Noname manuscript No.
(will be inserted by the editor)

Fuzzy matching and without Central Authority: Multi-authority Attribute Searchable Encryption

Binrui Zhu · Jiameng Sun · Jing Qin[*] Jixin Ma

Received: date / Accepted: date

Abstract Attribute-based Keyword Search (ABKS) support the access control on the search result based upon fuzzy identity over encrypted data, when the search operation is performed over outsourced encrypted data in cloud. However, almost ABKS schemes trust a single authority to monitor the attribute key for users. In practice, we usually have different entities responsible for monitoring different attribute keys to a user. Thus, it is not realistic to trust a single authority to monitor all attributes keys for ABKS scheme in practical situation. Although a large body of ABKS schemes have been proposed, few works have been done on multi-authority attribute searchable encryption. We propose a multi-authority attribute searchable encryption without central authority in this paper. Comparing previous ABKS schemes, we extend the single authority ABKS scheme to multi-authority ABKS scheme and remove the central authority in multi-authority ABKS scheme. We analyze our scheme in terms of security and efficiency.

Keywords multi-authority, without central, searchable encryption, attribute-based encryption, fuzzy identity

1 Introduction

Cloud computing has been widely recognized as the next big thing in this era. Outsourcing data to cloud servers, while providing service economic savings and various convenience for users. Cloud server may be honest but curious, in order to ensure the security, data is usually stored as encrypted form in the cloud. At the same time, it also brings a new question that how users get encrypted data without decrypt of them. Searchable encryption is a primitive, which enables data users to search over the encrypted data. Both keywords privacy and data privacy are protected in this procedure.

Recently, Li and Zhang (2014) proposed an ABKS scheme in which a data owner can control the search result and outsource encrypted data according to the access control policy. Attribute-based keyword search scheme can be generated based on attribute-based encryption(ABE) scheme. Sahai and Waters (2005) proposed a fuzzy identity encryption scheme in which the sender can encrypt the message by the specified attribute set (user's fuzzy identity) with the threshold value d, such that only the authorized person with at least d of the given attribute can decrypt the message. In order to get the attribute key corresponding to each attribute, authorized persons must go to the trustworthy attribute authority to prove that they have these attributes. This means we must trust a single attribute authority to issue the authorized persons attribute

This work is supported by the National Nature Science Foundation of China under Grant No:61272091.

B. Zhu, J. Sun
School of Mathematics, Shandong University, Jinan, Shandong, China
E-mail: zhubinrui1509889@163.com,
sunjia-meng1991@163.com

J. Qin(Corresponding author)
School of Mathematics, Shandong University, Jinan, Shandong, China
State Key Laboratory of Information Security Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
E-mail: qinjing@sdu.edu.cn

J. Ma
School of Computing and Mathematical Sciences, University of Greenwich, London,UK
E-mail: j.ma@greenwich.ac.uk

keys all of records such as ID number, drive license number, and student number etc. But in practice, we usually have many attribute authorities responsible for monitoring different attribute keys of a person (ID number is managed by the Public Security Bureau, student number is managed by the School Office and driving license number is managed by Department of Motor Vehicles. For this reason, many multi-authority ABE schemes (Lin et al (2008); Chase and Chow (2009); Chow (2010); Xu et al (2016))are proposed.

In an ABKS scheme, ciphertexts and authorized persons attribute keys are labeled with sets of descriptive attributes. Authorized persons attribute keys are distributed by attribute authority. Similar with ABE, ABKS scheme also trust a single attribute authority to issue the authorized persons attribute keys. Obviously, single attribute authority is not practical in real life and increase the burden of attribute authority work.

Considering a scenario: Patients upload the encrypted Personal Health Record (PHR) to the cloud server. It allows efficient sharing of medical information among researchers while we should keep the data security. Meanwhile, it must be achieved fine grained access control of data and support the search operation. For the attribute set{Medical Association Membership, Chief Physician, Medical Researcher, Police}, there are four different entities, charged by the {Medical Association, Hospital, Scientific Research Institution and Public Security Bureau} respectively. It is difficult to solve it by single authority. How to design a searchable encryption scheme to guarantee the confidentiality of PHR data and allow authorized person to search encrypted data in multi-authority environment is a challenging problem.

