CPDLC), which are applicable throughout the phases of the aircraft operation. At the same time, the consumption of satellite communications (e.g., CPDLC) is required for the smooth flight operations and air traffic control. Detail-sharing services provide a common data exchange platform for varying weather conditions and air movement information through Traffic Alert and Collision Avoidance System (TCAS), Aircraft Communication Addressing and Reporting System (ACARS), Flight Information System Broadcast (FIS-B), and Traffic Information System Broadcast (TIS-B). In 1978 ACARS was used to carry Aeronautical Operational Control (AOC) thereby facilitating the reporting of aircraft OOOI (out of the gate, off the ground, on the ground, into the ground) events. These systems enable the transmission of short messages containing phrases, codes and unique identifiers (as per the rules of the International Civil Aviation Organization (ICAO)).

Although there are significant divisions between airline structures around the globe, vulnerabilities can often be attributed to technological implementations; thus, we classify technology appropriately into two categories: (a) ATC contains technologies that upkeep air traffic services. This consists of pilots, controllers, and technology to monitor air traffic; (b) Detail sharing services is a technology that provides pilots statistics for alerts regarding conditions sich as traffic and weather information. The technologies serve a unified core purpose; thus, a deeper look at the whole system is required, hence we refer readers to [6].

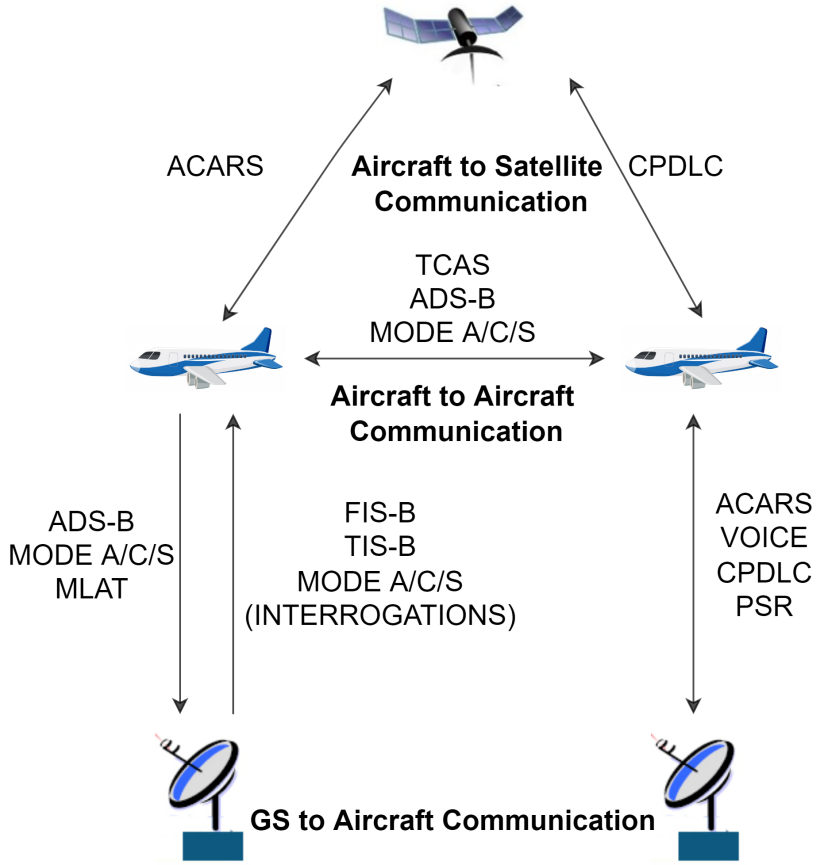

Figure 1. Aircraft Communication Wireless Technologies.

System Model

The system model comprises a pilot/ airplane, ground station (GS), and personnel. In the proposed system, the data stored on the Ground Station (GS) are communication between the Aircraft Pilots and GS Personnel, flight details, or passenger information. Therefore, securing all this information placed openly in the Ground Station is essential.

A client-server model will be used in the proposed system where the GS personnel functions as client and wishes to outsource Aviation Data (AD) to the cloud. Aviation Data (AD) contains many plaintext documents (PT Doc) and audio files that must be securely stored in the CSP. The GS personnel indexes the document (ID) based on keywords in the form of roots associated with the identifiers of Aviation Data. The secure ID and the encrypted data are stored in the CSP. Whenever the GS personnel needs a specific document, he/she would generate a trapdoor Q_w and then forward it directly to the CSP. This trapdoor is then used by the CSP to find the exact identifiers of the Aviation Data from the index of the document (ID) and returns all those documents associated with the generated trapdoor. Whenever the query is released from the GS Personnel to retrieve the required keyword's data segments, the CSP will consider this specific index of the document (ID) as a lookup table. The search process identifies several documents associated with the search query. The CSP returns the identified data segments to the GS personnel, which decrypts the data locally. There may be data users who would be willing to retrieve relevant data in some instances. The data users may share the keyword with the GS personnel to generate the trapdoor, send it to the CSP, and obtain the results. The identified data segments will be shared with the data user by the GS personnel. Figure 2 depicts the system model. To have a deeper look into our innovation, the system model and its description may be read in conjunction with the system schematic flow presented in the Section 5.

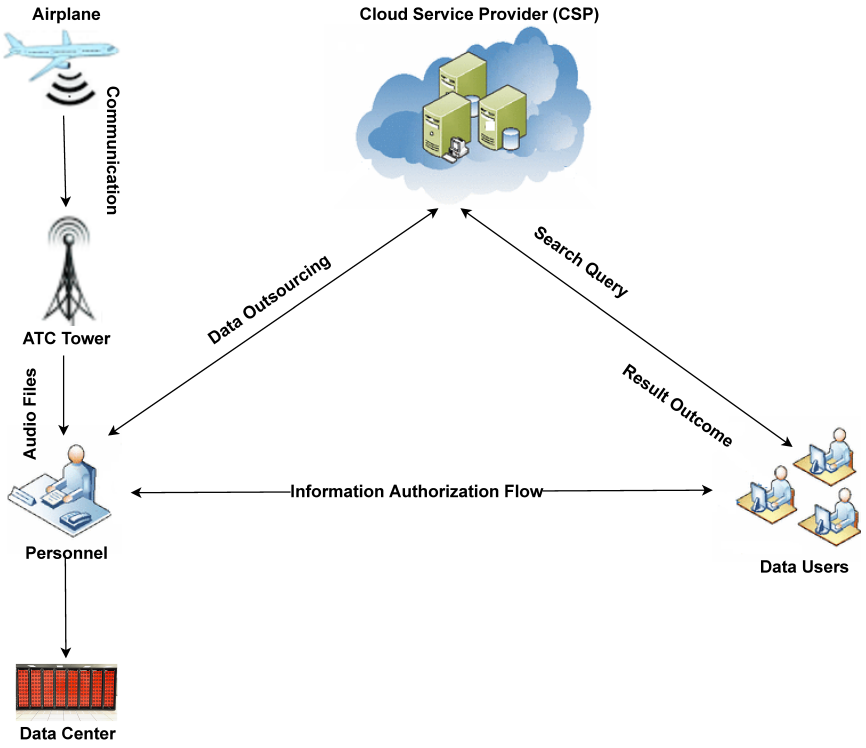

Figure 2. System Model.

3. Literature Review

3.1. Aviation Security

This section highlights the security issues in the aviation sector and revisits the existing literature on aviation security.

Tommaso et al. in [1] highlight cyber-attacks performed on the Civil Aviation Systems. A Milanese firm specializing in surveillance software suffered a cyber attack followed by the online distribution of products sold only to police and intelligence agencies. They discuss all cyber-attacks and vulnerabilities found in the systems placed in an aircraft and highlight those easily exploitable systems. However, the limitation of their paper is that it only describes the possible physical attacks which can occur to the aircraft but do not describe any leakage of wireless communication, which could lead to security and privacy risks.

Xu Lu in [7] describes the activity of Aircraft Communication Addressing and Reporting Systems (ACARS), a digital data link system used to transmit short messages by radio or satellite between aircraft and ground stations. As ACARS transmission messages use plain text, anyone can monitor and decode them. The author built a new certificate-less key isolation signature and encryption scheme that does not need any digital management of the user's public key and can effectively manage it to reduce the harm caused by leakages. The limitation of their work is that it will work only for digital data, particularly for concise messages.

Krishna Sampigethaya in [8] has focused on the security threats faced by an airplane operating under air traffic systems by investigating the risks and presenting their mitigation strategies. The proposed model comprises a cyber-physical risk assessment framework to achieve system security that takes into consideration impact of compromised air traffic control data on flights. The technique detects compromised air traffic data through the physics of the aircraft physics, onboard data to protect flight position confidentiality, leveraging flight patterns, airspace restrictions, and information hiding techniques that prevent advanced threats. The research describes the risk assessment of different types of threats, attacks, and vulnerabilities related to the aircraft. The limitation of the research is

that it only describes the possible physical attacks that can occur but does not describe any leakage of information from the end-user, leading to security and privacy breaches.

Ehud Ezroni and Gilad Dafna in [9] minimize the miscommunication gap between the air controllers and pilots and present a system for providing unambiguous communication between an aircraft pilot and a communicator. The system includes a communication link and a visually sensible display providing an output indication. The speech synthesizer receives a piece of digital information from the speech recognizer. The limitation of this study is that it is used only for removing miscommunication or unambiguous errors from the communication but does not describe any leakage of information from the end-user, which can cause confidentiality and privacy concerns.

Martin Strohmeier and Kellogg College in [10] interviewed 242 international aviation experts aiming to analyze the community's knowledge regarding the awareness and potential impact to aviation due to insecure wireless system technologies. This systematic analysis would result in proposing novel mechanisms to improve the security and safety of air traffic communication networks. The paper gives a detailed overview of the technologies, vulnerabilities, and existing attacks, but does not describe any information leakage at the ground station.

