CMSC5726
Computer and Network Security Project

Searchable Encryption

by

Shenghao XU

1155148147

May. 2021

Abstract

Cryptography is a young and ancient discipline. Its use to protect military and diplomatic communications dates back thousands of years. In modern times, due to the rapid development of computers and communication systems, the need for digital information protection and security services for individuals has further increased. Modern cryptography originated from a large number of related theories that appeared in the middle and late 20th century. In 1949, C. E. Shannon published a classic paper entitled *"Communication Theory of Secure Systems"* [18], marking the beginning of modern cryptography. Ciphers can be classified into *Symmetric Cryptography* and *Asymmetric Cryptography* by determining whether the same key is used. In addition to the encryption and decryption of information, there is also the *One-Way Hash Function* technology used to confirm data integrity. Combining encryption and decryption techniques in cryptography and one-way hashing techniques, digital certificates have been proposed for preventing tampering and forgery.

In recent years, with the rise of GPU and cloud computing, the security of traditional cryptography has been challenged, which leads to the emergence of new cryptography technologies. In this report, the state-of-the-art cryptography technology-*searchable encryption* (SE) will be illustrated and discussed.

Contents

Chapter 1

Introduction

In this chapter, the background of searchable encryption (SE) will be presented, while we will introduce the schemes of searchable encryption and its different algorithms. Then the basic algorithm for SE schemes will illustrate.

1.1 General Introduction

In this era of information, we deal with a tremendous amount of data on a daily basis, whether for private or commercial use. The emergence of cloud technology has streamlined information processing, and users no longer need to have all their data or software installed on a local computer. With cloud technology, users can get the data they want on any machine, anytime, anywhere. Cloud service suppliers store data and set up virtual system environments for users on cloud-based servers, and carry out data transmission and operation with users through the network. Since the data is stored in the cloud out of the user's physical control, cloud server administrators and unauthorized users (e.g., hackers and other users who do not have access privileges) can access the data to obtain the information contained in the data, which will potentially cause the leakage of data information and user privacy.

In order to ensure the security of the data, users can encrypt the data before uploading it to the cloud and store the data in the form of ciphertext on the cloud server. However, although this approach enhances the security of the data, however, when users need to find relevant files containing a certain keyword, how to search the ciphertext in the cloud server will become a problem that users need to face. The traditional method to solve this problem is to download all ciphertext data locally for decryption and then perform a keyword search on the plaintext, but this method not only wastes network and storage overhead by downloading a large amount of unrelated data but also reduces computational efficiency due to decryption and search operations.

To address the aforesaid problem better, searchable encryption is proposed. Searchable encryption is intended to provide both confidentiality and searchability. The data proprietor can entrust specific query tokens to allow the cloud server to implement queries on encrypted data. The ciphertext search mechanism was first proposed in [12], the authors proposed Oblivious RAMs that are used to support complex query processing with adaptive security, however, this solution requires multiple rounds of interaction between the user and the server-side, which makes it very inefficient. In 2000, the authors in [19] proposed a searchable encryption method based on symmetric cryptography, which pioneered the use of a practical searchable encryption mechanism to implement keyword search on ciphertexts. In 2014, the public key cryptography based searchable encryption was proposed by authors in [6] . Based on these researches, not only more searchable encryption algorithms are proposed theoretically, but also searchable encryption approaches are applied in real application scenarios. Such as the method proposed in [20], the proposed method implements a ciphertext search on the log data.

1.2 Searchable Encryption Categories

Searchable encryption research content can be broadly divided into two categories: (1) searchable encryption schemes, (2) searchable encryption model. The searchable encryption schemes contain two major branches.

First is the keyword support, which focuses on the research about what kind of keyword format that users can search with. Single keyword, multi keywords, conjunctive keywords, and flexible query are the four major categories within the domain of keyword support. The second is encryption, the major techniques are searchable symmetric encryption (SSE) and public-key encryption with keyword search (PEKS).

The second category is the searchable encryption model that searchable encryption schemes are built upon the client and server model. For the various types of models, the server stores encrypted data on account of single or multiple clients (i.e., owner). If one or more clients (i.e., searcher) want to request the data from the server, those clients need to be able to generate the token for the server, the server then performs the search on behalf of the client [7]. The resulting SE model can be divided into the following four structures:

1. Single owner-Single searcher

2. Multi owner-Single searcher

3. Single owner-Multi searcher

4. Multi owner-Multi searcher

where, for single owner-single searcher structure is suitable for data outsourcing and for other three structures perform better on data sharing.