1.1 Related works

Song et al (2000) proposed the first keyword search on ciphertext with symmetric encryption methold. It can only support single keyword search and search requires linearly scan each file document word by word. The most important thing is that it is not fully secure and only supports user-sever-user model. After this paper, many searchable encryption schemes(Goh (2003); Curtmola et al (2006); Chang and Mitzenmacher (2005)) focusing on this model based on symmetric encryption. Symmetric searchable encryption schemes only supports user-sever-user model and unsuitable for three party situation, which is unsuitable in the cloud environment. Boneh et al (2004) solved this problem and proposed the first public key encryption keyword search(PEKS). Their scheme provides a solution for the third party user to search on the encrypted data. However, Boneh's scheme requires a secure channel and can not achieve indistinguish of trapdoor. Following Boneh's work, Baek et al (2008) designed a PEKS scheme without secure channel by adding key pairs for the cloud server. Park et al (2004) proposed a conjunctive keyword search scheme based on PEKS scheme and solved the shortcomings of single keyword and increase the scheme practicality. But it can not ensure the security of trappdoor indistinguishability and keyword guess attack. Abdalla al (2008) proposed a new definition consistency of keyword search in ciphertext and designed a new PEKS scheme from identity based encryption. Golle et al (2004) proposed the conjunctive keyword searchable encryption. Although Golle's scheme belonged to the symmetric searchable encryption, it also supports multi-keyword search.

Sahai and Waters (2005) proposed a transformation from IBE to ABE. Then following Sahai's researcher, Goyal et al (2006) proposed a key policy attribute based encryption(KP-ABE) scheme which supports any monotonic access formula. ABE has two forms of encryption, one called KP-ABE, another called cipher policy attribute based encryption(CP-ABE). KP-ABE is that the key decides the access control policy, while ciphertext and attribute are associated. CP-ABE is that the ciphertext decides the access control policy, while key and attribute are associated. Bethencourt et al (2007) proposed the first CP-ABE scheme, which supports tree-based access structure. Recently, many attribute based encryption schemes(Li et al (2017); Ma et al (2016); Wang et al (2016, 2017)) have been proposed, but they only support single authority. Chase and Chow (2009) made extension to Sahai and Waters scheme from another view. They proposed a multi-authority ABE scheme that achieves the practical requirements. Following the Chase's work Chase and Chow (2009), many researchers focus on multi-authority ABE schemes(Xu et al (2016); Zhong et al (2016))that satisfy the practical requirements. For resisting collusion attack between attribute authorities. Lin et al (2008) proposed the multi-authority ABE scheme without the central authority, but it can only achieves m resiliance. Chow (2010) proposed a new privacy-preserving architecture for multi authority ABE without a central authority.

Searchable encryption gives a method to securely searching operation on encrypted data in cloud environment. When the Data owner uploads the encrypted data and shares the data privately to the third party user, it should first know the third party users identity in order to encrypt data with the corresponding encryption key. Han et al (2014) proposed an ABKS scheme that ensures attribute privacy and supports multi-user situation. Recently, many attribute based encryption with keyword search schemes(Shi et al (2014); Li and Zhang (2014); Liu et al (2014); Zheng et al (2014); Wang et al (2013); Koo et al (2013)) have been proposed, but they only support single keyword search and these schemes also leak the receivers identity. Liu et al (2016); Xhafa et al (2014) proposed a single-authority ABE with fuzzy keyword search scheme respectively. For searching operation over ABE, it is not realistic to trust a single authority to monitor all attributes keys in practical situation. In multi-authority ABE scheme, the central authority can assign a portion of the decryption key according to the user's global identifier. Once the central authority is broken, the security does not exist. So, how to design the multi-authority ABKS scheme without a trusted central authority is a challenging problem.

1.2 Our contribution

We propose a model of multi-authority attribute searchable encryption scheme in which a data owner can encrypt keywords specifying an attribute set, such that only an authorized person who has adequate attribute keys from authorities can search and decrypt the message in cloud environment. In our multi-authority attribute searchable encryption scheme, we assume that attributes can be divided into n disjoint sets. Each set will be mastered by a different authority. The main idea of our scheme is to find a way to extend the single authority ABKS scheme to multi- authority ABKS scheme. We make the following contributions over existing research:

A general transformation from multi-authority ABE to multi-authority ABKS is proposed. We also give a concrete multi-authority ABKS without central authority scheme based on the multi-authority ABE.

The scheme supports the access control based upon fuzzy identity over the search result and provides a multi-authority attribute searchable encryption with multi-owner/multi-user architecture, achieves the security of user anonymity, indistinguishes of keywords and trapdoor, keyword guessing attack.

