# An Efficient Fuzzy Searchable Encryption Scheme based Attribute for Medical Data

Feijiao Shao, Ruijuan Zheng

School of Henan University of Science and Technology, Luoyang 471023, China

## Abstract

**With the development of medical information technology, most of the patient medical data in unidentifiable form is stored in the cloud server to ensure secure storage of medical data. However, there are other problems such as the secure search of medical data. Search Encryption (SE) technology supports data ciphertext searchable. After that, many scholars have applied SE technology to solve the secure storage and secure search of medical data. When doctors search medical data, malicious users can obtain patient diagnosis and treatment information through doctor attribute information. This paper proposes a secure search encryption method for medical data. The method uses Symmetric Encryption algorithm with simple and fast, Ciphertext-Policy Attribute-Based Encryption (CP-ABE) algorithm to encrypt medical data. The hospital local server constructs a binary index tree according to alphabetical order of disease symptom keywords, and calculates the edit distance to return the correct disease symptom keywords when doctors misspell them. To achieve secure search of medical data ciphertext, the hospital local server constructs a disease symptom keyword search token through user attributes. The scheme also supports outsourced decryption for returned medical data ciphertexts. Finally, this scheme proofs that data security is selectively chosen plaintext attack secure (CPA-secure). The experimental results show that the search time of users searching medical data is less and the computational efficiency of users obtaining medical data is higher.**

## Keywords

**CP-ABE; Symmetric Encryption; Fuzzy Searchable; Outsourced Decryption.**

## 1. Introduction

With the deepening of medical informatization, medical institutions can produce a large amount of electronic medical data. It can save storage space and computing time, and also facilitate hospital management for hospital administrators in a cloud server. Medical data involving multiple professional disciplines is the main feature of patient medical records. After the integration of massive medical data, doctors can conduct more in-depth research and understanding of specialized diseases. Since the epidemic prevention and control, the application of medical data in cloud storage has become the mainstream application in terms of research on new coronary pneumonia, collection of patient medical data, and cross-department and cross-hospital collaboration. However, cloud server as a third-party server may face the risk of privacy leakage. Under this security risk, most medical institutions encrypt the collected medical data after integration which is securely stored in the cloud server in the form of ciphertext to ensure that a large amount of medical data. For encrypted medical data, doctors or researchers in other medical institutions cannot directly search the medical data ciphertext. When they need to search a specific medical data, they need to decrypt all medical data ciphertext in advance. In the face of a large number of medical data ciphertexts, this method to search for the required medical data will waste a lot of computing time. How to realize the safe and efficient search of medical data in the form of ciphertext is an urgent problem to be solved in the development of medical informatization.

One of the most common ciphertext searchable ways is Search Encryption (SE) technology, which selectively searches encrypted files based on keywords entered by data user rather than searching all encrypted files. First, the data owner encrypts data and saves data ciphertext in the cloud server. Only the data user holds the decryption key corresponding to the encryption key, the data ciphertext can be searched and decrypted securely. In order to prevent unauthorized users who have decryption keys from accessing medical data, most researchers use SE technology with access policies and privacy protection mechanisms. After the doctors attributes satisfy the access policy, the encrypted medical data can be accessed in this way. The doctor enters the relevant searchable keywords to search the ciphertext of medical data. Although SE technology can search securely and efficiently, most searches only support precise keyword searches. Considering the large range and strong professionallism of medical data after integration in this paper, there may be spelling errors and inaccurate content descriptions during search. It is necessary to implement fuzzy search for encrypted medical data files.

This paper proposes a scheme that can ensure data security in the search process. This scheme solves the problems of medical data access control and patient privacy security through Symmetric Encryption method and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) method. This scheme also reduces user search time by building a keyword binary index tree. It realizes fuzzy keywords through calculating edit distance between keywords in binary index tree and entered keywords. Finally, this research scheme decrypts some medical data files through the cloud server in the decryption stage, which greatly reduces the user decryption time.

