Encrypted Data Retrieval and Sharing Scheme in Space-Air-Ground Integrated Vehicular Networks

Haoyang Wang, Kai Fan, *Member, IEEE,* Kuan Zhang, *Member, IEEE,* Zilong Wang, *Member, IEEE,* Hui Li, *Member, IEEE,* and Yintang Yang, *Senior Member, IEEE*

Abstract—As a smart transportation application of the Internet of Things (IoT), the Internet of Vehicles (IoV) depresses the chances of traffic accidents, while improving transportation efficiency and user driving experience. However, as the number of vehicles continues to grow, the original ground-based IoV system is difficult to meet the ever-increasing demand. To this end, Space-Air-Ground Integrated Network (SAGIN) incorporates satellite systems, aerial network and terrestrial communications. However, because SAGIN integrates multiple network services and communication modes, which makes SAGIN more vulnerable to various types of attacks and security threats. This paper first presents the dominating security threats in data storage, transmission and sharing of space-air-ground integrated vehicular network (SAGIVN). Moreover, for guaranteeing the safety and effectiveness of the model, we advance a safe and effective encrypted data retrieval and sharing scheme in SAGIVN(ERDSS) for possible threats, the ERDSS can execute fuzzy retrieval over misspelling keywords and sort results by relevance scores to realize precise retrieval. We perform a comprehensive security discussion and execute experiments based on real-world data sets. The consequences demonstrate that the ERDSS is safe and efficient.

Index Terms—IoV, SAGIN, Natural Language Process, Searchable Encryption, Data Privacy.

1 Introduction

THE emergence and development of IoV has promoted the integration of Internet of Things (IoT) and road traffic, making road traffic control safer and more efficient, vehicles more intelligent as well. According to statistics [1], IoV can bring average of 1400 dollars in revenue to vehicles each year. It is estimated that by 2020, the global market size of IoV will reach 115.26 billion Euros [2], with huge market potential. Nevertheless, there are still many shortcomings in the present IoV architecture. With the continuous growth of global vehicles, these weaknesses have become increasingly prominent. For example, the privacy and security of data transmission and storage cannot be guaranteed, and the coverage of network services is limited, etc. The current IoV structure utilizes dedicated short-range communications (DSRC) or cellular network (CN) as the roadside units (RSU) in the network. However, the service coverage of the two is limited, and both require high construction costs, high-speed moving vehicles need to frequently switch the connected RSU, which also makes the two support high physical mobility faced with huge challenges.

With the continuous expansion of the IoV market, more and more vehicles and related facilities are integrated to it. It only has an impact on the network capacity, but also makes the network topology more complicated. Due to these reasons, data sharing and transmission in network-intensive areas may be lost or congested, which reduces network reliability and increase transmission delay.

The aforesaid disadvantages in the available IoV technologies are primarily because of the fact that the existing IoV architectures are ground-based to supply network access services for vehicles and in-vehicle applications. In response to this principal problem, the emergence of SAGIN [3], [4] offers new ideas for the promotion of IoV. SAGIVN is to add low-earth orbit satellites (LEOS), unmanned aerial vehicles (UAV), and high-altitude platforms (HAP) to the ground network to construct a ground-based, multi-dimensional and different level integrated network architecture. taking advantage of various network equipment in the aerospace field for vehicles and in-vehicle applications under diverse circumstances to offer real-time network access services.

Compared with the existing IoV architecture, SAGIVN provides a wider range of real-time network access. Meanwhile, its broadcast or multicast function supports a large number of users at the same time. Moreover, in extreme cases, such as natural disasters(earthquakes, tsunami), it also can sustain reliable access to vehicles. Based on the above advantages, it can be seen that SAGIN can provide efficient, low-cost and stable network access services when facing complex environments. **Figure.1** demonstrates the

Haoyang Wang is with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, Shaanxi, 710126 China e-mail: why19970701@163.com.

Kai Fan is with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, Shaanxi, 710126 China e-mail: kfan@mail.xidian.edu.cn. Kai Fan is the corresponding author

Kuan Zhang is with the department of Electrical and Computer Engineering, University of Nebraska-Lincoln, NE 68588 USA, e-mail:kuan.zhang@unl.edu.cn.

Zilong Wang is with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, Shaanxi, 710126 China e-mail: zlwang@xidian.edu.cn.

Hui Li is with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, Shaanxi, 710126 China e-mail: lihui@mail.xidian.edu.cn.

Yintang Yang is with the Key Lab. of Minist. of Wide Band-Gap Semicon, Xidian University, Xi'an, Shaanxi, 710126 China e-mail:ytyang@xidian.edu.cn

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3062626, IEEE Internet of Things Journal

architecture of SAGIVN as follows:

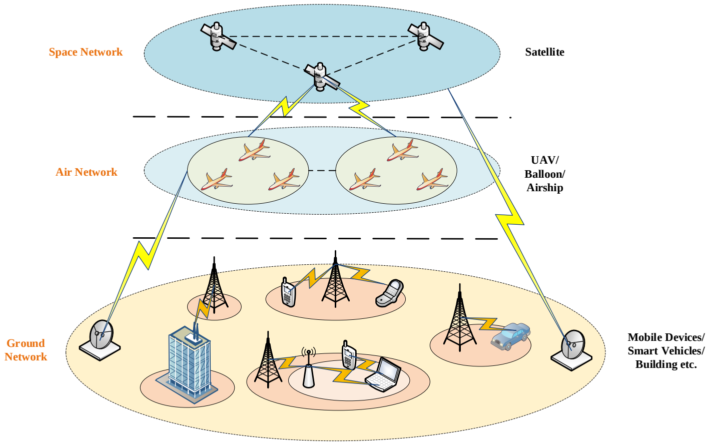


Fig. 1: SAGIN Architecture

It is worth noting that as SAGIVN contains a variety of diverse types of network interfaces and communication modes, it is prone to various types of attacks and threats. We need to fully consider various network attacks and security threats against SAGIVN.

- The attacker pretends to be a legitimate vehicle in SAGIN to transfer wrong information to other vehicles or perform a replay attack, thereby occupying network bandwidth and causing network communication jams.
- When the vehicle interacts with the high-altitude communication platform such as satellite, UAV, HAP to perform location query or navigation services, the transmission information contains the vehicle's private information, for instance, location, speed etc. If the intruder intercepts this kind of data, it could infer the vehicle's driving route, resident location, hence posing threats to vehicle privacy and security.
- Massive data in SAGIN are stored on the cloud server in architecture. For preventing attackers from directly invading the cloud server and filching data, it is necessary to preserve the privacy of the stored data.

In this paper, to protect SV data privacy when data storing on DC and SV querying for any information. Moreover, to reduce communication latency and computational overhead, we present a data privacy-preserving scheme with fuzzy ranked and multi-keyword search based on Paillier cryptography and attribute encryption. Our contributions can be summarized as follows:

- **Fuzzy Search**. We put forward an efficient method of keyword transformation based on the uni-gram algorithm and paillier encryption. For misspelling of one letter, the ERDSS can acquire the fuzzy value by utilizing the misspelling keyword and the correct keyword. Additionally, the presented method is effective against other spelling mistakes.
- **High Search Precision**. In order to improve query precision, we introduced natural language process in the trapdoor generation part of the ERDSS. Before generating query trapdoors, we need to utilize the dependency grammar and phrase structure tree to calculate the keyword weight in query keyword set, which plays an important role in query part to improve query accuracy.
- **Cross-Language**. We encode keywords in dictionary based on ASCII code and encrypt them into ciphertext with paillier encryption, which can realize across-language query in arbitrary languages environment.
- **Fine-grained management of access control**. Most of the existing IoV architectures utilize secure broadcast channels to realize data transmission and sharing, which will not only generate huge communication overhead, but also cannot perform fine-grained management of user authorizations. Based on attributed cryptography, the ERDSS implements fine-grained management of user access control.
- **Performance and Security**. We realize and evaluate the ERDSS on the light of a real SAGIVN environment. The consequences indicate that the ERDSS realizes high precision effectively and meets the proposed security requirements.

The remainder of this paper is organized as follows: Section 2 introduces some relevant preliminaries. The problem formulation in Section 3 includes system model architecture, design goals and security definitions. Section 4 presents the definition of the ERDSS algorithms as well as algorithms construction in detail. Section 5 gives the security analysis of the ERDSS in two aspects. Section 6 evaluates the performance of the ERDSS and compares it with some other related schemes. Section 7 makes a brief conclusion.

2 Related Work

2.1 SAGIN

In the original SAGIN framework, the space-air layer required to furnish access services to the ground through specific equipment, the communication between the space-air layer and the ground was relatively independent. With the continuous replacement of terrestrial mobile communication systems, the integration of space-air and ground-based networks also being continuously promoted.

