The Journal of Supercomputing (2020) 76:3013–3042
https://doi.org/10.1007/s11227-019-03087-y

A scoping review of searchable encryption schemes in cloud computing: taxonomy, methods, and recent developments

Umasankararao Varri[1,2] · **Syamkumar Pasupuleti**[2] · **K. V. Kadambari**[1]

Published online: 22 November 2019

Abstract
With the emergence of cloud computing, data owners are showing interest to outsource the data to the cloud servers and allowing the data users to access the data as and when required. However, outsourcing sensitive data into the cloud leads to privacy issues. Encrypting the data before outsourcing provides privacy, but it does not provide search functionality. To achieve search over encrypted data without compromising the privacy, searchable encryption (SE) schemes have been proposed. It protects the user's sensitive information by providing searchability on encrypted data stored in the cloud. In this paper, we surveyed different SE schemes which are existed in the cloud domain. In this survey, we presented the taxonomy of the SE schemes: symmetric searchable encryption, public key searchable encryption, and attribute-based searchable encryption schemes, and then provided a detailed discussion on the SE schemes in terms of index structure and search functionality. A comparative analysis of SE schemes is also provided on security and performance. Furthermore, we discussed the challenges, future directions, and applications of SE schemes.

Keywords Cloud storage · Searchable encryption · Privacy preserving · Search functionality · Security

✉ Umasankararao Varri
vusankararao@idrbt.ac.in; shankarit54@gmail.com

Syamkumar Pasupuleti
psyamkumar@idrbt.ac.in

K. V. Kadambari
kv.kadambari@gmail.com

[1] National Institute of Technology (NIT) Warangal, Hyderabad, Telangana, India

[2] Institute for Development and Research in Banking Technology (IDRBT), Hyderabad, Telangana, India

1 Introduction

With the rapid development of cloud computing, many people are showing interest to outsource their data to the cloud [25, 31, 48, 49]. Cloud storage provides benefits including easy access to data, less usage of physical storage devices, and reducing the infrastructure to maintain the data. Moreover, cloud users can access their data anywhere and on any device with an Internet connection.

Although cloud storage provides many benefits to users, the privacy of sensitive data is still a challenging issue (sensitive data include personal health records, confidential documents, private photographs, and any document which is personal to the user). This issue is because once the data are outsourced into the cloud server, the user loses physical control over data. Most of the cloud servers are *honest but curious*, i.e., we can trust cloud service providers for their services, but they may be interested in accessing our data. Hence, it is now necessary to protect the privacy of sensitive data in the cloud. The most common solution to guarantee user privacy is the encryption of user data before outsourcing into the cloud. However, the encryption of data cannot produce the required benefits due to the difficulty in searching over encrypted data. To address this issue, the searchable encryption (SE) schemes are used. The SE schemes allow the cloud server to search over encrypted data without knowing the information about plaintext or keywords.

Thereafter, many authors have proposed different SE schemes based on encryption techniques: symmetric searchable encryption (SSE), asymmetric/public searchable encryption (PSE), and attribute-based searchable encryption (ABSE). SSE deals with the private key, and it allows the user who holds the private key to generate a trapdoor. PSE deals with a public key as well as private key by allowing data owners to perform encryption with public keys of users and generate trapdoor with the private key of users. In ABSE, trusted third-party authority (TA) generates the private keys, and data owners encrypt the documents under defined access policies, such that the users decrypt the documents only when the attributes are matched with the access policy.

A comprehensive survey on searchable encryption is presented in [24, 45, 46, 63]. The authors in [24] discussed different search functionalities, and they classified SE into three categories, such as Server–User (S–U) model, User–Server–User (U–S–U) model, and UserA–Server–UserB (U_A–S–U_B) model. However, they have not elaborated on future research directions. The authors in [63] discussed the model for SE and outlined different schemes based on SSE and PSE. However, they have not considered ABSE as a primary classification since ABSE is efficient in terms of user access control. The authors in [46] discussed in detail about the symmetric searchable encryption in terms of design goals, structure, and query functionality. They also identified challenges from the survey and provided a few future directions. However, the paper is limited only for SSE schemes. The authors in [45] provided a framework for cloud-based SE systems and surveyed different existing schemes based on security measurements. The authors also discussed several applications of SE and identified useful future research directions. Still, the taxonomy of the paper is based on search functionality, and they have not considered encryption techniques.

In this paper, we provide a scoping review of searchable encryption schemes in cloud computing. The main contribution of this paper provides the following:

- Various searchable encryption (SE) schemes based on encryption techniques such as SSE, PSE, and ABSE are reviewed,
- SE schemes are reviewed in terms of search functionality such as single-keyword search, multi-keyword search, verifiable keyword search, dynamic keyword search, and attribute-based keyword search.
- Different attacks possible on SE schemes are defined and the security model of SE schemes is provided in tabular form.
- Performance of different SE schemes based on index creation time, search time, and trapdoor time is analyzed.
- The applications of SE are identified and introduced challenges and future directions to address them.

The rest of the paper is organized as follows: Searchable encryption and its architecture are described in Sect. 2. Taxonomy of SE and different SSE, PSE, and ABSE schemes is given in Sect. 3. Security analysis of different SE schemes is given in Sect. 4. Performance analysis of different SE schemes is given in Sect. 5. Applications of searchable encryption are given in Sect. 6. Challenges and future directions are presented in Sect. 7. Finally, the conclusion is presented in Sect. 8.

2 Searchable encryption

Searchable encryption is a cryptographic technique that allows data users to search over encrypted data using keywords securely without decrypting it. In SE schemes, the data owner encrypts the index and document collection before outsourcing it into the cloud server. The index contains the set of keywords from all the documents. To perform a search, the user generates a trapdoor and sends it to the cloud server. The cloud server returns the documents which are related to the generated trapdoor. SE scheme was first proposed by Song et al. [53] to allow users to access the documents stored in the cloud securely. Their scheme provides controlled search, hidden queries, and query isolation to improve privacy further. SE allows the data users to shoot their queries on the ciphertext, which in turn guarantees the privacy of the data. By using SE schemes, we ensure that the adversary is not able to attack the data so easily. SE schemes are widely used in the cloud computing platform because privacy and security are frequently measured parameters in the cloud.

2.1 Architecture of SE schemes

SE schemes are built based on *client/server model*, where cloud server (CS) acts as a server and data owners (DOs) and data users (DUs) act as clients during storage and retrieval. Figure 1 depicts the architecture of symmetric searchable encryption.

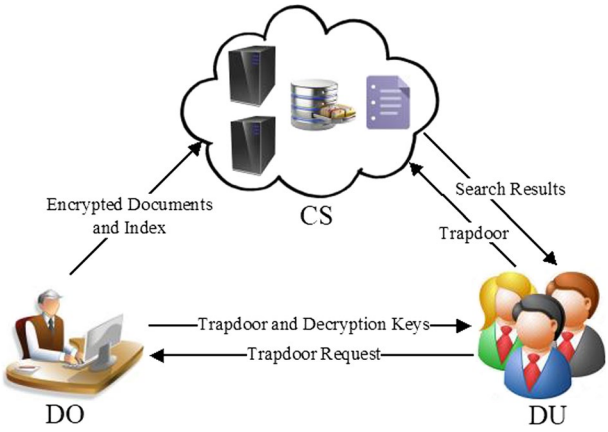

Fig. 1 Architecture of symmetric searchable encryption

Data owner The role of DO is to outsource the collection of documents $D = d_1, d_2, \ldots, d_k$ and a list of keywords $w = w_1, w_2, \ldots, w_n$ in an encrypted form. The DO must encrypt the documents and keyword collection before outsourcing into a cloud server so that the encrypted keyword set is searchable. However, the number of data owners is independent of the SE scheme itself.

Cloud server Cloud server stores the documents sent by the data owner and further also performs search tasks. When the DU submits trapdoor of a query keyword, it searches over the encrypted keywords, and once the search is completed, then CS returns the documents that contain the keyword to the DU. We believe that the CS is honest but curious. It means that the CS will follow the protocols, but it may evaluate the data, or search query patterns.

Data user Data user is authorized to retrieve data from the cloud by sending trapdoor to the CS. Once the results are obtained, the DU can decrypt the results.

2.2 Design goals of SE

We discuss some of the design goals of searchable encryption schemes.

Data privacy The data stored in the cloud should not be disclosed to unauthorized parties.

Index privacy Index privacy indicates that the server should not be aware of the keywords embedded in the index.

Keyword privacy The server should not be able to learn the keywords in trapdoor or authentication tags generated by the user.

Search pattern The search pattern is defined as the information that can be extracted from knowledge on whether two or more search results are from the same keyword.

Access pattern Access pattern is defined as the information that can be extracted from knowledge of a sequence of search results that contains the keyword.

Efficiency The user should be able to generate trapdoors and get search results efficiently. On the other hand, the cloud server should be able to fulfill the keyword search efficiently.

Verifiability In spite of data privacy, the results obtained from the cloud must be verified to examine data integrity.

Access control Access control prevents unauthorized access of data from the users who are not allowed to access the data. It can be achieved by performing user revocation.

3 Taxonomy of SE

Several searchable encryption schemes have been proposed in the later time based on Song et al. [53] scheme. Figure 2 shows the taxonomy of searchable encryption. In this section, we discussed in detail each of the encryption techniques. Further, in each of the encryption techniques, we examined different schemes; most of these schemes are published between 2015 and 2019 and divided these schemes based on keyword search functionality.

3.1 Symmetric searchable encryption

In SSE, the documents are encrypted by using symmetric/private key encryption techniques. As shown in Fig. 1, the cloud server sits in between the DO and the DU. In this, DO encrypts the documents along with the index and sends them to the cloud server, and later DU can access encrypted data by generating the trapdoor to the cloud server. The cloud server performs a search over encrypted data and sends results to DU, and then, the results obtained can be decrypted by using the secret key.