1.3 Organizations

The remaining parts of the paper are organized as follows: Some necessary preliminaries are provided for the proposed schemes in section 2. We analyze the relationship between multi-authority ABE and multi-authority ABKS schemes in section 3. The multi-ABKS scheme is proposed and analyzed in section 4. The security and performance comparison are performed and analyzed in section 5. Section 6 concludes this paper.

2 Preliminaries

In this section, we introduce the formal definition of access tree and access structure. Then we give the description of pseudorandom functions, anonymous key issuing, the bilinear map and complexity assumptions.

2.1 Access control structure

We use the access structure of the tree Υ. Each non-leaf node of the access structure tree can be described by its children node and a threshold value. n_x denotes the number of children node x and k_x represents the threshold value, then $0 < k_x \leq n_x$. The tree leaf node associated with a attribute value $att(x)$ and $k_x = 1$.

In order to simply the operation of the access tree. We define some functions about access tree. Function *parent(x)* denote the node x of parent node. We build an children node index for every node and children node index number from *1* to n_x. Another function *index(x)* return node x number. r denote root of the access tree and Υ_x denote subtree root at the node x. Hence Υ can be denote as Υ_r. If $\Upsilon_r(\gamma) = 1$, it is denoted a set of attributes satisfy the access tree. We take recursively way to compute $\Upsilon_x(\gamma)$, if x is a non-leaf node, evaluate all children nodes values $\Upsilon_{x'}(\gamma)$, returns 1 if and only if at least k_x children node return 1. If x is a leaf node, then $\Upsilon_x(\gamma)$ returns 1 if and only if $att(x) \in \Upsilon$.

2.2 Pseudorandom functions

We use techniques for distributed pseudorandom functions (PRF) similar in Chase and Chow (2009). The main purpose is to use the PRF to make the key user-specific (otherwise, user can share their keys from the authority to their friend). So the work require that every user have a unique global identifier and prove that it is the owner of the global identifier. But the user presents the same global identifier to the authority, it is easy to build a complete file about global identifier for all authorities. Alternatively, we can interact with a server via a pseudorandom function and obtain attributes keys without revealing one's full global identifier. In Chase and Chow (2009), they use the anonymous key issuing to protect the user privacy. In this anonymous credential system, the user can get and prove the credential while remaining anonymous.

2.3 Bilinear map and complexity assumptions

We give formal definitions on bilinear map and our complexity assumptions. Let G_1, G_2 and G_T are three cyclic multiplicative groups of prime order p. A bilinear map $e : G_1 \times G_2 \rightarrow G_T$.

which satisfies:

1. Bilinear: For any $x, y \in Z_p$, $h \in G_1$, $j \in G_2$, $e(h^x, j^y) = e(h, j)^{xy}$.

2. Non-degenerate:exist $h \in G_1$, $j \in G_2$, $e(h, j) \neq 1$.

Assumption 1 (HDH). The Hash Diffile-Hellman problem: given the four tuple $(g, g^x, g^y, H(g^z)$ and hash function H, $x, y, z \in Z_p$, $G =< g >$, decide $z = xy(modp)$ whether or not.

Assumption 2 (DBDH). The Decisional Bilinear Diffile-Hellman problem: given the five tuple $(g_1, g_2^x, g_2^y, g_2^z, Z)$, $x, y, z \in Z_p$, $Z \in G_T$, $G_1 =< g_1 >$, $G_2 =< g_2 >$, decide $Z = e(g_1, g_2)^{xyz}$ whether or not.

Assumption 3 (BDH). The Bilinear Diffile-Hellman problem: given the four tuple $(g_1, g_2^x, g_2^y, g_2^z)$, $x, y, z \in Z_p$, $G_1 =< g_1 >$, $G_2 =< g_2 >$, compute the value $Z = e(g_1, g_2)^{xyz} \in G_T$.

Assumption 4 (XDH). The external Diffle-Hellman assumption: the DDH problem is hard in G_1 and isomorphism $\Phi' : G_1 \rightarrow G_2$ is not exist.

Assumption 5 (q-DDHI). The q-Decisional Diffile-Hellman Inversion(q-DDHI) problem: given the $q+2$ tuple $(g, g^x, g^{x^2} .....g^{x^q}, g^y)$, $G =< g >$, decide $y = 1/x$ whether or not.

In this paper, we construct a multi-authority ABKS scheme based on these assumptions.