## 2. Related Work

With the development of medical informatization, most medical institution administrators store the aggregated medical data in the cloud. This storage method can not only reduce the maintenance of medical data by medical institutions, but also reduce the computational overhead of medical data. In order to ensure the safety of medical data during storage, most medical institutions use encrypted security technology for medical data, and store medical data in the cloud in a form that users cannot identify at will. However, when searching and accessing encrypted medical data, most medical institutions first decrypt the ciphertext of the medical data. Although this method achieves the purpose of data search, there are problems of excessive computational consumption and leakage of privacy and security in the process of transmission and storage. How to accurately and safely search medical data from difficult-to-identify medical data is an increasingly concerned issue in the field of data search.

Song et al. [1] firstly proposed a Searchable Encryption (SE) scheme to deal with the untrusted mail server problem. He divides a file into multiple words or multiple sentences, and encrypts each word or sentence separately. When searching, each word or sentence needs to be decrypted and scanned. Since the ciphertext of all file sets that may contain words or sentences needs to be scanned when implementing the search, the search overhead of this method is relatively high. Later, in order to solve the problem of low computational efficiency, literature [2] proposed a secure indexing mechanism, which ensures the security of data by using pseudo-random functions, establishes an index in plaintext data, and compares the index with the keywords in the file. It is added to the Bloom filter (Bloom Filter, BF), and it is judged whether the BF contains search keyword information, so as to realize the fast search of the data ciphertext.

According to different encryption key types, SE schemes are divided into two types: Symmetric Searchable Encryption (SSE) and Asymmetric Searchable Encryption (ASE). Boneh et al. [3] proposed a public key encryption scheme (Pubic-Key Encryption with Keyword Search, PEKS) that can implement secure search for keywords by inputting search keywords. In this scheme, the data user encrypts the keyword with his own private key, generates a keyword trapdoor, and compares the keyword trapdoor with the keyword ciphertext in the file to obtain the corresponding password. text data, so as to realize the search of a single keyword. Based on the PEKS scheme in the literature [4, 5, 6], many researchers have made improvements according to different user needs to improve the efficiency of the algorithm and protect the privacy of users.

In the Internet environment, the realization of data sharing among multiple users is the current development status. ABE encryption scheme not only ensures the security protection of data, but also realizes data access control. In 2014, Li et al. [7] proposed an Attribute-Based Pubic-Key Encryption with Keyword Search (ATT-PEKS) scheme. The program is based on the ABE program for keyword search. Before searching the ciphertext, first determine whether the user attributes satisfy the defined access rights. The scheme can not only ensure fine-grained access control of users, but also efficiently search ciphertext data. Zheng et al. [8] proposed a verifiable ATT-PEKS scheme. This scheme outsources the computationally complex bilinear pairing search to the cloud server, and constructs the keyword BF to verify the correctness of the returned results. After Guo et al. [9] used the ATT-PEKS scheme in the medical and health field, used the CP-ABE scheme to encrypt patient medical data, and used the patient's identity ID as the search key to search for the ciphertext of the medical data. Zhou et al. [10] combined the CP-ABE scheme and the PEKS scheme. This scheme is the first to prove the security of keywords in the search process in a security-defining way. Due to the large number of people accessing medical data and many keywords being searched, Wu et al. [11] proposed a multi-keyword search scheme based on large attribute domain CP-ABE. By comparing the keyword set index with the search keyword set trapdoor, the scheme determines whether the user's search is correct, returns the search results and decrypts them. It efficiently realizes multi-keyword search for medical data by medical institution staff.

In order to meet the different needs of users, researchers have developed more types of keyword searches, such as wildcard keyword searches [12, 13, 14] and fuzzy keyword searches. Combining the idea based on fuzzy identity encryption, Sun Ting et al. constructed a fuzzy keyword search scheme based on CP-ABE. First, the data owner encrypts the keyword, and then the data user generates a trapdoor according to the search keyword, and judges the edit distance between the trapdoor keyword and the ciphertext. If the edit distance is small, the corresponding keyword ciphertext is returned and decrypted. However, there are many encryption keywords in this scheme, and the problem of low encryption and decryption efficiency will occur. Liu et al. [15] designed a secure search scheme for medical data. The scheme realizes data security protection by using symmetric key algorithm and CP-ABE algorithm, and realizes fuzzy keyword search through binary tree. However, the CP-ABE algorithm of this scheme uses disease symptoms to form a binary tree as an access structure, and the flexibility of access control is low. However, in the face of medical data, if the wildcard method is used to realize the fuzzy keyword search of medical data ciphertext, there may be a problem of large storage consumption.