If KeyGen, BuildIndex, Enc, genTrapdoor, Search, and Dec are the polynomial-time algorithms over the keyword set *W*, then SSE algorithms description is as follows:

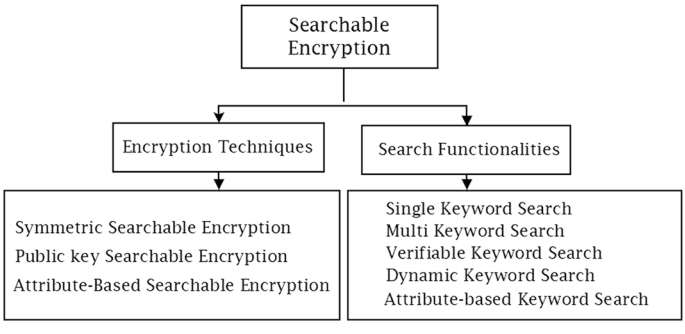

Fig. 2 Taxonomy of searchable encryption

$KeyGen(s) \rightarrow K$: The *KeyGen* algorithm is the very first algorithm in SE, and it is initiated by the DO. It takes security parameter s as an input and outputs the private key K.
$BuildIndex(w) \rightarrow I$: This algorithm is run by DO. It takes a set of keywords w where $w = w_1, w_2, \ldots, w_n$ from a document set as input and produces the searchable index I as output.
$Enc(D, I, K) \rightarrow E_d, E_i$: This algorithm is initiated by DO. It accepts document set D where $D = d_1, d_2, \ldots, d_k$, index I, and key K as input and produces the encrypted documents E_d and encrypted index E_i as output.
$genTrapdoor(K, q) \rightarrow T$: This algorithm is run by DU. It accepts secret key K and query keyword(s) q as input. If it is a single-keyword search, then q contains only one keyword; otherwise, it contains more than one keyword. Then, it outputs trapdoor T by encrypting q with K.
$Search(E_i, T) \rightarrow E_d$: This algorithm run by CS. It accepts the trapdoor T and encrypted index E_i as input and performs search operation with the trapdoor on encrypted index before producing the encrypted search results E_d as an output.
$Dec(E_d, K) \rightarrow D$: This algorithm is initiated by DU. It takes the search results E_d and secret key K as input and gives the actual documents D as output by decrypting E_d.

3.1.1 Single-keyword search

Single-keyword search allows users to search over encrypted data stored in the cloud server with only one keyword. The user query must contain exactly one keyword to generate the trapdoor, and then, CS returns the result related to that keyword. There have been many works proposed based on a single-keyword searching mechanism. Some of these schemes are as follows.

Goh [23] defined secure indexes and formulated security schemes to achieve index security and then introduced a secure index construction scheme called Z_ IDX based on bloom filters and pseudo-random functions. Bloom filters are memory-efficient data structures, and pseudo-random functions are deterministic and efficient functions that produce pseudo-random values indistinguishable from random sequences. Z_IDX was tested to implement searches over encrypted data. It is an efficient method when considering index updates but not efficient in search time at the cloud server.

To improve the search efficiency, Chang and Mitzenmacher [9] proposed the scheme to retrieve the files by searching with the indexed keywords efficiently. In this scheme, they have used pseudo-random bits to mask the index keywords and sent them to the server. However, the entire database must be searched for a specific query, which increases computation overhead and also requires additional storage overhead. Further, Chase and Kamara [10] proposed schemes for performing queries on structured data. In this, search queries are applied to the labeled data. While performing encryption, the data become labeled data by padding the data elements to be of the same length. However, only one label is assigned to the entire database. Thus, it requires padding the entire result set to achieve security. But Cash [8] used one label for each document that

the keyword contains i.e., If the keyword contains k documents, then it uses k different labels. It avoids making use of extra padding and also enables the parallel search.

All of the schemes discussed above are relevant to retrieve the search results based on an exact match. That is, there is no advantage of typo mistakes and other kinds of minor mistakes done by the user. So, Li et al. [33] proposed a first fuzzy keyword-based scheme that overcomes the stated problem. This scheme returns the documents of matched keywords. If the keyword has some minor typos, it returns the documents which are closest to the keyword by using keyword similarity semantics. They have used edit distance to measure the similarity semantics and also used security and privacy protocols to ensure privacy. However, this scheme cannot guarantee high service-level requirements like user searching experience and system usability. Also, it is specific to a certain distance measure. Then, Kuzu et al. [32] used locality-sensitive hashing (LSH) techniques for more generic solutions over distance measures, and a secure LSH index is constructed to witness privacy and security. The typical use of LSH is to solve the approximate or near neighbor search in the high-dimensional spaces.

However, a single-keyword search is used to search on the cloud server with only one keyword in the query, but it is not suitable for real-time applications. It is because searching with only one keyword may not identify the required file accurately. So, the multi-keyword search was introduced to improve search efficiency and get accurate results.

3.1.2 Multi-keyword search

Multi-keyword search is a popular searching scheme in cloud applications, and it allows users to search with multiple keywords in the query instead of only one keyword. With these multiple keywords, the efficiency of search and accurate results can be achieved. Based on multi-keyword search, several schemes are proposed and still many are contributing to provide privacy preserving of search in cloud platforms. We categorized these schemes into several types as ranked keyword search, fuzzy keyword search, synonym keyword search, semantic keyword search, and conjunctive keyword search.

3.1.2.1 Ranked keyword search In this keyword search, the documents are retrieved by considering the weights of the keywords. Further, these weights are assigned by using the TF $*$ IDF ranking model. Here, TF is a term frequency in a specific document and IDF is an inverse document frequency. IDF can be calculated as:

$$\mathrm{IDF}_t = \log \frac{N}{\mathrm{DF}_t}$$

where N is the number of documents and DF_t is the number of documents having term t. Similarity score between document D_j and query Q can be calculated as:

$$\text{Score}(D_j, Q) = \frac{1}{|D_j|} \sum_{w_i \in Q} \text{TF}_{i,j} \,.\text{IDF}_i$$

$$|D_j| = \sqrt{\sum_{w_i \in D_j} (\text{TF}_{i,j})^2}$$

where $|D_j|$ denotes the Euclidean length of document D_j and w_i is the keyword in document D_j.

Wang et al. [61] defined and solved a secure ranked keyword search over cloud data, which is encrypted by implementing order-preserving encryption (OPE). In this scheme, relevance scores and file id are used to build a searchable index securely. To protect these sensitive relevance scores, a one-to-many order-preserving mapping technique is developed. The scheme also guaranteed as-strong-as-possible security when compared with previously ranked search schemes. OPE is suitable for an efficient comparison of encrypted items without decrypting them. However, Yu et al. [72] addressed that OPE is subject to leakage of data privacy and proposed a scheme named two-round searchable encryption (TRSE), which is to support top-k multi-keyword search using homomorphic encryption. In this scheme, the vector space model is used to build the index. TRSE guarantees data security and the elimination of information leakage. However, the scheme is not dynamic. Du et al. [16] identified that file-injection attacks arise when the keyword update is done dynamically and investigated the schemes with forward-privacy functions one step ahead in file-injection attacks. Then, they proposed a bucket-encrypting index structure with a random generator (BEIS-I) to secure the index by encrypting the data identifier vectors (DIVs) and bit vectors. They also proposed bucket-encrypting index structure with a homomorphic generator (BEIS-II) to control bandwidth during query processing. The scheme guarantees privacy by using adaptive chosen keyword attack (CKA2) security model and forward-privacy model which support multi-keyword query processing, but the scheme does not verify the search results.

Ranked keyword-based search schemes provide multi-keyword search over encrypted data, but they do not fault tolerant i.e., they are not accurate when a user makes typographic errors.

3.1.2.2 Fuzzy keyword search Fuzzy keyword search allows users to search with minor mistakes in the keyword. Let distance $: F \times F \rightarrow R$ be a function that defines the distance between two keywords α and β be the threshold values for the used similarity metric s.t. $\alpha < \beta$. Then, the fuzzy keyword search is defined as follows.

FuzzyKeywordSearch(I, Q) It conducts search on index I based on the trapdoor of the keywords $f(Q)$ and outputs set of documents D, which are in the encrypted form. Suppose Fj is the keyword set related to Dj. Then, with the high probability, $D_j \in D$ if $\exists f_i(\text{distance}(f_i, f) \leq \alpha)$ and $D_j \notin D$ if $\forall f_i(\text{distance}(f_i, f) \geq \beta)$ where $f_i \in F_j$.

Z Fu et al. [20] proposed an efficient fuzzy-based multi-keyword search scheme based on [60] scheme, which was the first scheme on fuzzy-based search. The scheme proposed a keyword transformation method based on uni-gram to handle spelling mistakes and to improve accuracy. Furthermore, the stemming

algorithm is used to query the keywords with the same root and index vector with keyword weights is used to search over the index. But still, this scheme suffers from various security attacks [50]. To further improve security, Ahsan et al. [1] constructed a transformed keyword set to enable finding the original word from typo error and further used Jaccard similarity matrix to achieve maximum similarity. In this, the authors are limited to certain accuracy. To improve accuracy, Yuan et al. [73] designed three schemes to accomplish similarity search, security, accuracy, and scalability. Collision counting locality-sensitive hashing (LSH) is applied for secure similarity search, frequency hiding query schemes are used to witness security, and result sharing query schemes are implemented to improve scalability.

3.1.2.3 Semantic/synonym keyword search Consider a scenario where a user is searching for a keyword that is not present in the index, but by considering the similar meaning or synonym of that keyword, the user can retrieve the documents. This scenario shows the importance of semantic keyword search and synonym keyword search. Many authors have proposed schemes related to semantic- and synonym-based keyword search and some of these schemes performed better efficiency related to search results along with privacy. Fu et al. [21] identified that the grammatical relationship between query keywords is important when considering the user's perspective, and proposed the first-ever scheme to consider the relationship between the keywords in a query. Then, they designed a keyword weighting algorithm to exhibit the importance of the divergence among them. Further, they also designed a semantic keyword scheme to provide accuracy and efficiency in localizing the central keyword that the user is interested in. To further improve efficiency, Fu et al. [22] proposed two-cloud-server-based semantic search scheme (ECSED) based on concept hierarchy to make semantic-based search or content-aware-based search more effective. In this, one cloud server is used to store the encrypted documents and to return the trapdoor results to the user and the other cloud server is used to calculate the similarity scores and to return these scores to the first server. This scheme uses a tree-based index to organize all the documents efficiently and to process the query effectively.