3 Multi-authority ABE and Multi-authority ABKS

3.1 Multi-authority ABE Scheme Definition

Now, we will discuss the multi-authority ABE and give a general transformation from multi-authority ABE to multi-authority ABKS. A general multi-authority ABE scheme include four algorithms:

- **Setup**$(\lambda, N) \rightarrow (pa, (PK_i, SK_i)_{\{i=1...N\}})$: This algorithm inputs the number of authorities N and a security parameter λ, the system outputs a public parameter pa and N key pairs $(PK_i, SK_i)_{\{i=1...N\}}$, every key pairs for one attribute authority.
- **Enc**$(M, pa, A_{i\{i=1...N\}}) \rightarrow CT$: The Data sender inputs the set of attributes A_i, the message M and the the public parameter pa, computes the ciphertext CT, Where A_i represents a subset of attributes controlled by the attribute authority i.
- **KeyGen**$(GID, A_i, SK_{i\{i=1...N\}}, \Upsilon) \rightarrow msk_i$: This algorithm inputs the user's identity GID, the user attributes set A_i, the authority secret key SK_i and the access structure Υ, outs a decryption key msk_i corresponding the access structure.
- **Dec**$(CT, pa, msk_{i\{i=1...N\}}) \rightarrow M$: This algorithm inputs the ciphertext CT, the system public parameter pa and the user decryption key msk_i. Outputs the message M, if attributes satisfies the access structure tree.

3.2 Security Model

About the security in N-authority ABE scheme, we can define against compromising at most n authorities by the experiment as follows:

$Exp_{N-ABE}^{N-ABE-saa}(\lambda)$
$\mathcal{A} \rightarrow (I_{corr} \subset [1, N], A^C = \{A_1^C, ...A_N^C\})$;
if $|I_{corr}| > n$, Then return 0;
$\phi \rightarrow U_i$, $i \notin I_{corr}$;
$Setup(\lambda, N) \rightarrow (pa, \{PK_i, SK_i\}_{\{i=1...N\}})$;
$\mathcal{A}^{KeyGenO}(find', pa, PK_{i\{i=1...N\}}, SK_{i\{i \in I_{corr}\}})$
$\rightarrow (m_0^*, m_1^*)$;
$Enc(A^c, m_b^*) \rightarrow \mathcal{CT}, b \in \{0, 1\}$;
$\mathcal{A}^{KeyGenO}('guess', \mathcal{CT}, st) \rightarrow b'$;
if $b = b'$, then 0 else return 1.

The attribute-key generation oracle $\mathcal{A}^{KeyGenO}$ is defined as:
if($i \in I_{corr}$), return $\perp$;
if($\exists A_i^{u'}$ s.t. $(GID, A_i^{u'}) \in U_i$), return$\perp$;
if($|A_i^{u'} \bigcap A_i^C| \geq d_k) \bigwedge \{\forall j \neq k,$ $[(j \in I_{corr})$ $\bigcup(\exists A_j^u s.t.((GID, A_j^{u'}) \in U_j \bigwedge |A_j^u \bigcap A_C^j| \geq d_j))]\}$, return $\perp$;
$U_i \cup (GID, A_i^u) \rightarrow U_i$, return $\mathcal{A}^{KeyGenO}$.

Definition 1 (Selective-attribute attack): An N-authority ABE scheme can against selective-attribute attack if for any probability polynomial time adversary $\mathcal{A}$, there is a neglible function $\epsilon(\lambda)$ such that $Adv_{\mathcal{A}}^{saa} = |Pr[Exp_{N-ABE}^{N-ABE-saa}(\lambda) = 1]| \leq \epsilon(\lambda)$.

3.3 Multi-authority ABKS

In this section, we will propose a general construction method and security reduction for multi-authority ABKS through study of multi-authority ABE. Data owners encrypt the keyword set with attribute sets. Data users get the user key from attribute authorities, and server matches the ciphertext with trapdoor, figure 1 shows the entire system framework. A multi-authority ABKS scheme consists algorithms as follows:

