Research on Searchable Encryption Scheme for Internet of Vehicles Based on Trusted Authorization and Keyword Dynamic Update

Pengshou Xie, Xiaoye Li, Yongping Kang, Jiafeng Zhu, Wanjun Shao, and Cunhuan Tan

(Corresponding author: Xiaoye Li)

School of Computer and Communications & Lanzhou University of Technology
No. 36 Peng Jia-ping Road, Lanzhou, Gansu 730050, China
Email: 976339400@qq.com

(Received Oct. 23, 2023; Revised and Accepted May 20, 2024; First & Second Online Oct. 16 & Nov. 26, 2024)

Abstract

To solve the problem of dynamic updating of Internet of Vehicles data in search services and the communication and computational overheads of data sharing users, and to address the complex update process in existing search schemes concerning dynamic updating of data and keywords, a study of a searchable encryption scheme for Internet of Vehicles based on trustworthy authorization and dynamic updating of keywords is proposed. The file information is classified and stored by the K-mean clustering algorithm in the trusted authorization organization to reduce the storage overhead, combined with the broadcast encryption mechanism to provide appropriate permissions for each group of users in the encryption phase; Generate index vectors and query vectors by locally sensitive hash hashing and Bloom Filter, construct index tree and utilize Top-k algorithm to improve efficiency and file updates given k files related to a query; Evaluate the relevance of documents to queries with the help of TF-IDF. Based on the difficulty assumption, the security of this scheme can be proved under an adaptive keyword selection attack. The experimental results show that this scheme realizes dynamic keyword updating and satisfies multi-client multi-keyword search ranks the results based on access control, improves the retrieval efficiency, and reduces the computational overhead.

Keywords: Internet of Vehicles; Keyword Retrieval; Searchable Encryption; Trusted Authorization

1 Introduction

Cloud services have now become a common and economical platform for data outsourcing and sharing, and most users (both business and individual users) use cloud services for data storage and management. However, when data is migrated to cloud servers, users lose direct control over the data. To secure the data, encryption is applied to the data before outsourcing to ensure the privacy of the data [**?**, **?**]. However, by outsourcing the data in encrypted form, how to access the data efficiently becomes a new challenge, thus giving rise to searchable encryption [**?**].

In practical application scenarios, searchable encryption techniques need to provide users with appropriate search permissions so that different users can access different files. The most basic solution is that the data owner encrypts different files using different keys, after which the data owner gives each user the encryption/decryption keys for all the files authorized for searching, which is cumbersome and inefficient [**?**]. In Internet of Vehicles applications, users store huge amounts of data in the IoV cloud servers to save local resources and also utilize the cloud storage to retrieve the required data [**?**], however, there are also data security problems while bringing convenience. The cloud storage part of the enterprise or organization the cloud to establishes its hierarchical system [**?**], to ensure that users can efficiently search for data at the same time, but also for different users to carry out hierarchical control so that different levels of users can only retrieve the information under the level of data to which they belong to, but the data stored in the cloud at any time may be leaked to the cloud service provider, the security problems caused by such user privacy data leakage Such security issues caused by user privacy data leakage are endless and cause serious consequences to users. In the Internet of Vehicles environment, due to the wide range of data applicability, there is the problem of data waste and leakage caused by the cross-use of data among Internet of Vehicles users. To protect the security of Internet of Vehicles data, Internet of Vehicles users usually encrypt the data before outsourcing it to cloud servers, but how to manipulate the ciphertext domain after encryption becomes a new challenge [**?**, **?**, **?**]. Existing research uploads the ciphertext data shared to the user along with the encrypted keyword list to the

cloud server at the same time, and the user generates a keyword trapdoor, which is utilized to directly perform keyword searches on the ciphertext on the cloud server, without the need to download the ciphertext file locally and decrypt it before searching it, which effectively improves the speed of operating the data on the cloud under the premise of realizing the protection of the private data and frees up the user's local resource space [**?**, **?**], but there is still room for improvement in user authorization and search efficiency.

In this paper, in response to the above problems, based on trusted authorization and dynamic update of keywords, the results obtained from the research on the Internet of Vehicles search service scheme will have a theoretical contribution to the dynamically updated keyword retrieval in the Internet of Vehicles environment and will have a propulsive effect on enhancing the security of the Internet of Vehicles [**?**], popularizing the Internet of Vehicles applications, and improving the quality of people's lives.

2 Related Theories

2.1 Searchable Encryption

Suppose that the Internet of Vehicles user wants to store his files in an honest but curious external server. To effectively protect the privacy of their files, the Internet of Vehicles user chooses to encrypt their files before storing them. If traditionally grouped passwords are used to encrypt the files, this restricts the ability to decrypt the files to only the key owner, which means that when an Internet of Vehicles user performs a keyword-based search query, he or she must first download all the ciphertext files that have been uploaded and retrieve them after completely decrypting them, which is extremely inefficient. Searchable Encryption (SE) [**?**] is a technology that has both encrypted data storage and can be directly retrieved on the ciphertext domain, and its typical application is cloud storage. SE utilizes the strong computational power of cloud servers to perform keyword retrieval of ciphertexts, and the technology does not disclose any user's privacy to the servers, which not only enables the data user's privacy to be protected but also retrieval efficiency is greatly improved because of the server's computational power of the server is greatly improved [**?**].

To realize ciphertext searching of outgoing packet data, basic searchable encryption schemes usually contain the following basic algorithms [**?**].

1) KeyGen: The KeyGen algorithm is used to initialize the system, enter security parameters, the encryption method for generating indexes and type of file encryption, and generate keys. If symmetric encryption is used to generate the index, the KeyGen algorithm is used to generate the set of symmetric keys. If public key encryption is used, the public key and secret parameters are generated.

2) BuildIndex PK, KW: BuildIndex Algorithms are used to generate searchable encrypted indexes corresponding to the document keyword collection KW. DO extracts the keywords from the document collection, encrypts the generated indexes with the key PK, and outsources the encrypted keyword indexes to the server (in some cases, the indexes include relevance scores from the keyword list). At the same time, the documents are encrypted and outsourced to the cloud server. If symmetric encryption is used, the symmetric key K is entered.