Krishna et al. in [11] have presented an indepth study of security issues and throw light on the complexities associated with protecting safety service IPv6 air-to-ground communication. ATN/OSI and ACARS protocols are currently used for air-to-ground communication. This protocol has transitioned to IPv6 from existing ACARS and ATN/OSI. The authors have described the architectural goals that should be considered while during the deployment of IPv6 safety services between air and ground communications. This technique is particularly challenging to implement since there is need to change the entire aviation infrastructure worldwide.

By analyzing the existing literature, it is obvious that the state-of-the-art mechanisms within the aviation industry are prone to security and privacy breaches. These breaches lead to a compelling case for the aviation sector to design robust, secure, and privacy-preserving schemes. Thus, this paper proposes the design of a novel searchable encryption technique for aviation sector that allows semantic searching capabilities. The presented scheme is novel considering it provisions semantic analysis while preserving keyword-trapdoor and trapdoor-index indistinguishability over a real-world aviation dataset.

3.2. Searchable Encryption

In [12], Song et al. were the foremost to introduce searchable encryption thus they also established formal security definitions for searchable encryption. The authors followed a layered approach where symmetric encryption was used to encrypt the data, and linear scanning over the documents was performed. The limitation of this system was that it could only operate for restricted document sizes.

Eu-Jin Goh in [13] proposed an index-based searchable encryption approach. The author adopted a Bloom filters-based approach to generate an index against each file and a list of keywords. In his approach, the author created a searchable index for each file containing trapdoors of all unique terms. This approach made the search operations generically quicker and well adapted to big data scenarios. The limitation was that it had false positives due to the underlying choice of data structures. Curtmola et al. in [14], described a new approach that provided a solution to the false-positive results due to Bloom filters. The author kept a single hash table index for all documents containing entries and a trapdoor of a word mapped to a set of file identifiers for the documents in which the keyword appeared. This method is comparatively faster and takes constant time to access related files. However, the limitation of their approach is that it is less secure and reveals different facets of data to the potential attackers.

Tahir et al. in [15] present a lightweight searchable encryption scheme that employs probabilistic trapdoors. Their scheme is lightweight, and uses modular inverses. Although, the work is able to prevent distinguishability attacks, the scheme lacked the capability of se-

mantic analysis, hence the index table would contain all the keywords in the dictionary (not only the root words). The authors also presented novel security definitions for searchable encryption that are widely adopted in the literature.

Grigoris Karvounarakis et al. in [16] introduced the concept of searching semantically. The author described the Resource Query Language (RQL) technique to formulate queries. The method used query modification and an ontology structure to define related terms to achieve their semantic nature. The ontology structures often need to be significant and custom-tailored to their specific use cases or domain, making them very dependent and unadaptable to different areas. So, using some formal language or form leads to exact searching that is inappropriate for naïve or everyday users.

The authors in [5], present a semantic searchable encryption technique. The authors use a stemming algorithm for identifying the roots. The authors have built their scheme over [14], hence they provide the same security guarantees and do not hide the search pattern. Liu et al. in [17] propose an efficient semantic and authorized search mechanism over the cloud. The authors have used conceptual graphs for the semantic analysis. The proposed scheme is independent of the mechanism used to calculate the scores of the sentences. The authors have claimed to propose an efficient variant however there is a lack of experimentation over a real-world dataset to verify their claim. Furthermore, the authors claim to achieve indistinguishability but there are no security proofs presented.

In [18], Sun et al. have proposed a scheme through a combination of semantic searching with searchable encryption. To capture the semantics of queries the authors have used data mining and indexing method over encrypted file metadata. Stemming methods in terms of index and query provide more general matching. These approaches used keyword-based searching methods, often returning irrelevant results or false positives, failing to disambiguate unstructured text.

In [19] Mingwu et al. proposed a searchable encryption technique that supports privacy-preserving fuzzy multi-keyword search (SE-PPFM) in cloud systems, which builds by asymmetric scalar-product-preserving encryption and Hadamard product operations, which effectively improves the efficiency of the scheme but using asymmetric encryption slows down the process and risks widespread security compromise. In [20] Yang et al. have created a quick multi-keyword semantic ranking search scheme. The authors have introduced the concept of domain weight scoring in document scoring and semantically expanded search keywords to improve the accuracy of the document index. Moreover, the document vector has been divided into blocks in an effort to filter out many irrelevant documents, thereby improving the scheme's efficiency.

In [21] Redwan et al. present a novel approach that facilitates searchable encryption of large EHR systems using Attribute-based Encryption (ABE) and multi-keyword search techniques. The framework outsources key search features to the cloud side, and the system can perform keyword searches on encrypted data while significantly reducing network bandwidth and client-side computation costs. However, the problem with the attribute-based encryption (ABE) scheme is that the data owner needs to use every authorized user's public key to encrypt data. The application of this scheme is restricted in the real environment because it uses the access of monotonic attributes to control the user's access to the system.

A recent research [22] has studied an effective fuzzy semantic searchable encryption scheme (FSSE) which is able to support the searching of multi-keywords over encrypted data stored in the cloud. However, a drawback is that the fuzzy rationale is not constant. So the outcomes are dependent on suspicions and may not be generally acknowledged. Approval and verification of a fuzzy information-based framework need comprehensive testing with equipment.

Payal et al. in [23] present KeySea, a keyword-based searching scheme that allows searching in attribute-based encrypted data. While searching among the documents in search for target keyword(s), maintaining receiver anonymity and ensuring data privacy are essential features in healthcare, bureaucracy, and social engineering applications. However,

the problem with the attribute-based encryption (ABE) scheme is that the data owner needs to use every authorized user's public key to encrypt data. In [24] Awais et al. have used a single reader single writer searchable symmetric encryption technique to hide search, size, and access patterns from an adversary. In there model, the cloud is considered semi trusted and the cloud functions as a storage container only. The trapdoor generated from the client side is also probabilistic; ensuring that a single keyword can have multiple encrypted forms instead of just a single one (as in the case of a deterministic technique). The authors do not enable semantic search over the encrypted data.

Recent researches [25,26] have presented ranked semantic-based searchable encryption schemes. The presented schemes use inverted index tables and TFxIDF for achieving ranked searching. The scheme in [25] does not preserve the trapdoor unlinkability, hence it is not privacy preserving. However, the scheme presented in [26] has the feature of trapdoor unlinkability but it has not been verified against the widely adopted security definitions of indistinguishability presented in [15].

The readers are also referred to the following papers that present surveys on searchable encryption techniques [27–31].

4. Preliminaries

4.1. ATC Speech Corpora

In air traffic communication (ATC), most communications are in English, but the speakers are primarily multilingual and non-natives. Speech quality is also poor due to noise/static and syntax limited to International Civil Aviation Organization (ICAO) phraseology. The only benefit is that the vocabulary is limited because aviation has their phrases, codes, and unique identifiers according to the ICAO rules and procedures. Hence, to determine the similarity between this specific Aviation Data (AD), we need to find the semantic relation between the phrases, codes, abbreviations, and the unique identifiers used in aviation communications. The semantic search over Aviation Data (AD) in an encrypted domain can address almost all the issues discussed in Section 1.

4.2. Semantic Search and Stemming Algorithm

Semantic search is an information retrieval process, and there are many information retrieval techniques and algorithms available in the literature, including HS (Hierarchy Spread) algorithm, BDOS (Bi Direction One-Step) algorithm, and tree-based concept hierarchy [32–34]. The most common algorithm used to perform the semantic search is the stemming algorithm. A stemming algorithm generates roots of phrases, codes, abbreviations, and unique identifiers, which could be challenging in aviation communication. A stemming algorithm is a morphological way of analyzing the words having approximately the same meaning and, in some cases, those words having the same roots related to the same set of words. Here, there is a need to clarify that the purpose of the stemming algorithm is not to identify the correct meaningful roots of the words but to extract the exact root of the same relatable keywords.

Many stemming algorithms commonly use n-grams of Mayfield and McNamee's, Hidden Markov Models [35,36] and Porter Stemming algorithms. Our proposed approach uses Porter Stemming algorithm to enable extraction of roots of keywords. The Porter Stemming algorithm has a less error rate, hence it is most suitable for this research. Readers are also referred to [37] that presents a comparison of different stemming algorithms. Mostly semantic search is performed on the plaintext, resulting in high performance and efficiency in the information retrieval [38]. We intend to perform the same task and achieve similar performance and efficiency over the Aviation Data (AD) but in the encrypted domain. So, we combine the stemming algorithm technique with the Searchable Encryption scheme for encrypted data hosted on the public cloud platform.

Regarding semantic search, there is a need to extract all the keywords present in the dataset. Then, a root is generated by forming a group of those keywords related to each other. For example, if we have keywords such as squawk, squawked, squawking, and

squawks, they all can make one group, and the root of this group will become "squawk". In another example of Aviation data (AD), the keywords maintain, maintains, maintaining, and maintained can make one group. The root of this group will become "maintain", similarly, the keywords gamet, sigmet, airmet, and volmet can make one group, and the root of this group will become "met". The Figure 3, all the related keywords have the same roots, and when we organize all the related keywords, we have to store the roots of the keyword in the secure index table. To generate a query, the user now searches his query with the roots instead of keywords. So, we can say that we are performing a root-based/semantic-based searchable encryption scheme instead of traditional keyword-based searching.