CoRaSat [5] was launched in 2012, CoRaSat allocates underutilized communication resources to satellite services to achieve flexible and intelligent resource utilization. SANSA [6] is a research project initiated by the European Union in 2015. It proposed to combine space-air-ground networks to increase the throughput and flexibility of the backhaul network, while providing more effective network coverage for high-density and low-density areas. VITAL [7] was launched in 2015. By introducing network function virtualization and software-defined networking into the space-air layer, it offers joint resource management in a hybrid space-air-ground network to realize a flexible and flexible future network. SATNEX IV [8] was initiated by the European Space Agency in 2017. Its dominating goals include early exploration and scientific evaluation and testing of space-air communication networks, and assessment of the application of ground-based communication technologies in space-air networks. SATis5 [9] was activated in 2018. The project established a large-scale, real-time, end-to-end functional verification test platform for SAGIN based on 5G communication technology.

Despite the development of SAGIN has been continuously promoted, researchers have not yet proposed the SAGIN architecture with smart transportation as the application background.

2.2 Searchable Encryption

Searchable encryption (SE) has been evolved as an significant branch of cryptography since it was proposed by Song et al. [10] in 2000. SE could mainly separated into symmetric searchable encryption (SSE) and asymmetric searchable encryption (ASE) on the basis of construction method. In parallel, SE can be partitioned into fuzzy search, dynamic search, ranked search and so on.

So as to solve the problem of misspelling keywords and typographic mistakes in query, Li et al. [11] first proposed a fuzzy keyword SE scheme. Wang et al. [12] utilized LSH function and bloom filter to solve the problem of fuzzy multi-keyword search over encrypted data. Subsequently, Fu et al. [13] improved the Wang et al. scheme for higher precision.

Ranked search is presented to order the retrieved results built on a concrete method of similarity, so that users can acquire more relevant retrieved results quicker. Wang et al. [14] first utilized inverted indexes to fabricate a preserving order SE scheme. Dai et al. [15] presented a ranked multi-keyword SE scheme based on keywords splitting algorithm and binary tree in a hybrid cloud on the light of an equally divided $k-means$ clustering.

For resolving dynamic update in dictionary and database, Kamara et al. [16] firstly designed a parallel and dynamic SE scheme. Xia et al. [17] designed a secure ranked retrieval scheme over encrypted data that supports multi-keyword search and dynamic update, in which KBB-tree is built and greedy depth-first search algorithm is put forward to retrieve top-k results.

Although the SE technology has been developed relatively well, it lacks a relatively complete SE scheme for the specific background of smart transportation, IoV and SAGIN.

2.3 Natural Language Process

Natural language process (NLP) is an interdisciplinary subject that includes computer science, artificial intelligence (AI) and linguistics. As an essential segment of the field of AI, NLP promotes the continuous development of various fields of AI.

The research fields of NLP generally incorporate lexical analysis [18], syntactic analysis [19], semantic analysis [20] and pragmatic analysis [21]. Lexical analysis mainly cover word segmentation, part-of-speech tagging, named entity distinction and word sense disambiguation. Lexical basically implemented through rules-based, statistical and machine-based methods. The main goal of syntactic analysis is to determine the relationship between the components in a sentence, that is, the syntactic structure, which is mainly realized by the analysis of rhetorical structure and dependence relationship. Semantic analysis, as the current focus of NLP research, has different meanings for different language units. At this level, semantic analysis refers to word sense disambiguation, and at the sentence level it refers to the labeling of semantic roles. Pragmatic analysis mainly corresponds to the description in the text and reality, forming a dynamic ideographic structure.

2.4 Privacy-Preserving in IoV

Zhou et al. [22] proposed a novel deferentially privacy-preserving location-based service usage framework deployed on the edge node, designed to provide an adjustable privacy protection solution to balance the utility and privacy. Kong et al. [23] put forward an efficient and location-secure novel data sharing scheme with anti-crosstalk in the IoV environment. This scheme can collect and distribute data captured by vehicle sensors and utilize an improved Paillier cryptosystem to realize the sensory data aggregation preserving location privacy. Wu et al. [24] presented a privacy protection scheme based on physical layer security tools for eavesdropping attacks on vehicle users in the communication process under the IoV. The scheme can investigate resource management for secrecy provisioning when the vehicle user offload computation tasks. Moreover, it discusses three promising technologies, all of which help to enhance the confidentiality of eavesdropping attacks.

The above schemes provide effective solutions for privacy-preserving in the IoV, but the privacy emphasized in these are mainly the location and traffic data of the vehicle itself, while the outsourcing storage and sharing of data belonging to the vehicle has not yet been availably resolved. This is exactly the direction that the ERDSS will discuss under the emerging environment SAGIVN.

3 Preliminaries

3.1 Paillier Cryptography

Paillier encryption is a probabilistic public key encryption algorithm designed by Paillier in 1999. The security of this algorithm is based on the difficult problem of composite residual classes. The algorithm is a homomorphic encryption, which satisfying addition and number multiplication homomorphism.

3.2 Cross-Language

In many previous SE schemes, the system design only considered one language, and in reality, there will be multiple languages in the database. If the search mode in the existing SE scheme is used, cross-language search cannot be achieved. In order to achieve cross-language search, the ERDSS first utilizes the Uni-gram algorithm to split the search keywords into individual characters according to the language(Latin language, Stroke language) to which keywords belong, and then executes subsequent operations on the resulting characters. The definition of the Uni-gram algorithm and the the processes of diverse languages are as follows:

- **Uni-gram Algorithm** The Uni-gram algorithm belongs to the N-gram algorithm in the statistical language model. The basic idea of the N-gram algorithm

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3062626, IEEE Internet of Things Journal

is to perform a sliding window operation of size N on the content of the text according to bytes, forming a byte of length N for fragment sequence, the Uni-gram algorithm adjusts the sliding window value to one character to form a byte fragment with a length of 1. In ERDSS, we implement the Uni-gram algorithm by Python.

- **Encode & Encrypt (1)Latin Language.** For languages based on Latin alphabets such as English, Italian, Portuguese and Spanish, we directly calculate ASCII codes with each single letter, then multiply the ASCII codes by the corresponding multiples based on the position of the letters in the word, finally add integers of each letter together and encrypt it. **(2)Stroke Language.** For languages composed of strokes as basic elements, such as Chinese, Korean and Japanese, etc, we construct the relevance between strokes and ASCII codes, then calculate the sum of the ASCII codes of each text. The detailed process is demonstrated in figure.2 and figure.3 respectively.

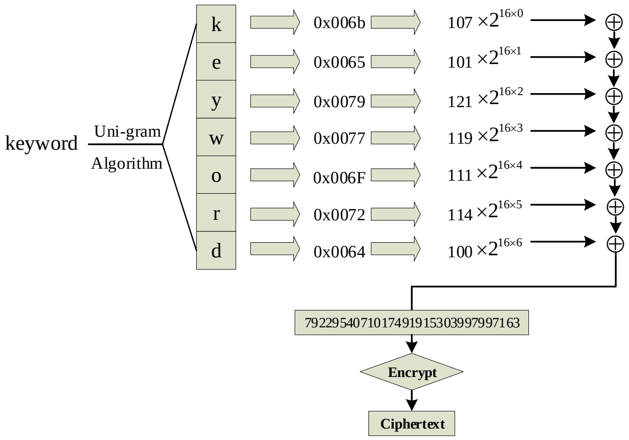


Fig. 2: Latin Language

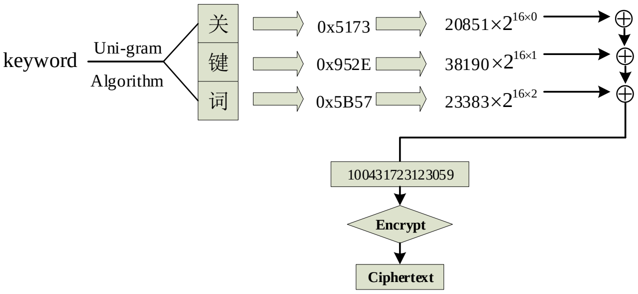


Fig. 3: Stroke Language

3.3 Dependency Grammar

The main purpose of dependency grammar is to analyze the syntactic structure of a sentence to better understand the meaning of the sentence. Dependency grammar has a general assumption that the syntactic structure essentially contains the relationship between words, and this relationship is called a dependency relationship. In the dependency grammar, it can accurately identify the part of speech of the keywords and the relationship between the keywords in the sentence and the subordinates. The keywords that belong to the dominant position are called dominant words, and the keywords that are in the dominant position are called dependent words. The relationship between the keywords in the sentence is one-way, and the dependency relationship between them is linked through a semantic arc.

3.4 Phrase Structure Tree

Phrase structure tree and dependency grammar are two typical analysis methods in natural language process. Phrase structure tree is used to express the syntactic structure of sentences. Among them, only the leaf nodes are related to the words in the input sentence, and the other intermediate nodes are all marked phrases.