1.3 Basic Algorithms for Searchable Encryption Scheme

The numerous searchable encryption schemes available today can be divided into two categories depending on the construction algorithm: symmetric-key cryptography based SE schemes and public-key cryptography based SE schemes. The former is mainly built using the pseudorandom generator, pseudorandom function, hashing algorithm, and symmetric encryption algorithm. For public-key cryptography based SE schemes mainly uses algebraic tools such as bilinear projections and base security on the intractability of some complex problems. In the following, we will discuss two types of SE schemes based on symmetric cryptography algorithms and public-key cryptography algorithms, respectively,

Although different SE schemes have different construction methods, the SE scheme basically consists of the following four algorithms [5]:

1. *Setup.* An algorithm that utilizes security parameters as input and output keys. For public-key cryptography based SE schemes, the algorithm base on the security parameters to generate the public key and private key. On the other hand, private keys will be generated (e.g. key of pseudorandom function) for symmetric-key cryptography based SE schemes .

2. *Generate Token.* Enter the query keyword to the algorithm, which will output the token.

3. *Build Index.* The data owner will select the appropriate set of keywords based on the content of the data and create an index table using SE schemes. For symmetric-key cryptography based SE schemes, the data owner uses the public key to encrypt the keyword set for each data. For public-key cryptography based SE schemes, the data owner will encrypt the keyword set using a symmetric key or using a key-based hashing algorithm. The main body of the data will be encrypted using the symmetric encryption algorithm.

4. *Query.* The server will take the received token and the index table in each data as input. Judge whether the data satisfies the search request by determining whether the algorithm output is the same as the predefined result. If the output meets the pre-defined results, then the server will send the search results back to the data searcher.

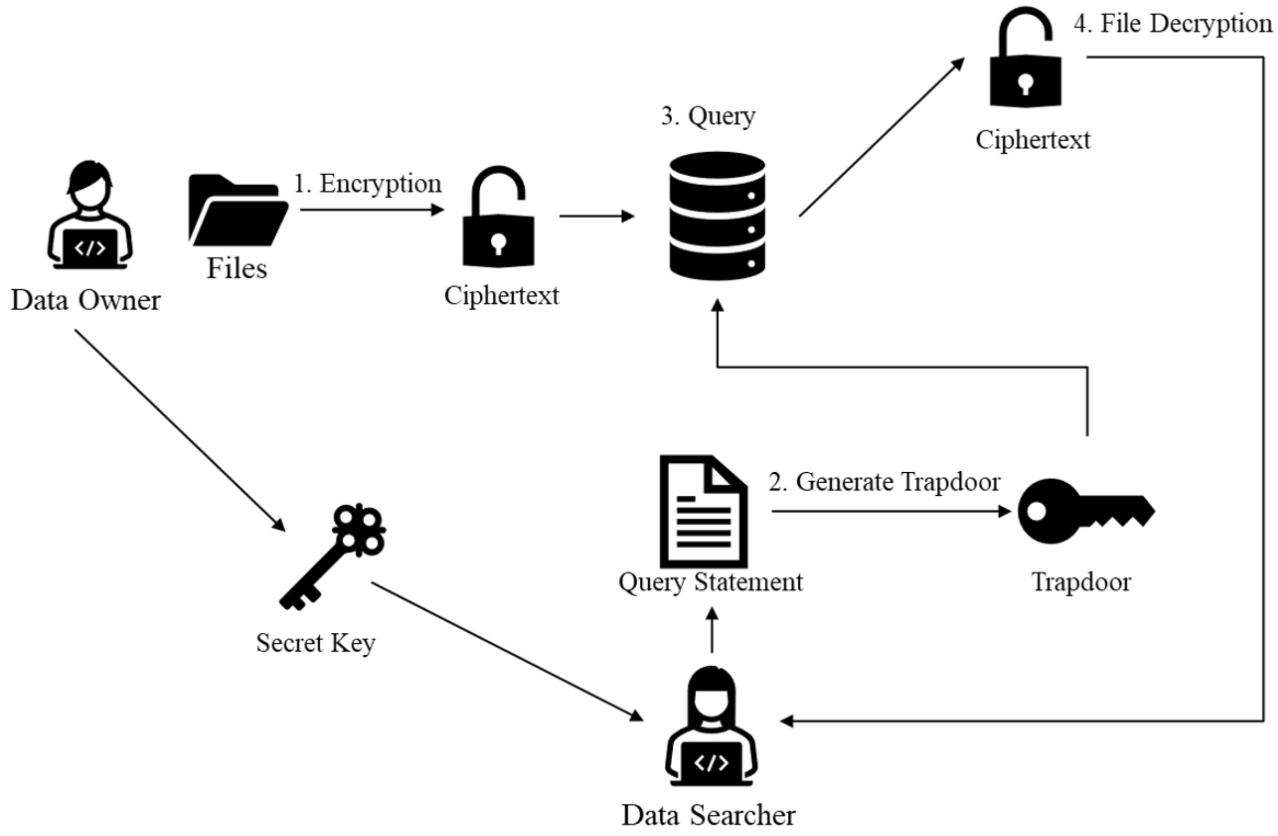


Figure 1.1: Searchable encryption procedures

With the above basic algorithm, the general process of searchable cryptography can be divided into 4 main steps, as shown in Figure 1.1:

1. Encryption: The data owner encrypts the file to be uploaded using the encryption key and uploads the ciphertext to the server.