- **Setup**$(\lambda, N, U) \to (gp, PK_i, SK_i, PK_s, SK_s)$: This algorithm inputs the number of authorities N, a security parameter λ and the attributes universe U, the system outputs the public parameter gp, the sever public and secret key pairs (PK_s, SK_s) and attribute authority key pairs $(PK_i, SK_i)_{\{i=1...N\}}$, every key pairs for one attribute authority.
- **Keygen**(N, U, u, Υ) $\to msk_i$: This algorithm takes the number of attribute authorities, the attributes universe U, user identity u and the access structure Υ, outputs the decryption key $msk_i, \{i = 1...N\}$, corresponding the access structure Υ.
- **Enc**$(gp, M, W, PK_s, A_i) \to CT$: The Data owner inputs the set of attributes A_i, the sever public key PK_s, the message M, the keywords W, and the system public parameter gp, computes the ciphertext CT.
- **Trapdoor**$(W, msk_i, PK_s) \to T_W$: The Data user inputs keyword W, the attribute key msk_i, the server public key PK_s, and computes trapdoor T_W corresponding the access structure.
- **Test**$(T_W, CT, SK_S) \to 1$ *or* 0: The server inputs the ciphertext CT, a trapdoor T_W and the server secret key SK_s. If the ciphertext attributes satisfy the access structure tree, outputs 1, else outputs 0.
- **Dec**$(CT, gp, msk_{i\{i=1...N\}}) \to M$: The Data user inputs the ciphertext CT, the system public parameter gp and the user key msk_i, outputs the message M, if attributes ciphertext satisfies the access structure tree.

3.4 Security Model of Multi-authority ABKS

About N-authority ABEKS scheme, we can define by the selective-attribute ciphertext attack experiments as follows:

- **Setup:** $\mathcal{A}$ is assumed to be an polynomial time attack algorithm, running time is bounded by t. $\mathcal{A}$ give the challenge attribute sets γ to the challenger. The public parameter are given to the $\mathcal{A}$.

- **Phase 1-1:** $\mathcal{A}$ makes the queries of the trapdoor T_w, adaptively makes the attribute keys queries for corresponding to any access structures tree Υ, where attribute set γ is not satisfy access structures tree Υ.

- **Phase 1-2:** $\mathcal{A}$ gives challenger $\mathcal{B}$ two be challenged keywords, w_0 and w_1. $\mathcal{B}$ randomly picks bit b, and computes a attribute ciphertext for w_b under attribute sets γ and returns it to $\mathcal{A}$.

- **Phase 1-3**: $\mathcal{A}$ continues making trapdoor queries of the form w and the attacker can not ask for the trapdoors w_0 and w_1.

- **Phase 1-4:** $\mathcal{A}$ outputs its guess b'.

In this attack experiments, the advantage of the adversary $\mathcal{A}$ is:

$$Adv_{\mathcal{A}}^{saca} = |Pr[b = b'] - 1/2|.$$

Definition2 (Selective-attribute ciphertext attack): An N-authority ABKS scheme is secure against selective-attribute ciphertext attack if for any probability polynomial time adversary $\mathcal{A}$, there is a neglible function $\epsilon(\lambda)$ such that

$$Adv_{\mathcal{A}}^{saca} = |Pr[b = b'] - 1/2| \leq \epsilon(\lambda).$$

About N-authority ABEKS scheme, we can define by the trapdoor indistinguishability experiments as follows:

- **Setup:** $\mathcal{A}$ is assumed to be an polynomial time attack algorithm, running time is bounded by t, the public parameter and sever public key are given to the $\mathcal{A}$.

- **Phase 1-1:** $\mathcal{A}$ makes the queries of the trapdoor T_w, adaptively makes the attribute keys corresponding to any access structures tree Υ.

- **Phase 1-2:** $\mathcal{A}$ gives challenger $\mathcal{B}$ two be challenged keywords, w_0 and w_1. $\mathcal{B}$ randomly picks bit b, and computes a trapdoor T_{w_b} for w_b under attribute sets γ and returns it to $\mathcal{A}$.

- **Phase 1-3:** $\mathcal{A}$ continues making trapdoor queries of the form w and the attacker can not ask for the trapdoors w_0 and w_1.

- **Phase 1-4:** $\mathcal{A}$ outputs its guess b'.