3.5 Optimized $TF \times IDF$ Algorithm

In the ERDSS, we introduce the optimized $TF \times IDF$ algorithm. Compared with the traditional $TF \times IDF$ algorithm, the optimized one adds the weight of the keyword position in the file when calculating the relevant score, so that the score between keywords and files becomes more accurate. The detailed process is depicted as Eq.1:

$$s = \sum_{w_i \in Q} \frac{(n_{w_i}/L_{f_i}) \times \sum_1^k (\gamma_{f_j} \times tf_{j,w_i})}{\sqrt{\sum_{w_i \in W} ((n_{w_i}/L_{f_i}) \times \sum_1^k (\gamma_{f_i} \times tf_{j,w_i}))^2}} \times \frac{\ln(1+N/N_{w_i})}{\sqrt{\sum_{w_i \in W} (ln(1+N/N_{w_i}))^2}} \tag{1}$$

where γ_{f_j} represents the weight coefficient of the j-th part of the document, and its sum is 1; tf_{j,w_i} represents the number of keyword w_i in the j-th part; n_{w_i} represents the number of keyword w_i; L_{f_i} is the length of the whole file f_i.

4 Problem Formulation

In this section, we put forward the system model architecture of the ERDSS as shown in figure.4, we define the threat model and the security definition. Furthermore, on the light of the infrastructure of the system model, the goals to be achieved by the ERDSS are put forward.

4.1 System Model

As presented in figure.3, system model consists of four entities, we design system model on the background of SAGIVN.

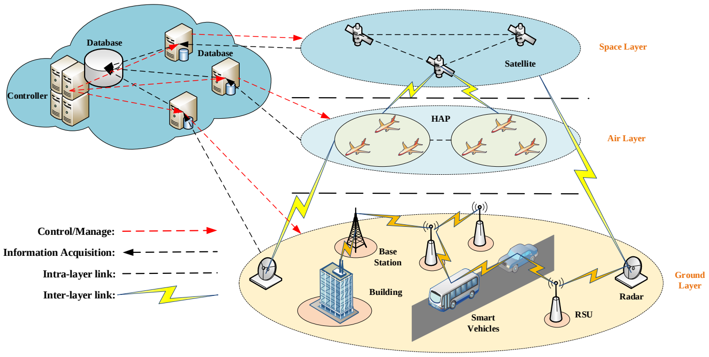


Fig. 4: SAGIVN Architecture

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3062626, IEEE Internet of Things Journal

- **Smart Vehicles**. As owners and users of data in SAGIVN, smart vehicles (SV) play an essential role in the system. The data privacy of vehicles will be included in the data uploaded to the system database. Adducing the road navigation in SAGIVN as a concrete instance, the navigation request issued by smart vehicle contains its own ID number, hodiernal speed, location and destination, etc. The navigation instruct obtained by vehicle will exist navigation route, passing point and other information.

  When SV initiates data query or communication requisition, it requested to produce trapdoors based on query keyword set. SV also be required for calculating weight of keywords in accordance with dependency grammar and phrase structure tree in NLP. SV transfers trapdoors combined with weight sets to the network access point (NAP).
- **Network Access Point**. As the basic unit for SV to provide computing and interconnection services, the NAP includes RSU on the ground layer, HAP on the sky layer and satellite cluster on the space layer. RSU offers services for areas with dense communication networks, while HAP and satellite clusters provide services for areas with underdeveloped communication networks because of their larger-scale service faculty.

  When the NAP receives the service request from the SV, it converts the information in request then transfers them to the database cluster (DC). The controller in the DC obtains results from corresponding sub-databases on the basis of the instruction and returns results to the SV through the NAP.
- **Database Cluster**. The DC contains multiple sub-databases, which stores a large amount of SV information and traffic data. After receiving the NAP instructions from diverse layers, the sub-databases pass them to the controller in DC, then the controller queries results from corresponding sub-databases and eventually returns to NAP.
- **Authority Platform**. The authority platform (AP) is a fully trusted entity in system. It provides SV with encryption index key and attribute key. It uploads SVs' encryption data and file bi-directional hash table to the DC, and transmits encryption matrix and keyword uni-directional hash table to the NAP.

4.2 Design Goals

- **Cross-Language Retrieval**. The ERDSS should support cross-language retrieval over encrypted data. The SV can issue queries in various languages and limit the language of the returned results.
- **Query Semantic Analysis**. The keywords in query are semantically analyzed before the trapdoor generation to calculate the weight of each keyword in query.
- **Ranked Search**. In accordance to NLP, the ERDSS can calculate and assign corresponding weights to query keywords, so that achieve higher retrieval precision and transmit results to the SV after sorting.
- **Security Goals**. We make the assumption that AP is fully trusted entity, DC and NAP are honest-but-curious, i.e., they are honest to execute the protocols but curious about SVs' data and will attempt to deduce any useful information during operations. But they will not be compromised simultaneously. The security goals of the ERDSS are defined as follows: (1). **Confidentiality of outsourced data**. The ERDSS should be able to protect the confidentiality of documents, indexes, languages information, file IDs and file symmetric key. DC and NAP should not deduce any useful information from the uploaded tuples and encrypted documents. (2). **Unlinkability of query**. The ERDSS should ensure that even if the SV issues two identical queries, they can be transformed into two completely different trapdoors. DC and NAP cannot deduce the relationship between trapdoors. (3). DC and NAP cannot learn any useful information from the returned results.
- **Dynamic Topological Network**. Due to the high-speed mobility of SVs, the network topology of SAGIVN will be frequently changed with the movement of entities. Furthermore, because of the strict standards of the SAGIVN model for communication response delay, the design of ERDSS needs to be adapted to the real-time transformation of the network topology structure to ensure efficient communication on the premise of safety.
- **Support Updating**. The ERDSS supports the addition and deletion of data in DC and the updating of keywords in dictionary stored on NAP.

4.3 Security Definition

For designing reliable schemes, the security definition of most presented SE schemes allows leaking some valued information to the adversary. Such access pattern, i.e. the identifiers of the corresponding documents, search pattern, i.e. whether the keyword w has been searched before. The ERDSS strengths the security definition by preventing the NAP and DC from learning anything about the search pattern and access pattern. The security definitions of the ERDSS can be demonstrated as follows:

Definition 1. ***Outsourced data security****. The presented SE scheme does not reveal any information to the NAP and DC pertaining to the stored documents more than their number, sizes and document identifiers.*

Definition 2. ***Index Security****. The protection of index privacy is twofold. Firstly, the index should be encrypted so that the cloud server cannot learn the content of the index. Secondly, by analyzing the security index, the cloud server cannot infer information about the file, including whether a file contains certain keywords and whether different files contain the same keyword.*

Definition 3. ***Trapdoor Privacy****. Trapdoor contains retrieval information in encrypted form. Trapdoor privacy requires that given a search trapdoor NAP cannot learn any information about query, including query keywords,* ***search pattern****(whether the same query has been searched) etc.*

Definition 4. ***Access Pattern***. *The association between the searched keywords and their relevant document identifiers should be indistinguishable for NAP.*

Definition 5. ***Semantic Security***. *Given the keyword set Ω and the security index $\tilde{I}$, the adversary A forges a series of queries in any probability polynomial time(PPT). The following describes the security of the scheme formally through security experiments.*

Setup: *Challenger C creates a keyword set ω. C selects some subsets from the files corresponding to set ω to form file set Σ. Firstly, Challenger C runs* ***setup*** *algorithm to generate key, then* ***indexgen*** *algorithm generates encrypted matrix index $\tilde{I}$ for A. Finally, C transfers keyword set ω and security index $\tilde{I}$ to adversary A.*

Queries: *Adversary A is allowed to request challenger C for trapdoor T for query $Q \in \Omega$. A can utilize T to perform* ***search*** *algorithm on security index, thereby acquiring the result P_R.*

Challenger: *Adversary A selects two non-empty queries $V_0, V_1 \subset \Omega$, where $V_0 - V_1 \neq \phi$ and $V_1 - V_0 \neq \phi$. Then transfers the query request to challenger C.*

After receiving V_0 and V_1, C selects $b \overset{R}{\leftarrow} \{0, 1\}$, and then produces trapdoor T_b for V_b. Allow A to use T_b to generate the search result P_R for T_b.

Response: *Adversary A outputs its conjecture b' about b. The advantage of adversary A winning in this security experiment is defined as Eq.2:*

$$Adv_A = |\Pr[b = b'] - \frac{1}{2}| \tag{2}$$

If no polynomial time adversary can win the above security experiment with a non-negligible advantage, the scheme satisfies semantically secure.

5 Algorithm Construction

In this section, we divide the algorithms in ERDSS into five parts. As the first part, the "system initialization" section is mainly responsible for producing correlative parameters and keys for each entity, and completing the data processing. The "index construction" aims to build indexes based on the dictionary and dataset generated by initialization. The main participants of the above two parts are AP and SV. When SV queries or shares data, it needs to execute "trapdoor generation" algorithm to get trapdoors, then NAP and DC execute "search" algorithm after receiving trapdoors and return results to SV. SV decrypts and verifies whether the results are valid after getting the results. When the system needs to be updated, AP and SV performs the algorithms in "update" part to accomplish it.