2. Trapdoor generation: Authorized data searchers use the secret key to generate trapdoors for the keywords to be queried and send them to the server.

3. Query: The server retrieves the index table of each uploaded file and trapdoor submitted by the data searcher, then returns the ciphertext that containing the trapdoor keywords.

4. Decryption: The data searcher uses the decryption key to decrypt the ciphertext returned by the server.

Note that owner does not necessarily have to create the index on the plaintext. Based on this point we can divide the SE scheme into two categories:

- Index-based schemes: For each keyword, an index is created that contains pointers to all files containing that keyword. Since this scheme makes a lot of changes to the indexes when updating data, so that it is suitable for mostly read-only modes.

- Sequential scan: Due to there is no index in this scheme so that each query requires a scan of the files to figure out the match file. Undoubtedly, the efficiency of this scheme will be much lower, but it is more suitable for storage systems that are updated frequently.

Chapter 2

Symmetric-Key Cryptography Based SE Schemes

In this chapter, the symmetric-key cryptography based SE schemes will be illustrate and formulate. followed by the efficiency and security of symmetric-key cryptography based SE schemes.

2.1 Formulation

In symmetric-key cryptography, encryption and decryption are performed using the same key, which means that it requires the sender and receiver to mutually agree on the key before communicating securely. The security of symmetric-key cryptography algorithms relies on the key and compromising the key means that anyone can decrypt the messages they send or receive, so the confidentiality of the key is critical to communication. The advantage of symmetric-key cryptography algorithm is that the encryption and decryption are done by the same key, so the computational overhead is low, and it is suitable for encrypting large blocks of data, while the disadvantage is that the encryptor (i.e. sender) and decryptor (i.e. receiver) need to negotiate the key in advance, and the key needs to be transmitted through a secure channel. Since the key needs to be transmitted, then the security risk in the transmission process reduces the security of the symmetric-key cryptography algorithm.

Based on the symmetric-key cryptography, searchable symmetric encryption (SSE) has been proposed (also known as symmetric-key cryptography based SE schemes) that can perform query functions on encrypted files and in the process do not reveal any plaintext information about the data. By the definition proposed in [9], the searchable symmetric encryption can be defined as follows:

Definition: Searchable symmetric encryption (SSE)

$\Delta = (w_1, \cdots w_d)$ denote a dictionary containing d words in alphabetical order, and 2^Δ denote the set of all possible documents containing the words Δ.
For an SSE scheme over a dictionary Δ is the ensemble of five polynomial-time algorithms $SSE = (GenKey, Encrypt, Trapdoor, Search, Decrypt)$:

1. $K \leftarrow GenKey\,(\alpha)$: An algorithm run by the owner of data to set up the scheme. A security parameter α and a secret key K are the input and output of the algorithm, respectively.

2. $(I, c) \leftarrow Encrypt(K, D)$: The secret key and set of the plaintexts $D = (D_1, \cdots D_n)$are the input for this algorithm and the output index and ciphertexts set $c = (c_1, \cdots c_n)$. For SSE schemes do not adopt searchable index that $I = \emptyset$.

3. $T_W \leftarrow Trapdoor\,(K, W)$: when secret key K and keyword W is entered into the algorithm,it will calculate and outputs the trapdoor T_W of keyword W.

4. $X \leftarrow Search\,(I, T_W)$: The algorithm is run by the server in order to find which plaintexts in D contain the keyword W. With the input and the trapdoor T_W, a set of plaintext identifiers denote by X will be the result of the algorithm.

5. $D_i \leftarrow Decrypt\,(K, c_i)$:If the searcher wants to recover a plaintext, he/she needs to provide the secret key K and the ciphertext c_i to the algorithm to receive the output plaintext D_i.

For the definition presented in [9], An SSE scheme is correct that for $\forall \alpha \in \mathbb{N}$, $\forall n \in \mathbb{Z}$,$\forall w \in \Delta$, $\forall D \subseteq 2^\Delta$ and $\forall K$ output by $GenKey\,(\alpha)$, $\forall\,(I, c)$ output by $Encrypt(K, D)$, the following equation holds:

$$Search\left(I, Trapdoor\left(K, W\right)\right) = D(W) \tag{2.1}$$

$$Decrypt\left(K, c_i\right) = D_i \tag{2.2}$$

where, $1 \leq i \leq n$. Symmetric searchable encryption usually processes the keywords first, mostly using methods such as pseudorandom functions or hashing algorithms. When the user performs a keyword query, the query keyword is processed identically as the keyword of the document. If the keyword for query is matched with the keyword of the document and satisfies a certain format, the match is successful and the corresponding document will be returned.

2.2 SWP scheme

The first practical SE scheme SWP was proposed in [19]. The SWP scheme searches encrypted data by using a particular two-layer encryption structure that enables the ciphertext to be searched using sequential scanning which means that if you want to find the corresponding ciphertext, you need to scan all the files. The main ideology of the SWP scheme is to separately encrypt each word and then imbed a hash value with a particular pattern in the ciphertext. SWP Scheme performs the search by uses the server to compare the extracted specially formatted hash value from both query and ciphertext.