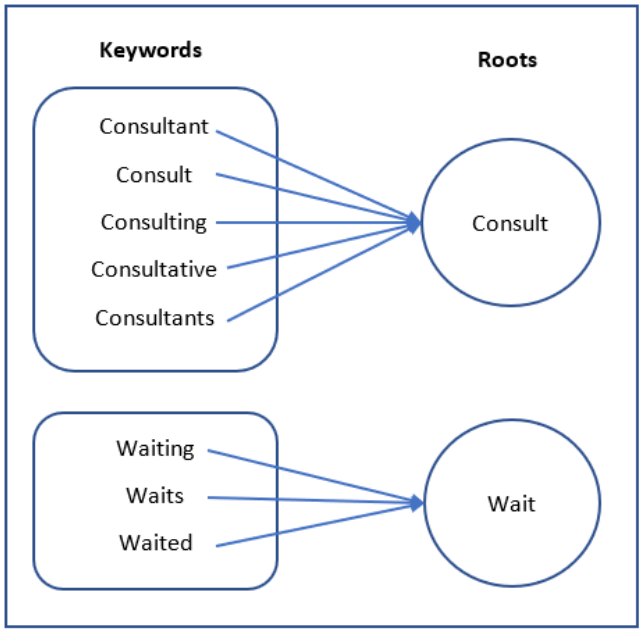

Figure 3. Stemming Process.

5. Scheme Overview

This scheme consists of two sides, one is client-side, and the other is the server. On the client side, we have performed all functions except searching and document retrieval, which is performed on the server-side. The first presented challenge is that of data format since the data set comprises of only audio files containing the communication between the aircraft pilots and ATC controllers. So these audio files need to be transformed into transcriptions over which we implement our proposed scheme.

In our proposed scheme, the transcriptions are used to extract the keywords, and then the stemming process on these keywords is applied to generate the roots of most similar words in the Aviation Data (AD). It is necessary to generate a secure index for the roots and their corresponding plaintext documents. After that, we have to encrypt the Aviation Data (AD) secure index and send these two encrypted segments to the Cloud Server (CS), as shown in Figure 4. Hence, whenever any Ground Station (GS) personnel wants any document from the encrypted set of data placed at the server-side, they need to generate a probabilistic trapdoor. The trapdoor is based on probabilistic encryption by considering every third person as an eavesdropper or intruder during the communication between GS personnel and the cloud server (CS). To prevent the system from being attacked by an adversary, the concept of randomization has been implemented in the scheme when the trapdoor generates. So, if the exact query is searched twice, different trapdoors are generated. By using the randomization process, we can achieve more security against the attacks because, in that scenario, the attacker cannot extract the length of the trapdoor or the exact trapdoor itself. Nevertheless, the attackers fail to extricate the exactly generated

trapdoor of any searched query. So this secure trapdoor is sent to the cloud server (CS) to retrieve the desired document.

On the server-side, the server takes this trapdoor and executes its searching mechanism by using the secure index sent earlier to the server with the encrypted data. When the server receives the plaintext documents (*PT*) identifiers from the secure index, it will go to the database where all the encrypted plaintext documents (*Enc_PT*) are stored against the generated trapdoor and sent back to the GS personnel.

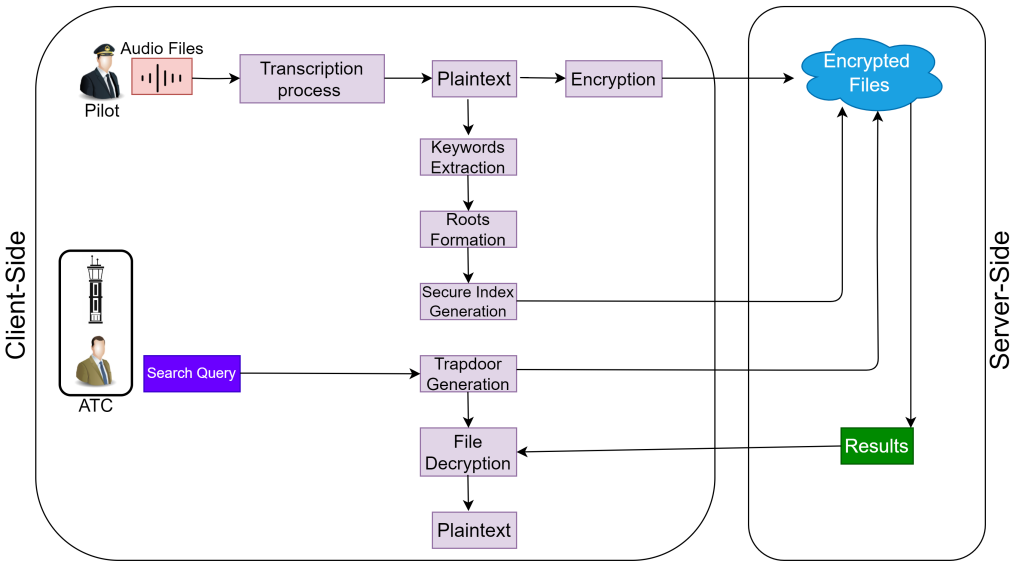

Figure 4. The system schematic flow.

Correctness and Soundness

The proposed scheme is correct and sound if, for our security parameter λ and the master key K, the search outcome against the query trapdoor Q_w using the secure index should retrieve the correct document identifiers and induce no false positives.

(1) If the searched keywords k_w belongs to the search outcome PT_i, then the following should hold true with an overwhelming probability.

$$Result = S_O(ID, Q_w) = (PT_i) \ \ where \ \ 1 \leq i \leq n \tag{1}$$

(2) If the searched keywords k_w do not belong to the search outcome PT_i, then the following should hold true with an overwhelming probability.

$$Result = S_O(ID, Q_w) = 0 \tag{2}$$

Table 1 highlights the important abbreviations and notations used in the proposed scheme.

Table 1. Notations & Abbreviations.

AD	Aviation Data
GS	Ground Station
CSP	Cloud Service Provider
CS	Cloud Server
ID	Index of the Document
PT Doc	Plaintext Documents
Enc_PT	Encrypted Plaintext Document
ATC	Air Traffic Communication
ICAO	International Civil Aviation Organization
CSP	Cloud Service Provider

Table 1. *Cont.*

λ	Security Parameter
K	Master Key
Ω	Random Number
K_w	Keyword
Q_w	Trapdoor
Dec(Enc_PT)	Decryption of encrypted plaintext
S_o	Search Outcome
R	Roots extracted from keywords
R_list	List of Roots/keywords
st_A	State of Adversary
$A_i(st_A)$	Current state of Adversary
$TR_1 \dots TR_i$	Range of Trapdoors
TR_c	Trapdoor Calculated after query generated
A_{m+1}	Next state of Adversary
C'	Trapdoor calculated by the adversary
SE	Searchable Encryption

6. Security Definitions

To achieve the security and privacy of aviation data, we must take the indistinguishability definitions for searchable encryption into account. Definitions proposed in [15] are referred to here and used to analyze the security and privacy of the scheme presented in this paper. These security definitions are well accepted and referred to in the literature [24].

6.1. Keyword-Trapdoor Indistinguishability for Searchable Encryption

Keyword-trapdoor indistinguishability states that whenever the searching process should happen, and the user generates a query for the keyword, the trapdoor should be indistinguishable for the keywords searched repeatedly. This should hold even if an adversary A has a history. If the adversary A wants to retrieve the respective keyword in polynomial time, he needs to extract large amounts of data.

6.1.1. Description

Referring to the scenario in our proposed scheme, the challenger generates the index table ID containing the roots of all the keywords extracted from the plain text dataset. Adversary A would select a root R and send this root to the challenger C. Then C will create the encrypted trapdoor for that root and send it back to the adversary A. This process will be repeated so that the adversary A can obtain enough encrypted trapdoors.

Soon after, the adversary A will have the option to choose two roots R_1, R_2 belonging to R and send these selected roots to the challenger C. Now, the challenger C will send a trapdoor R_i corresponding to that root, where i is the outcome of a fair coin tossed. Afterwards, the adversary must decide the exact root corresponding to the respective root in polynomial time. If the adversary can extract that root with the probability greater than ½, then it means that the adversary wins, and our scheme does not hold the property of keyword-trapdoor indistinguishability. If the adversary does not satisfy the property as mentioned above, the challenger is considered a winner with the property of keyword-trapdoor indistinguishability.

Keygen, Encryption, Index Generation, Trapdoor Generation, Search Outcome, Decryption are considered the phases of a searchable encryption scheme having the list of roots $R = R_1, R_2 \dots, R_n$ extracted from the keywords $kw = kw_1, kw_2 \dots, kw_n$, from the plaintext dataset $PT = (PT_1, PT_2, \dots, PT_n)$ and having the λ as a security parameter. Considering a probabilistic function of Keyword Trapdoor (SE, A) (λ):

$$(K) \leftarrow Keygen(\lambda)$$
$$(ID) \leftarrow Build\ Index\ (Enc(PT), R_list$$
$$for\ 1 \leq i \leq m$$
$$(st_A, R_i) \leftarrow A_i(st_A), (TR_1 \dots TR_i)$$

$(TR_i) \leftarrow$ *Build Index* (KR_i)
$C \leftarrow^{\$} \{0, 1\}$
$(st_A, R_0, R_1) \leftarrow A_0(\lambda)$
$(TR_c) \leftarrow Build_T rap\ (K, \Omega, R_c)$
$C' \leftarrow A_{m+1}(st_A, TR_c)$
$(T'R_i) \leftarrow Build_T rap\ (R_j)\ ; j \in N$
if $C' = C$, *output 1 other wise output is 0.*

st_A is the representation of string which holds the adversary's state. The concept of keyword-trapdoor indistinguishability will only be satisfied if:

$$\Pr\Big[\ Keyword_{Trapdoor_{(SE,\ A)}}(\lambda) = 1\Big]\ \leq\ \frac{1}{2} + negl(\lambda)$$

and it highly depends on the choice of challenger C which is a fair coin toss. This is explained with the help of a game.