5.1 System Initialization

- **Key Initialization**. NAP sets the fuzzy value of query $acc = x$ (acc means the text distance between two keywords, x is a integer). Moreover, NAP inputs security parameter α to AP, AP outputs a string K which its length is α, K is the key to operate symmetric encryption. NAP inputs security parameter μ to AP, AP outputs a pair key of paillier which process is illustrated as follows:
  - AP randomly chooses two big prime integer $0 <$ p $<2^\mu$, $0 <$ q $<2^\mu$ and then computes $n = p \times q$.
  - AP computes $\sigma = e^{-1} \bmod n$, which $e = lcm(p-1, q-1)$.
  - AP transfers public-private key pairs: $pk = n$, $sk = (e, \sigma)$.
- **File Initialization**. SV extracts w_i from file collection to construct dictionary W. Furthermore, SV generates the corresponding file identifier $id_y (1 < y < n)$ for each file and transfers W, plaintext data set to AP. AP utilizes symmetric key K to encrypt files in file collection. Moreover, AP executes attribute encryption for K, attribute policy defined by SV. AP makes use of public parameter *pp*, a random number $s \in Z_p^*$ to calculate key ciphertext as Eq.3:

$$\begin{aligned} C_k = [&C_1 = g_1^s \cdot g^{-s \cdot \sum_{i=1}^{n} H_2(att_i)}, \\ &C_2 = e(g, g)^s, \\ &C_3 = K \cdot e(g, h_1)^{-s}] \end{aligned} \tag{3}$$

The key ciphertext will be stored on the AP. When the SVs need to decrypt, they send their attribute sets to the AP, and AP will return the decrypted K to the SV.

5.2 Index Generation

- AP utilizes optimized $TF \times IDF$ algorithm to compute relevance scores between files and keywords.
- AP executes preconditioning operation for keywords in dictionary, which detailed process has illustrated in section 2.2.
- AP uses the SV's paillier key pk_c to encrypt the keywords in the dictionary and the letters in keyword (Latin Language) or one single character(Stroke Language) to acquire tuples as Eq.4:

$$\begin{aligned} &CI_{w_1} = ([w_1]_{pk}, ([w_{1_1}]_{pk}, [w_{1_2}]_{pk}, \cdots, \\ &[w_{1_{len}}]_{pk})) \\ &\vdots \\ &CI_{w_m} = ([w_m]_{pk}, ([w_{m_1}]_{pk}, [w_{m_2}]_{pk}, \cdots, \\ &[w_{m_{len}}]_{pk})) \end{aligned} \tag{4}$$

Then AP uses the hash functions $H_w(\cdot)$, $H_f(\cdot)$ the random function $R(\cdot)$, the obtained keyword ciphertext tuple and file identifier to generate an encrypted index, which its creating steps are illustrated as follows:

  - Initializing a $(m' \times n')$ dimensional matrix η, where $m \leq m'$, $n \leq n'$(m is the maximum number of keyword set, n is the maximum number of data set), setting all elements in matrix to 0.
  - Encrypting matrix index η for $x = 1, \cdots, n'$, $y = 1, \cdots, m'$. AP making use of hash functions $H_w(\cdot)$ to produce hash table $\alpha_w(\cdot)$, which contains mapping result $\alpha_w([w_i]_{pk})$. Moreover, AP taking advantage of $R(\cdot)$ and $H(\cdot)$ to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3062626, IEEE Internet of Things Journal

generate hash table $\alpha_f(\cdot)$ that contains mapping result $\alpha_f(f_y)$, which the process is illustrated as Eq.5:

$$\begin{aligned} R(id_y) &= f_y \\ \alpha_f(f_y) &\to t \end{aligned} \tag{5}$$

- Constructing the correspondence between files and keywords based on initializing matrix η. If keyword $w_x(1 \le x \le m)$ appears in file $id_y(1 \le y \le n)$, the corresponding element in matrix η is set to the optimized $TF \times IDF$ value of w_x and id_y, otherwise is set to 0.

AP only transfers the encrypted index to NAP, the hash functions $H_w(\cdot)$, $H_f(\cdot)$ and the random function $R(\cdot)$ are reserved by itself. Figure.5 shows the construction process of encrypted matrix index.

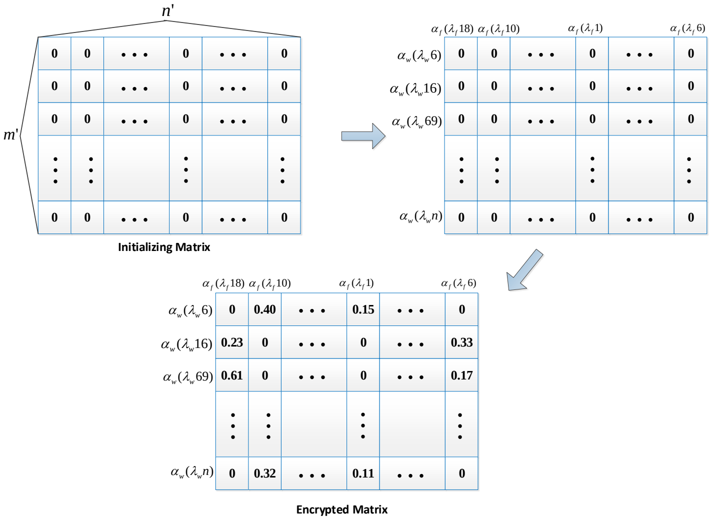


Fig. 5: Matrix Index

5.3 Trapdoor Generation

Trapdoor generation consists of two parts. Firstly, SVs calculate the weights for all keywords in query keyword set according to dependency grammar and phrase structure tree. Secondly, SVs take use of the paillier keypair to produce trapdoors for keywords, then combine the weights and trapdoors to get the final tuple.

- **Weight Calculation**. For each keyword in query set, the initial keyword relationship is 1, if the keyword has a syntactic relationship with other keywords, its weight becomes 1+R, where R represents the syntactic relationship.
  For the two search keywords q_1 and q_2, the syntactic relationship is expressed as $R(q_1, q_2)$. If there is syntactic relationship between them, the relationship of q_1 and q_2 increases by $\frac{d_2}{d}R(q_1, q_2)$ and $\frac{d_1}{d}R(q_1, q_2)$ respectively, where d_1 and d_2 represent the distance between two keywords and their common ancestor nodes, and d represents the distance between keywords, that is, $d = d_1 + d_2$.
  For any query $Q = \{q_1, q_2, \cdots, q_z\}$, z is the number of search keywords. For keyword q, its weight value is $p \times z$, where p is the weight ratio of search keyword q, that is depicted as Eq.6,

$$\frac{1 + \sum\limits_{j=2}^{z} R(q_i, q_j)}{\sum\limits_{i=1}^{z}(1 + \sum\limits_{j=2}^{z} R(q_i, q_j))} \tag{6}$$

Therefore, the weight KW of q is expressed as Eq.7:

$$KW(q) = p \times z = \frac{(1 + \sum\limits_{j=2}^{z} R(q_i, q_j)) \times z}{\sum\limits_{i=1}^{z}(1 + \sum\limits_{j=2}^{z} R(q_i, q_j))} \tag{7}$$

Taking "multiple keyword search encryption" as an query set for example, the phrase structure tree and dependency grammar are depicted in figure.6 and figure.7 respectively.

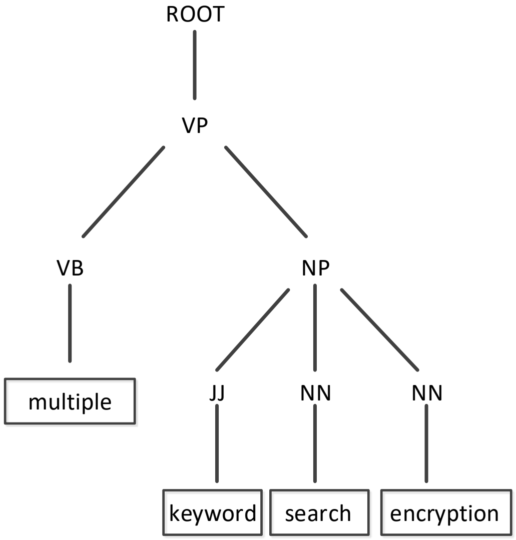


Fig. 6: Phrase Structure Tree

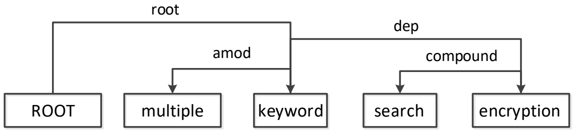


Fig. 7: Dependency Grammar

For example, the relationship between "encryption" and "multiple" is gained from the syntactic diagram, the distance between them is 5 from the phrase structure tree, the syntactic relationship between them is $R(amod) = 1/(ln5)$, the distance from "encryption" and "multiple" to its common ancestor are 3 and 2, so the weights of "encryption" and "multiple" are $2(ln5)/5$ and $3(ln5)/5$ respectively. Similarly, there are two syntactic relationship between "multiple" and "keyword", "keyword" and "multiple". Taking advantage of the above method, the final weights of "encryption" and "multiple" are calculated as $KW(multiple) = 1.23$ and $KW(encryption) = 0.95$ respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3062626, IEEE Internet of Things Journal

- **Trapdoor Generation**. SV makes use of their paillier keypair to encrypt the query keyword w. Furthermore, SV encrypts the letters or single character in keyword w to acquire the ciphertext tuple $T_w = \{C_w || C_{w_1}, \cdots, C_{w_{len}}\}$. Eventually, SV adds the weight relevant with the keyword w to the tuple.