Details of Scheme. As shown in Figure 2.1, in order to create a searchable ciphertext, first the content of the file is split into fixed-size word blocks W_i, then encrypt the W_i through the deterministic algorithm $E\left(\cdot\right)$. The encrypted word block $E\left(W_i\right)$ is divided into two parts, L_i and R_i, respectively, by using the X_i denote the encrypted word block, where $X_i = E\left(W_i\right) = \langle L_i, R_i \rangle$. Using the pseudorandom generator can generate pseudorandom values S_i with the help of stream cipher. Then using a pseudorandom function $f\left(\cdot\right)$ to compute the key $k_i := f_{k'}\left(L_i\right)$, and the key k_i is used for the hash function $F\left(\cdot\right)$ to perform the hashing on S_i. Based on the above operation, we can get $T_i = \langle (S_i, F_{k_i}\left(S_i\right) \rangle$. The ciphertext C_i can be generate through perform the XOR operation on the T_i and X_i, where $C_i = X_i \oplus T_i$.

In order to implement the searching, the trapdoor is required. The trapdoor comprises

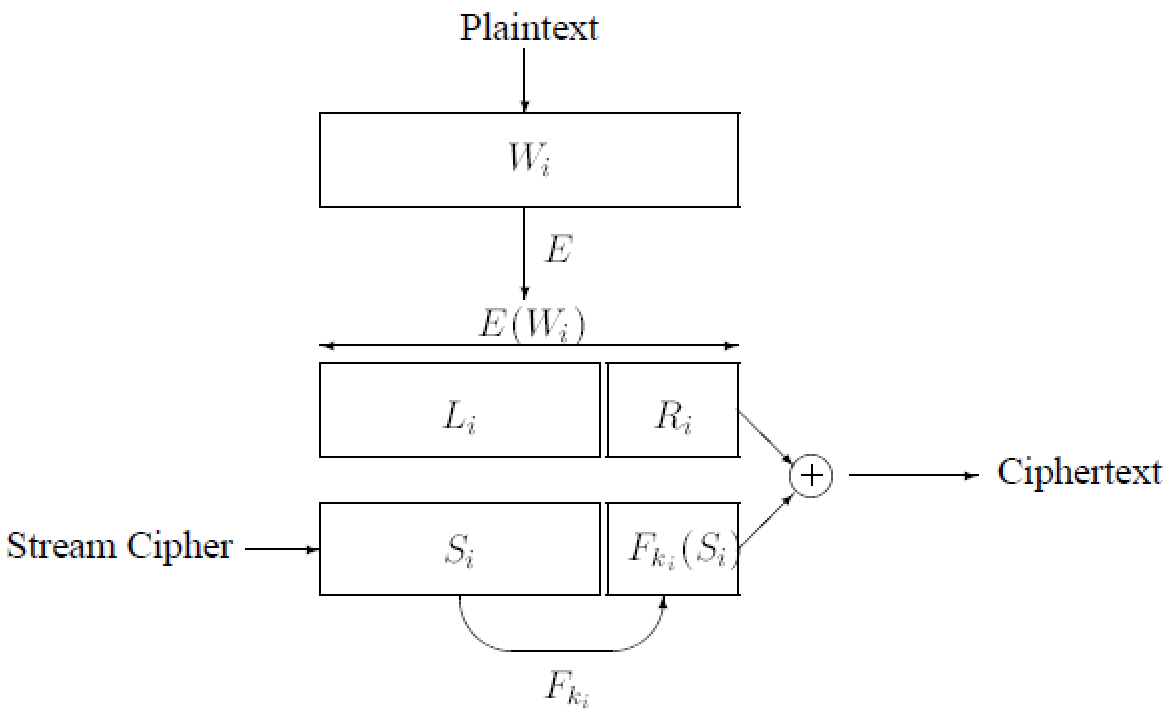


Figure 2.1: Encryption of SWP in [19]

the encrypted keywords for searching $X = E(W) = \langle L, R \rangle$ and the key $k := f_{k'}(L)$. As shown in Figure 2.2, based on this trapdoor, the server can scan by examining all stored ciphertext C_i, If the result of $C_i \oplus X$ is the $\langle s, F_k(s) \rangle$ form of certain s, the keyword is found.

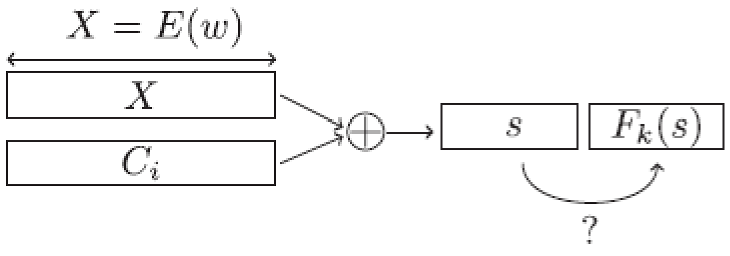


Figure 2.2: Sequential search of SWP in [19]

Efficiency. For the SWP scheme, the complexity of the encryption and search algorithms has a linear relationship with the total number of words per file (i.e., in the worst case, the entire file needs to be traversed). From this point we can see that the SWP scheme is less efficient, a single word query needs to scan the entire file, taking up considerable computing resources on the server.