The advantage of the adversary $\mathcal{A}$ in this game is defined as:

$$Adv_{\mathcal{A}}^{Trapdoorindistinguishability} = |Pr[b = b'] - 1/2|.$$

Definition3 (Trapdoor indistinguishability): An N-authority ABKS scheme is trapdoor indistinguishability secure for any probability polynomial time adversary $\mathcal{A}$, there is a neglible function $\epsilon(\lambda)$ such that

$$Adv_{\mathcal{A}}^{Trapdoorindistinguishability} = |Pr[b = b'] - 1/2| \leq \epsilon(\lambda).$$

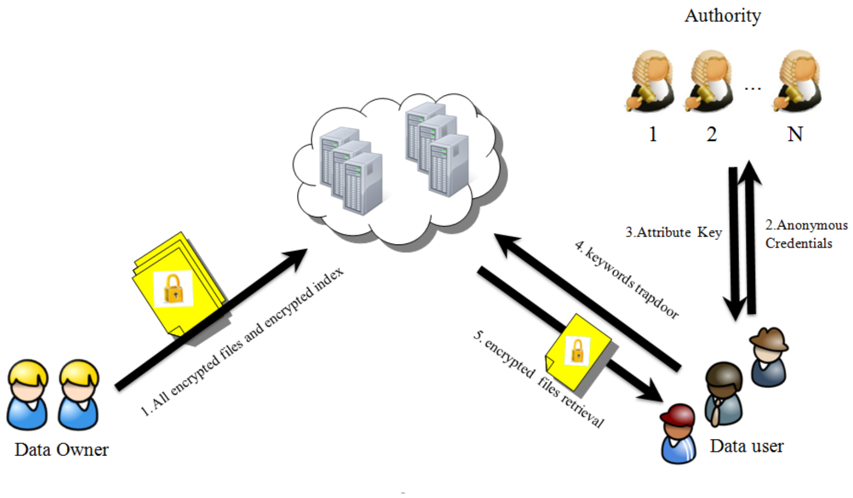

Fig. 1 Multi-authority ABKS

4 A concrete Multi-authority ABKS scheme

4.1 Scheme Description

We will give a concrete a multi-authority ABKS scheme that is based on the above multi-authority ABE scheme. This scheme consists algorithms as follows:

- **Setup**$(\lambda, N, \Im, U)$: This algorithm inputs the number of authorities N, a security parameter $1^{\Im}$, and the attributes universe U, the system outputs a system parameter

  $$gp = \{e, q, g, g_0, g_1, g_2, G_1, G_2, G_T, H, H_1, H_2\}, \quad (1)$$

  G_1, G_2 and G_T are three cyclic multiplicative groups of prime order p, $g, g_1 \in G_1, g_1 = g^y, g_0, g_2 \in G_2, g_2 = g_0^{\partial}, y, \partial \in Z_{p^*}$, $H : \{0,1\}^* \rightarrow G_1, H_1 : \{0,1\}^* \rightarrow G_1$, $H_2 : \{0,1\}^* \rightarrow G_T$, the server public and secret key pairs

  $$(PK_s, SK_s) = (g_2^a, a) \quad (2)$$

  the user secret key

  $$SK_u = \partial \quad (3)$$

  and N master key pairs

  $$(MPK_i, MSK_i) = (Y_i, v_i) = (e(g_1, g_2)^{v_i}, v_i), \quad (4)$$

  $\{i = 1...N\}$, every key pairs for one attribute authority. For each attribute $j \in \{1, 2, ...n_i\}$, authority i picks $x_i, t_{i,j} \in Z_p$, each authority i stores attribute private key

  $$< x_i, \{s_{i,t}\}_{t \in \{1,2...N\} \backslash \{i\}}, \{t_{ij}\} >, \quad (5)$$

  s_{it} is only know to authority $\{i, t\}$, $s_{i,t} = s_{t,i}$. Finally the system outputs the authority public parameters as follows:

  $$\begin{aligned} &< Y = e(g_1, g_2)^{\sum_i v_i}, \\ &\{y_i = g^{x_i}, T_{1,ij} = g_1^{t_{ij}}, T_{2,ij} = g_0^{t_{ij}}\} > . \end{aligned} \quad (6)$$

- **Keygen**(N, U, u, Υ): Each user u invokes an anonymous key distribution protocol and obtains the attribute key as follows: authority i randomly chooses $R_{it} \in Z_p$ and computes

  $$\begin{aligned} D_{it} &= g_1^{R_{it}} PRF_{it}(u), PRF_{it}(u) = g^{x_i x_t / s_{it} + u}, i > t \\ D_{it} &= g_1^{R_{it}} / PRF_{it}(u), PRF_{it}(u) = g^{x_i x_t / s_{it} + u}, i < t \end{aligned}$$

  and sends D_{it} to user u. User u can compute

  $$\begin{aligned} &D_u = \prod D_{it} = g_1^{R_u} \\ &R_u = \sum R_{it} \\ &(i, t) \in \{1, 2, ...N\} \times \{1, 2, ...N\} \backslash \{i\}. \end{aligned} \quad (7)$$