3.1.2.4 Conjunctive keyword search A single-keyword search usually matches with a huge number of documents; among these, only a few are related to the user search. Instead, conjunctive keyword search (CKS) allows users to search with different keywords to produce individual search results and then intersects all individual results to generate a final result. This final result contains documents related to the user search.

Cai et al. [6] identified the vulnerability involved in inclusion relation (IR) attacks and came up with the scheme called secure-CKS based on bloom filters. This scheme transforms the query into a randomized and integrated form to make it hard to find any relationships among the different queries. Although the scheme is efficient for inclusion relation attacks, it does not address other attacks. To further improve the security, Ali and Lu [2] proposed a keyword field-free CKS

scheme to avoid the ordering of keyword in the trapdoor. It then introduced a security model indistinguishability under chosen keyword attack (IND-CKA) to secure the index. The security model is based on bloom filters and pseudo-random functions. However, the scheme does not provide dynamic update operations.

3.1.3 Verifiable keyword search

Most of the SSE schemes are based on the honest-but-curious cloud server model. Unfortunately, in practice, this assumption does not hold all the time since cloud servers may lead to external attacks, software bugs, internal misconfiguration errors, and even insider threats [57]. All these influences may cause the cloud server to operate beyond the honest-but-curious model. On the other hand, to address these issues, verifiable search schemes have been proposed to guarantee data integrity. In the verifiable search schemes, the documents obtained from the search are verified to decrypt the results. The verifiable algorithm is as follows:

KeywordTest$(pk, T_{w_i}, K) \rightarrow (K(w_i) \quad \text{or} \quad reject)$: It takes public key pk, trapdoor T_{w_i}, and file's ciphertext k as inputs and outputs $K(w_i)$, if w_i contained in the set otherwise it outputs *reject*.
VerifyDecrypt$(sk, K) \rightarrow (f \quad \text{or} \quad reject)$: If KeywordTest is passed, it takes user's private key sk, files' ciphertexts K as input, if accepted the client can be able to decrypt the files, otherwise it outputs *reject* and send this information to the server.

Cheng et al. [12] proposed a scheme to prevent malicious servers and to provide an environment for verifiable search results. This scheme is built on secure indistinguishability obfuscation (iO) to provide verifiability along with efficient keyword search. Obfuscation is referred to as a process applied to information to make it difficult to reverse without knowing the algorithm which is applied. However, the scheme does not provide forward privacy. Then, [4] proposed a scheme with same functionality as [12]. Besides, forward privacy is achieved to make the search secure against active adversaries. And also Ogata and Kurosawa [43] proposed a method to transform any SSE method to no-dictionary verifiable SSE scheme. Further, it addresses the issues of keeping keywords with a cuckoo hash table. However, all the schemes [4, 12, 43] are single-user models i.e., two-party models. However, cloud service providers enable service for multi-user models, i.e., three-party models. With this intention, [78] proposed the first generic verifiable symmetric searchable encryption (GSSE) scheme to support a single-owner, multi-user model. This scheme further supports the verifiability of any SSE scheme. GSSE constructs and maintains its proof index by using an Incremental Hash (IH) [3] and dynamic Merkle Patricia Tree, which further guarantees the data integrity. Also, to maintain the data freshness over multiple users, they proposed a time-stamp chain. The idea of IH is that if we already computed the hash for some document, and if some part of the document is modified in the later time, then instead of computing the update hash value for the complete document, we just need to compute the updated hash value

for the modified part. Merkle Patricia Tree is a tree of hashes. Each node in the tree contains a hash of its children along with the node's value.

3.1.4 Dynamic keyword search

All the schemes discussed so far support static operations only, i.e., there is no provision to add or delete documents without re-indexing the entire data. Besides, dynamic keyword search allows users to update their data flexibly. Some of the dynamic keyword search schemes in SSE are as follows.

Cao et al. [7] designed the first multi-keyword ranked search over encrypted data (MRSE) in the cloud, in which the documents and queries are represented as vectors. They have used secure cloud data utilization techniques to assure privacy preserving. In this scheme, coordinate matching is chosen to check the similarity, i.e., it returns the documents as many matches as possible. To further improve security, Yan et al. [70] presented a new multi-keyword dynamic search to make the data search more secure and privacy preserved. They used function-hiding inner product encryption to intensify security by avoiding the search pattern leakage. This scheme uses a binary tree structure for indexing to guarantee efficiency and effective dynamic update operations. Similarly, Xia et al. [66] presented a secure scheme to support dynamic insertions and deletions of documents also used the TF*IDF weightage model and vector space model to build the index as well as to construct the query. Alongside, a greedy depth-first search and a special tree-based index structure are used to provide an efficient multi-keyword ranked search dynamically. It uses the secure KNN encryption algorithm to encrypt the index and query keywords, and phantom terms are added to the query as well as to an index to counter the statistical attacks.

Further, Fu et al. [19] contributed work toward synonym-based ranked search. In this, the synonym search supports the search of keywords with similar meaning, and the index structure is built based on the binary trees in dynamic nature. Then, Li et al. [35] proposed a dynamic SSE scheme to support the conjunctive keyword search with the authentication mechanism. The scheme searches by considering a single keyword at a time, and the results obtained by each search are intersected. In this, the Merkle tree is used to ensure the correctness of integrity; the bilinear map accumulator is responsible for the final result, which is a subset of all the searches. The scheme is unforgeable against adaptive adversaries. In the adaptive adversary model, they make continuous queries for the secret key. The scheme is also secure under adaptive chosen keyword attack. However, the scheme is inefficient with more security parameters. To provide efficiency, Wan and Deng [59] proposed verifiable privacy-preserving keyword search (VPSearch) by incorporating privacy-preserving search schemes and homomorphic message authentication codes (MAC) technique, to achieve verifiable results as well as privacy. They offered data users to verify their search results efficiently without storing the results locally, and they have used a bit vector to construct the document index and the query. In this, the client encrypts the data and then it is authenticated by using homomorphic MAC. The encrypted authenticated data are outsourced into a cloud server. With the authenticated trapdoor, the client can search the cloud.

To further improve the security and to work with big data, Wang et al. [74] proposed a scheme to support large-scale similarity search. This scheme uses a high-dimensional feature vector as a search criterion. These feature vectors are mapped to fuzzy bloom filters which further utilize LSH to encode the index. The scheme is secure under adaptive chosen query attack (AQA2) and forward privacy. This scheme is not a keyword-based search scheme; instead, it is a high-dimensional feature vector-based search.

Some of the recent SSE schemes with their search functionalities as well as index structures are shown in Table 1. We observe that every search functionality has its own advantages. Still, in comparison with other search functionality, dynamic keyword search gains more attention as in many real-world applications, the data change dynamically, so the scheme should be flexible to work with such data. Moreover, SSE schemes require an extra secure channel to exchange the secret key between the data owner and data users. However, in the real world no such secure channel exists.

3.2 Public searchable encryption

SSE schemes require an extra secure channel to share the secret key between DO and DU, but we cannot guarantee that the secure channel is not compromised. PSE schemes use public key encryption techniques in which two keys are used for encryption and decryption: the public key and private key, respectively. As shown in Fig. 3, DO encrypts the documents along with document index with the user's public key and outsources the documents along with an index to the cloud server. Now DU can generate the trapdoor to the cloud server to obtain results. Once DU gets the results, it can decrypt the results by using the user's private key.

Table 1 Comparison of different modern SSE schemes based on search functionality and index structure

S. no.	Scheme name	Year	Search functionality	Index structure
1	Xia et al. [66]	2016	Ranked keyword search	Tree
2	Fu et al. [20]	2016	Fuzzy keyword search	Vector
3	Bost et al. [4]	2016	Verifiable keyword search	Tree
4	Yan et al. [70]	2016	Ranked keyword search	Tree
5	Ali and Lu [2]	2016	Conjunctive keyword search	Tree
6	Fu et al. [21]	2017	Semantic keyword search	Vector
7	Yuan et al. [73]	2017	Fuzzy keyword search	Inverted index
8	Liu et al. [39]	2017	Single-keyword search	Tree
9	Li and Liu [34]	2017	Conjunctive keyword search	Indistinguishable bloom filter
10	Ahsan et al. [1]	2017	Fuzzy keyword search	Inverted index
11	Li et al. [35]	2018	Conjunctive keyword search	Inverted list
12	Du et al. [16]	2018	Ranked keyword search	Inverted index
13	Fu et al. [22]	2018	Semantic keyword search	Tree
14	Zhu et al. [78]	2018	Verifiable keyword search	MPT
15	Wu and Li [65]	2019	Conjunctive keyword search	Virtual binary tree

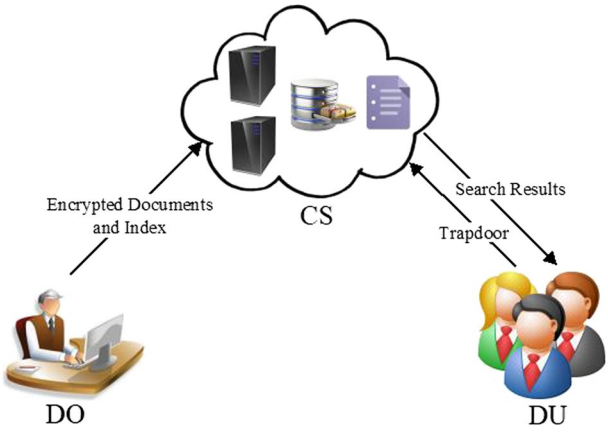

Fig. 3 Architecture of public searchable encryption

If KeyGen, BuildIndex, Enc, genTrapdoor, Query, and Dec are the polynomial-time algorithms over the keyword set W, then PSE algorithms description is as follows:

$KeyGen(s) \rightarrow (PK, SK)$: The *KeyGen* algorithm is the very first algorithm in SE process, and it is initiated by the DO. It takes security parameter s as an input and outputs the public parameter PK and secret key SK.
$BuildIndex(w) \rightarrow I$: This algorithm is run by DO. It takes a set of keywords w where $w = w_1, w_2, \ldots, w_n$ from a document set D as input and produces the searchable index I as output.
$Enc(PK, D, I) \rightarrow E_d, E_i$: This algorithm is initiated by DO. It accepts document set D where $D = d_1, d_2, \ldots, d_k$, index I, and key PK as input and produces the encrypted documents E_d and encrypted index E_i as output.
$genTrapdoor(SK, q) \rightarrow T$: This algorithm is run by DU. It accepts secret key SK and query keyword(s) $q \in I$ as input. If it is single-keyword search, then q contains only one keyword; otherwise, it contains more than one keyword. Then, it outputs trapdoor T by encrypting q with SK.
$Query(E_i, T) \rightarrow E_d$: This algorithm run by CS. It accepts the trapdoor T and encrypted index E_i as input and performs search operation with the trapdoor on encrypted index before producing the candidate set of encrypted documents E_d as an output.
$Dec(E_d, SK) \rightarrow D$: This algorithm is initiated by DU. It takes the search results E_d and secret key SK as input and gives the plaintext version of documents as output by decrypting E_d.