Security. SWP was the first scheme propose for searchable encryption, and at that time there was no formal definition of searchable encryption security to analyze the security of it. There is a threat of statistical attacks on the security of SWP. For example, an attacker can guess whether a keyword is a common term by counting the number of times it appears in a file.

Summary of SWP. The major drawback of the SWP scheme is that it must use fixed-size words W_i which do not compatible with existing file encryption standards. Secondly, due to the unique two-layer encryption approach of SWP, this methodology can only be used for plain text data and cannot be used for other data such as compressed data [7].

While this work is difficult to meet today's rich and rigorous definition of security. However, it should be noted that it was the presentation of the SWP scheme that drew the attention of academia and industry to searchable encryption and a series of enduring studies.

2.3 Z-IDX Scheme

In order to address the problems of the SWP scheme mentioned above, *Goh* proposed a method in [11] to index each file based on Bloom filter (BF) [4] called Z-IDX.

Details of Scheme. Bloom filter is consist of a binary array M (m bits) and a hash functions h_t, used to determine whether or not the keyword exists in a set, where $h_t : \{0,1\}^* \rightarrow [1,2,\ldots,m], t = 1,2,\ldots,r$. At the beginning, each positions of arrary M are set to 0. If the keyword k exist in the set $S = \{k_1,\ldots,k_m\}$, then set bits value of the positions $h_1(k_i),\ldots,h_r(k_i)$ to 1. Therefore, to determine whether the keyword k belongs to S, we only need to check the bits $M[h_1(k)],\ldots,M[h_r(k)]$, if all the bits are 1, then k belongs to S; otherwise, k does not belong to S.

Efficiency. Since index generation must generate a Bloom filter for each file, the algorithm is linearly related to the number of different words in each file. Therefore, the search efficiency for the Z-IDX scheme is linearly related to the number of documents, while the search efficiency in the SWP scheme is proportional to the number of keywords. In terms of spatial efficiency, in addition to storing the ciphertext of the file, the server also needs to record the file index. When the length of the file is short, its index may be several times the length of the file, so the spatial utilization rate is low.In terms of time efficiency, the query requires the server to calculate and judge each file, and the time consumption of the whole keyword query operation is $O(n)$, where n is the number of files stored on the server, which is time inefficient.

Security. The Z-IDX is proven semantic security against adaptive chosen keyword attacks (**IND-CKA**) security [11]. However, such a definition does not guarantee the safety of trapdoor security. That said, the possibility exists under the **IND-CKA** definition that the server can recover the keywords queried from the trapdoor.

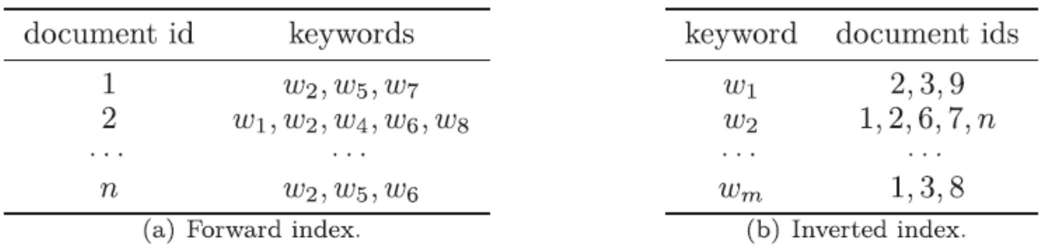
2.4 Parallel and dynamic SSE

Figure 2.3 illustrates two different ways of constructing an index. The construction method called forward index is shown in (a), which is the index construction method adopted by Z-IDX [11]. Forward index constructs a data structure with a file as the basic unit (e.g. Bloom filter in Z-IDX). When querying, the operation of the data structure is used to determine whether the keyword to be retrieved exists in a file. If there exist n files in the server so that the search complexity is $O(n)$. The second method is the inverted index, as shown in (b). The key idea of an inverted index is to construct an index for each keyword in the server instead of each file. Under the optimal situation, the search complexity is $O(|D(W)|)$, where $|D(W)|$ is the number of files that contain the keyword W.

Compared with these two methods, the inverted index method is more time-efficient than a forward index method. However, for the inverted index, the index needs to be reset when the file is updated, so the update overhead of this method is very high. The forward index method is more efficient for file updates but its query speed is slower. How to combine the advantages of the two methods to support efficient querying and file updating at the same time has been a long-standing concern in academia. Next we will introduce the state-of-the-art scheme in SSE - Parallel and dynamic SSE.