6.1.2. Game 1

Suppose that there are at most kw keywords where $kw = kw_1, kw_2, \ldots, kw_n$ in the plaintext dataset $PT = (PT_1, PT_2, \ldots, PT_n)$. The game comprises of three phases: Phase 1, Challenge Phase and the Outcome Phase, and it is played between an adversary A and a challenger C.

Phase 1: The adversary A selects and sends a root R to the challenger C. The challenger C returns adversary A a trapdoor corresponding to R. This continues between A and C for a while.

Challenge Phase: A selects two distinct roots $R_0, R_1 \in R$ belonging to two distinct keywords and sends them to C. C in response tosses a fair coin $b \leftarrow \{0, 1\}$ and generates two trapdoors TR_b corresponding to the values of b, such that the challenge has been completed. If the adversary wishes, the Phase 1 is run again allowing the adversary to generate trapdoors for the same keywords that were searched previously.

Outcome Phase: A is given the generated trapdoors TR_0, TR_1 and has to guess and output b. If the adversary is able to distinguish between the two trapdoors such that it can identify the index entries corresponding to the keywords, the adversary wins, otherwise the challenger wins and the scheme provides trapdoor-index indistinguishability.

6.2. Trapdoor-Index Indistinguishability for Searchable Encryption Scheme

Another important aspect of the Searchable Encryption Scheme is Trapdoor-Index Indistinguishability. This property is used to hide or protect information sent from the user to the cloud server and has a chance to be exploited. This information includes trapdoor, index table, and search queries. So there is a need to intricate these statistics so that no information is exposed related to the index table that could lead to the search query identification. It states that if a single keyword searches repeatedly, the trapdoor generated must be indistinguishable enough even if the adversary maintains a history of the past trapdoors and corresponding index table entries.

6.2.1. Description

In this scenario, the challenger C generates an index table with the documents' names along with the roots and keywords extracted from each plaintext dataset PT_i in PT where PT represents a plain text dataset. The challenger sends the list of trapdoors created from the roots R to the adversary and tosses a fair coin. After that adversary selects two roots (R_1, R_2) and sends them back to the challenger, now the challenger will provide the trapdoor associated to that root R_c, and the adversary will have to identify the corresponding index value and send the output bit c.

Keygen, Encryption, Index Generation, Trapdoor Generation, Search Outcome and Decryption are considered the phases of a searchable encryption scheme having the list of roots $R = R_1, R_2, \ldots, R_n$ extracted from the keywords $kw = kw_1, kw_2, \ldots, kw_n$, from the

plaintext dataset $PT = (PT_1, PT_2, \ldots, PT_n)$. Adversary A is going to exploit our communication channel.

By considering a probabilistic function of index trapdoor *(SE, A)* (λ):

$(K) \leftarrow Keygen(\lambda)$
(ID) $\leftarrow$ *Build Index (Enc(PT),* R_list
for $1 \leq \mathrm{i} \leq m$
let $I' = I[0][x]$
let $R = \{R_1, R_2, \ldots, R_i\}$
$(st_A, R_i) \leftarrow A_i(st_A, TR_1 \ldots TR_i)$
$(TR_c) \leftarrow Build_Trap\ (K, \Omega, R_c)$
$C \leftarrow^{\$} \{0, 1\}$
$(st_A, R_0, R_1) \leftarrow A_0(\lambda)$
$(TR_c) \leftarrow Build_Trap\ (K, \Omega, R_c)$
$C' \leftarrow A_{m+1}(st_A, TR_c)$
$(T'R_i) \leftarrow Build_Trap\ (R_j); \mathrm{j} \in \mathrm{N}$
if $C' = C$, output 1 other wise output is 0.

st_A is the representation of string which holds the adversary's state. The concept of keyword-trapdoor indistinguishability will satisfy if:

$$\Pr\Big[\ Index_{Trapdoor_{(SE,\ A)}}(\lambda) = 1\Big]\ \leq\ \frac{1}{2} + negl(\lambda)$$

and it depends on the choice of challenger C which is a fair coin toss. This is explained with the help of a game.

6.2.2. Game 2

Suppose that there are at most kw keywords where $kw = kw_1, kw_2, \ldots, kw_n$ in the plaintext dataset $PT = (PT_1, PT_2, \ldots, PT_n)$. The game comprises of three phases: Phase 1, Challenge Phase and the Outcome Phase, and it is played between an adversary A and a challenger C.

Phase 1: The adversary A selects and sends a root R to the challenger C. The challenger C returns adversary A a trapdoor and the index table entry I corresponding to R. This continues between A and C for a while.

Challenge Phase: A selects two distinct roots $R_0, R_1 \in R$ belonging to two distinct keywords and sends them to C. C in response tosses a fair coin $b \leftarrow \{0, 1\}$ and generates two trapdoors TR_b and index entries $I[0][R_b]$ corresponding to the values of b, such that the challenge has been completed. If the adversary wishes, the Phase 1 is run again while allowing the adversary to generate trapdoors for the same keywords.

Outcome Phase: A is given the generated trapdoors TR_0, TR_1, index table entries $I[0][R_b]$ and has to guess and output b. If the adversary is able to distinguish between the two trapdoors such that it can identify the index entries corresponding to the keywords, the adversary wins, otherwise the challenger wins and the scheme provides trapdoor-index indistinguishability.

7. Proposed Scheme

The proposed scheme and its detailed analysis is presented in this section. As discussed previously, the proposed scheme comprises six polynomial-time algorithms which are mentioned in detail in this section.

7.1. KeyGen Phase

The Algorithm 1 is utilized to generate a master key K for the client. λ is viewed as a security parameter and the client generates a master key and random number as K, $\Omega \leftarrow (0, 1)^{\lambda}$. The generated key is used with AES-256 bits and only in possession of the client.

Algorithm 1: Keygen

Input: A security parameter λ
Generate Key K , $\Omega \leftarrow (0,1)^{\lambda}$
Output: Master Key K and Random number Ω

7.2. Encryption Phase

In the Algorithm 2, all documents in the dataset have to be encrypted which is achieved through AES-256 bits. After encryption, the documents are sent to the CS.

Algorithm 2: Encryption

Input: Plain Text Dataset (*PT*);
Output: Encrypted Plain Text files;
for *PT* in *PT_set* do:
    read *PT*;
    encrypted ← encrypt*(PT,K)*
    encrypted_PT_set.append(encrypted)

7.3. Index Generation

The index generation phase (Algorithm 3) is crucial because it enables the searching capability and the accuracy of our searching phase is dependent on the index table. In this phase, the tokenization process isolates the keywords from the rest of the data. After that, the punctuation marks and all stop words are removed from the data set because they are not valuable for the search phase. Now stemming is performed on these extracted keywords, which are gathered after tokenization. Stemming retrieves our desired roots and is used in our searching phase as a keyword. Now we take the hash of all these newly generated roots or keywords iteratively and store in parameter "a" Parameter "b" uses to store the AES encryption of all the roots iteratively. Compute the inverse of these two parameters "a" and "b" and store the multiplication of these inverses in parameter c'. The value c' is stored in the 1st column of our Index of the Documents "*ID*", whereas the AES encryption of the plain text "*PT*" of our aviation data are stored in the 2nd column of the index table "*ID*". Our secure index is now accessed for document retrieval whenever a search query is generated. This is also explained through a toy example presented in Figure 5.

Algorithm 3: Index Generation (*BuildIndex*)

Input: Document ids,keywords k_w
Output: Inverted Index (ID);
for *PT* in *PT*_set do:
    tokens ← word_tokenize(PT)
    keywordlist.append ← (tokens)
    for word in keyword list do:
        $a = H_K$(keyword k_w);
        $b = Enc_K$(keyword k_w);
        compute inverse of a & b and store in c';
        $c' = a' * b'$;
        saves in 1st column of ID 'col 1'
    for PT in PT_set do:
        AES(PT_id's);
        store in 2nd column of ID 'col 2'
        flist.append([col 1][col 2])

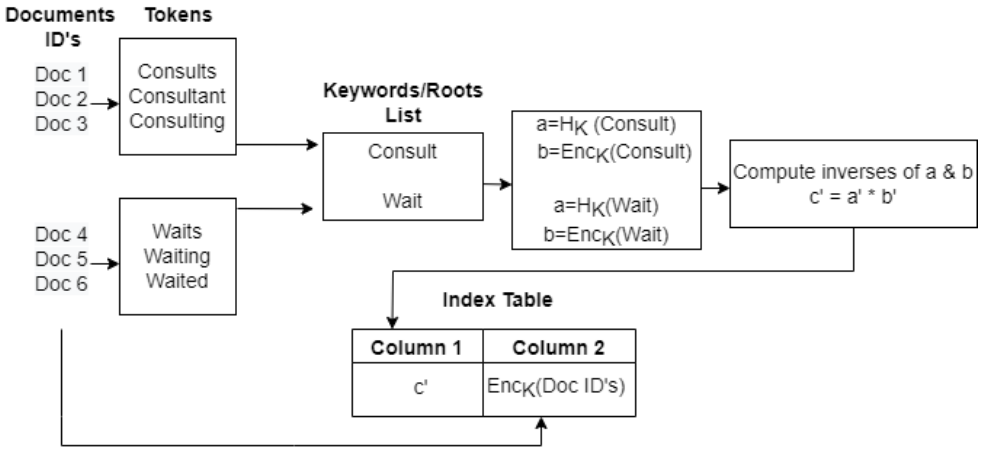

Figure 5. Toy Example-Index Generation.