3.2.1 Single-keyword search

Jeong et al. [30] showed that the construction of secure PSE scheme for keyword guessing attack is not possible when a polynomial bounds the number of keywords.

Further, Xu et al. [68] proposed a scheme to support a fuzzy keyword search. In this scheme, the number of keywords shares the unique fuzzy keyword trapdoor, by that the cloud server does not learn the exact keyword. However, this scheme has limitations over efficiency and security. To provide security, Chen et al. [11] proposed a scheme called dual-server PSE scheme by using smooth projective hash functions to provide security against inside keyword guessing attack. Single-keyword search is recommended for real-time applications; multi-keyword search schemes have significant attention.

3.2.2 Multi-keyword search

3.2.2.1 Ranked keyword search Zhang et al. [75] proposed schemes for privacy-preserving multi-keyword ranked search in a multi-owner model (PRMSM). In this scheme, novel secure search protocols have been constructed to provide security for both keywords and trapdoors. They also proposed a novel addictive order family and privacy-preserving function family to rank the search results and to provide privacy for relevance scores of keywords and documents. Additionally, they proposed a new data user authentication protocol and dynamic secret key generation protocol to authenticate data users and to prevent attacks on secret keys. To further improve efficiency, Pasupuleti et al. [44] proposed a scheme to reduce the computational overhead while performing encryption and decryption. The scheme used the ranking process to retrieve top-ranked documents based on relevance scores without leaking information about documents and the search keywords. To improve security, Zhang et al. [76] proposed a first deterrent-based scheme using the Paillier cryptosystem which belongs to public key encryption with a multi-owner model based on the ranking procedure. In this scheme, the cloud server does not know how many data owners, or which data owners are exchanging the data. However, these schemes are not fault tolerant, i.e., user typological errors in the trapdoor keywords are not considered.

3.2.2.2 Fuzzy keyword search Xu et al. [69] formalized public key encryption with a fuzzy keyword search (PEFKS). In PEFKS, the trapdoor is divided into two parts, i.e., the exact keyword search trapdoor and fuzzy keyword search trapdoor. Only the fuzzy keyword search trapdoor is given to the cloud server to retrieve matched documents, and then, the user can further filter the results by issuing the exact keyword search trapdoor locally. Further, they proposed a transformation scheme that can transform any identity-based searchable encryption scheme into the PEFKS scheme. The scheme is secure against keyword guessing attack.

3.2.2.3 Conjunctive keyword search Ding et al. [15] proposed a scheme to provide a conjunctive keyword search by using public key encryption. In this scheme, no pairing operations are involved while performing encryption as well as when generating trapdoor. Further, Hwang et al. [29] identified that existing schemes are vulnerable against offline keyword guessing attacks and proposed a scheme to enable conjunctive keyword search by using bilinear pairing. The scheme is semantically secure against offline keyword guessing attacks even if the user is using the weak

device. However, the schemes [15, 29] require the complete list of keywords in the index when the trapdoor is generated. This leads to leakage of information and a lack of query privacy. Then, Yang and Ma [71] proposed a novel approach called Re-dtPECK, a time-dependant searchable encryption scheme to support conjunctive keyword search; this is secure against chosen keyword attacks, chosen time attacks, and also offline keyword guessing attacks. The security of the scheme is built based on the standard model instead of a random oracle model. However, in this scheme, the accuracy of search results is not guaranteed [41]. Further, Xu et al. [67] proposed a security-enhanced scheme by using the composite-order bilinear group as property. The scheme guarantees the correctness and consistency of the search results. Furthermore, keyword guessing attacks are resisted. To further improve efficiency, Farras and Gonzalez [17] proposed a scheme which supports conjunctive keyword search to improve the size of the index as well as the trapdoor generation time. The scheme is efficient compared with existing schemes in most critical operations like size, time, and performance.

3.2.3 Verifiable keyword search

Shen et al. [51] proposed the third-party-enabled searchable and verifiable scheme for big data applications. In this paper, a cube data structure is used for convenient storage and access. Further, it introduced a data protection scheme by using digital signatures and key agreement protocols for an efficient searchable verifiable scheme. However, the scheme is efficient in secure data protection, but security in data sharing is not guaranteed. Further, Wu et al. [64] proposed a secure verifiable scheme based on homomorphic encryption standards in a multi-user platform. In this, the

Table 2 Comparison of different modern PSE and ABSE schemes based on search functionality and index structure

S. no.	Scheme name	Year	Encryption	Search functionality	Index structure
1	Liang and Susilo [36]	2015	ABSE	Attribute-based keyword search	Matrix
2	Yang and Ma [71]	2016	PSE	Conjunctive keyword search	Vector
3	Zhang et al. [75]	2016	PSE	Ranked keyword search	Vector
4	Chen et al. [11]	2016	PSE	Single-keyword search	Vector
5	Pasupuleti et al. [44]	2016	PSE	Ranked keyword search	Tree
6	Sun et al. [58]	2016	ABSE	Attribute-based keyword search	Inverted index
7	Miao et al. [41]	2016	PSE	Conjunctive keyword search	Vector
8	Ma [40]	2016	PSE	Single-keyword search	Vector
9	Xu et al. [67]	2017	PSE	Conjunctive keyword search	Vector
10	Cui et al. [13]	2018	ABSE	Attribute-based keyword search	Tree
11	Miao et al. [42]	2018	PSE	Verifiable keyword search	Tree
12	Huang et al. [27]	2017	PSE	Verifiable keyword search	Vector
13	Zhang et al. [76]	2018	PSE	Ranked keyword search	Vector
14	Sun et al. [55]	2018	PSE	Verifiable keyword search	Vector
15	Farras and Gonzalez [17]	2019	PSE	Conjunctive keyword search	Vector

inverted index structure is used for authenticated data structure, which further improves the correctness of the search results.

Some of the recent PSE and ABSE schemes with their search functionalities, as well as index structures, are shown in Table 2. We observe that PSE schemes are more secure compared to SSE because there is no extra communication between DO and DU for the secret. However, the limitations with PSE are (1) the key management is taken care of by DO and it needs the DO to be online, every time a new user registers with the system; and (2) user authentication, user access control, and user revocation are difficult operations in PSE schemes. Attribute-based searchable encryption (ABSE) is a solution for above-said limitations.

3.3 Attribute-based searchable encryption

SSE and PSE can provide privacy for the data, but they are not efficient when talking about authorized access. Any user with a private key can be able to access the data in SSE and PSE mechanisms. Attribute-based searchable encryption (ABSE) is quite different, and it integrates an access policy along with the encrypted documents to control the user access. The access policy contains information about all the users who can be able to access the document. If user attributes (e.g., name, department, course, gender) are not satisfied with the defined access policy, then the user is not authorized to access the documents. Figure 4 depicts the architecture of attribute-based searchable encryption, where trusted authority (TA) is responsible for generating public parameters and secret keys for the user.

ABSE is divided into two types: Key-policy ABSE (KP-ABSE) and Ciphertext-policy ABSE (CP-ABSE). The key generation and encryption procedure in both policies make the difference. In key-policy ABSE, the secret key is generated by using access policy P and the documents and index are encrypted using a set of attributes ω. In ciphertext-policy ABSE, the secret key is generated by using a set of attributes ω and the documents and index are encrypted using an access policy P.

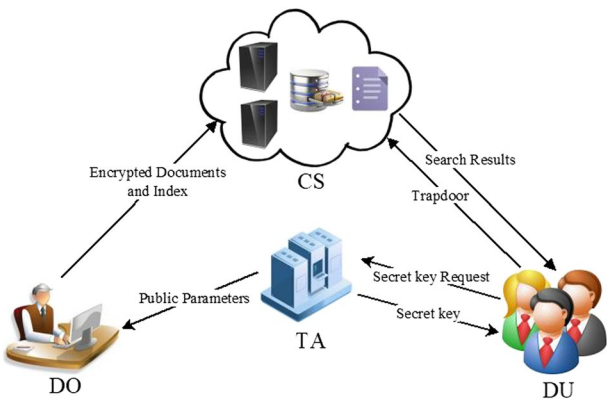

Fig. 4 Architecture of attribute-based searchable encryption

If Setup, KeyGen, Enc, Trapdoor, and Query are the polynomial-time algorithms over the keyword set W, then CP-ABSE algorithms description is as follows:

$Setup(\lambda, \cup) \rightarrow (PK, MK)$: This algorithm is initiated by TA. It takes security parameter λ and universal attributes $\cup$ as an input and outputs the public parameter PK and master secret key MK.
$KeyGen(PK, MK, S) \rightarrow sk$: This algorithm is initiated by TA. By accepting the PK, MK and user attributes S as an input; the algorithm outputs secret key sk for users.
$Enc(W, P) \rightarrow cp$: This algorithm is initiated by DO. It performs encryption of keyword set W with access policy P to obtain ciphertext cp.
$Trapdoor(sk, w) \rightarrow t$: This algorithm is initiated by DU. It allows to generate a search trapdoor t according to sk and keyword w.
$Query(cp, t) \rightarrow 0, 1$: This algorithm is initiated by CS. This algorithm returns 1 if the keyword in the index and the keyword in the trapdoor match, otherwise returns 0.