# 3. System Model

## 3.1 Model Entity Description

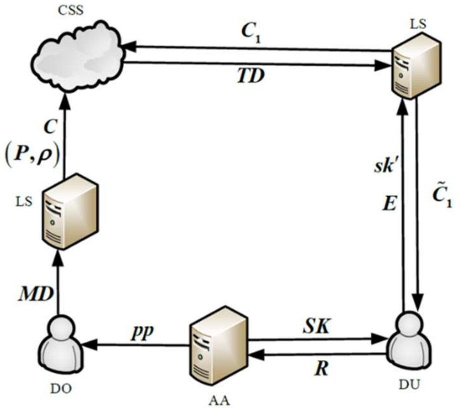

**Fig. 1** System Model

In the hospital information system, the system model is shown in Fig. 1. The scheme model of the paper mainly includes five entities whose main tasks are as follows.

Data Owner (DO): DO generally refers to a doctor, nurse, or hospital server administrator. The main task of DO is to organize the medical records of patients with the same disease.

Data User (DU): DU is primarily doctor or researcher who accesses medical data. If DU is an authorized user, secure search and decryption of medical data ciphertext is possible. Otherwise, DU cannot perform access operations.

Cloud Storage Server (CSS): In the hospital information system, CSS is an untrusted entity. The main task of CSS is to store medical data ciphertext and disease symptom keywords ciphertext, as well as perform medical data ciphertext search.

Attribute Authority (AA): The main task of AA is to generate and distribute secret keys according to the key distribution protocol discussed by the hospital and AA.

Local Server (LS): It is a trusted server in the hospital. LS is mainly used for keyword matching, generating keyword trapdoor, as well as decrypting, encrypting partial ciphertext.

### 3.2 Model Definition

In the model, the fuzzy search of disease symptom keywords based on attribute ciphertext is realized, which is mainly composed of six algorithms. The detailed description of six algorithms is shown as below.

(1) Setup Phase

$Setup(1^{\kappa}) \rightarrow \{pp, msk\}$. In the phase, AA first sets a security parameter $\kappa$. AA can obtain the public parameter $pp$ and master secret key $msk$ through calculation.

(2) Key Generation Phase

$KeyGen(pp, msk, R) \rightarrow \{SK\}$. According to input $pp$ and $msk$, AA generates the key $SK$ related to the user attribute $R$ by running the algorithm.

(3) Encrypt Phase

$Encrypt(pp, MR, (P, \rho)) \rightarrow \{ct\}$. In the encryption algorithm, DO inputs $pp$, medical data $MR$ and access policy $(P, \rho)$. Finally, the ciphertext $ct$ which is related to medical data is output.

$Enc\_keyword(pp, F, (P, \rho)) \rightarrow \{CK\}$. In this algorithm, DO inputs $pp$, the keyword $F$ in the medical data, and the access policy $(P, \rho)$. Finally it outputs the corresponding keyword ciphertext $CK$.

(4) Tokendoor Generation Phase

$Tokendoor(pp, F', SK) \rightarrow \{T\}$. LS corresponding to DU runs the algorithm to output the keyword search token $T$ according to $pp$, $SK$ and the search keyword set $F'$. It also sends $T$ to CSS.

(5) Search Phase

$Search(CK, T) \rightarrow \{ct\}$. CSS judges whether the user attribute $R$ conforms to the access policy in ciphertext according to $T$ and $CK$. CSS outputs ciphertext $ct$ corresponding to the search keyword $D'$ in $T$.

(6) Decrypt Phase

$Decrypt(pp, SK, ct) \rightarrow \{MR\}$. In the model, the following decryption algorithm is run according to $pp$, $SK$ and $ct$.

- $GenTK_{out}(SK) \rightarrow \{sk', osk\}$. LS corresponding to DU generates the outsourced decryption key $osk$ and the user transformation key $sk'$ according to $SK$.

- $Transform_{out}(sk',ct) \to \{\widetilde{ct}\}$ . LS corresponding to DU runs the outsourced transformation algorithm. It inputs $sk'$ and part of the medical data ciphertext $ct$ , and finally generates an intermediate value $\widetilde{ct}$ .