7.4. Trapdoor Generation

In this phase (Algorithm 4), there is a need to generate an encrypted search query to send it to the CS for retrieving the required encrypted data segments containing similar words. For this, the user uses his randomly generated number and stores the hash of this randomly generated number in parameter d. Another parameter c is used to store the product of parameters a, b, d. The parameters a and b are the same as used in the index generation phase. Parameter e stores the cryptographic hash value of parameter d. The parameters c and e are considered search queries or trapdoors.

Algorithm 4: Trapdoor Generation ($Build_{Trap}$)

Input: Keyword;
Output: Trapdoor (c, e);
$d \leftarrow H(\Omega)$
$c \leftarrow a * b * d$
$e \leftarrow H(d)$ $Q_w \leftarrow (c, e)$

7.5. Search Outcome

Now, in the Algorithm 5, CS takes the trapdoor sent by the client and performs its searching functions. The server initializes its secure index list and performs the multiplication between c and c'. The server has already contained the value of c' in the 1st column of the index table. Then the CS takes the hash of the result after the multiplication process and analyzes whether this value is equal to parameter "e" or not. If it is equal to the parameter "e", it is considered an optimistic hit, and it can save/add the document name to that file which will be sent to the client later.

Algorithm 5: Search Outcome

Input: Trapdoor;
Output: Encrypted Documents;
Initialize a 2D array *A[ ]*
for $1 \leq b \leq$ size(ID) :
Set *a = ID[1][b]*
for number of columns in *ID* do:
    if ($e == H(c * c')$) then:
        do filename $\leftarrow$ *row [0]*
        outcome.append *A[ ]* $\leftarrow$ (Enc Plain text file)

7.6. Decryption Phase

In the Algorithm 6, the required encrypted file or retrieved file is sent to the client. Decryption will be performed using the client's Master Key K and AES decryption.

Algorithm 6: Decryption

Input: Encrypted PT Documents;
Output: Decrypted PT Documents;
1 Initialize a List decrypted_PT_set.
2 decrypted_PT_set.append(decrypt(enc(PT)))

7.7. Security Evaluation of Proposed Scheme

This section presents an analysis of the leakage profiles of the proposed scheme by identifying information easily accessible to the adversary, or we can say that our scheme's leaked information is either encrypted or plaintext, relevant or irrelevant according to the content. Our scheme has the Index of the document (ID), our search query Q_w, and the search outcome S_o revealed to the adversary. Moreover, the adversary can do whatever he wants to that leaked information by using standard models; we are not limiting the adversary to any technique but restricting the execution of time as all the processes should be performed in the polynomial time.

Leakage L_1: The first leakage relates to the index of the document (ID), which is exposed to all the entities including the user, the cloud server (CS), and the adversary A. Hence the 1st leakage L_1 is defined as:

$$L_1(ID) = \{(c') \| \ AES(PT\_id's)\} \tag{3}$$

Leakage L_2: The second leakage relates to the search query or trapdoor which is generated when a user is in search of a specific Root (R), which is also exposed to all the entities which includes the user, the Cloud server (CS), and the adversary A can be placed. The second leakage L_2 is thus defined as:

$$L_2(Q_w) = \{(c = a * b * d) \| \ e = H(d)\} \tag{4}$$

Leakage L_3: The third leakage relates to the search outcome S_o when the search query is performed on the Cloud Server (CS). Assuming that this outcome is exposed to all entities, the 3rd leakage L_3 can be defined as under:

$$L_3(S_o) = \{AES(PT_i) \forall \ Q_w \in ID\} \tag{5}$$

7.7.1. Discussion on Leakage

We have discussed in our proposed scheme that we are using probabilistic encryption in our scheme for trapdoor generation. So, the generated search query is also probabilistic because every time the generated keyword is the multiplication of the hash of a random number, the hash of the keyword with the master key, and the encryption of the keyword. The random number used in our algorithm is newly generated every time to make the query probabilistic. So, the attacker cannot retrieve the information from that keyword and relate it to the index of the document ID because we have stored the inverses of the actual values in our index table. If we consider the worst-case scenario that the attacker has succeeded in acquiring the query generation process, the query generation in the future is still secure because it is probabilistic; that is why it shows an independent behavior every day. We can say that our scheme is secure in terms of search patterns. It is impossible to hide the access pattern in the index formation environment, although using some additional techniques and algorithms can minimize the attack ratio. So query trapdoor unlinkability and indistinguishability are not impacted by this leakage.

As mentioned above, we can assume that Leakage L_1 and Leakage L_3 may be related to the security and privacy issues of the users. However, a formal security analysis describes that these leakages do not reveal any data outsourced to the cloud server as these variables represent hash or encrypted values.

7.7.2. Informal Security Analysis

The informal security analysis will describe and map our proposed scheme with the security definitions mentioned in Sections 6.1 and 6.2, similar to [15].

Lemma 1. *It proves that Leakages L_1, L_2, L_3 described previously are secure under the security definitions mentioned in Sections 6.1 and 6.2. In which L_1 leakage linked with Index of a document ID, L_2 leakage linked with the trapdoor generated whereas the last leakage L_3 linked with the search outcome of the search query.*

Proof Sketch: In the security definitions mentioned in Sections 6.1 and 6.2, the proposed solution must be secure to prevent numerous attacks. This goal can be achieve if the index of document ID is secure and the query generation is probabilistic. By using probabilistic encryption in our scheme to generate a trapdoor from the keyword, every time whenever the exact keyword searches, a different trapdoor will be generated, and it becomes impossible for the adversary to retrieve an exact keyword from an encrypted trapdoor and this can also create a problem for an adversary to make a relationship between a trapdoor, keyword, and the index table before searching. However, if the adversary has a large enough stock of search queries and search outcomes, it may add additional help for an adversary, but it is still impossible to perform these processes in polynomial time. So, it can be concluded that the proposed system is in coherence with security definitions.

As we have earlier mentioned, the leakages are meaningless and achieve a high level of security. The reason behind this statement is that all the information related to these leakages is encrypted and hashed, and the master key K is fully secure. So, the adversary cannot regenerate the hash or decrypt the file in polynomial time. The leaked information is also encrypted with complex encryption algorithms that can make it almost impossible for an adversary to extract a piece of the original information. This technique can make deciphering impossible and secure our system against indistinguishability attacks. □

Lemma 2. *It proves that Leakages L_1, L_2, L_3 described in the previously is privacy-preserving while considering the security definitions mentioned in Sections 6.1 and 6.2. The L_1 leakage links with the index of documents in which encrypted plain text files identifiers and the multiplicative inverse of the hash of keyword with the master key and encryption of keywords stores. The leakage L_2 links with the trapdoor generated, whereas the leakage L_3 links with the search outcome of the searched query.*

Proof Sketch: It is already mentioned in the previous sections that our system is secured against the leakages L_1, L_2, L_3. Our trapdoor is generated using probabilistic techniques and is not distinguishable from the list of keywords and the index of the document because we have stored inverses of keywords in our index table. So, it states that the trapdoor Q_w which generates for any keyword k_w may not map to the index of the document with equal probability, and the result is entirely indistinguishable before searching. □

8. Computational Analysis

8.1. Dataset Description

The dataset is genuinely collected from the Civil Aviation Authority from 2019 to 2021. The dataset comprises live recordings of communications between aircraft pilots and air traffic controllers. These recordings have been converted into plaintext to perform our scheme's functions and become a total of 1020 files in number.

8.2. Implementation Details

The practical work has been performed on Python in Windows 10 for the client-side, whereas Windows server 2019 is used as the CSP. The dataset used in this scheme is a real-world dataset. To generate graphs, we used Microsoft Excel 2013 to provide the practical aspect of the scheme. This scheme used Client-Server architecture, and to fulfill this aspect, we used two machines were deployed to serve as the client end and server end. So, transfer of files, index tables, and trapdoor through the network to the cloud server will also incur a cost which has not been considered in the research.

To obtain the security of the dataset, 256-bit AES-CBC and SHA-384 are used for the cryptographic hash function. The system we used for the implementation was core i5 3rd Gen, 2.9 GHz processor, and 8 GB RAM for client-side, whereas for server-side, we use workstation Xeon (R) CPU of 2.5 GHz processor and 12 GB RAM.

8.3. Algorithmic Complexity

The asymptotic analysis is presented in this section thus proving the algorithmic performance of the proposed scheme.

The complexity is based on the number of files denoted by n, number of keywords denoted by m, hash function is represented by h and complexity of encryption function is represented by e. The complexity of roots/fragments is denoted by R and random number by r. The proposed scheme consists of six phases including Keygen, Encryption, Index Generation, Trapdoor Generation, Search Outcome, and Decryption phase. The complexity analysis of these phases are given as follows:

The complexity of the schemes are denoted by $O()$, read as "big oh". This is called the asymptotic upper bound complexity. It tells the running time of the algorithm when the size and number of input parameters is maximum. It is a relationship among input parameters and the required time to process those input parameters. In the case of proposed scheme, Keygen and Decryption phases are fairly constant functions and require the same amount of time. Therefore, the time complexity for Keygen and Decryption functions remain the same.

For the Index Generation algorithm, the function takes an index table as input parameter and gives the ciphered index. The generation of index table ID depends on the keywords and the dataset used. The ID generation algorithm involves the AES encryption and Hash functions. Since the index generation is directly dependent on the roots, the complexity of Index Generation phase is $O((nR + (mn))$.

In the Trapdoor Generation function, the user gives a root/keyword, to the query generation algorithm, to search for specific file. The Trapdoor Generation function uses two Hash functions and one AES encryption function along with multiplication function. The time complexity for each is constant individually. If the number of keywords increases the time complexity will be constant i.e., $O(2h + e)$.