3.3.1 Attribute-based keyword search

Liu et al. [38] proposed a novel method for verifying the results obtained from the cloud server by using KP-ABSE (key-policy attribute-based searchable encryption). This scheme can efficiently verify the integrity and correctness of the search results. Additionally, the scheme assures that offline keyword guessing attacks are prevented. Then, Hur and Noh [28] proposed a ciphertext-policy attribute-based encryption scheme to achieve access control policies with user revocation capability along with efficient attribute-based search capability. Further to improve efficiency, Zheng and Xu [77] proposed a novel verifiable attribute-based search scheme to solve the problems such as search over the encrypted outsourced data, verifying whether the cloud server has faithfully performed the search operations. In this scheme, access trees are used for access control. However, it fails to support data sharing of encrypted data. For efficient data sharing, and flexible keyword updates, Liang and Susilo [36] used the abilities of attribute-based keyword search and attribute-based proxy re-encryption. With this integration, data owners can efficiently share their data with the users who satisfy the access policy. Then, Miao et al. [42] proposed an attribute-based verifiable keyword search scheme to allow users to check the correctness of the search results. The scheme achieves fine-grained access control, and also attribute-based priority tree is used to provide access control of the same data. Besides, the scheme is secure against CKA. However, the scheme used less computationally efficient tree structure [55].

To improve efficiency, Sun et al. [58] designed a scalable keyword search scheme based on ABSE supporting multi-user and multi-owner model. The scheme can run in linear search with fine-grained search authentication at the file level by providing better scalability. User revocation can be computationally efficient as the data owner can manage the most critical tasks of the cloud server effectively. This scheme is secure against CKA. However, the scheme is not suitable for other types of security attacks. Further, Cui et al. [13] proposed a multi-owner, multi-user search scheme

called attribute-based keyword search with an efficient revocation scheme (AKSER). This scheme is efficient with regard to user revocation and provides fine-grained authorized keyword search. This scheme also achieved security goals like keyword secrecy, keyword semantic security, trapdoor unlinkability, and collusion resistance. However, the user needs to perform new registration each time the revocation happens. Then, Wang et al. [62] proposed the first hierarchical attribute-based encryption scheme for the document collection. With the attribute-based search schemes, resources like ciphertext storage space and encryption/decryption time are saved. In this scheme, the index is constructed with attribute-based retrieval features (ARF) tree for the document collection. The search efficiency is improved by using a depth-first search traversal algorithm for the ARF tree and by using parallel computing.

We observed that the ABSE schemes have more features compared to SSE and PSE schemes in terms of security. Further, the dynamic keyword search with ABSE is a useful observation for real-world applications. Table 3 shows different SE schemes along with the search functionality they contained. It is observed that some of the schemes are having more than one search functionality. If the scheme contains a particular search functionality, then it is represented as *YES*, otherwise *NO*.

4 Security model of searchable encryption schemes

The SE system needs to prove that it can preserve the privacy of the user's data and prevent information leakage. In this section, we provide the security strengths of SE schemes, mainly focusing on different attacks possible in SE schemes and the solutions given by different authors to avoid these following attacks.

Chosen plaintext attack (CPA) In this attack, the attacker may choose random plaintexts that are encrypted to obtain the corresponding ciphertexts. Several works are proposed to secure against this attack [14, 52, 54]. In the known-plaintext attack model, the cloud server can acquire plaintext–ciphertext pairs from the dataset. Using this information, it attempts to resolve the secret key to decrypt the ciphertext.

Chosen keyword attack (CKA) In this attack, the adversary may attack the keywords selectively to obtain the decryption of selected keywords. Wang et al. [60], Ali and Lu [2], Xia et al. [66], Yuan et al. [73], Fu et al. [21], Du et al. [16], Li et al. [35], Farras and Gonzalez [17], and Curtmola et al. [14] proposed schemes to secure against CKA. Indistinguishably under the chosen keyword attack (IND-CKA) model captures the notion that the documents or index is not revealed. Initially, bloom filters and pseudo-random functions are used as a background for this model.

Keyword guessing attack (KGA) In PSE schemes, the adversary may generate encrypted tags corresponding to all the possible keywords to generate trapdoor to determine the documents further. The keyword guessing attacks are highly possible when a sender and a receiver usually query commonly used keywords such as "important," "action," and "happy." It is known that the number of commonly used keywords is not so big. Several authors have proposed schemes [1, 11, 27, 29, 30, 68, 69, 71] to secure against KGA.

Known ciphertext model (KCM) In this model, the CS can be able to know the secure indexes, encrypted files, and trapdoors [20, 21]. Moreover, CS can also know

Table 3 Comparison of different SSE, PSE, and ABSE schemes based on search functionality

S. no.	Scheme name	Year	Encryption	SKS	FKS	RKS	SeKS	VKS	CKS	ABSE	DKS
1	Cao et al. [7]	2014	SSE	No	No	Yes	No	No	No	No	Yes
2	Fu et al. [19]	2014	SSE	No	No	Yes	Yes	No	No	No	Yes
3	Hwang et al. [29]	2014	PSE	No	No	No	No	No	Yes	No	No
4	Wang et al. [60]	2014	SSE	No	Yes	No	No	No	No	No	Yes
5	Cash et al. [8]	2014	SSE	Yes	No	No	No	No	No	No	No
6	Liang and Susilo [36]	2015	ABSE	No	No	No	No	No	No	Yes	No
7	Yan et al. [70]	2016	SSE	No	No	Yes	No	No	No	No	Yes
8	Sun et al. [58]	2016	ABSE	No	No	No	No	Yes	No	Yes	Yes
9	Ali and Lu [2]	2016	SSE	No	No	No	No	No	Yes	No	No
10	Xia et al. [66]	2016	SSE	No	No	Yes	No	No	No	No	Yes
11	Chen et al. [11]	2016	PSE	Yes	No	No	No	No	No	No	No
12	Fu et al. [21]	2017	SSE	No	No	No	Yes	No	No	No	No
13	Miao et al. [42]	2017	PSE	No	No	No	No	Yes	No	No	No
14	Cui et al. [13]	2017	ABSE	No	No	No	No	No	No	Yes	Yes
15	Li et al. [35]	2017	SSE	No	No	No	No	Yes	Yes	No	Yes
16	Xu et al. [67]	2017	PSE	No	No	No	No	Yes	No	No	No
17	Wan and Deng [59]	2018	SSE	No	No	Yes	No	Yes	No	No	Yes
18	Fu et al. [22]	2018	SSE	No	No	Yes	Yes	No	No	No	No
19	Wang et al. [74]	2018	SSE	No	Yes	No	No	No	No	No	Yes
20	Wang et al. [62]	2018	ABSE	No	No	Yes	No	No	No	Yes	Yes

SKS single-keyword search, *RKS* ranked keyword search, *VKS* verifiable keyword search, *FKS* fuzzy keyword search, *SeKS* semantic keyword search, *CKS* conjunctive keyword search, *DKS* dynamic keyword search, *ABSE* attribute-based keyword search

and record the search results. In the chosen ciphertext attack model, the adversary can collect information by getting the decryption of ciphertext by choice [36, 37, 40, 44, 55].

Known background model (KBM) In this model, the CS performs statistical analysis to obtain keyword specific information, which can be further combined with background information to know the keyword in a user query. With this statistical analysis, the cloud server can be able to perform term frequency (TF) statistical attack to deduce or even recognize the keywords by analyzing histograms [56]. There several other authors proposed the schemes against this attack [21, 22, 66].

Chosen message attack (CMA) This attack is possible in signature schemes where the adversary can get the signature of the chosen number of messages [4]. In many cases, this attacking model is used to protect message authentication codes (MAC).

There are many other kinds of attacks like inclusion relation attacks, chosen identity attacks, file-injection attacks, and brute-force attacks, which are vulnerable to privacy. Tables 4 and 5 show different SSE and PSE schemes, respectively, and different attacks in which the scheme is secure against.

5 Performance analysis

In this section, we analyzed the performance of different SE schemes concerning index creation time, search time, and trapdoor time.

Index creation time Index creation time is based on the number of keywords from all the documents. Efficient index construction guarantees that the search process takes less amount of time. There are different ways to create an index

Table 4 Security model used in different SSE schemes

S. no.	Scheme name	Security model
1	Xia et al. [66]	KBM
2	Fu et al. [20]	KCM, KBM
3	Bost et al. [4]	CMA
4	Yan et al. [70]	CKA
5	Ali and Lu [2]	CKA
6	Fu et al. [21]	KCM, KBM
7	Yuan et al. [73]	CKA
8	Liu et al. [39]	CKA
9	Li and Liu [34]	CKA
10	Ahsan et al. [1]	CKA
11	Li et al. [35]	CKA
12	Du et al. [16]	CKA
13	Fu et al. [22]	KBM
14	Zhu et al. [78]	CKA
15	Wu and Li [65]	CKA

KBM known background model, *KCM* known ciphertext model, *CMA* chosen message attack, *CKA* chosen keyword attack

Table 5 Security model used in different PSE and ABSE schemes

S. no.	Scheme name	Security model
1	Liang and Susilo [36]	CCA
2	Yang and Ma [71]	KGA
3	Zhang et al. [75]	CKA
4	Chen et al. [11]	KGA
5	Pasupuleti et al. [44]	CCA
6	Sun et al. [58]	CKA
7	Miao et al. [41]	KGA
8	Ma [40]	CCA
9	Xu et al. [67]	KGA
10	Huang et al. [27]	KGA
11	Shen et al. [52]	CPA
12	Miao et al. [42]	CKA
13	Zhang et al. [76]	CKA
14	Sun et al. [55]	CCA
15	Farras and Gonzalez [17]	CKA

CCA chosen ciphertext attack, *KGA* keyword guessing attack, *CKA* chosen keyword attack, *CPA* chosen plaintext attack

with the keywords of documents. An inverted index is a popular technique to construct the index. The vector space model is a widely used technique in multi-keyword search scenarios. Tree-based index construction improves efficiency in the search process.

Search time Search time depends on search functionality. The efficiency of any SE scheme depends on search time; thus, this computational measurement becomes an important metric. The schemes with the inverted index require $O(r)$ time, where r is the number of documents matched with the search keyword. On the other hand, schemes with tree-based index require $O(\log n)$ time, where n is the total number of documents. Hence, search time should be efficient when designing SE schemes.