5.4 Search

We introduce two ciphertext comparison algorithms used in the search phase: paillier ciphertext field integer greater than or equal to comparison algorithm(CGE) and paillier ciphertext field integer equal to comparison algorithm(CE).

CGE. SV and NAP have a pair of paillier key pairs (pk, sk) and (pk_{NAP}, sk_{NAP}) respectively, given two integers a and b, encrypted with the same public key pk to get two ciphertext integers $A = [a]_{pk}$, $B = [b]_{pk}$. The purpose of the CGE algorithm is to obtain the relationship between a snd b by performing certain calculations on A and B, that is, $a \geq b$ or $a < b$. The CGE algorithm is demonstrated in **Algorithm.1**:

Algorithm 1 CGE Algorithm

```
Input: $A = [a]_{pk}$: the paillier ciphertext of integer $a$; $B = [b]_{pk}$: the paillier ciphertext of integer $b$;
Output: result
SV:
X = (A/B)^r, where r is random in N^+.
```

```
Input: $A = [a]_{pk}$: the paillier ciphertext of integer $a$; $B = [b]_{pk}$: the paillier ciphertext of integer $b$;
Output: result
$SV$:
$X = (A/B)^r$, where $r$ is random in $N^+$.
Transfer $X$ to NAP.
$NAP$:
Receive $X$ from NAP.
$x = Dec_{sk}(X)$.
if $x \geq 0$ then
    result = $Enc_{pk_{NAP}}(1)$.
else
    result = $Enc_{pk_{NAP}}(0)$.
end if
Transfer result to NAP.
$SV$:
Receive result from NAP.
result = $Dec_{sk_{NAP}}(result)$.
Output result.
```

The execution process of the CGE algorithm is described in detail as follows:

1) SV chooses a relatively small positive integer r and computes $X = (\frac{A}{B})^r$. In accordance to the additive homomorphism of the paillier encryption algorithm, there exists $(\frac{A}{B})^r = (a-b) * r$. $(a-b)$ and $(a-b) * r$ have the same sign because r is a small positive integer. Eventually, SV transfers X to the NAP.
2) NAP decrypts X utilizing the private key sk to acquire the plaintext x. If $x \geq 0$, let result = $Enc_{pk_{NAP}}(1)$, otherwise let result = $Enc_{pk_{NAP}}(0)$, and sends the result to the SV.
3) After receiving the result, the SV decrypts it with the private key sk_{cp} and acquires result = $Dec_{sk_{NAP}}(result)$.

CE. The purpose of the CE algorithm is to determine whether the plaintext a is equal to b by calculating the two ciphertext integers $A = [a]_{pk}$ and $B = [b]_{pk}$, so as to realize the function of accurate search. The main idea of the CE algorithm is to implement the CGE algorithm twice, as shown in **Algorithm.2**:

Algorithm 2 CE Algorithm

```
Input: $A = [a]_{pk}$, $B = [b]_{pk}$
Output: result
$temp1 = \text{CGE}(A, B)$;
$temp2 = \text{CGE}(B, A)$;
$result = temp1 * temp2$;
Output result;
```

In the search phase, after acquiring the trapdoor tuple TTS, NAP first executes fuzzy search, puts the items matching TTS into the candidate set TTS', then NAP makes use of TTS' for accurate search to get the final result set TTS''. The fuzzy search and accurate search are demonstrated in **Algorithm.3** and **Algorithm.4** respectively:

Algorithm 3 Fuzzy Search Algorithm

```
Input: Trapdoor tuple set $TTS$, Keyword number $n$, Encrypted index $\eta$, Fuzzy value $acc$
Output: Candidate keyword set $TTS'$
for each $i \in [1, n]$ do
    $temp1 = CGE(C_{w_i}, ([w_j]_{pk})/([acc * 128]_{pk}))$;
    $temp2 = CGE(([w_j]_{pk})/([acc * 128]_{pk}), C_{w_i})$;
    $temp = temp1 * temp2$;
    if $temp = 1$ then
        $TTS' = TTS' \cup C_{w_i}$
    end if
end for
Return $TTS'$
```

After NAP acquiring the final result set TTS'', NAP takes the row vectors $(v_{w_i}, v_{w_j}, \cdots, v_{w_z})$ relevant with all items in TTS'' from encrypted matrix η. Then NAP multiplies the row vectors by the correlation coefficient on the light of keyword weight in trapdoor T_w sent by SV. Finally, NAP adds all row vectors to get v_{result} as shown in figure.8, and transfers it to DC.

5.5 File Decryption

SV transmits its own attribute set Att to AP after getting the query result. AP decrypts the related key ciphertext on the basis of Att as Eq.8:

$$\begin{aligned}
&C_3 \cdot e(C_1, h_{U_i}) \cdot C_2^{SK_{U_i}} \\
&= e(g_1^s \cdot g^{-s \cdot \sum_{i=1}^{n} H_2(att_i)}, \\
&(h_1 \cdot g^{-SK_{U_i}})^{1/(\alpha - \sum_{i=1}^{n} H_2(x_i))}) \cdot C_3 \cdot C_2^{SK_{U_i}} \\
&= C_3 \cdot e(g, h_1)^s \cdot e(g, g)^{-s \cdot SK_{U_i}} \cdot e(g, g)^{s \cdot SK_{U_i}} \\
&= K \cdot e(g, h_1)^{-s} \cdot e(g, h_1)^s \\
&= K
\end{aligned} \tag{8}$$

SV utilizes the symmetric key K to decrypt the ciphertext data and obtain the plaintext set of the query result. The process is as Eq.9:

$$\begin{aligned}
&Dec_K(C_{id_y}) = f_{id_y} \\
&F = F \cup f_{id_y}
\end{aligned} \tag{9}$$

Algorithm 4 Accurate Search Algorithm

```
Input: Trapdoor tuple set $TTS$, Candidate keyword set $TTS'$, Candidate keyword number $t$, Fuzzy value $acc$, the length of search keyword $len$, the length of $w_i$ $L_i$
Output: The final result set $TTS''$
for each $i \in [1, t]$ do
    $temp = 0$;
    if $L_i \geq len$ then
        $p = len$,$q = L_i$;
    else
        $p = L_i$,$q = len$;
    end if
    for each $j \in [1, p]$ do
        for each $k \in [1, q]$ do
            if $CE(C_{w_j}, [w_i]_{pk})$ then
                $L_i - -$;
                break;
            end if
            if $k == L_i$ then
                $temp + +$;
            end if
            if $temp > acc$ then
                break;
            end if
        end for
    end for
    if $temp \leq acc$ then
        $TTS'' = TTS'' \cup C_{w_j}$
    end if
end for
Return $TTS''$
```

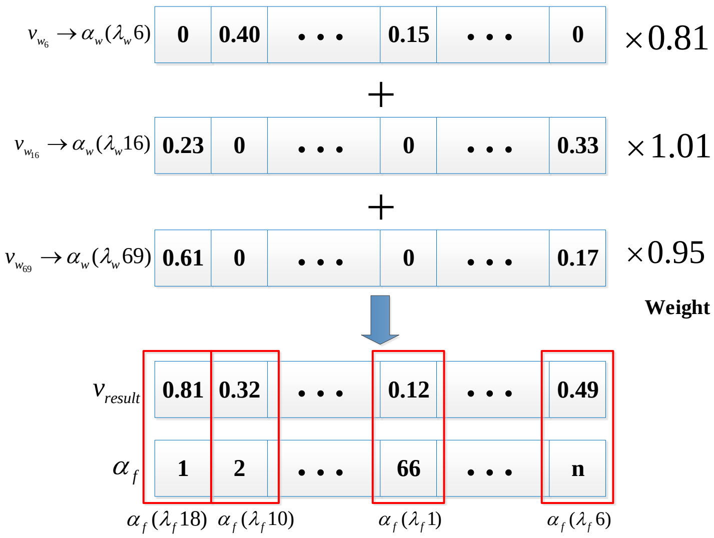


Fig. 8: Query Matching

5.6 Update

The update of the ERDSS is divided into two aspects: keyword update and data update. In this part, we will introduce two aspects respectively:

- **Keyword Update** Because only the file-based bidirectional hash table is stored on the DC, the SV does not need to transmit the status of the updating to the DC

  1) **Keyword Addition**. When SV adds keywords to the dictionary, they need to create an update tuple, which includes the operation instruction $op = "addition"$ and the added keyword w_{add}. After AP receiving the tuple from SV, AP first computes $TF \times IDF$ value between w_{add} and each file. Then AP encrypts w_{add} with paillier encryption to obtain its ciphertext $[w_{add}]_{paillier}$ and generates a vector $V_{w_{add}}$ based on these. Eventually, AP transfers the op and the vector as a new tuple to NAP. NAP makes use of its stored hash function $H_w(\cdot)$ to map $V_{w_{add}}$ to its position in matrix index and add it in.
  2) **Keyword Deletion**. The deletion operation of keywords in dictionary is basically similar to the addition operation. The operation instruction of SV is $op = "deletion"$. The difference with addition operation is that when NAP receives the ciphertext $[w_{del}]_{paillier}$ of the keyword w_{del}, NAP maps its position in matrix index through the hash function $H_w(\cdot)$, and then deletes its relevant vector v_{del} from matrix index.