  Anonymous key issuing protocol is similar in Chase's paper. Authority i chooses a degree d_i polynomial at random

  $p_i(.)$ with $p_i(0) = v_i - \sum_{t \in \{1,2,....N\} \backslash i} R_{it}$

  and issues

  $$S_{ij} = g_1^{p_i(j)/t_{ij}} \quad (8)$$

  for the user u attribute j.
  About attribute keys $D(x)$ and R_x corresponds the tree Υ, user first defines a polynomial q_x for each node x. These polynomials are started from the root node r to leaf node. First for each node

x, user sets polynomial q_x degree $d_x = k_x - 1$, which d_x is the threshold value. Now user sets value $q_r(0) = y$ and chooses other q_r polynomial points at random. So, the q_r polynomial has been defined completely. For any other node x similar with root node r, sets $q_x(0) = q_{parent_x}(index(x))$ and chooses random other points of the polynomial q_x to define it completely.
According to the above method, user defines the polynomial completely. User gives the value $D(x)$ and R_x to the user for each leaf node x:

$$D(x) = g^{q_x(0)} T_{1,ij}^{r_x}, R_x = g_1^{r_x}, \tag{9}$$

which r_x is randomly choose in Z_P. Finally authorities outputs the attribute key $msk_i = (D_x, R_x, D_u, S_{ij})$ for the user, $i \in \{1, 2, ...N\}$.

- **Enc**$(M, W, PK_s, A_{i\{i=1...N\}})$: The Data owner takes the set of attributes A_i, the keywords set W and the server public key PK_S, chooses random numbers s and computes:

$$\begin{aligned} &E_0 = MY^s, E_1 = g_0^s, C_{ij} = T_{2,ij}^s = g_0^{st_{ij}}, \\ &j \in A_{i\{i=1...N\}}, E_2 = H_2(t) \cdot e(g_1, g_2)^s, \\ &t = e(H_1(W), PK_s^s), E_3 = g_2^s. \end{aligned} \tag{10}$$

- **Trapdoor**(W, msk_i, PK_s, SK_u): Data user inputs keyword W, the attribute key msk_i, the server public key PK_s, the user secret $SK_u = \partial$, chooses random number r' and computes trapdoor

$$\begin{aligned} &T_1 = H_1(W) \cdot H(PK_s^{r'}), T_2 = g_2^{r'}, T_3 = D_x^{\partial}, \\ &T_4 = R_x^{\partial}, T_W = [T_1, T_2, T_3, T_4]. \end{aligned} \tag{11}$$

- **Test**(T_W, CT, SK_s): Server inputs the ciphertext CT, a trapdoor T_W and the server secret key SK_s. When the ciphertext attribute satisfy the access structure tree γ. Algorithm can compute $Z_x = e(T_3, E_1)/e(T_4, C_{ij}) = e(g, g_2)^{sq_x(0)}$, when the node x is a leaf node. If x is a non-leaf node: computes the node v that are children of x

$$\begin{aligned} &i = index(v), S'_x = \{index(v) : v \in S_x\} \\ &Z_x = \textstyle\prod_{v \in S_x} (e(g, g_2)^{sq_v(0)})^{\triangle_{i,S'_x}(0)} \\ &= \textstyle\prod_{v \in S_x} (e(g, g_2)^{sq_{parent(v)}(index(v))})^{\triangle_{i,S'_x}(0)} \\ &= \textstyle\prod_{v \in S_x} e(g, g_2)^{sq_x(i)\triangle_{i,S'_x}(0)} \\ &= e(g, g_2)^{sq_x(0)}. \end{aligned}$$

Server takes recursively way to compute and get the value $e(g, g_2)^{sy} = e(g_1, g_2)^s$. Finally it verifies

$$E_2/e(g_1, g_2)^s = H_2(e(T_1/H(T_2^{SK_s}), E_3)^{SK_s})$$

or not. If they are equal, output 1 else 0.

- **Dec**$(CT, pa, D_x, R_x, D_u, S_{ij})$: For each authority $i \in [1, 2...N]$: for any d_i attributes $i \in A_i^C \bigcap A_u^C$, where A_u denotes the attribute set of the data user, compute $e(S_{ij}, C_{ij})^{\partial} = e(g_1, g_2)^{sp_i(j)}$. Interpolate all the valuese$(g_1, g_2)^{sp_i(j)}$ to get $B_i = e(g_1, g_2)^{sp_i(0)}$, multiply B_i together to get $C = Y^s/e(D_u, g_2^s)$ and compute $e(D_u, E_3) \cdot C = Y^s$, finally data user can recover M by E_0/Y^s.