2.4.1 KPR: Dynamic SSE

The authors in [15] proposed a scheme called KPR which is an extension of the SSE-1 scheme [9]. KPR provides the database with the ability to add or delete data, and modify files for efficient updates.



document id	keywords
1	w_2, w_5, w_7
2	w_1, w_2, w_4, w_6, w_8
...	...
n	w_2, w_5, w_6

(a) Forward index.

keyword	document ids
w_1	$2, 3, 9$
w_2	$1, 2, 6, 7, n$
...	...
w_m	$1, 3, 8$

(b) Inverted index.

Figure 2.3: Unencrypted forward and inverted index in [7]

Details of Scheme. KPR redefines arrays and tables in SSE-1 as search array A_s and search table T_s and in conjunction with the inverted index construction method, introduce an additional deletion array A_d and deletion table T_d to record and track the keywords contained in each file. When adding or deleting a file, the user sends the corresponding token to the server, and the server uses these tokens to update the relevant data and ciphertext.

Efficiency. For the KPR scheme, it takes 8 pseudo-random functions to generate the index of a keyword. To perform a search in KPR, the server performs a table lookup on the first node and performs an XOR operation on the others to decrypt them, where the file containing the keywords queried by the user is treated as a node.

Security. KPR defines a generalization adaptive security against chosen-keyword attacks (**CKA2**) security [15]. Under this definition, allow small amounts of information to be leaked in updates. Contrary to formerly known schemes, the KPR is secure in the random oracle model (RO).

2.4.2 KP: Parallel and Dynamic SSE

KP is a new highly parallelized dynamic SSE scheme that takes advantage of the CUP multi-core architecture [14].

Details of Scheme.The authors' proposed method in [14]is based on a new tree-based multi-map data structure called keyword red-black (KRB) tree. The KRB tree has a pointer to the file as a leaf, if at least one of the nodes below is the path to the file identifier containing the keyword, then each node stores the information [7].

Efficiency. Under the condition that for n files indexed by m keywords and p cores of CUP available, the KP need spend $O((r/p)\log n)$ parallel time and $O(r \log n)$ sequential time for searching a keyword w, where r denote the how much files contain the keyword

w.

Security. In KP, a variant of **CKA2** security was defined that compare with the KPR's **CKA2** when performed the update operation, no leak of information anymore

2.5 Summary of SSE

In this chapter, we first introduce and define the SSE, and then discuss the different SSE schemes. The compression of the four SSE schemes introduced above is shown in Table 2.1.

Scheme	*dynamism*	*search efficiency*	*security*	*index size*
SWP [19]	static	$O(\frac{n}{p})$	CPA	Ø
Z-IDX [11]	dynamic	$O(\frac{n}{p})$	CKA1	$O(n)$
KPR [15]	dynamic	$O(r)$	CKA2	$O(mn)$
KP [14]	dynamic	$O((r/p)\log n)$	CKA2	$O(m+n)$

Table 2.1: Comparison between different SSE schemes.

In the Table 2.1, n denote the number of files, m denote the number of keywords. Furthermore, r is using to represent the number of files that contain the keyword w and p denote the number of cores.

From Table 2.1, we can see that SWP is the only schemes that does not build the index, all other scheme construct the index to speed up the searching process. Also in addition to SWP, other schemes can dynamically add or remove files, where dynamic means no need to re-index the entire data collection. There are Table 2.1 can be seen that SWP and Z-IDX achieve the same search efficiency for parallel time ($O(n)$ for sequential time). For KPR scheme is CKA2 secure and achieves optimal search efficiency. Compare the KP with KPR, the former has a smaller index size and eliminates information leakage during updates.

Chapter 3

Public-Key Cryptography Based SE Schemes

In this chapter, the public-key cryptography-based SE schemes will be illustrated and discussed.

3.1 Formulation

The origin of asymmetric searchable encryption (i.e., public key-based searchable encryption) techniques can be traced back to the untrustworthy server routing problem. *UserA* needs to send emails to *UserB* via the mail server. To ensure the privacy of emails, that the email content can be sent to *UserB* correctly under the premise that the mail server does not know the email content.

The authors in [6] first applied searchable cryptography to asymmetric cryptography and proposed the concept of public-key encryption with keyword search (PEKS). According to the definition state in the [6], the public-key encryption with keyword search scheme can be described as follows:

Definition: public-key encryption with keyword search (PEKS)

$$PEKS = (GenKey, Encrypt, Trapdoor, Test)$$

1. $(K_p, K_s) \leftarrow \mathit{GenKey}\,(\alpha)$: The algorithm uses a security parameter α as the input and outputs a public key K_p and a private key K_s.

2. $C_W \leftarrow \mathit{Encrypt}(K_p, W)$: Using the generated public key K_p and the keyword W of the encrypted file, the ciphertext C_W of the keyword W is generated.

3. $T_W \leftarrow \mathit{Trapdoor}\,(K_s, W)$: Input Private key K_s and keyword W,the algorithm outputs the trapdoor T_W of keyword W.