- $Decrypt_{out}(\widetilde{ct},osk,ct) \to \{MR\}$ . After DU receives the intermediate values $\widetilde{ct}$ , $osk$ and $ct$ , it decrypts and generates medical data $MR$ .

### 3.3 Security Model Definition

The paper defines a security game between challenger and adversary. The game is shown as below.

- Init: In this phase, the adversary $\mathcal{Y}$ generates a challenge access structure $(P^*,\rho^*)$ and sends it to the challenger $\mathcal{R}$ .

- Setup: Firstly, $\mathcal{R}$ runs $Setup$ phase query and generates a pair secret key $(pp,msk)$ . $\mathcal{R}$ can send $pp$ to $\mathcal{Y}$ .

- Phase 1: $\mathcal{R}$ sets up an empty set $S$ and empty table $B=(R,sk,sk',osk)$ . $\mathcal{Y}$ adds user attribute $R$ to $S$ and sends $R$ to $B$ , but $R$ does not satisfy the initial challenge access policy $(P^*,\rho^*)$ . $\mathcal{R}$ first scans $B=(R,sk,sk',osk)$ to determine whether there is $sk$ and $sk'$ corresponding to $R$ . If they exist, $\mathcal{R}$ returns $sk$ and $sk'$ to $\mathcal{Y}$ . Otherwise, $\mathcal{R}$ executes $KeyGen$ phase query to generate $sk$ , and $GenTK_{out}$ phase query generates $sk'$ and $osk$ . $\mathcal{R}$ stores $sk$ , $sk'$ and $osk$ in table $B$ and sends $B$ to $\mathcal{Y}$ .

We assume that $\mathcal{Y}$ does not issue a transform key query against the same set of attributes when user issues a key query with a set of user attributes.

- Challenge: $\mathcal{Y}$ sends two equal-length challenge medical data $MR_0$ and $MR_1$ to $\mathcal{R}$ . $\mathcal{R}$ randomly selects a value $b \in \{0,1\}$ and encrypts $MR_b$ to obtain the challenge ciphertext $ct^*$ . $\mathcal{R}$ extracts the disease symptom keyword in the medical data, encrypts the keyword to generate a related challenge keyword ciphertext $CK^*$ , and sends the challenge ciphertext $C^*=(ct^*,CK^*)$ to $\mathcal{Y}$ .

- Phase 2: The query in the phase is same as the query in phase 1. However, the user attribute of the phase is different from that of phase 1, and the user attribute key $sk$ obtained by $\mathcal{Y}$ is different from that of phase 1. In addition, the user attributes at the phase also does not satisfy the challenge access policy $(P^*,\rho^*)$ .

- Guess: $\mathcal{Y}$ outputs a guess $b' \in \{0,1\}$ . If $b'=b$ , it knows that $\mathcal{Y}$ wins the game.

In the end, if the advantage of polynomial time adversary $\mathcal{Y}$ to win the game is $|Pr[b'=b]-\frac{1}{2}|$, the proposed system model is Chosen Plaintext Attack Secure (CPA).

## 4. Construction

(1) Setup Phase

$Setup(1^\kappa) \to \{pp,msk\}$ . Firstly, AA inputs the security parameters $\kappa$ into the group generator $\mathcal{G}$ . It generates public key $pk=(p,G,G_1,e)$ , where $G$ and $G_1$ are cyclic group with a prime number $p$ , $e$ is a bilinear map. AA randomly select $g,m,l,h,n,\beta \in G$ and $d \in Z_p$ . And $H:\{0,1\} \to Z_p$ is a hash function. Then the algorithm generates some public parameters and master secret key as follow:

$$pp=\{g,m,l,h,n,H,pk,g^\beta,e(g,g)^d\},msk=\{d,\beta\}$$

At this phase, each LS also creates a binary index tree of disease symptom keywords in which the disease symptom keywords are arranged in alphabetical order.