For the Search Outcome function, simple multiplication and one Hash function are involved. As the keyword is searched, the time complexity will be the same for each encrypted query. Therefore, we can conclude that the time complexity for Search Outcome function is $O(mn)$. Table 2 compares algorithmic complexities with those seen in the state of the art.

Table 2. Complexity Comparative Analysis.

Schemes	Encrypt	Index Generation	Trapdoor Generation	Search Outcome
SSED [5]	$O(mn * ke)$	$O(mn * e)$	$O(e + h)$	$O(mn * R)$
ESAS [17]	$O(ne * r)$	$O(nm^2 * e)$	$O(m^2 * e)$	$O(nm^2)$
SESAPL [24]	$O(2n + h + e)$	$O((mn) + n)$	$O(h + e)$	$O(m * k + R)$
Proposed Scheme	$O(ne)$	$O((Rhe) + (mne))$	$O(2h + e)$	$O(mn)$

8.4. Performance Evaluation

This section presents an efficiency and effectiveness evaluation of the used algorithms. We analyze all algorithms and our whole scheme to perform the complexity analysis.

With the stemming algorithm, traditional keyword searching replaces the root-based searching scheme on the encrypted data. Since the roots are smaller than or equal to the keywords, set $R \leq K$, which means that this method has consumed less storage space in practice than the traditional keyword approach. Furthermore, searching for the root is efficient as compared to keyword searching. The bottleneck of the computation is assumed on the server-side because CS is responsible for answering the queries coming from the client side.

After gathering all the audio files, we now have to face the challenge of organizing the dataset; we need to convert all the audio communication between pilots and ATC controllers into plain text. To do this, we have used one of the best transcription algorithms, namely Sonix.ai, which gives us the most accurate result. The method used by almost all transcription algorithms is that they first convert audio signals into digital signals through an Analog-Digital converter which is based on Pulse Code Modulation (PCM).

PCM utilization represents the analog signal into the digital signal format, which is possible by performing three basic steps: sampling, quantization, and encoding. The amplitude of the analog signals is sampled at uniform intervals, and each sample quantizes to its closest value within a predetermined range of digital levels. The second step is quantization, which changes a continuous amplitude signal with discrete amplitudes. The last step, encoding, assigns the binary numbers or bits to amplitude values [39].

At that moment, our audio signals are transformed into digital signals. It is time to receive our output in plaintext by transcribing our digital signals into plaintext. We use speech-to-text algorithms to take an input and process signal to obtain the same information in the text file as the audio. The process to convert audio to text is shown in Figure 6. The time it takes to convert 1000 audio files into plain text is almost 194.58 min. The value time graph is shown in Figure 7.

The searching process is based on root-based searching instead of typical keyword searching. So, we need to extract the roots from the plain text dataset, which we have to achieve after the audio transcription process to plain text.

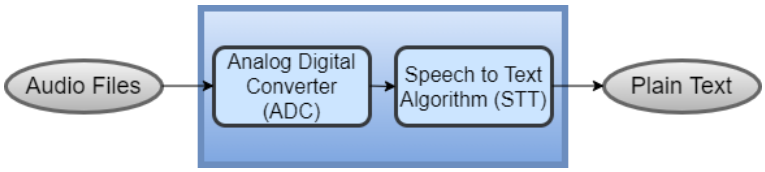

Figure 6. Audio to Plaintext Transcription Process.

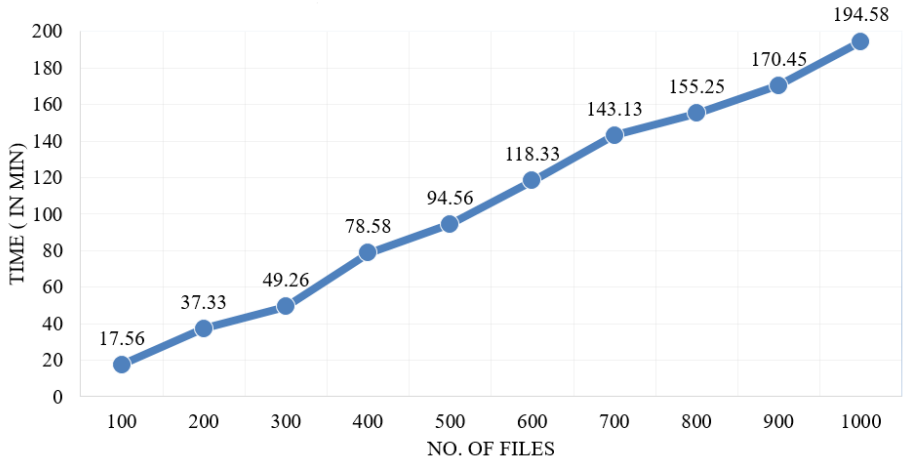

Figure 7. File Conversion Time Graph.

The root extraction time graph shows the time it takes to extract the roots from the dataset mentioned in Figure 8. The graph depicts linear growth with increasing number of files. For extracting roots from 1000 files, a total of 3.3875 s were consumed.

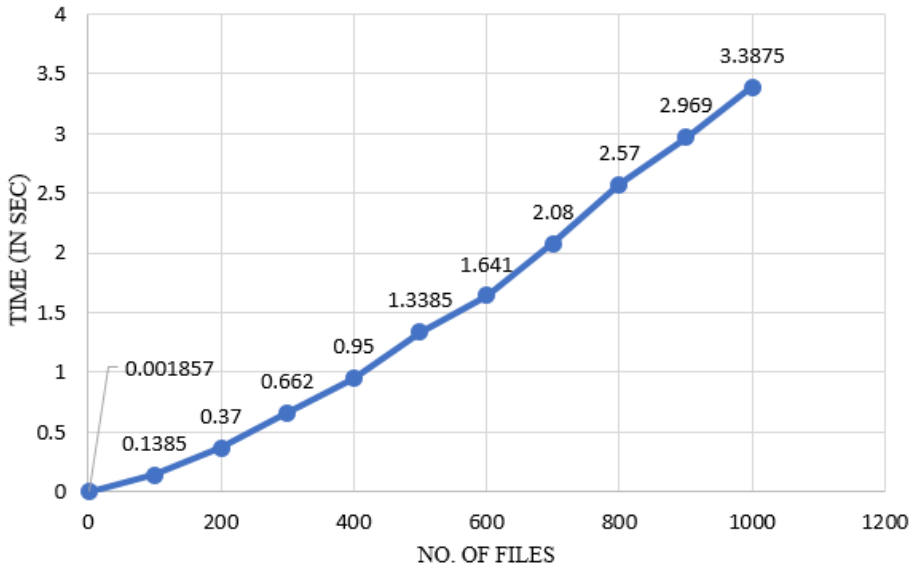

Figure 8. Root Extraction Time Graph.

The Figure 9 shows the total time taken to encrypt all the files. It is clear from the graph that execution time increases as the number of files increases, and it took almost 19 s to encrypt all the 1000 files. After the file encryption phase, now it is time to extract roots from the files because we are performing root-based searching instead of keyword searching.

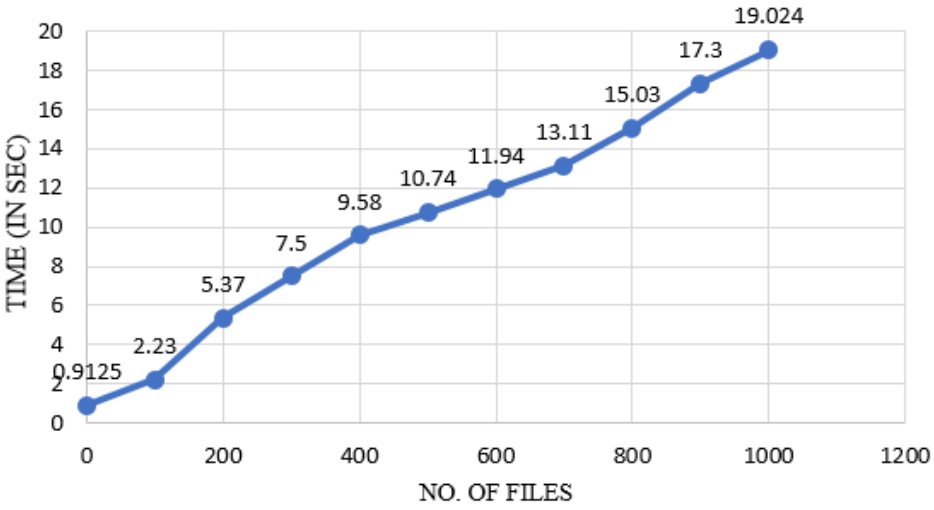

Figure 9. File Encryption Time Graph.

The Index generation was performed, which includes extracting the roots from all the documents, hashing, and encryption. This index consists of the name of fragments of all the documents in the dataset, which will be used as an identifier when the client's query results are going to be retrieved. Here we see in Figure 10 the linear growth of the graph, and as the number of files increases, the time will also increase.

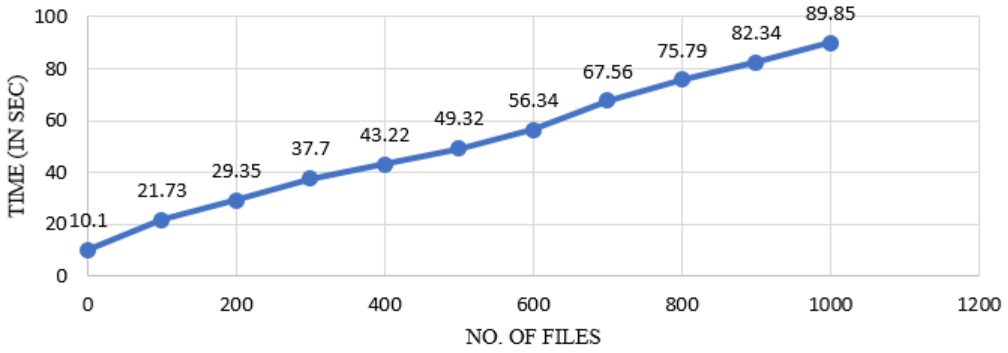

Figure 10. Index Formation Time Graph.