- **Data Update** The data update of the ERDSS is different from the keyword update. The keyword update only needs to be operated on NAP. However, the data update needs to be operated on NAP and DC simultaneously. The specific process is depicted as follows:

  1) **Data Deletion**. In the ERDSS, the file update is different from the keyword update. The file update requires AP to transfer update instructions to both NAP and DC. Firstly, taking the file deletion as an example. The tuple generated by SV includes the deleting file identifier f_{del}, deletion operation $op = "deletion"$. After receiving the tuple, AP takes advantage of the stored hash function $H_f(\cdot)$ mapping to get the position $pos_{f_{del}}$ of $[f_{del}]_{paillier}$. AP sends the new tuple $(op, pos_{f_{del}})$ to NAP and DC respectively, then NAP and DC performs the deletion operation in $\alpha_f(\cdot)$ and matrix index stored by them.
  2) **Data Addition**. The data addition operation also first requires SV to transfer update instructions $op = "addition"$, added file identifier f_{add} and keywords $\{[w_n]_{paillier}||(n = i, j, z, m, n, p)\}$ relevant with f_{add} to AP. AP makes use of $H_f(\cdot)$ mapping to gain the position $pos_{f_{add}}$ of $[f_{add}]_{paillier}$. Furthermore, AP computes the $TF \times IDF$ values between keywords $\{[w_n]_{paillier}||(n = i, j, z, m, n, p)\}$ and f_{add}, and generates a tuple set as Eq.10

$$\{([w_n]_{paillier}, (TF \times IDF)_n)||(n = i, j, z, m, n, p)\}. \quad (10)$$

AP transmits $(op, [f_{add}]_{paillier})$ to DC, which is used to update the bi-directional hash table $\alpha_f(\cdot)$ stored by DC. Simultaneously, AP deliveries tuples as Eq.11

$$(op, [f_{add}]_{paillier}, \{([w_n]_{paillier}, (TF \times IDF)_n) \tag{11}$$

$||(n = i, j, z, m, n, p)\})$ to NAP, which is used to update the matrix index stored by NAP.

6 Security Discussions

In this part we investigate to which extent the ERDSS fulfills the security goals depicted in subsection 3.2 and security definition described in subsection 3.3.

6.1 Confidentiality of outsourced data

- **Outsourced data Security**. For protecting the outsourced data, AP applies AES-encryption on these plaintexts before producing matrix index. Compared with other symmetric encryption algorithms, AES has the characteristics of high security and efficiency. However, the number of files and the size of each encrypted file are permitted to be revealed in the ERDSS.
- **Index Security**. To preserve the index information, the ERDSS makes use of hash function to create keyword hash table after encrypting keywords from dictionary with paillier algorithm. Meanwhile, for preventing file identifier from statistical attacks, we suggest random function and hash function to produce file hash table. What remains to be leaked is relevance scores, but NAP cannot execute statistical attacks based on them without other information. Such scheme not only preserves the privacy of keywords and files information, but also avoids revealing the distribution of these values to NAP.
- **Access Pattern**. The ERDSS distributes the security index and the outsourced data among distinct, not colluding NAPs to avoid revealing the access pattern information. NAP learns the query trapdoors but not the identifiers of retrieved files, while DC learns the file identifiers but no the query trapdoors. For this reason, both NAP and DC cannot learn the relationship between the trapdoors and retrieved results.

6.2 Semantic Security

The security of the ERDSS depends on the semantic security of the paillier homomorphic encryption algorithm. If the paillier homomorphic encryption algorithm based on the ERDSS is a semantically safe, then the ERDSS is also semantically safe.

Assuming that A in the polynomial time algorithm can win the security experiment in section 3.3 with a non-negligible advantage, then A can be fabricated an algorithm ς, which can undermine the semantic security of the random prediction model(ROM) encryption algorithm. ς can access the oracle machine O_f, where the method f is either a random algorithm or a homomorphic encryption algorithm. Replacing the encryption operation in scheme with the calculation of ς, and then proving the security of the scheme through the security experiment given in section 3.3.

The specific process of the security experiment is depicted in **Algorithm.5**:

Algorithm 5 Security Experiment

```
Input: $(\Omega, \Sigma, I) \leftarrow \varsigma(k)$
Output: $b'$
$MK \leftarrow Setup(k)$;
$(M'_D, \tilde{I}) \leftarrow GenIndex(MK, I)$;
for each $i \in [1, q]$ do
    one query each time: $q_i \leftarrow A(Q_1, Q_2, \cdots, Q_q)$;
    $T_{Q_i} \leftarrow Trapdoor(MK, Q_i)$;
    $P_R(x) \leftarrow Search(\tilde{I}, T_{Q_i})$;
end for
$b \leftarrow A(V_0 \in \Omega^*, V_1 \in \Omega) \in \{0, 1\}$;
$T_{Q_b} \leftarrow Trapdoor(MK, V_b)$;
$P_R(x) \leftarrow Search(\tilde{I}, T_{Q_b})$;
Output $b'$;
```

A outputs b' as the adversary's conjecture against b. If A outputs 0, then ς guesses that f in O_f is a random function, expressed as $\varsigma_f = 0$. Otherwise, ς guesses f is an encryption algorithm, which is expressed as $\varsigma_f = 1$. Obviously, if f is a random function, there is a probability $\Pr[\varsigma_f = 0] = 1/2$. If f is an encryption algorithm, then ς and A have the same probability to output 1. Therefore, ς has the same advantages as A in winning the security experiment in distinguishing the semantic security encryption algorithm from the random prediction model. Nevertheless, based on the definition of the semantic security encryption system, ς does not exist. Consequently, there is no algorithm A with a non-negligible probability to win a security experiment.

7 PERFORMANCE EVALUATION

We compare the ERDSS with two other similar data preserving schemes Wang et al. [25] and Fu et al. [13] In this section, we evaluate the performance of these schemes. A comparison of the performances of Wang et al. and Fu et al. are demonstrated in **table.1**.

Moreover, the movement of SVs in SAGIVN will modify the network topology of the system in real time, the topology will continue to affect SV communication. The ERDSS adapts dynamic topology by reducing the impact of topology on SV communication based on cloud-edge architecture. ENs in the ERDSS as forwarding points for communication between SVs or between SVs and DC, is the key to guarantee efficient communication. Therefore, deploying a sufficient number of ENs in an area can provide services for multiple SVs simultaneously. We fixed other variables, utilizing diverse numbers of ENs in a region as variables to measure the communication delay, the results are depicted in figure.12:

It can be observed from **figure.12** that the introduction of ENs can enable efficient communication among entities. Furthermore, it can be noted that the communication consumption is the lowest when the number of EN is 6, while the communication efficiency gradually decreases when the number is greater than 6. Since the SV transfers communication data to all ENs in the region, the traffic flow in a certain

TABLE 1: Comparison of the three schemes

Scheme	Fine-grained control	Fuzzy search	Verifiable result	Multi-keyword	High accuracy
ERDSS	√	√	√	√	√
Wang et al.	×	√	×	√	√
Fu et al.	×	√	×	√	×

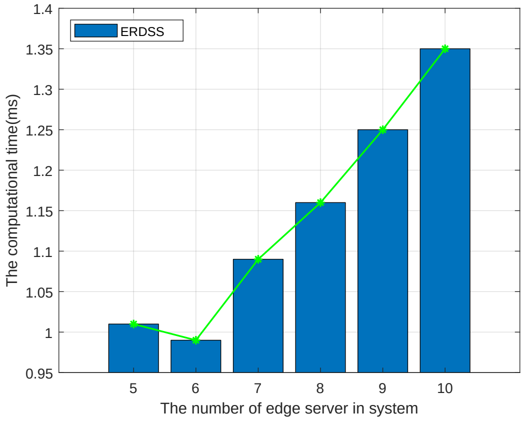


Fig. 9: Communication Delay

area is constant, and the number of ENs is too large. When the service provided for the SV is over-saturated, the data transmission will become the SV's burden.

In 2013, Cheng et al. [26] summarized and proposed solutions to the impacts of vehicle WiFi offloading caused by high physical mobility and fluctuation of mobile channels in the IoV, which provided ideas for our subsequent research on how to balance computation overhead and communication consumption.

We simulate the ERDSS from four aspects of index construction, trapdoor generation, search and attribute encryption, and plotted related data statistical graphs. The dataset used in the simulation is a dataset for a large movie review called "Learning Word Vectors for Sentiment Analysis". We realized the ERDSS in the compiler **Eclipse** using the **java** language(the main scheme) and compiler **Pycharm** using the **python** language(natural language process). The emulated hardware environment is Intel(R) Pentium(R) CPU G3220 3.00GHz, and the operating system is Win10 64bit.