3) TrapGen SK, KW: Authorized DUs use the TrapGen algorithm to generate search traps corresponding to the keywords they are interested in. Enter the private keys SK and KW and output the trapdoor TR. In some cases DO may generate a search trapdoor on behalf of the DU. If a symmetric key is used, the algorithm inputs the symmetric key K.

4) SearchText PK, I, TR: The SearchText algorithm for the CS receives the search trapdoor TR, compares the TR with the keyword ciphertext in index I, generates a list of matching documents, and finally sends the matching documents to the DU. In addition, the retrieved documents can be sorted if the index carries relevance score information.

2.2 Broadcast Encryption

The model of a broadcast encryption scheme generally consists of the following 4 components.

1) System Establishment Setup (r, u): taking the security parameter λ and the maximum number n of users receiving broadcast messages as inputs, it outputs the system master key K_0 and the system public key set K_1, which K_0 is kept secretly by the private key generation center.

2) Private key extraction Extract (K_1, K_0, $\mathrm{d_i}$): The public key K_1 , the master key K_0 and the user identity d_i are taken as input and the corresponding private key Y_i of the user is output.

3) Broadcast encryption Encrypt (K_1, S): Collaborate the public key K_1, the set of broadcast message recipients $S = \{d_1, d_2, .., d_\mathrm{s}\} s \leq n$ as input and output (h_k, k_2), where $\mathrm{h_k}$ is the ciphertext header and $\mathrm{k_2}$ is the session key. When the sender wants to broadcast the message M, the plaintext M is encrypted to C_k using the session key $\mathrm{k_2}$ and assembled into the final ciphertext $C'k = (h_k, S, C_k)$, where (h_k, S) is the complete ciphertext header and C_k is the ciphertext body.

4) Broadcast decryption Decrypt$(s, d, Y_\mathrm{i}, h_k)$: Take as input the ciphertext header(h_k, S), the public keyK_1, the receiver's identity d, and its private keyY_i. If $\mathrm{d} \in \mathrm{S}$ so, output the session key $\mathrm{k_2}$. When the receiver wants to decrypt the ciphertext body C_k, it

decrypts C_{k} using the session key k_2 obtained above and outputs the message plaintext M.

2.3 Clustering Algorithm

Clustering is used to categorize documents into different clusters before creating an index. Clustering of documents should reduce the number of documents in each cluster to increase the efficiency of the search process. To perform clustering, the data owner collects all the documents into a dataset $F = \{f_1, f_2, ..., f_n\}$.Then, the data owner applies the k-means algorithm on F to generate k clusters, which are $L = \{l_1, l_2, ..., l_m\}$. $\mathrm{l_t}$ is a set of similar documents, where $\mathrm{t} = 1, 2, ..., k$. This means that all documents in the same cluster should be as similar as possible, and documents in different clusters should be as dissimilar as possible to documents in all other clusters. The k-means algorithm as a divisive clustering algorithm is highly applicable for most of the data. The algorithm is relatively scalable and computationally simple and efficient; the space complexity is low and linear. The Single-Pass incremental clustering algorithm is sensitive to the order of the data at the time of data input as compared to the Single-Pass incremental clustering algorithm which is also a divisive clustering algorithm. Hierarchical-based clustering algorithm has higher time complexity as compared to the k-mean algorithm [**?**].

2.4 Linear Hypothesis

Bilinear pair Definition [**?**]: G and G_T are two cyclic groups of order p. The existence of a mapping e:$G \times G \rightarrow G_T$ on G and G_T should have the following properties:

1) Bilinear: For any $a, b \in Z_P$, g $\in G$, remain $e(g^a, g^b) = e(g, g)^{ab}$.
2) Countability: For any $a, b \in Z_P$, g $\in G$, there exists a polynomial time algorithm capable of computing.
3) Non-degeneracy (assume propositions without loss of generality): For any element g, $\mathrm{e}(g, g) \neq 1$ in G, where 1 is a unit element in the group G_T.

2.5 Inverse Text Frequency TF-IDF

TF-IDF is a common technique for information retrieval, which is mainly used to weigh common keywords in documents.TF-IDF is commonly used in the field of information retrieval as a statistical method to assess the importance of a word to a document set or one of the documents in a corpus, so this technique is more applicable to top-k ranked search [**?**]. In this technique, it is not only the frequency of occurrence of words that is of interest, because for example, the words "the" appear frequently in the text, but they do not have much significance. Therefore, the importance of a word in the TF-IDF technique does not only increase proportionally to the number of times it appears in the document but also decreases inversely to the frequency of its appearance in the corpus.

3 System Model and Algorithm Formal Definition

3.1 System Model

Traditional encryption techniques can encrypt data to ensure that it is not leaked or tampered with during storage and transmission. However, in an Internet of Vehicles scenario, a large amount of data often needs to be retrieved, and traditional encryption techniques cannot directly perform search operations on encrypted data. Searchable encryption provides a solution to search and query encrypted data without exposing plaintext content. The traditional searchable encryption system model is shown in Figure **??**, specifically including the following key elements:

1) Encrypted data structure: Searchable encryption techniques use specific data structures to organize and store encrypted data to support search operations. Common data structures include symmetric searchable encryption, public-key searchable encryption, etc [**?**, **?**].
2) Search algorithm: Searchable encryption techniques will define search algorithms that allow specific search operations to be performed on encrypted data, such as searching by keywords, filtering by attributes, and so on.
3) Cryptographic protection: Searchable encryption techniques use appropriate encryption algorithms to encrypt and protect the data to ensure the security of the data during storage and transmission. Common encryption algorithms include symmetric encryption algorithms (e.g. AES), asymmetric encryption algorithms (e.g. RSA), etc. [**?**].

The searchable encryption for car networking realizes the protection of sensitive data while searching and querying operations on encrypted data to improve the data security and privacy protection level of the car networking system. This technology has a wide range of application prospects in the field of the Internet of Vehicles and can protect the security of sensitive data such as vehicle location information and owner's personal information.

The scheme model of this paper consists of several parts, including data owner, Internet of vehicles cloud server, data user, and trusted authorization organization, and through the mutual collaboration of these parts, it can realize the scope-qualified search between user groups, and through the policy configuration, it can authorize the search scope owned by the Internet of vehicles user group to legitimate users to conduct the search query, and the system model of this scheme is shown in Figure **??**.