Trapdoor time Trapdoor time deals with the time taken to generate trapdoor to search keywords. The number of keywords in the query decides the overhead of trapdoor.

When the security of the SE scheme is increased, the performance is decreased because of the added overheads. So, there must be a balanced trade-off between security and performance. Tables 6 and 7 show performance of different SE schemes based on the metrics discussed above where M represents the total number of keywords, N represents the total number of documents, D represents the size of dataset, B represents the number of buckets, L represents the number of rows of matrix, A represents the number of attributes in attribute set, C represents the computational cost of bilinear pairing, V represents the vector dimension, F represents the file number, T represents the distinct keywords in the query, R represents the number of keywords in the query, and S represents the size of the dictionary.

Table 6 Performance analysis of different SSE schemes based on search time, index creation time, and trapdoor time

S. no.	Scheme name	Search time	Index creation time	Trapdoor time
1	Xia et al. [66]	$O(M \log N)$	$O(MN^2)$	$O(M^2)$
2	Fu et al. [20]	$O(N)$	$O(M)$	$O(M)$
3	Bost et al. [4]	$O(N \log N)$	$O(M)$	$O(S)$
4	Yan et al. [70]	$O(D)$	$O(N)$	$O(S)$
5	Ali and Lu [2]	$O(D)$	$O(N)$	$O(M)$
6	Fu et al. [21]	$O(S)$	$O(SV)$	$O(S)$
7	Li et al. [35]	$O(M \log M)$	$O(N)$	$O(F + R \log T)$
8	Yuan et al. [73]	$O(M)$	$O(N)$	$O(M)$
9	Liu et al. [39]	$O(R)$	$O(N)$	$O(M)$
10	Li and Liu [34]	$O(MlogN)$	$O(NlogN)$	$O(M)$
11	Wang and Deng [59]	$O(N + M)$	$O(N)$	$O(S)$
12	Du et al. [16]	$O(N)$	$O(NB)$	$O(C)$
13	Fu et al. [22]	$O(D)$	$O(MnlogN)$	$O(M)$
14	Wu and Li [65]	$O(RlogN)$	$O(MlogN)$	$O(S)$
15	Zhu et al. [78]	$O(M)$	$O(MlogN)$	$O(M)$

Table 7 Performance analysis of different PSE and ABSE schemes based on search time, index creation time, and trapdoor time

S. no.	Scheme name	Search time	Index creation time	Trapdoor time
1	Liang and Susilo [36]	$O(AC)$	$O(N^2)$	$O(L^2)$
2	Yang and Ma [71]	$O(D)$	$O(MN)$	$O(M)$
3	Shen et al. [51]	$O(M)$	$O(N)$	$O(M)$
4	Chen et al. [11]	$O(R)$	$O(MN)$	$O(C)$
5	Pasupuleti et al. [44]	$O(R)$	$O(N)$	$O(M)$
6	Sun et al. [58]	$O(N)$	$O(M)$	$O(R)$
7	Miao et al. [41]	$O(M)$	$O(D)$	$O(R)$
8	Ma [40]	$O(R)$	$O(N)$	$O(R)$
9	Xu et al. [67]	$O(D)$	$O(N)$	$O(M)$
10	Cui et al. [13]	$O(M)$	$O(A)$	$O(C)$
11	Miao et al. [42]	$O(AN)$	$O(M)$	$O(A)$
12	Huang et al. [27]	$O(M)$	$O(M)$	$O(C)$
13	Zhang et al. [76]	$O(M)$	$O(MN)$	$O(M)$
14	Sun et al. [55]	$O(C)$	$O(M)$	$O(C)$
15	Farras and Gonzalez [17]	$O(C)$	$O(MC)$	$O(C)$

6 Applications of searchable encryption

In this section, we present real-world situations where the SE schemes can be applied.

Healthcare clouds In healthcare cloud computing, the patients can outsource their medical data remotely. Outsourcing medical data into cloud were becoming the trend in many medical applications as they reduce the cost of maintaining Electronic Health Records (EHR). Since the health data of individuals are sensitive, they must be encrypted before outsourcing into the cloud. However, the encryption does not provide searchability to the users. Hence, SE is considered to be a feasible solution to provide the privacy of the data. Liang and Susilo [37] proposed a mechanism to provide a healthcare outsourcing platform. Likewise, many schemes in recent times are falling into this area.

Mobile clouds Nowadays, many of the mobile applications are running in a cloud environment. Mobile devices are major cloud vendors. Increasing the use of mobile devices, the cloud has turned from desktops to mobiles. Mobile cloud delivers all the mobile computing operations securely with mobile cloud resources. To provide privacy while searching for information from the cloud, the SE is a solution, and SE is still growing in mobile cloud computing.

Banking sector Banking sectors are adopting cloud computing to improve their strength in information technology (IT). However, most of the banking data are sensitive; hence, they face many privacy issues. To preserve the privacy of data, banking organizations are adopting SE schemes to securely provide search operations over encrypted data stored in the cloud.

Crowdsourcing Crowdsourcing is come up with the pay-for-performance model. We do not pay for resources, but pay for solutions. Crowdsourcing cloud computing has many benefits from a reduction in cost to many security concerns. However, enabling users to provide searchability along with protecting the privacy of data is challenging. Hence, SE is a solution to the above-stated problem.

Internet of Things (IoT) IoT is increasingly one of the popular technology trends. IoT needs strong privacy preserved data handling capability because it produces a huge amount of data through sensors and stores it in the cloud. The ultimate solution for this scenario is searchable encryption in the cloud. Data from the sensors are uploaded into the cloud to allow users to access through search operations.

Fog computing Fog computing is an extension of cloud computing, and it outsources encrypted sensitive data in multiple fog nodes to avoid network congestion and latency on the edge of IoT. However, the privacy of data is also a concern in fog computing. Hence, SE is a desirable solution to provide privacy to the data.

Big data Big data have a huge impact on our lives today, and they draw the attention of cloud computing for the storage and processing of huge amounts of data. Security and privacy are major challenges for big data. Hence, searchable encryption is a leading solution to provide the privacy of big data.

There are many other applications of searchable encryption, which effect many real-world scenarios by providing better efficiency and accuracy.

7 Challenges and future directions

In this section, we discuss the challenges and research directions of searchable encryption schemes.

7.1 Efficiency in build index

From section 3, we observed that most of the schemes use a well-defined data structure to build an index from dictionaries to balanced binary search trees. From the literature, inverted index [1, 16, 35, 58, 73] is widely used data structure in single-keyword search techniques and vector space model achieved many advantages by using TF*IDF factor to rank the keyword frequency [11, 17, 20, 21, 27, 40, 41, 55, 67, 71, 75, 76]. Likewise, in later times, binary search trees decrease the computational overhead with the time complexity $O(\log n)$ where n is the number of documents [2, 4, 39]. It seems that the inverted index and vector space model still serve as core indexing techniques. However, further, the tree and graph structures can be used together to explore new properties and to improve the efficiency in building an index.

7.2 Efficient search functionality

From the literature, we observed that the early stage of SE schemes is built based on a single-keyword search functionality, but these schemes are limited to less accurate results [8–11, 23, 68]. Later, many SE schemes are proposed in a multi-keyword scenario where the query contains multiple keywords instead of only one. A ranking-based keyword search is one of the widely used functionalities where the documents are retrieved by considering the ranks of given keywords [16, 44, 61, 66, 70, 72, 75, 76]. Similarly, small kinds of user typological mistakes are avoided and can be able to retrieve documents by using a fuzzy keyword search [1, 20, 32, 33, 50, 60, 69, 73]. As well as a single-keyword search is performed on each query keyword, and the results of all searches are intersected to produce a final result to make conjunctive keyword search [2, 6, 15, 29, 41, 71]. However, exploring different other search mechanisms for richer query types such as subset queries and range queries is further research investigation.

7.3 Secure encryption techniques

Initially, SE schemes are built based on symmetric searchable encryption (SSE) technique. These schemes need an extra secure channel to share the private key among owners and users. However, there is no guarantee that the secure channel is safe. Then, the public searchable encryption (PSE) technique made attention that they never require a secure channel to communicate between owner and users because PSE uses two keys: one for encryption (public key) and another for decryption (private key). Based on the PSE domain, the identity-based searchable encryption (IBSE) and attribute-based searchable encryption (ABSE) contributed security metrics as user revocation and fine-grained access control, respectively. However, with the existence of the post-quantum era, all the schemes which are

build based on bilinear pairing assumption can be attacked by quantum computers. Hence, new schemes must be built by considering quantum attacks.

7.4 Secure dynamic search

The addition of new keywords into an already constructed index is not possible in static SE schemes without re-indexing the entire data. From the survey, we observed that dynamic SE schemes allow users to perform dynamic update operations like insert, delete, and modify on the data which is already outsourced [7, 13, 18, 19, 35, 58, 59, 62, 66, 70, 74]. On the other hand, updating messages dynamically can be easily observed by the cloud server irrespective of the index structure; this implies that the dynamic search schemes are suffering from forward and backward privacy. The dynamic search schemes are said to have forward privacy and backward privacy if the server cannot be able to know that the newly added message has a keyword that is previously searched and the server cannot perform queries on deleted messages, respectively. Dynamic search is suitable for real-world applications because the data in real-world applications are variable and also dynamic schemes in general leak more information compared to static schemes. Hence, the construction of secure dynamic schemes is a possible area of study.

7.5 Verifiability

Along with data privacy, data integrity is also important to make the SE scheme more secure. Data integrity can occur in many ways as software bugs, insider threats, and external attacks. Since the results obtained from the cloud server may not be correct all the time, the verification of search results must happen to witness correct data. We observed from the survey that many authors proposed different verifiable schemes to check the results obtained from the cloud are not integrated [4, 12, 38, 42, 43, 51, 55, 57, 64, 78]. Verifiable search schemes should be built without sacrificing essential functionalities like dynamic data updates and some critical search functionalities. Moreover, the verification cost should also be nominal and affordable to users irrespective of the collection of large data. Hence, verifiable search schemes with minimal cost and without losing essential functionalities are demanded.