After the dataset is encrypted and stored in the cloud server, index generation and roots extraction is also performed. Then the next step is the searching phase, where a client or user performs a search query. Our algorithm does all the searching in the encrypted domain. This searching function is executed on the server side. The search graph also shows a linear growth with the increase in the number of files in the database. The server takes 0.98 s to search for the keyword "section" across 150 files. Suppose as a result of a search query 1000 files are to be returned, the server takes 6.89 s. It is impossible for a query to return all the files, however, for demonstrating the upper-bound complexity we assume that all the documents are retrieved in a particular scenario and server requires 6.89 s for the searching. This scenario is depicted in Figure 11.

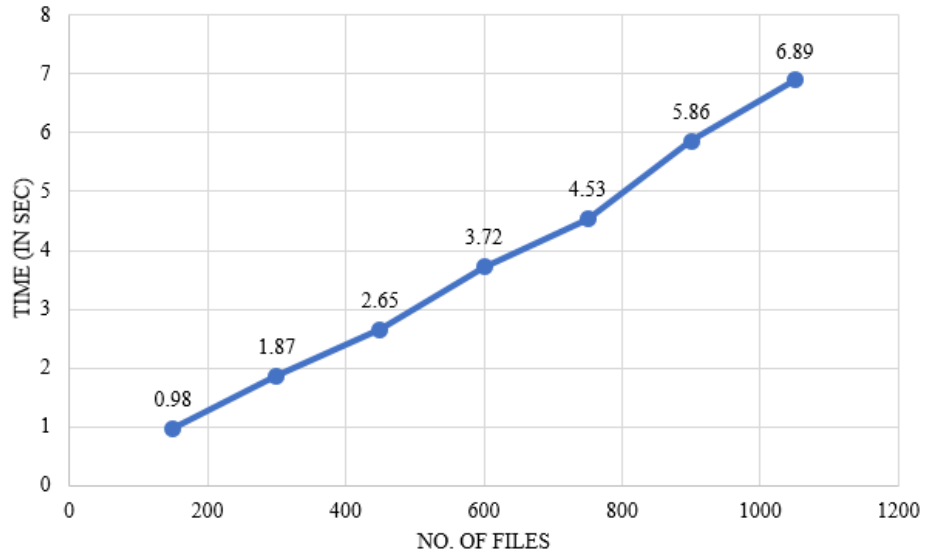

Figure 11. Search Time Graph.

Our algorithm needs some time to decrypt all the resulting files that are returned as a result of a search query. Figure 12 presents the decryption time of our algorithm. This decryption is the last function which needs to be analyzed. The decryption graph also shows a linear behavior, as seen in the previous graphs. The decryption of 100 files is 2.36 s, when considering the scenario where 100 files are returned to the client. Retrieval of all 1000 files is an impossible query, but for the sake of efficiency and complexity analysis, we assume that all the documents are retrieved in a particular scenario and require 19.7873 s for the decryption.

To examine the efficiency of the proposed scheme, we have compared it with the state of the art by implementing [24] and testing it over our dataset. The Figure 13 shows the results and demonstrates that the proposed scheme is highly efficient. The query time of the proposed scheme is lower than SESAPL scheme [24]. The SESAPL costs an average

search time of 23.95 s for all the 1000 files, whereas the suggested scheme costs an average time of 21.55 s for all 1000 files. This is presented in Figure 13.

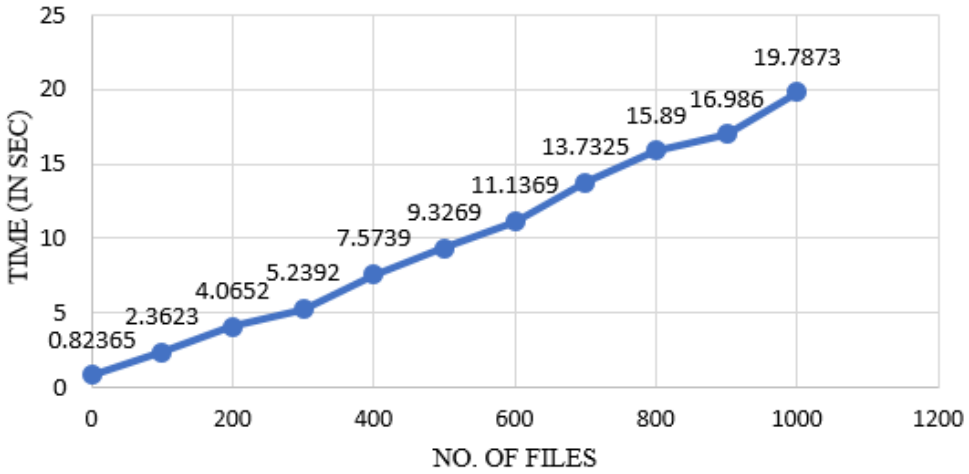

Figure 12. File Decryption Time Graph.

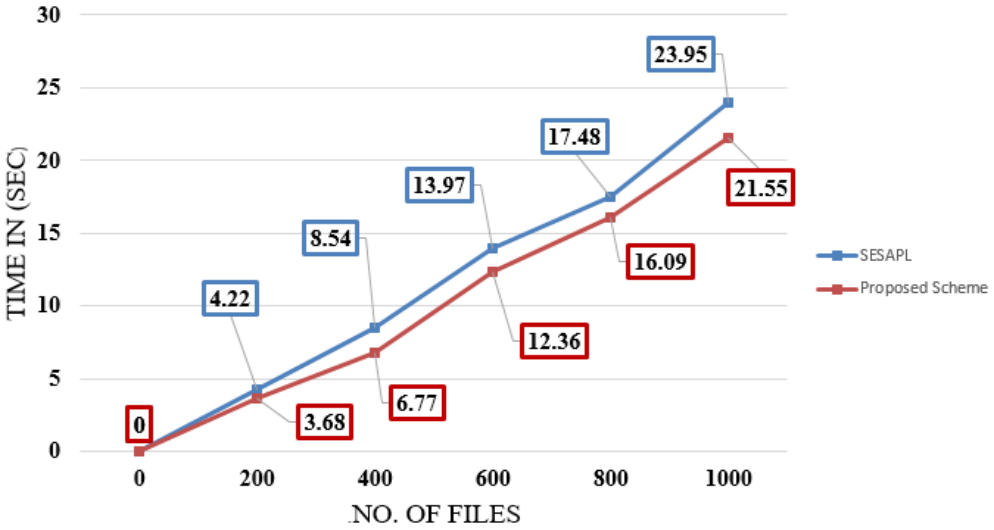

Figure 13. Comparative Computational Complexity.

9. Conclusions and Future Work

Amplification of attack surfaces and many attack vectors, civil aviation is vulnerable to cyber attacks. Data encryption is an industrial best practice and widely deployed to prevent data theft and privacy loss. While this is sufficiently effective, it hampers the ability to search over the encrypted data thus impacting data usability and system utility. In this research, a novel approach has been presented that enables search within the encrypted domain by identifying semantically related keywords. The scheme prevents search pattern attacks by generating probabilistic trapdoors. A stemming algorithm is introduced in the scheme, and the search process uses the roots of the keywords instead of the keywords themselves, which has the added advantage of being computationally less expensive and reduces the storage overhead. A contribution of this research is that it introduces a secure way of hiding information and search pattern by using a probabilistic approach for third-party adversaries. While embedding the concept of indistinguishability the scheme is designed to be lightweight yet provisions better security, and is applicable to practical scenarios because of its complete client and server architecture. The scheme has also been tested on a real-world data set of Aircraft Communication data to validate the proof

of concept. Although the proposed scheme has the capability of effectively preserving search patterns, the access patterns can be revealed to the cloud. The scheme will be enhanced further to preserve the access patterns in the future. Furthermore, in the future, the proposed scheme can be incorporated with access control, multiparty computation and key management to enable key management and authorizations between the GS personnel and data users.

Author Contributions: Conceptualization, M.A.S., S.T. and M.A.; methodology, M.A.S., S.T., M.A. and H.T.; software, M.A.S.; validation, M.A.S. and M.A.; formal analysis, F.K. and R.T.; investigation, S.T. and F.K.; resources, S.S. and A.M.A.; data curation, M.A.S.; writing—original draft preparation, S.T., M.A.S. and R.T.; writing—review and editing, S.T., H.T., R.T. and F.K.; project administration, H.T.; funding acquisition, S.S and A.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the SAUDI ARAMCO Cybersecurity Chair at Imam Abdulrahman Bin Faisal University in Saudi Arabia.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: This research is supported by the National Centre for Cyber Security (NCCS) and Higher Education Commission (HEC) in Pakistan, as part of the project "Privacy-Preserving Search over Sensitive Data Stored in the Cloud".