1) Data owners: In this system is a group of Internet of Vehicles users with a plurality of sub-user groups, the Internet of Vehicles user is a data owner or can be a data user, and at the same time, the data owner

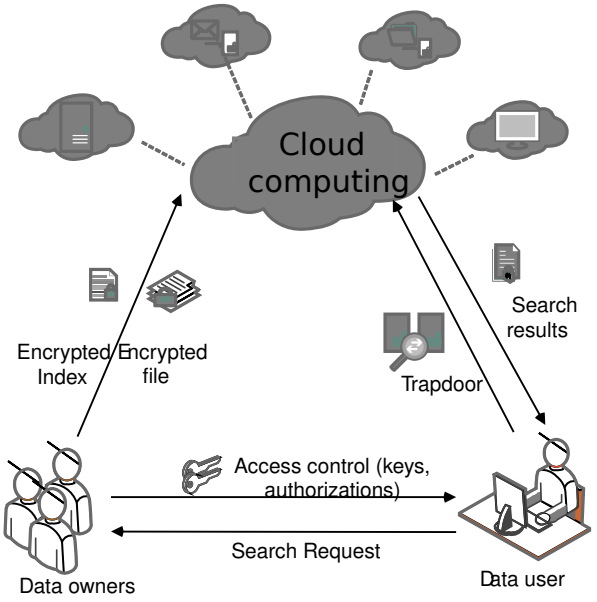

Figure 1: Model of traditional searchable encryption system

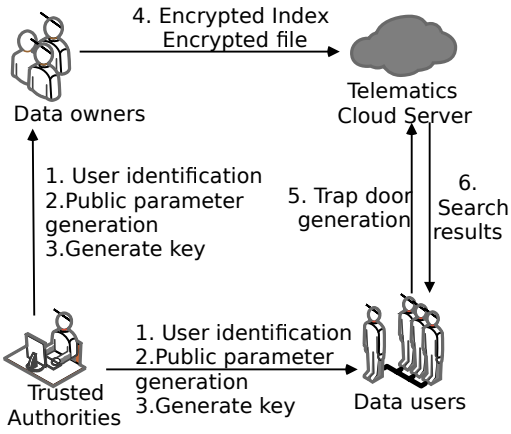

Figure 2: Dynamic keyword model of searchable encryption system for the Internet of Vehicles

is also the owner of the access policy configuration of the group of user groups, the data user submits the access policy request to the member of the data owner, and the member of the data owner configures the corresponding privilege information for the applicant of the Internet of Vehicles data and updates it in an index.

2) Internet of Vehicles Cloud Server: The cloud server of the Internet of Vehicles is mainly responsible for storing keyword indexes, storing ciphertext files, and performing search functions in the system. The cloud server does not have the relevant key, so the cloud server is not able to obtain any information related to the plaintexts.

3) Data users: Data users can search the corresponding secret files by their private keys, keywords, and access policies.

4) Credible Authorized Organizations: Trusted authorization organizations are trusted to classify users using policies for access control, generating public parameters, and distributing keys for legitimate users.

3.2 Formal Definition

In this paper, we design a searchable encryption scheme for the Internet of Vehicles based on trusted authorization and dynamic updates of keywords, authorize the corresponding roles of users and their owned search scope through access policy configuration, and combine with the broadcast encryption mechanism to realize encrypted file sharing, allow users to search for keywords within a subset of files they are authorized to access, divide the data into several clusters through clustering algorithm, and take the keyword set of each cluster as the user's access Permissions. The interaction of the algorithms can be visualized through the following formal definitions. Search execution process.

1) KeyGen(ε): The key generation algorithm is executed by a trusted authorization organization and inputs a security parameter ε and the number of the Internet of Vehicles users n. It outputs the public parameters of the system p, the master private key SK, and the master public key PK.

2) Accredit(U): The authorization algorithm is run by the Trusted Authorization Organization and uses access control to classify the Internet of Vehicles users into different groups denoted by U_{n} and generates a unified user identity and creates a unified Internet of Vehicles user identity U_{id} information table (consisting of U_{id}) on the cloud server.

3) BuildIndex(D, W, PK): The index generation algorithm takes as input the system-generated public parameters p, the owner of the Internet of Vehicles data U_{id}, the keyword information W contained in the file set D, and the policy configuration information I. Before creating the Index the file information is categorized and stored by applying the K-means clustering algorithm into different clusters to categorize the keywords and store them, and the hidden structure is used to interrelate the files that contain the same keyword by selecting an independent LSH function H=$\{\mathrm{H} : \{0,1\}^{160}\}$ from the Locally Sensitive Hash Hash (LSH) family. An m-bit Bloom filter ID is constructed as an index for each file d . The index tree TE is used to evaluate the relevance score of each file. Subsequently, all the individual indexes are integrated into an index tree T, which creates encrypted index vectors in each node and finally returns the encrypted index tree TE. The relevance scores of

the files are evaluated according to the TF-IDF rule, which in turn sorts the files.

4) Encrypt(SK, T): The encryption algorithm traverses the binary tree T and encrypts all vectors stored in the corresponding node u with the key SK to generate the ciphertext CK.

5) Trapdoor(SK, Q): This trapdoor construction algorithm inputs new information about the access policy configuration of the Internet of Vehicles user, the query keyword W_{i}, and the user's private key SK_{i} to generate the keyword trapdoor Tr(w). An m-bit long Bloom filter is generated for query Q. For each query keyword A, the same LSH function $h_j \in H$, $1 \leq \mathrm{j} \leq l$ is used to map to the Bloom filter to construct a Bloom query vector Q, which is encrypted to finally return the query vector trapdoor TD.

6) Search(TD, TE): This query algorithm inputs trapdoor and index, policy configuration information I, query the corresponding keywords, and performs policy configuration identification to return the k-file clusters of legitimate access ranges to the search user. Member U_{AX} of the Internet of Vehicles cluster user U_A, who applies for the search range owned by the Internet of Vehicle cluster user U_B will be rejected. The conditions for permission generation are determined by the trusted authorization organization. The cloud server executes the search algorithm during the search process and returns the top k encrypted files with the highest relevance score to the user after a successful match.