4. $D \leftarrow \mathit{Test}\,(K_p, T_W, C_W)$: Given a public key K_p, a searchable encryption $C_W = \mathit{Encrypt}(K_p, W)$ and a trapdoor T_W, the algorithm can output a decision value denote by D, where $D \in \{0, 1\}$.

3.2 INHJ Scheme

The authors in the [13] proposed a scheme called INHJ that provides a method for constructing public-key encryption with delegated search (PKEDS) which is based on the ElGamal public-key encryption (PKE) [10].

Details of Scheme. The authors propose in PKEDS to allow the server to search each portion of the encrypted data, instead of the former scheme in which only the metadata is searchable. In this mechanism, the server has the ability to verify whether the encrypted data contains a malware signature and does not need to decrypt it. In addition, the INHJ scheme allows two distinct kinds of trapdoors. The first one allows searching for keywords within the trapdoor and the second one supports searching for keywords directly by the server.

Efficiency.The construction of INHJ utilizes the prime order bilinear group. Since INHJ is built on PKE, the encryption efficiency of both is the same with each keyword requiring two exponentiations in the group $\mathbb{G}$ [13].

Security. Under the symmetric external Diffie-Hellman (SXDH) [3] assumption, the INHJ is proven that the scheme is ciphertext indistinguishable and trapdoor indistinguishable [13], which means that no one except the server can know anything about the plaintext keyword. But at the same time, since the server is allowed to retrieve all encrypted data, so the server can generate any trapdoor, which results in INHJ being weaker than PEKS in terms of security.

3.3 Research and Progress of PEKS

In 2004, *Boneh et al.* [13] first introduced the concept of public key searchable encryption and gave a PEKS construction scheme based on identity-based encryption (IBE). This scheme was applied to mail systems to solve the mail routing problem of untrustworthy servers. A year later, *Abdalla et al.* [1] analyzed the consistency problem of the PEKS and pointed out that the schemes in the [13] are computationally consistent. Also, they proposed a general conversion algorithm between the PEKS scheme to the IBE scheme.

In 2006, *Byun et al.* [8] found a serious security vulnerability in the current PEKS scheme: since the keyword space is much smaller than the key space, which makes it easy for an attacker to break the PEKS scheme through Off-Line Keyword Guessing Attacks (OKGAs).To defend against off-line keyword guessing attacks, *Rhee et al.* proposed a trapdoor-secure dPEKS (searchable public-key encryption with a designated tester) scheme in [17], which introduces the concept of trapdoor indistinguishability and strengthens the trapdoor security in the PEKS scheme.

In 2008, *Baek et al.*[2] pointed out that the scheme in the [13] needs to be built on top of a secure channel and therefore lacks practical value. To address the above problem, they proposed a PEKS scheme that does not require a secure channel, which introduces the concept of a designated tester and requires the server to have its own public-private key pair to ensure secure communication under a public channel.

A verifiable attribute-based keyword search (VABKS) scheme was proposed by *Zheng et al.* in 2014 [21], which outsources the search operation to the cloud server while verifying whether the cloud server performed the correct search operation, which achieves confidentiality and integrity of the search.

In 2016, *Liang et al* [16] proposed a PEKS scheme supporting regular language search, and the authors first gave the concept of S-DFA-FE (Searchable Deterministic Finite Automata-based Functional Encryption), and then provide a specific S-DFA-FE construction scheme. The scheme provides data integrity verification while supporting regular language search, and it does not require some special keywords from the data owner when constructing the keyword index structure.

Chapter 4

Conclusion

Searchable encryption is proposed to solve two basic problems of encryption: (i) the storage problem of untrustworthy servers; (ii) the routing problem of untrustworthy servers. The former has advanced the research progress of symmetric searchable encryption (SSE), while the latter drives the research progress of asymmetric searchable encryption (ASE), also known as public-key encryption with keyword search (PEKS). Both types of searchable encryption provide solutions for ciphertext search from untrusted servers and have been widely used in real-world applications, such as cloud storage ciphertext searching.

In this paper, we introduce searchable encryption technology, firstly, we briefly introduce the background knowledge of searchable encryption. Then we introduce different research categories of searchable encryption. We focus on the two main types of searchable encryption: symmetric searchable encryption (SSE) and public-key encryption with keyword search (PEKS). The state-of-the-art scheme in both types of searchable encryption is introduced and compared.

Bibliography

[1] Michel Abdalla et al. “Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions”. In: *Annual international cryptology conference.* Springer. 2005, pp. 205–222.

[2] Joonsang Baek, Reihaneh Safavi-Naini, and Willy Susilo. “Public Key Encryption with Keyword Search Revisited”. In: *Proceeding Sof the International Conference on Computational Science and Its Applications, Part I.* ICCSA ’08. Perugia, Italy: Springer-Verlag, 2008, pp. 1249–1259. ISBN: 9783540698388. DOI: 10.1007/978-3-540-69839-5_96. URL: https://doi.org/10.1007/978-3-540-69839-5_96.