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zan, T.D.; d'Amore, F.; Camillo, F.D. The Defence of Civilian Air Traffic. ISSN 2280-6164. IAI 2016. Available online: https://www.iai.it/sites/default/files/iai1523e.pdf (accessed on 10 November 2022).
2. Deepa, K. A Novel Sentimental Based Semantic Search of Cloud Encrypted Data. *Int. J. Anal. Exp. Modal Anal.* **2019**, *XI*, 4251–4255. Available online: http://www.ijaema.com/gallery/499-september-2527.pdf (accessed on 10 November 2022).
3. Griffiths, J. Chinese hackers used tools leaked after the attack on Italian cybersecurity firm Hacking Team. In *South China Morning Post 2015*; South China Morning Post Publishers Limited: Causeway Bay, Hong Kong, 2015. Available online: http://www.scmp.com/node/1838426 (accessed on 10 November 2022).
4. Liesdonk, P.V.; Sedghi, S.; Doumen, J.M.; Hartel, P.H.; Jonker, W. Computationally efficient searchable symmetric encryption. In Proceedings of the 7th VLDB Workshop on Secure Data Management, Singapore, 17 September 2010. Available online: https://link.springer.com/chapter/10.1007/978-3-642-15546-8_7 (accessed on 11 October 2022).
5. Moataz, T.; Shikfa, A.; Boulahia, N.C.; Cuppens, F. Semantic search over encrypted data. In Proceedings of the 20th International Conference on Telecommunications (ICT), Casablanca, Morocco, 6–8 May 2013. Available online: https://ieeexplore.ieee.org/abstract/document/6632121 (accessed on 10 November 2022).
6. Prakash, P.; Abdelhadi, A.; Miao, M. Secure Authentication of ADS-B Aircraft Communications using Retroactive Key Publication. *arXiv* **2019**, arXiv:1907.04909.
7. Lu, X. Research on the security of communication addressing and reporting system of civil aircraft. *IOP Conf. Ser. Earth Environ. Sci.* **2019**, *295*, 032026. [CrossRef]
8. Sampigethaya, K. Aircraft Cyber Security Risk Assessment: Bringing Air Traffic Systems and Cyber-Physical Security to the Front. In Proceedings of the AIAA SciTech Forum, 7–11 January 2019. Available online: https://arc.aiaa.org/doi/abs/10.2514/6.2019-0061 (accessed on 10 November 2022).
9. Ezroni, E.; Dafna, G. Aircraft Communication System. U.S. Patent 6,720,890 B1, 13 April 2004.
10. Strohmeier, M.; College, K. *Security in Next Generation Air Traffic Communication Networks*; University of Oxford Trinity: Oxford, UK, 2016. Available online: https://www.bcs.org/media/2143/security-air-traffic.pdf (accessed on 10 November 2022).
11. Sampigethaya, K.; Poovendran, R.; Bushnell, L. A Framework for Securing Future e-Enabled Aircraft Navigation and Surveillance. In Proceedings of the AIAA Infotech@Aerospace Conference, Seattle, WA, USA, 6–9 April 2009. Available online: https://arc.aiaa.org/doi/abs/10.2514/6.2009-1820 (accessed on 10 November 2022).
12. Song, D.X.; Wagner, D.; Perrig, A. Practical techniques for searches on encrypted data. *IEEE Syst. Secur. Priv.* **2000**, *1*, 44–55.
13. Goh, E.J. Secure Indexes. *Cryptol. Eprint Arch.* **2003**, *216*, 1–18.
14. Curtmola, R.; Garay, J.; Kamara, S.; Ostrovsky, R. Searchable symmetric encryption: Improved definitions and efficient constructions. *J. Comput. Secur.* **2011**, *19*, 895–934. [CrossRef]
15. Tahir, S.; Ruj, S.; Rahulamathavan, Y.; Rajarajan, M.; Glackin, C. A New Secure and Lightweight Searchable Encryption Scheme over Encrypted Cloud Data. *IEEE Trans. Emerg. Top. Comput.* **2019**, *7*, 530–544. [CrossRef]

16. Karvounarakis, G.; Alexaki, S.; Christophides, V.; Plexousakis, D.; Scholl, M. RQL: A declarative query language for RDF. In Proceedings of the 11th International Conference on World Wide Web, Honolulu, HI, USA, 7–11 May 2002.
17. Liu, X.; Guan, Z.; Du, X.; Zhu, L.; Yu, Z.; Ma, Y. ESAS: An Efficient Semantic and Authorized Search Scheme over Encrypted Outsourced Data. In Proceedings of the International Conference on Computing, Networking and Communications ICNC, Honolulu, HI, USA, 18–21 February 2019. Available online: https://ieeexplore.ieee.org/abstract/document/8685554 (accessed on 10 November 2022).
18. Sun, X.; Zhu, Y.; Xia, Z.; Chen, L. Privacy-preserving keyword-based semantic search over encrypted cloud data. *Int. J. Secur. Appl.* **2014**, *8*, 9–20. [CrossRef]
19. Zhang, M.; Chen, Y.; Huang, J. SE-PPFM: A Searchable Encryption Scheme Supporting Privacy-Preserving Fuzzy Multikeyword in Cloud Systems. *IEEE Syst. J.* **2020**, *15*, 2980–2988. [CrossRef]
20. Yang, Y.; Liu, J.; Cai, S.W. Fast multi-keyword semantic ranked search in cloud computing. *Comput. Sci.* **2018**, *41*, 1126–1139.
21. Walid, R.; Joshi, K.P.; Choi, S.G.; Kim, D.-Y. Cloud-based Encrypted EHR System with Semantically Rich Access Control and Searchable Encryption. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020. Available online: https://ieeexplore.ieee.org/abstract/document/9378002 (accessed on 10 November 2022).
22. Liu, G.; Yang, G.; Bai, S.; Zhou, Q.; Dai, H. FSSE: An Effective Fuzzy Semantic Searchable Encryption Scheme Over Encrypted Cloud Data. *IEEE Access* **2020**, *8*, 71893–71906. [CrossRef]
23. Chaudhari, P.; Das, M.L. KeySea: Keyword-based Search with Receiver Anonymity in Attribute-based Searchable Encryption. *IEEE Trans. Serv. Comput.* **2020**, *15*, 1036–1044. [CrossRef]
24. Awais, M.; Tahir, S.; Khan, F.; Tahir, H.; Tahir, R.; Latif, R.; Umair, M.Y. A novel searchable encryption scheme to reduce the access pattern leakage. *Future Gener. Comput. Syst.* **2022**, *133*, 338–350. [CrossRef]
25. Xia, Z.; Zhu, Y.; Sun, X.; Chen, L. Secure semantic expansion based search over encrypted cloud data supporting similarity ranking. *J. Cloud Comput.* **2014**, *3*, 8. [CrossRef]
26. Hu, Z.; Dai, H.; Yang, G.; Yi, X.; Sheng, W. Semantic-Based Multi-Keyword Ranked Search Schemes over Encrypted Cloud Data. *Secur. Commun. Netw.* **2022**, *2022*, 4478618.
27. Bosch, C.; Hartel, P.; Jonker, W.; Peter, A. A Survey of Provably Secure Searchable Encryption. *ACM Comput. Surv.* **2014**, *47*, 18. [CrossRef]
28. Yunling, W.; Jianfeng, W.; Xiaofeng, C. Secure searchable encryption: A survey. *J. Commun. Inf. Netw.* **2016**, *1*, 52–65. [CrossRef]
29. Pham, H.; Woodworth, J.; Amini, M. Survey on Secure Search Over Encrypted Data on the Cloud. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e5284. [CrossRef]
30. Varri, U.; Pasupuleti, S.; Kadambari, K.V. A scoping review of searchable encryption schemes in cloud computing: Taxonomy, methods, and recent developments. *J. Supercomput.* **2020**, *76*, 3013–3042. [CrossRef]
31. Handa, R.; Krishna, C.R.; Aggarwal, N. Searchable encryption: A survey on privacy-preserving search schemes on encrypted outsourced data. *Concurr. Comput. Pract. Exper.* **2019**, *31*, e5201. [CrossRef]
32. Jiang, S.; Hagelien, T.F.; Natvig, M. Ontology-based Semantic Search For Open Government Data. In Proceedings of the IEEE 13th International Conference on Semantic Computing (ICSC), Newport Beach, CA, USA, 30 January–1 February 2019. Available online: https://ieeexplore.ieee.org/abstract/document/8665522 (accessed on 10 November 2022).
33. Li, Y.; Yuan, L.; Vasconcelos, N. Bidirectional Learning for Domain Adaptation of Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. Available online: https://ieeexplore.ieee.org/document/8954260 (accessed on 10 November 2022).
34. Yang, S.; Yu, W.; Zheng, Y.; Yao, H.; Mei, T. Adaptive Semantic-Visual Tree for Hierarchical Embeddings. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019. Available online: https://dl.acm.org/doi/abs/10.1145/3343031.3350995 (accessed on 10 November 2022).
35. Stefanovic, P.; Kurasova, O.; Strimaitis, R. The N-Grams Based Text Similarity Detection Approach Using Self-Organizing Maps and Similarity Measures. *Appl. Sci.* **2019**, *9*, 1870. [CrossRef]
36. Shi, H.; Li, Y.; Cao, H.; Zhou, X.; Zhang, C.; Kostakos, V. Semantics-Aware Hidden Markov Model for Human Mobility. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 1183–1194. [CrossRef]
37. Jivani, A.G. A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl.* **2011**, *2*, 1930–1938.
38. Jasna, K.K.; Shabna, M. An Efficient Semantic Aware Search Method over Encrypted cloud data. *Int. Res. J. Eng. Technol. IRJET* **2018**, *13*, 2359–2371. Available online: https://www.irjet.net/archives/V6/i2/IRJET-V6I2208.pdf (accessed on 10 November 2022).
39. Laskov, L.; Georgieva, V.; Dimitrov, K. Analysis of Pulse Code Modulation in MATLAB/Octave Environment. In Proceedings of the 55th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST), Nis, Serbia, 10–12 September 2020. Available online: https://ieeexplore.ieee.org/abstract/document/9232755 (accessed on 10 November 2022).