7) Decrypt(SK, CK): This decryption algorithm inputs a text key to decrypt the returned result CK to get a plaintext document of the search result.

8) Update(CK, TD, TE): This updated algorithm updates the index tree while updating the encrypted file data, which enables adding and deleting data information to achieve dynamism.

3.3 Security Model

The main idea of the security model is that the adversary is unable to obtain more valid information through partial information or is unable to access information that is beyond his authority [**?**]. The scheme is said to be Adaptive Choice of Keyword Attack (IND-CKA) secure if there is no polynomial-time adversary A, which can win the following game with a non-negligible advantage, and the adversary is unable to obtain valid information about the keyword corresponding to the trap through a known legitimate trap Tr(w).

System initialization: challenger C executes the KeyGen algorithm and generates the public parameter P. P is sent to adversary A. Challenger C secretly saves SK, and PK.

Inquiry and Challenge Stage 1: In this phase adversary A adapts to polynomially bounded subsubsequent queries. Adversary A randomly chooses a keyword w to generate a trapdoor to initiate a query to challenger C. The Internet of Vehicles Cloud Server computes the trapdoor Tr(w) for keyword w via the trapdoor generation algorithm Trapdoor and returns the computed result to Adversary A.

Inquiry and Challenge Stage 2: Adversary A again initiates a query and challenge on the ciphertext and the index, but the keyword initiating the query and challenge cannot be the keyword that appeared in the query and challenge Stage 1 and returns the computed trapdoor to Adversary A.

Interrogation stage 3. Adversary A again randomly selects one of the two keywords and initiates the query, where the queried keyword cannot be a keyword that has already been queried and is represented in a simple representation as: W_η, where $\eta \in (0,1)$, encrypted using the key of adversary A, is sent to the cloud server.

Output: Adversary A outputs guess $\eta^{'} \in (0,1)$. If $\eta^{'} = \eta$ Adversary A's guess is correct, then Adversary A is in an advantageous position throughout the game challenge, and the probability of winning this game is: $\Im = \left|\Pr\left[\eta^{'} = \eta\right] - \frac{1}{2}\right|$; otherwise, Adversary A loses.

However, based on the assumption that the bilinear problem is difficult, the advantage $\Im$ of winning the game at any polynomial time adversary A is a negligible function, then the search trap of this scheme satisfies trap indistinguish ability under adaptive keyword selection attack and the key information is secure.

4 Program Structure

Existing keyword search schemes are mainly based on single-keyword, multi-keyword [**?**, **?**], and fuzzy keyword search problems to unfold, and there are relatively few researches on dynamically updated keyword retrieval and high computational overhead. Therefore, we need to solve the problem of how to divide the storage files by K-mean clustering to solve the overhead problem; how to construct the index vector by bitmap locally sensitive hashing LSH to complete the process of dynamically updating the data as well as the keyword dynamic updating process, and reduce the occupied space to accelerate the search efficiency; how to construct the query vector by mapping the keyword to the Bloom filter by using the locally sensitive hashing LSH, which can ensure the search efficiency and accuracy; how to utilize the broadcast encryption mechanism to effectively control the access rights of Internet of Vehicles users in the encryption phase; how to utilize the TF-IDF to evaluate the relevance of documents and

queries; how to combine the indexing tree and top-k algorithm to improve the efficiency and document updating is the key of this paper. A searchable encryption scheme for the Internet of Vehicles based on trusted authorization and dynamic update of keywords is realized by improving the traditional searchable service scheme. The scheme is constructed as follows:

1) Key generation algorithm KeyGen(ε,n)→ (p,sk,pk)

   The safety parameter ε and the number of connected Internet of vehicles users n are used as input parameters to the system, G and G_T are cyclic groups of order prime q, g is a generating element of the group, and is a bilinear mapping: $G \times G \to G_T$.After that, a collision-free hash function $H\{0,1\}^* \to Z_{\mathrm{q}}^*$ is chosen randomly, the private key sk $\in Z_P^*$ of the system, and the public key of the system is $PK_{\mathrm{s}} = \mathrm{g}^{\mathrm{s}}$, and the public parameter P of the system is obtained as (G,G_T,g,e,PK_{s}, H), and P is disclosed to all users in the system. Input the public parameter P of the system and randomly select $z_1, z_2 \in Z_q^*$ such that $sk_1 = z_1$,$\mathrm{sk}_2 = z_2$. Calculate,$PK_1 = \mathrm{g}^{\mathrm{z}_1}$,$PK_2 = \mathrm{g}^{\mathrm{z}_2}$ Send ($\mathrm{sk}_1 PK_1$), (sk_2,PK_2) as a public-private key pair to the owner of the Internet of vehicles data and the data user respectively.

2) Index Building Algorithm BuildIndex (P,U_{id},W,I,PK) → TE

   This index generation algorithm inputs the system-generated public parameter p, the Internet of vehicles data owner U_{id}, the keyword information W contained in the file set D, and the policy configuration information I.The document information is categorized before creating the index and divided into different clusters by clustering. The Internet of Vehicles data owner collects all the document information into a data set $F = \{f_1, f_2, ..., f_n\}$. By K-mean algorithm data set F generates k clusters, which is $L = \{l_1, l_2, ..., l_m\}$ is a set of similar documents, where $t = 1, 2, ..., k$. This means that all documents in the same cluster should be as similar as possible and documents in different clusters should be as dissimilar as possible to documents in all other clusters. The K-means clustering algorithm is described as follows: ① Randomly select k documents as the center of mass. ② Calculate the Euclidean distance from each document to each center of mass. ③ Assign the documents to the closest cluster. ④ Recalculate the centers of mass of each category after the assignment is completed and observe whether the clustering converges or not. If it converges, the clustering algorithm is completed; otherwise, continue to perform steps ② to ④.