(2) Key Generation Phase

$KeyGen(pp, msk, R) \to \{SK\}$. In the key generation phase, AA generates user attribute keys through key distribution protocol. AA randomly select $a, a_1, a_2, \ldots, a_x \in Z_p$ and user attribute $R_x$, where $x$ represents the number of user attributes $R$. Then the algorithm generates user attribute secret key $sk = \{k, k_0, k_{1,x}, k_{2,x}, k_{3,x}\}$ used to decrypt data ciphertext as follow:

$$k = g^d m^a, k_0 = g^a, k_{1,x} = g^{a_x/\beta}, k_{2,x} = (l^{R_x} h)^{a_x/\beta} n^a, k_{3,x} = (l^{R_x} h)^{a_x} (n^a)^\beta .$$

In addition, AA also generates secret key $sk_t = \{tk_1, tk_2\}$ used to generate search keyword token as below:

$$tk_1 = g^\beta, tk_2 = n^\beta .$$

Finally, AA sends $SK = \{sk, sk_t\}$ to DU.

(3) Encrypt Phase

DO performs encryption operations for medical data $MR$ before medical data is stored in CSS and encrypts the disease symptom keywords extract from the medical data. The specific operations of the encryption phase are as follows:

- $Encrypt(pp, MR, (P, \rho)) \to \{ct\}$. The local server manager LS of medical institution firstly encrypts $MR$ with a symmetric key $sk_s$, and calculates the data ciphertext $C_1 = Enc_{sk_s}(MR)$. Next, LS formulates an access policy $(P, \rho)$ consisting of an access matrix $P$ of $b \times f$ and a mapping $\rho$ from row $P_i$ in matrix $P$ to attribute $\rho(i)$. LS randomly picks $\vec{s} = (t, s_2, s_3, \ldots, s_i)^\top$ and $v_i, w_i \in Z_p$, where $i \in [1, b]$. $\vec{s}$ can be calculated $\lambda_i = P_i \vec{s}$. LS can obtain the medical data ciphertext $CT_2 = \{ct_0, ct_1, (ct_{0,i}, ct_{1,i}, ct_{2,i}, ct_{3,i})_{i \in [1,b]}\}$ as follows:

$$ct_0 = sk_s e(g, g)^{dt}, ct_1 = g^t, ct_{0,i} = (g^{w_i})^\beta g^{v_i}, ct_{1,i} = g^{w_i}, ct_{2,i} = m^{\lambda_i} n^{v_i}, ct_{3,i} = (l^{\rho(i)} h)^{v_i} .$$

LS stores generated medical data ciphertext $ct = \{CT_1, CT_2\}$.

- $Enc\_keyword(pp, F, (P, \rho)) \to \{CK\}$. LS extracts disease symptom keyword set $F = (f_1, f_2, \ldots, f_m)$ from medical data $MR$. It randomly picks value $\theta \in Z_p$, and uses $(P, \rho)$ to encrypt $F$. Its ciphertext is $CK = \{ck_0, ck_1, (ck_{1,i}, ck_{2,i}, ck_{3,i})_{i \in [1,b]}\}$ and detail description as follows:

$$ck_0 = g^{H(F)}, ck_1 = g^\theta, ck_{1,i} = g^{v_i}, ck_{2,i} = g^{\lambda_i/\beta} n^{v_i}, ck_{3,i} = (l^{\rho(i)} h)^{-a_i \theta} .$$

Finally, LS stores generated medical data ciphertext and disease symptom keyword ciphertext $C = \{ct, CK\}$ to CSS.

(4) Tokendoor Generation Phase

$Tokendoor(pp, F', sk_t) \rightarrow \{T\}$. DU inputs the search keyword set $F' = (f_1', \ldots, f_m')$, and sends it to LS. LS calculates the edit distance $d$ between the search key and the binary index tree key. If the edit distance $d \le k$ ($k$ set by the medical institution administrator), LS returns the corresponding fuzzy keyword set. For any fuzzy search keyword , LS randomly selects a value $d \in Z_p$ for any fuzzy search keyword $f_x', x \in [1, m]$. LS generates keyword search token $T = \{t_0, t_1, t_2, t_3, (t_{1,i})_{i\in[1,b]}\}$ and detail descripts as follows:

$$t_0 = g^{H(F')} e(g,g)^{dt}, t_1 = g, t_2 = (tk_1)^d = g^{d\beta}, t_3 = (tk_2)^d = n^{d\beta}, t_{1,i} = (n^{\rho(i)} h)^{r_i}.$$