[3] Lucas Ballard et al. “Correlation-Resistant Storage via Keyword-Searchable Encryption.” In: *IACR Cryptol. ePrint Arch.* 2005 (2005), p. 417.

[4] Burton H. Bloom. “Space/Time Trade-Offs in Hash Coding with Allowable Errors”. In: *Commun. ACM* 13.7 (July 1970), pp. 422–426. ISSN: 0001-0782. DOI: 10.1145/362686.362692. URL: https://doi.org/10.1145/362686.362692.

[5] Dan Boneh and Brent Waters. “Conjunctive, Subset, and Range Queries on Encrypted Data”. In: *Theory of Cryptography.* Ed. by Salil P. Vadhan. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 535–554. ISBN: 978-3-540-70936-7.

[6] Dan Boneh et al. “Public Key Encryption with Keyword Search”. In: *Advances in Cryptology - EUROCRYPT 2004.* Ed. by Christian Cachin and Jan L. Camenisch. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 506–522. ISBN: 978-3-540-24676-3.

[7] Christoph Bösch et al. “A survey of provably secure searchable encryption”. In: *ACM Computing Surveys (CSUR)* 47.2 (2014), pp. 1–51.

[8] Jin Wook Byun et al. “Off-line keyword guessing attacks on recent keyword search schemes over encrypted data”. In: *Workshop on secure data management*. Springer. 2006, pp. 75–83.

[9] Reza Curtmola et al. “Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions”. In: *Proceedings of the 13th ACM Conference on Computer and Communications Security*. CCS ’06. Alexandria, Virginia, USA: Association for Computing Machinery, 2006, pp. 79–88. ISBN: 1595935185. DOI: 10.1145/1180405.1180417. URL: https://doi.org/10.1145/1180405.1180417.

[10] T. Elgamal. “A public key cryptosystem and a signature scheme based on discrete logarithms”. In: *IEEE Transactions on Information Theory* 31.4 (1985), pp. 469–472. DOI: 10.1109/TIT.1985.1057074.

[11] Eu-Jin Goh. *Secure Indexes*. Cryptology ePrint Archive, Report 2003/216. http://eprint.iacr.org/2003/216/. 2003.

[12] Oded Goldreich and Rafail Ostrovsky. “Software Protection and Simulation on Oblivious RAMs”. In: *J. ACM* 43.3 (May 1996), pp. 431–473. ISSN: 0004-5411. DOI: 10.1145/233551.233553. URL: https://doi.org/10.1145/233551.233553.

[13] Luan Ibraimi et al. “Public-Key Encryption with Delegated Search”. In: *Proceedings of the 9th International Conference on Applied Cryptography and Network Security*. ACNS’11. Nerja, Spain: Springer-Verlag, 2011, pp. 532–549. ISBN: 9783642215537.

[14] Seny Kamara and Charalampos Papamanthou. “Parallel and dynamic searchable symmetric encryption”. In: *International conference on financial cryptography and data security*. Springer. 2013, pp. 258–274.

[15] Seny Kamara, Charalampos Papamanthou, and Tom Roeder. “Dynamic Searchable Symmetric Encryption”. In: *Proceedings of the 2012 ACM Conference on Computer and Communications Security*. CCS ’12. Raleigh, North Carolina, USA: Association for Computing Machinery, 2012, pp. 965–976. ISBN: 9781450316514. DOI: 10.1145/2382196.2382298. URL: https://doi.org/10.1145/2382196.2382298.

[16] Kaitai Liang et al. “Privacy-Preserving and Regular Language Search Over Encrypted Cloud Data”. In: *IEEE Transactions on Information Forensics and Security* 11.10 (2016), pp. 2365–2376. DOI: 10.1109/TIFS.2016.2581316.

[17] Hyun Sook Rhee et al. "Trapdoor security in a searchable public-key encryption scheme with a designated tester". In: *Journal of Systems and Software* 83.5 (2010), pp. 763–771. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2009.11.726. URL: https://www.sciencedirect.com/science/article/pii/S0164121209003100.

[18] C. E. Shannon. "Communication theory of secrecy systems". In: *The Bell System Technical Journal* 28.4 (1949), pp. 656–715. DOI: 10.1002/j.1538-7305.1949.tb00928.x.

[19] Dawn Xiaoding Song, D. Wagner, and A. Perrig. "Practical techniques for searches on encrypted data". In: *Proceeding 2000 IEEE Symposium on Security and Privacy. S P 2000.* 2000, pp. 44–55. DOI: 10.1109/SECPRI.2000.848445.

[20] Brent R. Waters et al. "Building an Encrypted and Searchable Audit Log". In: *In The 11th Annual Network and Distributed System Security Symposium.* 2004.

[21] Qingji Zheng, Shouhuai Xu, and Giuseppe Ateniese. "VABKS: Verifiable attribute-based keyword search over outsourced encrypted data". In: *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications.* 2014, pp. 522–530. DOI: 10.1109/INFOCOM.2014.6847976.