   By selecting a separate LSH function from the locally sensitive hash LSH family H=$\{\mathrm{H} : \{0,1\}^{160}\} \to \{\{0,1\}^{\mathrm{m}}\}$. An m-bit Bloom filter ID is constructed as an index for each file d. The indexes are then integrated into an index tree T . Subsequently, all individual indexes are integrated into an index tree T . ① Extract the keyword set $W_D = \{W_1, W_2, ...\}$, $W_{\mathrm{i}} \in \{0,1\}^{160}$ from the file set D. ② For each keyword, insert the keyword weights into the index IDs using $\mathrm{h_j} \in H, 1 \le \mathrm{j} \le l$. ③ Construct the index tree using a single index $\{I_{D1}, I_{D2}, ...I_{D\mathrm{n}}\}$, create encrypted index vectors in each node, and finally return the encrypted index tree TE. evaluate the relevance scores of the files according to the TF-IDF rule, and thus sort the files.

3) File encryption algorithms Encrypt(SK,T) → CK

   The ciphertext CK is generated by traversing the binary tree T and encrypting all the vectors stored in the corresponding node u with the key SK. The owner of the connected Internet of Vehicles data generates two random vectors $\{D_{U1}, D_{U2}\}$ for the index vectors D_U in each node u based on the secret vectors S. Specifically, $D_{U1}[\mathrm{i}] = D_{U2}[\mathrm{i}] = D_U[\mathrm{i}]$, if S[i] = 0; otherwise $D_U[\mathrm{i}] = \frac{12}{D_U}[\mathrm{i}] + \mathrm{r}.D_{U2}[\mathrm{i}] = \frac{12}{D_{U2}}[i] - r$ and r is a random number. Then $\{D_{U1}, D_{U2}\}$ is encrypted with $\{M, M_2\}$. Finally, the encrypted index vector : I_U:$I_U = (M_1^T D_{U1}, M_2^T D_{U2})$ is created in each node.

4) Trapdoor construction algorithm Trapdoor(SK,Q) → TD

   An m-bit long Bloom filter is generated for query Q. For each query keywordW_{i}, the same LSH function $\mathrm{h_j} \in H, 1 \le \mathrm{j} \le l$ is used to map to the Bloom filter to construct a Bloom query vector Q. The query vector Q is divided into two vectors $\{Q_1, Q_2\}$.

   Where $S[\mathrm{i}] = 0$,$Q_1[\mathrm{i}] = \frac{12}{Q}[i] + r$, $Q_2[\mathrm{i}] = \frac{12}{Q}[i] + r$, r is a random number; otherwise,$Q_1[\mathrm{i}] = Q_2[\mathrm{i}] = Q[\mathrm{i}]$, $\{Q_1, Q_2\}$ and$\{M, M_2\}$ are encrypted to $TD = \{M_1^{-1}Q_1, M_2^{-1}Q_2\}$. The encryption finally returns the query vector trapdoor TD.

5) Query Search Algorithm Search(TD,TE) → CK_K

   Based on the generated trapdoor with index and policy configuration information I, the corresponding keywords are queried, and the policy configuration identification is executed to return the k file clusters of the legal access range to the search user. The member U_{AX} in the Internet of Vehicles cluster user U_A, applying for the search range owned by the Internet of Vehicles cluster user U_B.will be rejected. The conditions of permission generation trusted authorization organization determines. The cloud server executes the search algorithm during the search process, and the cloud server returns the top k encrypted files $CK_K$with the highest relevance score to the user after the successful matching of the ciphertext CK of the categorized storage files, the trapdoor TD, and the user's set U_{id}.

6) File decryption algorithm Decrypt(SK,CK) → FM

The connected Internet of Vehicles data user decrypts the returned result CK with a text key to get a plaintext document FM of the search results.

7) File update algorithm Update(CK,TD,TE)→ u_t

To facilitate this update and reduce communication costs, the Internet of Vehicles data owner retains the unencrypted index tree locally to generate updates and send them to the cloud. When updating encrypted cloud data, the index tree is synchronized and updated. Denote as the set of nodes that ma,y be changed during the update process. For example, if the file is f_2 deleted, then the subtree will be updated and$u_t = \{r, r_{11}, r_{12}, f_2\}$. Use$uodate \in \{delete, insert\}$. The Internet of Vehicles data owner sends a new file message requesting the Internet of Vehicles cloud server to add or delete a piece of specific file information, this message contains the ciphertext to be added, the updated index, and finally the updated information.

In the design of this scheme, Bloom filters can be used to check whether a given element is contained in a set or not. Bloom filters have high spatial and query efficiency, but they also have some false alarm characteristics [**?**]. In effect, the Bloom filter is an array of m bits, where each bit is initially set to 0. From $H = \{h_i | h_i : s \rightarrow [1, m], 1 \leq i \leq l\}$, there exists a set $S = \{a_1, a_2, ..., a_n\}$ and 1 independent hash function. For each element a_k in S, use l functions to map it to 1 position in the Bloom filter and change the value of position $h_i(a_k)$from 0 to 1. To determine whether an element a is in the set S, we can use the same method to map it to 1 position in the Bloom filter. If at least one of the positions is 0, then $a \in S$. Otherwise, $a \notin S$. or a false positive is generated, but the false positive is small. For example, for the m-position Bloom filter, the false positive is $(1 - e^{-\frac{\ln}{m}})^l$.

5 Program Analysis

5.1 Security Analysis

If the Decisional Bilinear Diffie-Hellman problem is hard in polynomial time, the proposed scheme in this paper for car search service based on trusted authorization and dynamic update of keywords is satisfying the Adaptive Choice Keyword Attack (IND-CKA) security.

Assume: that adversary A is a probabilistic polynomial time attacker. Challenger C can solve the Decisional Bilinear Diffie-Hellman problem by a probabilistic polynomial time algorithm. Under the DBDH assumption, Challenger C can obtain $(G, G_T, g, e, pk_s.H)$.

Proof.

Initialization setup: the challenger C gets the system's public parameters P to be $(G, G_T, g, e, pk_s.H)$. Then, randomly choose a collision-free hash hash function $H\{0,1\}^* \rightarrow Z_q^*$, private key sk $\in Z_P^*$, and public key as $PK_s = g^s$. Input the system public parameter P. Randomly choose $z_1, z_2 \in Z_q^*$, such that $sk_1 = z_1$,$sk_2 = z_2$. Compute $PK_1 = g^{z_1}$,$PK_2 = g^{z_2}$and send P, $(sk_1 PK_1)$, $(sk_2 PK_2)$, to the adversary A.