7.6 Key escrow problem

From the literature, we observed that the ABSE schemes decompose the user identity into a set of attributes and enable each user to have a unique set of attributes [13, 28, 36, 47, 58, 62, 77]. However, the most common problem found in ABSE schemes is a key escrow problem. It occurs when trusted authority (TA) who is known to generate a private key to the users has unusual access to the keys, i.e., TA decrypts the message and also behaves vulnerably. Hence, ABSE schemes should be designed without key escrow problem.

7.7 Blockchain technology in searchable encryption

As technology advances, blockchain technology is attractively promising aspects and attaining more attention from the SE approach. Several proposals utilized blockchain technology in searchable encryption [5, 26]. Cai et al. [5] is the first work integrating searchable encryption and blockchain. Nevertheless, it requires more study and experiments to develop more mature schemes in the future.

8 Conclusion

In this paper, we reviewed different searchable encryption schemes in cloud computing. The main objectives of the SE schemes are data privacy, efficiency, security, and query expressiveness. In this review, we presented an overview of searchable encryption by considering architecture, algorithms, and design goals. We classified SE into SSE, PSE, and ABSE schemes and discussed different SSE, PSE, and ABSE schemes in terms of search functionality, index structure, security metrics, and efficiency. Then, we analyzed the security of current searchable encryption schemes based on different types of attacks possible. We performed the performance analysis of current searchable encryption schemes based on search time, index creation time, and trapdoor time. Further, we discussed different applications of SE schemes.

Based on the review, we identified the challenges such as efficiency in building an index, secure encryption techniques, efficiency in search functionality, secure dynamic search, verifiability, key escrow problem, and blockchain technology in searchable encryption. In our future work, we are going to work with attribute-based searchable encryption to address the key escrow problem and to resist quantum attacks.

Acknowledgements The authors would like to thank the editor and the anonymous reviewers whose comments significantly helped to improve the quality of this paper.

Compliance with ethical standards

Conflict of interest The authors declare that they have no known competing for financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Ahsan MAM, Chowdhury FZ, Sabilah M, Wahab A, Idris B (2017) An efficient fuzzy keyword matching technique for searching through encrypted cloud data. In: International Conference on Research and Innovation in Information Systems (ICRIIS). https://doi.org/10.1109/ICRIIS.2017.8002456
2. Ali FS, Lu S (2016) Searchable encryption with conjunctive field free keyword search scheme. In: 2016 International Conference on Network and Information Systems for Computers (ICNISC), IEEE, pp 260–264. https://doi.org/10.1109/ICNISC.2016.064
3. Bellare M, Goldreich O, Goldwasser S (1994) Incremental cryptography: the case of hashing and signing. In: Annual International Cryptology Conference, Springer, Berlin, pp 216–233

4. Bost R, Fouque PA, Pointcheval D (2016) Verifiable dynamic symmetric searchable encryption: optimality and forward security. IACR Cryptology ePrint Archive p 62
5. Cai C, Yuan X, Wang C (2017) Towards trustworthy and private keyword search in encrypted decentralized storage. In: 2017 IEEE International Conference on Communications (ICC), IEEE, pp 1–7. https://doi.org/10.1109/ICC.2017.7996810
6. Cai K, Hong C, Zhang M, Feng D, Lv Z (2013) A secure conjunctive keywords search over encrypted cloud data against inclusion-relation attack. In: 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, IEEE, vol 1, pp 339–346. https://doi.org/10.1109/CloudCom.2013.51
7. Cao N, Wang C, Lia M, Ren K, Lou W (2014) Privacy-preserving multi-keyword ranked search over encrypted cloud data. IEEE Trans Parallel Distrib Syst 25(1):222–233. https://doi.org/10.1109/TPDS.2013.45
8. Cash D, Jaeger J, Jarecki S, Jutla CS, Krawczyk H, Rosu MC, Steiner M (2014) Dynamic searchable encryption in very-large databases: data structures and implementation. In: NDSS, Citeseer, vol 14, pp 23–26. https://doi.org/10.14722/ndss.2014.23264
9. Chang YC, Mitzenmacher M (2005) Privacy preserving keyword searches on remote encrypted data. In: International Conference on Applied Cryptography and Network Security, Springer, pp 442–455. https://doi.org/10.1007/11496137_30
10. Chase M, Kamara S (2010) Structured encryption and controlled disclosure. In: International Conference on the Theory and Application of Cryptology and Information Security, Springer, pp 577–594. https://doi.org/10.1007/978-3-642-17373-8_33
11. Chen R, Mu Y, Yang G, Guo F, Wang X (2016) Dual-server public-key encryption with keyword search for secure cloud storage. IEEE Trans Inf Forensics Secur 11(4):789–798. https://doi.org/10.1109/TIFS.2015.2510822
12. Cheng R, Yan J, Guan C, Zhang F, Ren K (2015) Verifiable searchable symmetric encryption from indistinguishability obfuscation. In: Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, ACM, pp 621–626. https://doi.org/10.1145/2714576.2714623
13. Cui J, Zhou H, Zhong H, Xu Y (2018) Akser: attribute-based keyword search with efficient revocation in cloud computing. Inf Sci 423:343–352. https://doi.org/10.1016/j.ins.2017.09.029
14. Curtmola R, Garay J, Kamara S, Ostrovsky R (2006) Searchable symmetric encryption: improved definitions and efficient constructions. In: 13th ACM Conference on Computer and Communications Security
15. Ding M, Gao F, Jin Z, Zhang H (2012) An efficient public key encryption with conjunctive keyword search scheme based on pairings. In: 2012 3rd IEEE International Conference on Network Infrastructure and Digital Content, IEEE, pp 526–530. https://doi.org/10.1109/ICNIDC.2012.6418809
16. Du M, Wang Q, He M, Weng J (2018) Privacy-preserving indexing and query processing for secure dynamic cloud storage. IEEE Trans Inf Forensics Secur 13(9):2320–2332. https://doi.org/10.1109/TIFS.2018.2818651
17. Farràs O, Ribes-González J (2019) Provably secure public-key encryption with conjunctive and subset keyword search. Int J Inf Secur. https://doi.org/10.1007/s10207-018-00426-7
18. Fu Z, Shu J, Sun X, Linge N (2014a) Smart cloud search services: verifiable keyword-based semantic search over encrypted cloud data. IEEE Trans Consum Electr 60(4):762–770. https://doi.org/10.1109/TCE.2014.7027353
19. Fu Z, Sun X, Linge N, Zhou L (2014b) Achieving effective cloud search services: multi-keyword ranked search over encrypted cloud data supporting synonym query. IEEE Trans Consum Electr 60(1):164–172. https://doi.org/10.1109/TCE.2014.6780939
20. Fu Z, Wu X, Guan C, Sun X, Ren K (2016) Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement. IEEE Trans Inf Forensics Secur 11(12):2706–2716. https://doi.org/10.1109/TIFS.2016.2596138
21. Fu Z, Wu X, Wang Q, Ren K (2017) Enabling central keyword-based semantic extension search over encrypted outsourced data. IEEE Trans Inf Forensics Secur 12(12):2986–2997. https://doi.org/10.1109/TIFS.2017.2730365
22. Fu Z, Xia L, Sun X, Liu AX, Xie G (2018) Semantic-aware searching over encrypted data for cloud computing. IEEE Trans Inf Forensics Secur 13(9):2359–2371. https://doi.org/10.1109/TIFS.2018.2819121
23. Goh EJ et al (2003) Secure indexes. IACR Cryptol ePrint Archive 2003:216