1) **Index Construction**. In this part, we take the same condition as the input for the simulation of the three schemes, and time consumption as the output. For the size of data set changed from 500 to 3000 with the fixed size of dictionary to 500. The simulation results among three schemes are demonstrated as follows in figure.9.
   It can be observed from figure.9 that the cost of index construction in Wang et al. and Fu et al. increases linearly with the number of files in data set. Although the time consumption of ERDSS is slightly higher than the other two schemes, it also maintains a linear growth. From the experimental data, ERDSS could implement rapid index construction.
2) **Trapdoor Generation**. In the ERDSS, trapdoor generation is divided into two parts: weight calculation and trapdoor calculation, where the weight calculation simulation is realized using **Python** language. We adopt the same condition as the input for the simulation of the three schemes, and time consumption as the output. For the size of query set changed from 2 to 14 with the fixed size of dictionary and data set to 500 and 3000, respectively. The results are depicted in figure.10.
   The time overhead of the three schemes in figure.10 basically retains a linear increase. When the number of keywords in the query of the ERDSS changes from 6 to 10, a period of rapid growth of time cost occurs. Due to the introduction of NLP in the ERDSS, the time consumption of the ERDSS is higher than that of Wang et al. However, because of paillier encryption taking place of bilinear pairs in the ERDSS, which time consumption is lower than Fu et al.
3) **Search**. In order to comprehensively verify the search performance of the ERDSS, we performed two corresponding simulations in this part. We mainly explore the impact of data set and query set size on search efficiency. Figure.11(a) changes the size of data set on the premise that the dictionary size is fixed, while figure.11(b) is opposite to the condition in figure.11(a).

   It can be acquired from the data in figure.11(a) that the search overhead of the ERDSS not be significantly affected by the size of data set. However, the search overhead will grow linearly with the increase in the number of keywords in query. It can also be inferred from figure.11(a) and (b) that the principal factor affecting the search efficiency is the size of the query set, which primarily owe to the search mechanism in the ERDSS.
4) **Precision**. In order to quantify search precision, we define the true positive by t_p and f_p represents false positive, and the precision is calculated by $\frac{t_p}{t_p+f_p}$. Meanwhile, to compare the search accuracy of three schemes more intuitively, we plot the fuzzy and exact search accuracy data of them as two line charts in figure.13 and figure.14 respectively.
   As the most essential indicator of this kind of scheme, the search precision directly reflects the strengths and weaknesses of scheme algorithms. From figure.11, we can see that the fuzzy precision of the ERDSS and Fu et al. basically the same, and higher than Wang et al.. It is worth noting that the search precision in the ERDSS is directly related to the acc value, which is set $acc = 1$ in this part.
   Figure.12 shows the accuracy of the exact search of three schemes. The accuracy of Wang et al. and the ERDSS is basically the same and sightly higher than

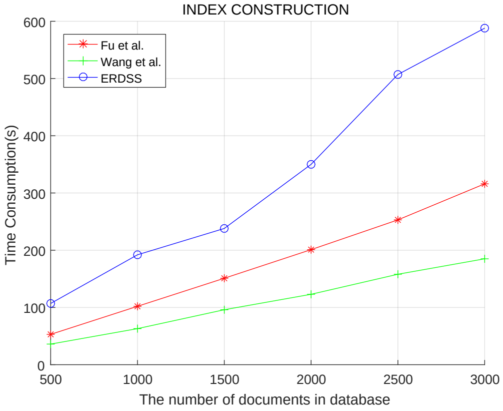


Fig. 10: Index Construction

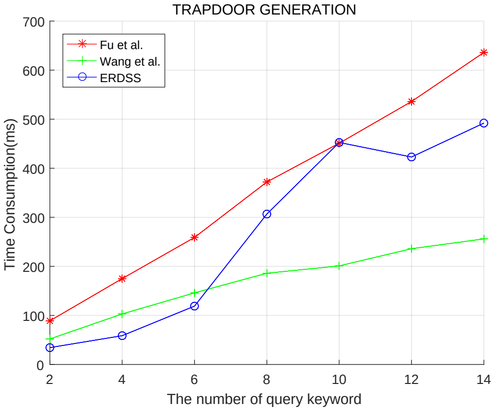
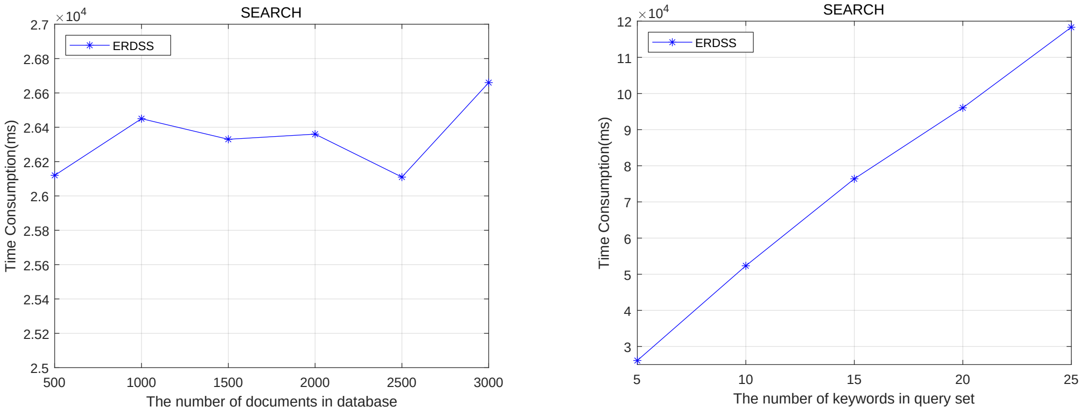


Fig. 11: Trapdoor Generation



(a)

(b)

Fig. 12: (a) For the size of data set varies from 500 to 3000 with the immobilized size of dictionary and query set to 500 and 5, respectively. (b) For the size of query set varies from 5 to 25 with the immobilized size of data set and dictionary to 1000 and 500, respectively.

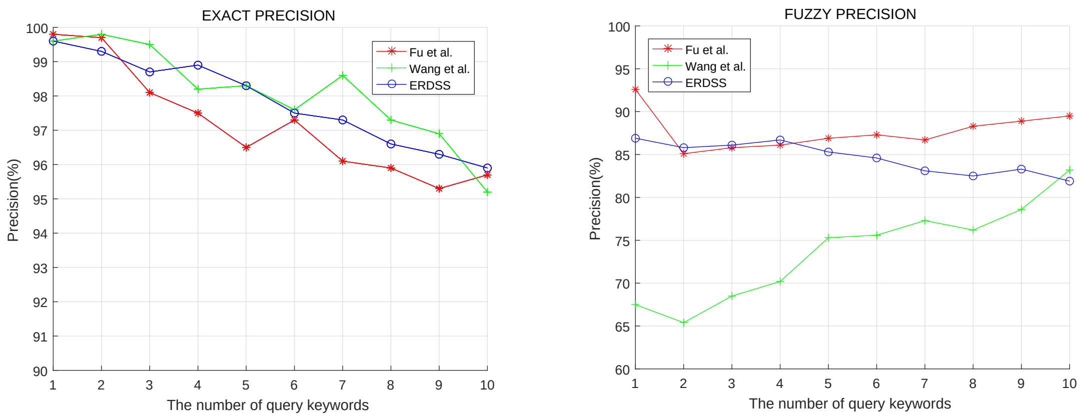
that of Fu et al. Simultaneously, it can see that as the number of keywords in query increases from 1 to 10, the precision of three schemes continues to decline. The ERDSS performs more accurate than the other two in both fuzzy and exact search, which also testifies that it is necessary to introduce NLP in the search stage to process the query set.

8 CONCLUSIONS

In this paper, we have investigated the problem existing in available IoV architectures, and derived SAGIVN architecture which in accordance with SAGIN. After identifying the security threats in SAGIVN , we put forward a scheme that supports fuzzy query of multiple keywords and secure data sharing on encrypted data. The ERDSS utilizes the dependency syntax and phrase structure tree in NLP to handle query requests, correct spelling errors in queries, calculate weights for query keywords, and execute secure result queries based on encrypted data index efficiently.

So as to ensure the security of the data sharing, the ERDSS integrates attribute encryption to realize the fine-grained management of the SV identity authority, avoiding the excessive communication overhead and delay caused by building large-scale encrypted broadcast channel. We also provide detailed security analysis and perform experiments using real data set in a simulated SAGIVN environment, which expounds the ERDSS's potential of practical usage.

The ERDSS principally proposes relevant solutions from the perspective of entity data storage and sharing in SAGIVN, but other aspects of security in SAGIVN also need to be guaranteed. Rory et al. [27] proposed a new network security research method for traffic detection and defense against network attacks in the Internet, which is also of great referential value for SAGIVN related scenarios. Meanwhile, Sun et al. [28], Liu et al. [29] and Lin et al. [30] respectively summarized the security threats existing in the Internet from the external, internal and software application perspectives and presented solutions, which can also provide practical assistance for SAGIVN's study in correlative field. Furthermore, communication as the fundamental and central part of SAGIVN, directly affects the overall structure and efficiency of SAGIVN. Wang et al. [31] reviewed the development of 6G technology in recent years from various aspects and summarized the security threats existing in 6G, which is also of guiding significance for the future research direction of SAGIVN.