Query Phase 1: Attacker A adaptively issues the following query request.

Adversary A submits at most q_h keyword queries to the randomized prediction machine, and challenger C executes the Trapdoor algorithm, which returns the trapdoor corresponding to the keyword to adversary A. Adversary A stores a list H_{list} of hashes.

$$H_{list} : \{w_i, coin_i; h_i, d_i, f_i, e_i; p_i\}$$

If a keyword w_i has already been queried, the hash value of that keyword is returned to the adversary A.

$$h_i = H_1(w_i), f_i = H_2(w_i), p_i = H_3(w_i)$$

If the keyword w_i is not queried, the challenger C chooses a random value $p_i \in Z_P^*$ for p_i and flips a random coin $coin_i \in \{0, 1\}$. This is followed by the generation of $h_i = g_1^{d_i}$,$f_i = g_1^{e_i}$. Finally, the challenger C returns the obtained h_i, f_i, p_i to the adversary A and adds $\{w_i, coin_i; h_i, d_i, f_i, e_i; p_i\}$ to the list H_{list}.

Inquiry Phase 2: An adversary A adaptively queries a series of trapdoors for a collection of keywords. Let the set of keywords for a particular query be W and the returned trapdoor be Tr(w). The challenger C obtains the corresponding tuples of the query keywords in the list H_{list}. If there is at least one $coin_u = 1$, C chooses to abstain; if all $coin_u$ are 0, C generates a random number $t \in Z_P^*$, and t is updated in the next successful trapdoor query. Subsequently, C outputs the trapdoor T.

$$T = \{T_a, T_b, T_c, q_1, q_2..., q_u\}.$$

Among these, $T_a = g_1^t, T_b = g_1^{t\left(\sum_{u=1}^{m} d_u\right)}, T_c = g_1^{t\left(\sum_{u=1}^{m} e_u\right)}$,Feed the trapdoor T back to the adversary A.

Challenge Phase 1: Adversary A selects a keyword set W^* and sends it to challenger C and additionally selects a keyword set R*. The keyword set R* and the keyword set W^* cannot appear in the query before the challenge by adversary A. Challenger C sets $W_0 = W^*$ and $W_1 = R^*$ and chooses a random bit $\eta \in \{0, 1\}$ to get the keyword set W_η.

$$W_\eta = \{w_{\eta,1}, w_{\eta,2}, ..., w_{\eta,m}\}$$

Challenger C asks the randomized prediction machine for all the keywords in W_η one by one and returns the corresponding tuple of H_{list}. If $coin_{\eta,u} = 1$

does not exist, then challenger C abstains. Instead, challenger C generates the challenge ciphertext S_η and sends the tuple (W_0, W_1, S_η) to adversary A.

Challenge Phase 2: Adversary A continues with a series of trapdoor queries for a set of keywords that cannot be W_0 and W_1. Challenger C performs the same action as in Challenge Phase 1, outputting the trapdoor Tr(w) and feeding it back to Adversary A.

Speculation phase: Eventually, adversary A outputs the result $\eta^{'} = 0$ or $\eta^{'} = 1$ as the judgment of challenger C on the random value. If $\eta = \eta^{'}$, adversary A wins the game, and vice versa, adversary A loses the game.

$$\Im = Adv_A(\lambda) = \left|\Pr(\eta = \eta^{'}) - \tfrac{1}{2}\right|$$

However, since the problem is assumed to be hard, then the probability $\Im$ that the adversary wins the game in polynomial time is negligible, so this paper's scheme is IND-CKA safe under the stochastic prediction model. □

5.2 Functional Analysis

The functions realized by the scheme in this paper are compared and analyzed with the schemes in literature [**?**], literature [**?**], and literature [**?**], as shown in Table **??**. The scheme in literature [**?**] can satisfy the access control function and multi-client settings, but it cannot realize multi-keyword search and keyword search result ranking and dynamic update of keywords, which is insufficient in search efficiency. The scheme in Literature [**?**] supports multi-keyword search and can rank the search results and realize the dynamic update of keywords, but does not have the access control function and multi-client settings. The scheme in literature [**?**] realizes multi-client settings and access control through a broadcast encryption mechanism, and can search for multiple keywords and rank the keywords, but cannot meet the dynamic update of keywords.

In this paper, the scheme improves the literature [**?**] to realize access control under the premise of satisfying multi-client, based on which it improves the problem of keyword search results can not be ranked and the dynamic update of keywords in the literature [**?**], and it reduces the number of comparisons between trapdoors and indexes through the advantage of clustering division to improve the efficiency of search.

5.3 Performance Analysis

In this paper, simulation experiments are conducted using the Python programming language, using the open-source Pycrptodome and pypbc modules to construct bilinear pair mappings and multiplication operations on groups and power operations; the hash function uses the SHA3-256 algorithm in the hashlib module. Intel(R) Core i5-6200U CPU @2.30 GHZ processor, 8 GB RAM, Microsoft Windows 10 Professional operating system. To analyze the execution time of a single cryptographic base operation, set the number of element bits in group G to 160 bits, repeat the execution of each operation 500 times, and take the average value, the execution time consumed is shown in Table **??**.

During the execution of the searchable encryption scheme, the time consumed to execute a single bilinear pair algorithm is much higher than the other underlying algorithms. Assuming that there are m documents in total and each document contains n keywords, the time taken by the Telematics user to query k keywords is shown in Table **??**. Table **??** compares and analyzes the time consumed in constructing the index, generating trapdoor and ciphertext retrieval in this paper's scheme with the schemes in literature [**?**], literature [**?**], and literature [**?**]. The time consumed in building the index is the shortest in document [**?**], followed by the schemes proposed in document [**?**] and document [**?**]. In this paper, document clustering is carried out in the whole process of index building to facilitate the subsequent retrieval, so the time consumed in index building is slightly higher than that in the other schemes. In the trapdoor generation stage, literature [**?**] involves the bilinear pair operation, which consumes relatively more time than other schemes, and literature [**?**] and literature [**?**] consume about the same amount of time, while this paper's scheme mainly generates traps through hash mapping and has already done a good job of clustering and dividing clusters when constructing indexes, which reduces the time consumption of trap generation. In the ciphertext retrieval stage, literature [**?**] only supports single keyword queries, and each query needs to compare the trapdoor with all indexes, resulting in time consumption. In this paper, the scheme has an advantage over other schemes in terms of computational efficiency in the ciphertext search phase due to clustering, which is analyzed in the following section.