(5) Search Phase

$Search(CK, T) \rightarrow \{ct\}$. After receiving the keyword search token $T$ sent by the DU, CSS first determines whether the user is an authorized user. If DU is an authorized user, CSS calculates the required ciphertext according to the search token $T$ and the keyword ciphertext $CK$. If DU is an unauthorized user, CSS does nothing. If:

$$\prod_{i\in[1,b],x\in[1,m]} \left(\frac{t_0 e(ck_{1,i}, t_3)}{e(ck_{2,i}, t_2) e(ck_1, t_{1,i}) e(ck_{3,i}, t_1)}\right)^{\varphi_i} = g^{H(F')}$$

holds, CSS returns ciphertext $ct$ related to disease symptom keyword.

(6) Decrypt Phase

$Decrypt(pp, sk, ct) \rightarrow \{MR\}$. DU needs to decrypt the symmetric key $sk_s$ firstly. Then it uses the symmetric key $sk_s$ to decrypt the medical data. The decryption algorithm consists of three steps:

• $GenTK_{out}(SK) \rightarrow \{sk', osk\}$. DU executes the algorithm to generate transformation key pair $(sk', osk)$. It calculates transformation key $sk' = \{sk_0', (sk_{1,x}', sk_{2,x}', sk_{3,x}')_{x\in[1,|R|]}\}$ and outsourced decryption key $osk = \{r\}$, where:

$$sk_0' = k_0^{1/r}, sk_{1,x}' = k_{1,x}^{1/r}, sk_{2,x}' = k_{2,x}^{1/r}, sk_{3,x}' = k_{3,x}^{1/r}.$$

The transformation key $sk'$ sends to CSS and the outsourced decryption key $osk$ stores to DU.

• $Transform_{out}(sk', CT_1) \rightarrow \{\widetilde{CT_1}\}$. It randomly picks constant $\varphi_i \in Z_p$. It runs:

$$\widetilde{CT_1} = \prod_{i\in[1,u]} \left(\frac{e(ct_{0,i}, sk_{2,\rho(i)}')}{e(ct_{1,i}, sk_{3,\rho(i)}') e(ct_{2,i}, sk_0') e(ct_{3,i}, sk_{1,\rho(i)}')}\right)^{\varphi_i} = e(m,g)^{\frac{-rs}{r}}$$

and finally generates an intermediate value $\widetilde{CT_1}$. Finally, It sends obtained intermediate value $\widetilde{CT_1}$ to LS where DU is located.

• $Decrypt_{out}(\widetilde{CT_1}, osk, ct) \rightarrow \{MR\}$. It inputs the outsourced decryption key $osk$, $\widetilde{CT_1}$ and $ct$ on the decryptor of LS where DU is located. Finally, it obtains $sk_s$ by calculating:

$$sk_s = \frac{ct_0}{e(ct_1, k)\widetilde{CT_1}^{osk}}.$$

DU obtains symmetric key $sk_s$ to decrypt the medical data $MR = Dec_{sk_s}(CT_1)$.

## 5. Security Analysis

(1) Data Security

The scheme in this paper first uses symmetric encryption to encrypt medical data to improve the efficiency of data encryption, and then uses CP-ABE scheme to encrypt symmetric keys to achieve fine-grained access control. In the CP-ABE scheme, the symmetric key can be decrypted only when the user attributes satisfy ciphertext access policy. Moreover, the medical data is stored in the form of ciphertext in all transmission processes, which ensures the confidential transmission of data. In addition, DU stores the user attribute key in this paper. When the DU sends a decryption request to CSS, AA generates an outsourced decryption key and sends it to outsourced cloud server, and then uses the outsourced decryption key to decrypt the ciphertext to obtain an intermediate value unrelated to ciphertext.

(2) Keyword Security

In order to improve the accuracy of disease symptom keyword search, the fuzzy keyword scheme in this paper constructs a binary index tree by alphabetizing disease symptom keywords. DU returns a more accurate medical data set by inputting disease symptom keywords and matching the keywords with the binary index tree of disease symptoms. In addition, according to the security game, it can be known that the adversary initiates the key query and searches for the token key query. The final adversary needs to distinguish between $CK_0^* = g^{H(F_0)}$ and $CK_1^* = g^{H(F_1)}$, that is $H(F_0)$ and $H(F_1)$. Since a negligible collision may occur, the collision probability is $1/2^{\kappa} \le 1/2$. Thus in the absence of a conflict, the adversary wins a safe game with an opposite probability. In this research scheme, it is proved that the adversary arbitrarily selects keywords to search, and its security is Indistinguishability Chosen Keyword Attack secure (IND-CKA).