(a)

(b)

Fig. 13: (a) Fuzzy Search (b) Exact Search

Acknowledgments

This work is supported by the National Key R&D Program of China(2017YFB0802300), the National Natural Science Foundation of China(No.61772403, U1836203), the Natural Science Foundation of Shaanxi Province(No.2019ZDLGY12-02), the Shaanxi Innovation Team Project(No.2018TD-007), the Xi'an Science and technology innovation plan(No.201809168CX9JC10) and National 111 Program of China B16037.

References

[1] MAI, A, SCHLESINGER, and D, "A business case for connecting vehicles," *Cisco Internet Business Solutions Group*, 2011.

[2] "The future economic and environmental costs of gridlock in 2030," 2014.

[3] Shen, Xuemin, (Sherman), Alhussein, Omar, Zhang, Ning, Zhuang, Weihua, and Y. and, "Software defined space-air-ground integrated vehicular networks: Challenges and solutions," *IEEE Communications Magazine Articles News & Events of Interest to Communications Engineers*, 2017.

[4] W. Zhang, L. Li, N. Zhang, T. Han, and S. Wang, "Air-ground integrated mobile edge networks: A survey," *IEEE Access*, vol. 8, pp. 125998–126018, 2020.

[5] K. Liolis, G. Schlueter, J. Krause, F. Zimmer, and A. Vanelli-Coralli, "Cognitive radio scenarios for satellite communications: The corasat approach," in *Future Network & Mobile Summit*, 2013.

[6] PerezNeira, I. Ana, Artiga, and Xavier, "Shared access terrestrial-satellite backhaul network enabled by smart antennas: Sansa," in *2015 European Conference on Networks and Communications*, 2015.

[7] M. Sina and C. Symeon, "Satellite network of experts (satnex-iv)." http://www.cttc.es/project/satellite-network-of-experts-iv/, 2017.

[8] L. Konstantinos, G. Alexander, S. Ray, S. Detlef, W. Simon, P. Georgia, E. Barry, W. Ning, V. Oriol, and T. J. a. Boris, "Use cases and scenarios of 5g integrated satellite-terrestrial networks for enhanced mobile broadband: The sat5g approach," *International Journal of Satellite Communications and Networking*, 2017.

[9] E. Lagunas, S. Chatzinotas, and B. Ottersten, "Carrier allocation for 5g integrated satellite-terrestrial backhaul networks," pp. 617–622, 2018.

[10] D. Xiaodong, S. David, and W. A. Perrig, "Practical techniques for searches on encrypted data," in *IEEE Symposium on Security & Privacy*, 2000.

[11] J. Li, Q. Wang, C. Wang, N. Cao, and K. Ren, "Fuzzy keyword search over encrypted data in cloud computing," in *INFOCOM, 2010 Proceedings IEEE*, 2014.

[12] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud," in *Infocom, IEEE*, 2014.

[13] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2706–2716, 2017.

[14] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in *2010 International Conference on Distributed Computing Systems, ICDCS 2010, Genova, Italy, June 21-25, 2010*, 2010.

[15] H. Dai, Y. Ji, L. Liu, G. Yang, and X. Yi, "A privacy-preserving multi-keyword ranked search over encrypted data in hybrid clouds," in *International Conference on Artificial Intelligence and Security*, 2019.

[16] K. Seny, P. Charalampos, and R. Tom, "Dynamic searchable symmetric encryption," in *ACM conference on computer & communications security*, 2012.

[17] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 1–1, 2015.

[18] R. Wilhelm, H. Seidl, and S. Hack, "Lexical analysis," *Compiler Design*, vol. 22, no. 2, p. N1, 2013.

[19] X. Lu, *Syntactic Analysis*. Springer Netherlands, 2014.

[20] D. Sarkar, *Semantic Analysis*. 2019.

[21] P. Bozek, A. Lozhkin, A. Galajdova, I. Arkhipov, and K. Maiorov, "Information technology and pragmatic analysis," *Computing and informatics*, vol. 37, no. 4, pp. 1011–1036, 2018.

[22] L. Zhou, L. Yu, S. Du, H. Zhu, and C. Chen, "Achieving differentially private location privacy in edge-assistant connected vehicles," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4472–4481, 2018.

[23] Q. Kong, R. Lu, M. Ma, and H. Bao, "A privacy-preserving sensory data sharing scheme in internet of vehicles," *Future Generation Computer Systems*, vol. 92, pp. 644–655, 2019.

[24] Y. Wu, L. P. Qian, H. Mao, X. Yang, H. Zhou, X. Tan, and D. H. Tsang, "Secrecy-driven resource management for vehicular computation offloading networks," *IEEE Network*, vol. 32, no. 3, pp. 84–91, 2018.

[25] J. Wang, X. Yu, and M. Zhao, "Privacy-preserving ranked multi-keyword fuzzy search on cloud encrypted data supporting range query," *Arabian Journal for ence & Engineering*, vol. 40, no. 8, pp. 2375–2388, 2015.

[26] N. Cheng, N. Lu, N. Zhang, X. S. Shen, and J. W. Mark, "Vehicular wifi offloading: Challenges and solutions," *Vehicular Communications*, vol. 1, no. 1, pp. 13 – 21, 2014.

[27] R. Coulter, Q. L. Han, L. Pan, J. Zhang, and Y. Xiang, "Data-driven cyber security in perspective–intelligent traffic analysis," *IEEE Transactions on Cybernetics*, vol. PP, no. 99, pp. 1–13, 2019.

[28] N. Sun, J. Zhang, P. Rimba, S. Gao, L. Y. Zhang, and Y. Xiang, "Data-driven cybersecurity incident prediction: A survey," *Communications Surveys & Tutorials, IEEE*, 2018.

[29] L. Liu, O. De Vel, Q. L. Han, J. Zhang, and Y. Xiang, "Detecting and preventing cyber insider threats: A survey," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2018.

[30] G. Lin, S. Wen, Q. L. Han, J. Zhang, and Y. Xiang, "Software vulnerability detection using deep neural networks: A survey," *Proceedings of the IEEE*, vol. PP, no. 99, pp. 1–24, 2020.

[31] M. Wang, T. Zhu, T. Zhang, J. Zhang, S. Yu, and W. Zhou, "Security and privacy in 6g networks: New areas and new challenges," *Digital Communications and Networks*, vol. 6, no. 3, pp. 281–291, 2020.

Haoyang Wang is a Master's student at the State Key Laboratory of Integrated Service Networks of Xidian University. He received his B.S. degree in software engineering from Northeastern University in 2018. Now he is studying for his M.S. degree in computer science from Xidian University. His research interests are cloud computing security, Searchable Encryption and IoV Security.

Kai Fan received his B.S., M.S. and Ph.D. degrees from Xidian University, P. R. China, in 2002, 2005 and 2007, respectively, in Telecommunication Engineering, Cryptography and Telecommunication and Information System. He is working as a professor in State Key Laboratory of Integrated Service Networks at Xidian University. He published over 70 papers in journals and conferences. He received 9 Chinese patents. He has managed 5 national research projects. His research interests include IoT security and information security.

Kuan Zhang is working as an assistant professor in Department of Electrical and Computer Engineering at University of Nebraska-Lincoln, USA. He received his B.S. and M.S. degrees from Northeastern University, P. R. China, in 2009 and 2011, respectively, in Communication Engineering and Computer Applied Technology. He received his Ph.D. degree from University of Waterloo, Canada, in 2016, in Electrical and Computer Engineering. He was a Postdoctoral Fellow from 2016-2017 at the University of Waterloo, Canada. He has published over 50 papers in journals and conferences. He was the recipient of Best Paper Award in IEEE WCNC 2013 and Securecomm 2016. His research interests include cyber security, big data, cloud/edge computing.

Zilong Wang received the B.S. degree from Nankai University, China, in 2005, and the Ph.D. degree from Peking University, China, in 2010, both in Mathematics. During 2008-2009, he was a visiting Ph.D. student in the Dept. of Electrical and Computer Engineering, University of Waterloo, Canada. He received a Postdoctoral Fellowship from University of Waterloo in 2012. Since 2010, he has been with the State Key Laboratory of Integrated Service Networks, Xidian University. He is currently a professor in school of Cyber engineering, Xidian University, Xi'an, China. His research interests are in the areas of sequence design, cryptography and information security.

Hui Li was born in 1968 in Shaanxi Province of China. In 1990, he received his B. S. degree in radio electronics from Fudan University. In 1993, and 1998, he received his M. S. degree and Ph. D. degree in telecommunications and information system from Xidian University respectively. He is now a professor of Xidian University. His research interests include network and information security.

Yintang Yang was born in 1962 in Hebei Province of China. He received his Ph.D. degree in semiconductor from Xidian University. He is now a professor at Key Lab. of Minist. of Educ. for Wide Band-Gap Semicon. Materials and Devices of Xidian University, Xi'an China. His research interests include semiconductor materials and devices, network and information security.