Figure **??** is a comparison of the search time consumed before and after clustering for a different number of keywords, when the document set is not divided using clustering, with the increase in the number of keywords the search time has been showing a gradual upward trend; when the document set is divided using clustering, the keyword retrieval process is only compared with the most similar clusters of the keyword set, and the search time will change with the increase in the number of keywords, but the fluctuation is small, and the search time is shorter, and the efficiency is significantly improved compared to that when the keyword set is not clustered.

Different searchable encryption schemes do not use the same time to search for the same number of keywords, and Figure **??** reflects that the search time used by literature [**?**] and literature [**?**] is significantly higher than that of this paper's scheme for the same number of keywords, which highlights the impact of the clustering division of the scheme in this paper on the search efficiency.

Table 1: Comparison of functions

programmatic	multi-keyword	multi-keyword	dynamic update (Internet)	access control	multi-client
*Literature*7	×	×	×	✓	✓
*Literature*2	✓	✓	✓	×	×
*Literature*10	✓	✓	×	✓	✓
Programofthispaper	✓	✓	✓	✓	✓

Table 2: Single cryptographic base operation execution elapsed time

symbolic	operation type	Execution time (ms)
t_p	bilinear pair operation (math.)	9.8
t_e	Power operations in group G	0.027
t_m	Multiplication operations in group G	0.003
t_a	Addition in group G	0.002
t_h	hash operation	0.015

Table 3: Comparison of program performance

programmatic	Time-consuming to build indexes (ms)	Time-consuming to generate trapdoors(ms)	dciphertext search is time-consuming(ms)
*Literature*7	$mn(t_h + 2t_m + 3t_a) = 0.027mn$	$k(t_p+2t_m+t_a+2t_h) = 9.838k$	$m(2t_p + t_h + 3t_m + 2t_a) = 19.628$m
*Literature*2	$2mt_e + mn(2t_h + t_a) = 0.054m + 0.032mn$	$k(t_a + 2t_h) + 3t_e = 0.032k + 0.081$	$3mt_p + t_e = 29.4m + 0.027$
*Literature*10	$2mt_e + mn(t_h + 2t_m) = 0.054m + 0.021mn$	$k(2t_m + t_h + t_e) = 0.048k$	$3mt_p = 29.4m$
Programofthispaper	$2mt_e + mn(3t_h + t_m) = 0.054m + 0.048mn$	$t_e + k(2t_h + t_m) = 0.027 + 0.33k$	$2t_p + t_h = 18.615$

6 Conclusions

Aiming at the dynamic updating of data and keywords in the Internet of Vehicles, as well as the communication and computation overhead in search service, this paper proposes research on searchable encryption schemes for the Internet of Vehicles based on trusted authorization and dynamic updating of keywords. In this paper, we pre-process the data classification in the trusted authorization center, combined with the broadcast encryption mechanism to provide corresponding permissions to each group of user roles in the encryption phase; generate index vectors and query vectors to construct the index tree by LSH and Bloom Filter and use Top-k algorithm to improve the efficiency and file update, and evaluate the relevance of the file and query to rank the query results with the help of TF-IDF. Based on the difficulty assumption it is proved that the scheme of this paper is IND-CKA secure. The experimental results show that this scheme realizes the access control based on satisfying dynamic update and multi-client multi-keyword search and ranking the results, which improves the retrieval efficiency and reduces the computational overhead. In future research, through further improvement of clustering algorithm to get more practical category classification, such as user rights can be cross-used in data categories, making the research more practical application value.

Acknowledgments

This research is supported by the National Natural Science Foundations of China under Grants No.61862040 and No.62162039.The authors gratefully acknowledge the anonymous reviewers for their helpful comments and suggestions.

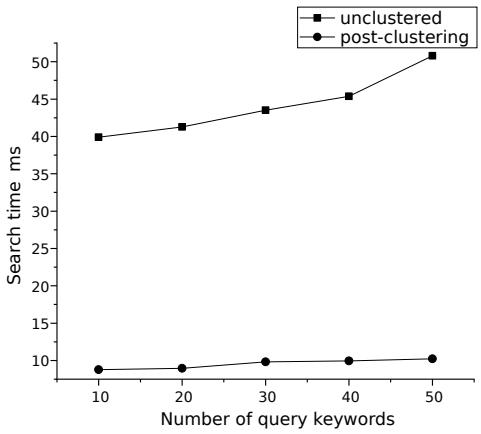

Figure 3: Search time before and after clustering based on number of keywords

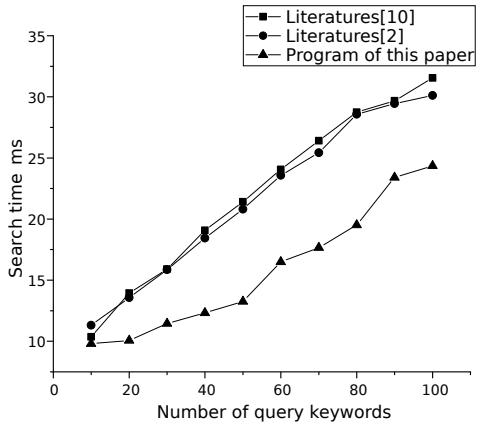

Figure 4: Search time before and after clustering based on number of keywords

References

[1] N Andola, R Gahlot, V K Yadav, S Venkatesan, and S Verma, "Searchable encryption on the cloud: a survey," *The Journal of Supercomputing*, vol. 78, no. 7, pp. 9952–9984, 2022.

[2] N Andola, S Prakash, V K Yadav, S Venkatesan, S Verma, et al., "A secure searchable encryption scheme for cloud using hash-based indexing," *Journal of Computer and System Sciences*, vol. 126, pp. 119–137, 2022.