## 6. Conclusion

In the paper, we introduce a data security protection method for keyword fuzzy search of medical disease symptoms. In this method, hospital staff who uploads data and encrypts medical data using symmetric encryption method and CP-ABE method to ensure secure storage of medical data. Each hospital local server constructs a library of disease symptom keywords, which are inserted into a binary index tree with alphabetical order to achieve fast search of disease symptom keywords. When hospital staff enter a disease symptom, hospital local server returns the disease symptom with a small edit distance from the binomial index tree. The returned disease symptoms are used to construct search tokens for secure search of disease symptom keywords. In addition, returned medical data ciphertext is decrypted by outsourced to reduce computation consumption. Compared with Liu scheme, it not only achieves fuzzy search of medical data, but also ensures the efficiency of encryption and decryption for medical data.

## References

[1] D. Song, D. Wagner,and A. Perrig. Practical techniques for searches on encrypted data. Proceedings of IEEE Symposium on Security and Prioacy, pp. 44-45 , 2000.

[2] E. J. Goh. Secure Indexes. IACR Cryptology ePrint Archive, pp. 216 , 2003.

[3] D. Boneh, G.D.Crescenzo, and R.Ostrov et al. Public key encryption with key word search. International conference on the theory and applications of cryptographic techniques. Springer, Berlin, Heidelberg, pp. 506-522 , 2004.

[4] Y. Miao, J. Ma, and X. Liu et al. Light weight fine-grained search over encrypted datain fog computing. IEEE Transactions on ServicesComputing, vol. 12, no. 5, pp. 772-785 , 2019.

[5] J. Wang, X. Yu, and M. Zhao, “Fault-tolerant verifiable keyword symmetric searchable encryption in hybrid cloud.” Int. J. Netw. Secur., vol. 17, no. 4, pp. 471–483, 2015.

[6] L. Zou, X. Wang, and S. Yin, “A data sorting and searching scheme based on distributed asymmetric searchable encryption.” Int. J. Netw. Secur., vol. 20, no. 3, pp. 502–508, 2018.

[7] S. Li and M. Xu, “Attribute attribute-based public encryption with keyword search,” Chinese Journal of Computers, vol. 37, no. 5, pp. 1017–1024, 2014.

[8] Q. Zheng, S. Xu, and G. Ateniese, “Vabks: verifiable attribute-based keyword search over outsourced encrypted data,” in IEEE INFOCOM 2014-IEEE conference on computer communications, pp. 522–530, 2014.

[9] C. Guo, R. Zhuang, Y. Jie, Y. Ren, T. Wu, and K.-K. R. Choo, “Fine-grained database field search using attribute-based encryption for e-healthcare clouds,” Journal of medical systems, vol. 40, no. 11, pp. 1–8, 2016.

[10] Y. Zhou, S. Zheng, and L. Wang, “Privacy-preserving and efficient public key encryption with keyword search based on cp-abe in cloud,” Cryptography, vol. 4, no. 4, p. 28, 2020.

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

[12] S. Sedghi, P. v. Liesdonk, S. Nikova, P. Hartel, and W. Jonker, “Searching keywords with wildcards on encrypted data,” in International Conference on Security and Cryptography for Networks, pp. 138–153, 2010.

[13] C. Hu and L. Han, “Efficient wildcard search over encrypted data,” International Journal of Information Security, vol. 15, no. 5, pp. 539–547, 2016.

[14] Y. Yang, X. Liu, R. H. Deng, and J. Weng, “Flexible wildcard searchable encryption system,” IEEE Transactions on Services Computing, vol. 13, no. 3, pp. 464–477, 2017.

[15] Z. Liu, J. Weng, J. Li, J. Yang, C. Fu, and C. Jia, “Cloud-based electronic health record system supporting fuzzy keyword search,” Soft Computing, vol. 20, no. 8, pp. 3243–3255, 2016.