24. Han F, Qin J, Hu J (2016) Secure searches in the cloud: a survey. Fut Gener Comput Syst 62:66–75. https://doi.org/10.1016/j.future.2016.01.007
25. Höfer C, Karagiannis G (2011) Cloud computing services: taxonomy and comparison. J Internet Serv Appl 2(2):81–94. https://doi.org/10.1007/s13174-011-0027-x
26. Hu S, Cai C, Wang Q, Wang C, Luo X, Ren K (2018) Searching an encrypted cloud meets blockchain: a decentralized, reliable and fair realization. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, IEEE, pp 792–800. https://doi.org/10.1109/INFOCOM.2018.8485890
27. Huang Q, Li H (2017) An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks. Inf Sci 403:1–14. https://doi.org/10.1016/j.ins.2017.03.038
28. Hur J, Noh DK (2011) Attribute-based access control with efficient revocation in data outsourcing systems. IEEE Trans Parall Distrib Syst 22(7):1214–1221. https://doi.org/10.1109/TPDS.2010.203
29. Hwang MS, Hsu ST, Lee CC (2014) A new public key encryption with conjunctive field keyword search scheme. Inf Technol Control 43(3):277–288. https://doi.org/10.5755/j01.itc.43.3.6429
30. Jeong IR, Kwon JO, Hong D, Lee DH (2009) Constructing PEKS schemes secure against keyword guessing attacks is possible? Comput Commun 32(2):394–396. https://doi.org/10.1016/j.comcom.2008.11.018
31. Kalapatapu A, Sarkar M (2012) Cloud computing: an overview. Cloud Comput Methodol Syst Appl. https://doi.org/10.1201/b11149-8
32. Kuzu M, Islam MS, Kantarcioglu M (2012) Efficient similarity search over encrypted data. In: 2012 IEEE 28th International Conference on Data Engineering, IEEE, pp 1156–1167. https://doi.org/10.1109/ICDE.2012.23
33. Li J, Wang Q, Wang C, Cao N, Ren K, Lou W (2010) Fuzzy keyword search over encrypted data in cloud computing. In: Proceedings 2010 IEEE INFOCOM, IEEE, pp 1–5. https://doi.org/10.1109/INFCOM.2010.5462196
34. Li R, Liu AX (2017) Adaptively secure conjunctive query processing over encrypted data for cloud computing. In: 2017 IEEE 33rd International Conference on Data Engineering (ICDE), IEEE, pp 697–708. https://doi.org/10.1109/ICDE.2017.122
35. Li Y, Zhou F, Qin Y, Lin M, Xu Z (2018) Integrity-verifiable conjunctive keyword searchable encryption in cloud storage. Int J Inf Secur 17(5):549–568. https://doi.org/10.1007/s10207-017-0394-9
36. Liang K, Susilo W (2015a) Searchable attribute-based mechanism with efficient data sharing for secure cloud storage. IEEE Trans Inf Forensics Secur 10(9):1981–1992. https://doi.org/10.1109/TIFS.2015.2442215
37. Liang K, Susilo W (2015b) Searchable attribute-based mechanism with efficient data sharing for secure cloud storage. IEEE Trans Inf Forensics Secur 10(9):1981–1992. https://doi.org/10.1109/TIFS.2015.2442215
38. Liu P, Wang J, Ma H, Nie H (2014) Efficient verifiable public key encryption with keyword search based on kp-abe. In: 2014 Ninth International Conference on Broadband and Wireless Computing, Communication and Applications, IEEE, pp 584–589. https://doi.org/10.1109/BWCCA.2014.119
39. Liu Z, Lv S, Wei Y, Li J, Liu JK, Xiang Y (2017) Ffsse: flexible forward secure searchable encryption with efficient performance. IACR Cryptol ePrint Archive 2017:1105
40. Ma S (2016) Identity-based encryption with outsourced equality test in cloud computing. Inf Sci 328:389–402. https://doi.org/10.1016/j.ins.2015.08.053
41. Miao Y, Ma J, Liu X, Liu Z, Shen L, Wei F (2016) Vmkdo: verifiable multi-keyword search over encrypted cloud data for dynamic data-owner. Peer-to-Peer Netw Appl. https://doi.org/10.1007/s12083-016-0487-7
42. Miao Y, Ma J, Jiang Q, Li X, Sangaiah AK (2018) Verifiable keyword search over encrypted cloud data in smart city. Comput Electr Eng 65:90–101. https://doi.org/10.1016/j.compeleceng.2017.06.021
43. Ogata W, Kurosawa K (2016) Efficient no-dictionary verifiable SSE. IACR Cryptol ePrint Archive 2016:981
44. Pasupuleti SK, Ramalingam S, Buyya R (2016) An efficient and secure privacy-preserving approach for outsourced data of resource constrained mobile devices in cloud computing. J Netw Comput Appl 64:12–22. https://doi.org/10.1016/j.jnca.2015.11.023
45. Pham H, Woodworth J, Salehi MA (2018) Survey on secure search over encrypted data on the cloud. arXiv preprint arXiv:181109767
46. Poh GS, Chin JJ, Yau WC, Choo KKR, Mohamad MS (2017) Searchable symmetric encryption: designs and challenges. ACM Comput Surv (CSUR) 50(3):40. https://doi.org/10.1145/3064005

47. Premkamal PK, Pasupuleti SK, Alphonse P (2018) A new verifiable outsourced ciphertext-policy attribute based encryption for big data privacy and access control in cloud. J Ambient Intell Human Comput 10:2693–2707
48. Qian L, Luo Z, Du Y, Guo L (2009) Cloud computing: An overview. In: IEEE International Conference on Cloud Computing, Springer, pp 626–631. https://doi.org/10.1007/978-3-642-10665-1_63
49. Sarga L (2012) Cloud computing: an overview. J Syst Integr 3(4):3–14. https://doi.org/10.20470/jsi.v3i4.131
50. Shen J, Shen J, Chen X, Huang X, Susilo W (2017a) An efficient public auditing protocol with novel dynamic structure for cloud data. IEEE Trans Inf Forensics Secur 12(10):2402–2415. https://doi.org/10.1109/TIFS.2017.2705620
51. Shen J, Wang C, Wang A, Ji S, Zhang Y (2018) A searchable and verifiable data protection scheme for scholarly big data. IEEE Trans Emerg Topics Comput. https://doi.org/10.1109/TETC.2018.2830368
52. Shen Z, Shu J, Xue W (2017b) Keyword search with access control over encrypted cloud data. IEEE Sens J 17(3):858–868. https://doi.org/10.1109/JSEN.2016.2634018
53. Song DX, Wagner D, Perrig A (2000) Practical techniques for searches on encrypted data. In: Proceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000, IEEE, pp 44–55. https://doi.org/10.1109/SECPRI.2000.848445
54. Su S, Teng Y, Cheng X, Xiao K, Li G, Chen J (2015) Privacy-preserving top-k spatial keyword queries in untrusted cloud environments. IEEE Trans Serv Comput. https://doi.org/10.1109/TSC.2015.2481900
55. Sun J, Wang X, Wang S, Ren L (2018) A searchable personal health records framework with fine-grained access control in cloud-fog computing. PloS One 13(11):e0207543. https://doi.org/10.1371/journal.pone.0207543
56. Sun W, Wang B, Cao N, Li M, Lou W, Hou YT, Li H (2013) Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking. In: Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, ACM, pp 71–82
57. Sun W, Liu X, Lou W, Hou YT, Li H (2015) Catch you if you lie to me: Efficient verifiable conjunctive keyword search over large dynamic encrypted cloud data. In: 2015 IEEE Conf Comput Commun (INFOCOM), IEEE, pp 2110–2118
58. Sun W, Yu S, Lou W, Hou YT, Li H (2016) Protecting your right: verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud. IEEE Trans Parallel Distrib Syst 27(4):1187–1198. https://doi.org/10.1109/TPDS.2014.2355202
59. Wan Z, Deng RH (2018) Vpsearch: achieving verifiability for privacy-preserving multi-keyword search over encrypted cloud data. IEEE Trans Depend Secure Comput 15(6):1083–1095. https://doi.org/10.1109/TDSC.2016.2635128
60. Wang B, Yu S, Lou W, Hou YT (2014) Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud. IEEE INFOCOM 2014-IEEE Conference on Computer Communications pp 2112–2120. https://doi.org/10.1109/INFOCOM.2014.6848153
61. Wang C, Cao N, Ren K, Lou W (2012) Enabling secure and efficient ranked keyword search over outsourced cloud data. IEEE Trans Parallel Distrib Syst 23(8):1467–1479. https://doi.org/10.1109/TPDS.2011.282
62. Wang N, Fu J, Bhargava BK, Zeng J (2018) Efficient retrieval over documents encrypted by attributes in cloud computing. IEEE Trans Inf Forensics Secur 13(10):2653–2667. https://doi.org/10.1109/TIFS.2018.2825952
63. Wang Y, Wang J, Chen X (2016) Secure searchable encryption: a survey. J Commun Inf Netw 1(4):52–65. https://doi.org/10.1007/BF03391580
64. Wu D, Gan Q, Wang X (2018) Verifiable public key encryption with keyword search based on homomorphic encryption in multi-user setting. IEEE Access 6:42445–42453. https://doi.org/10.1109/ACCESS.2018.2861424
65. Wu Z, Li K (2019) Vbtree: forward secure conjunctive queries over encrypted data for cloud computing. VLDB J 28(1):25–46. https://doi.org/10.1007/s00778-018-0517-6
66. Xia Z, Wang X, Sun X, Wang Q (2016) A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE Trans Parallel Distrib Syst 27(2):340–352. https://doi.org/10.1109/TPDS.2015.2401003
67. Xu K, Wang G, Wang S, Zhao Z, Wang J (2017) A secure channel free conjunctive keyword search without random oracle under simple assumption. In: 2017 IEEE 9th International Conference on

Communication Software and Networks (ICCSN), IEEE, pp 1467–1476. https://doi.org/10.1109/ICCSN.2017.8230352
68. Xu P, Jin H, Wu Q, Wang W (2013a) Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack. IEEE Trans Comput 62(11):2266–2277. https://doi.org/10.1109/TC.2012.215
69. Xu P, Jin H, Wu Q, Wang W (2013b) Public-key encryption with fuzzy keyword search: a provably secure scheme under keyword guessing attack. IEEE Trans Comput 62(11):2266–2277. https://doi.org/10.1109/TC.2012.215
70. Yan J, Zhang Y, Liu X (2016) Secure multi-keyword search supporting dynamic update and ranked retrieval. China Commun 13(20):209–221. https://doi.org/10.1109/CC.2016.7733045
71. Yang Y, Ma M (2016) Conjunctive keyword search with designated tester and timing enabled proxy re-encryption function for e-health clouds. IEEE Trans Inf Forensics Secur 11(4):746–759. https://doi.org/10.1109/TIFS.2015.2509912
72. Yu J, Lu P, Zhu Y, Xue G, Li M (2013) Toward secure multikeyword top-k retrieval over encrypted cloud data. IEEE Trans Depend Secure Comput 10(4):239–250. https://doi.org/10.1109/TDSC.2013.9
73. Yuan X, Wang X, Wang C, Yu C, Nutanong S (2017) Privacy-preserving similarity joins over encrypted data. IEEE Trans Inf Forensics Secur 12(11):2763–2775. https://doi.org/10.1109/TIFS.2017.2721221
74. Wang Q, He M, Du M, Chow SS, Lai RW, Zou Q (2018) Searchable encryption over feature-rich data. IEEE Trans Depend Secure Comput 15(3):496–510. https://doi.org/10.1109/TDSC.2016.2593444
75. Zhang W, Lin Y, Xiao S, Wu J, Zhou S (2016) Privacy preserving ranked multi-keyword search for multiple data owners in cloud computing. IEEE Trans Comput 65(5):1566–1577. https://doi.org/10.1109/TC.2015.2448099
76. Zhang W, Lin Y, Qi G (2018) Catch you if you misbehave: ranked keyword search results verification in cloud computing. IEEE Trans Cloud Comput 6(1):74–86. https://doi.org/10.1109/TCC.2015.2481389
77. Zheng Q, Xu S, Ateniese G (2014) Vabks: verifiable attribute-based keyword search over outsourced encrypted data. In: IEEE INFOCOM 2014-IEEE Conference on Computer Communications, IEEE, pp 522–530. https://doi.org/10.1109/INFOCOM.2014.6847976
78. Zhu J, Li Q, Wang C, Yuan X, Wang Q, Ren K (2018) Enabling generic, verifiable, and secure data search in cloud services. IEEE Trans Parallel Distrib Syst 29(8):1721–1735. https://doi.org/10.1109/TPDS.2018.2808283