[3] E. F. Cahyadi, C. Damarjati, M. S. Hwang, "Research on identity-based batch verification schemes for security and privacy in VANETs", *Journal of Electronic Science and Technology*, vol. 20, no. 3, pp. 1-19, 2022.

[4] E. F. Cahyadi, M. S. Hwang, "A comprehensive survey on certificateless aggregate signature in vehicular ad hoc networks", *IETE Technical Review*, vol. 39, no. 6, pp. 1265-1276, 2022.

[5] E. F. Cahyadi, M. S. Hwang, "An improved efficient anonymous authentication with conditional privacy-preserving scheme for VANETs", *Plos One*, vol. 16, no. 9, 2021.

[6] P. S. Chung, C. W. Liu, M. S. Hwang, "A study of attribute-based proxy re-encryption scheme in cloud environments", *International Journal of Network Security*, vol. 16, no. 1, pp. 1-13, 2014.

[7] Y Cui, F Gao, Y Shi, W Yin, E Panaousis, and K Liang, "An efficient attribute-based multi-keyword search scheme in encrypted keyword generation," *IEEE access*, vol. 8, pp. 99024–99036, 2020.

[8] S Guo, H Geng, L Su, S He, and X Zhang, "A rankable boolean searchable encryption scheme supporting dynamic updates in a cloud environment," *IEEE Access*, 2023.

[9] M. S. Hwang, T. H. Sun, "Using smart card to achieve a single sign-on for multiple cloud services," *IETE Technical Review*, vol. 30, no. 5, pp. 410–416, 2013.

[10] Z Jiang and Sh Chen, "Dual fine-grained public-key searchable encryption from lattices," *Computer and Information Science*, vol. 15, no. 1, pp. 1–66, 2022.

[11] H. Li, Q. Huang, and W. Susilo, "A secure cloud data sharing protocol for enterprise supporting hierarchical keyword search," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 3, pp. 1532–1543, 2020.

[12] W. Li, L. Xu, Y. Wen, and F. Zhang, "Conjunctive multi-key searchable encryption with attribute-based access control for ehr systems," *Computer Standards & Interfaces*, vol. 82, p. 103606, 2022.

[13] J Liu, Y Li, R Sun, Q Pei, N Zhang, M Dong, and V CM Leung, "Emk-abse: Efficient multikeyword attribute-based searchable encryption scheme through cloud-edge coordination," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 18650–18662, 2022.

[14] Zh Liu, Y Liu, J Xu, and B Wang, "Verifiable attribute-based keyword search encryption with attribute revocation for electronic health record system," *International Journal of Network Security*, vol. 22, no. 5, pp. 845–856, 2020.

[15] A M Manasrah, M Abu N, and M Salem, "A privacy-preserving multi-keyword search approach in cloud computing," *Soft Computing*, vol. 24, no. 8, pp. 5609–5631, 2020.

[16] N Polanco and B Cheng, "Situational crime prevention for automotive cybersecurity," in *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, pp. 562–568, 2022.

[17] J Ren, L Zhang, and B Wang, "Decentralized multi-authority attribute-based searchable encryption scheme," *International Journal of Network Security*, vol. 23, no. 2, pp. 332–342, 2021.

[18] D P Sharma and D C Jinwala. "Multi-keyword searchable encryption for e-health system with multiple data writers and readers,". in *Research Anthology on Securing Medical Systems and Records*, pp. 103–127. IGI Global, 2022.

[19] J Su, L Zhang, and Y Mu, "Ba-rmkabse: blockchain-aided ranked multi-keyword attribute-based searchable encryption with hiding policy for smart health system," *Future Generation Computer Systems*, vol. 132, pp. 299–309, 2022.

[20] B Wang, "An unlinkable key update scheme based on bloom filters for random key pre-distribution," *International Journal of Network Security*, vol. 22, no. 5, pp. 857–862, 2020.

[21] H Wang, Y Li, W Susilo, D H Duong, and F Luo, "A fast and flexible attribute-based searchable encryption scheme supporting multi-search mechanism in cloud computing," *Computer Standards & Interfaces*, vol. 82, p. 103635, 2022.

[22] P Wang, B Chen, T Xiang, and Zh Wang, "Lattice-based public key searchable encryption with fine-grained access control for edge computing," *Future Generation Computer Systems*, vol. 127, pp. 373–383, 2022.

[23] Y Wang, Y Wang, and J Wang, "Efficient self-adaptive access control for personal medical data in emergency setting," *International Journal of Computational Science and Engineering*, vol. 23, no. 4, pp. 341–351, 2020.

[24] Q Wu, X Ma, L Zhang, and Y Chen, "Expressive ciphertext policy attribute-based searchable encryption for medical records in cloud," *International Journal of Network Security*, vol. 23, no. 3, pp. 461–472, 2021.

[25] P Xie, X Tong, W, Y Zhao, T Feng, and Y Yan, "A trust assessment mechanism of the iov based on multi-factor analytic hierarchy process," *International Journal of Network Security*, vol. 24, no. 3, pp. 482–492, 2022.

Biography

Pengshou Xie was born in Jan. 1972. He is a professor and a supervisor of master student at Lanzhou University of Technology. His major research field is Privacy Protection, Security on Internet of Vehicles, Security on Industrial Internet. E-mail: xiepsh_lut@163.com.

Xiaoye Li was born in Oct. 1995. She is a master student at Lanzhou University of Technology. Her major research field is network and information security. E-mail: 976339400 @qq.com.

Yongping Kang was born in Feb. 1970. She is an Associate Professor at Lanzhou University of Technology. Her research interests include manufacturing information engineering, fault diagnosis of the production equipment. E-mail: 982542429@qq.com.

Jiafeng Zhu was born in Jan. 1997. He is a graduate student at Lanzhou University of Technology. His major research field is modern cryptography theory and information security technology. E-mail: zhujiafeng688@163.com.

Wanjun Shao was born in Jan. 1998. She is a master student at Lanzhou University of Technology. Her major research field is network and information security. E-mail: 2443404684@qq.com.

Cunhuan Tan was born in June. 2000. She is a master student at Lanzhou University of Technology. His major research field is network and information security. E-mail: 1635879818 @qq.com.