Correctness: When assuming the ciphertext $\{E_0, E_1, C_{ij}, j \in A_{i\{i=1...N\}}, E_2, E_3\}$ is valid for W' and the trapdoor T_W for W, we can verify Test algorithm correctness as:

For node x, when $i = att(x)$, according to the above analysis, we can get the value $E_2/e(g_1, g_2)^s = H_2(t)$ and continue computing

$H_2(e(T_1/H(T_2^{SK_s}), E_3)^{SK_s}) \overset{?}{=\!=} H_2(e(H_1(W'), g_2^{as}))$, test whether two values are equal by W and W'.

4.2 Security proof

The following theorem shows that the scheme is indistinguishes of keywords secure.

Theorem 1: Suppose the BDH problem is hard in group G_2, the multi-authority ABKS is secure under selective attribute ciphertext model attack.

Proof : Security proof similar paper Goyal et al (2006), adversary $\mathcal{A}$ is an an polynomial time attack algorithm can breake the muti-authority ABKS with the advantage ε . We construct an algorithm $\mathcal{B}$ that solves the BDH problem with $\varepsilon' = \varepsilon/eq_T q_{H_2}$, q_T and q_{H_2} are hash function queries to trapdoor and H_2. Let $G_1 =< g_1 >$. Algorithm $\mathcal{B}$ is given g_1, $g_2^a, g_2^b, g_2^c \in G_1$. It's goal is to output $v = e(g_1, g_2)^{abc} \in G_T$. Challenger randomly chooses $d \in \{0, 1\}$, output $< g_2^a, g_2^b, g_2^c, Z >$. When $d = 1$, challenger chooses $Z = e(g_1, g_2)^{abc}$, else randomly chooses $Z \in G_2$. Algorithm $\mathcal{B}$ simulates the challenger and interact with adversary $\mathcal{A}$ as follows:

•**Setup:** Algorithm $\mathcal{B}$ invokes adversary algorithm $\mathcal{A}$ choose attribute set , include elements in Z_P, Algorithm $\mathcal{B}$ sets $g'_1 = g_2^a, g'_2 = g_2^b$ and chooses randomly t_{ij}, B_{ij}. When $i \in \gamma$, $t_{ij} = B_{ij}$, else $t_{ij} \neq B_{ij}$.

• **Attributes key queries:** Adversary algorithm $\mathcal{A}$ adaptively makes the attribute keys queries for the attribute set γ is not satisfy access structures tree Υ, $\Upsilon(\gamma) = 0$. In order to generate keys, $\mathcal{B}$ must access the tree for each non leaf node determines a degree d_x polynomial $Q_x(x)$, $PolyUnsat(\Upsilon(x), \gamma, g^{\lambda(x)})$ is an algorithm for the root node x (attribute γ with unsatisfied root node x access tree, $\Upsilon(\gamma) = 0$). The algorithm $\mathcal{B}$ inputs access tree $\Upsilon(x)$, set of attributes

γ, an element $g^{\lambda(x)}$. Firstly, algorithm $\mathcal{B}$ defines root node x polynomial q_x, which degree is d_x. Because the attribute set is not satisfy access structures tree, $\Upsilon(\gamma) = 0$, no more than d_x children of x are satisfied the access structure tree. We assume that the children of node x has k_x children node x', which satisfies attributes γ of the node x, the algorithm chooses randomly point $\lambda(x') \in Z_p$, set $q_x(index(x')) = \lambda_{x'}$ and randomly choose point $d_x - k_x$, so q_x is defined completely. Finally, Algorithm $\mathcal{B}$ determines that the polynomial order is d_x . Notice that this when we know the value $g^{q_x(0)}$, $g^{q_x(index(x'))}$ can be obtained by interpolation and $q_{x'}(0) = q_x(index(x'))$. Algorithm $\mathcal{B}$ runs $PolyUnsat(\Upsilon(x), \gamma, g^{\lambda(x)})$ to define a polynomial q_x of degree d_x for the node x and satisfy $q_r(0) = a$. For the leaf node, if x satisfy the attribute sets γ, then algorithm can obtain $q_x(0)$, else algorithm can obtain $g^{q_x(0)}$.

Algorithm $\mathcal{B}$ defines $Q_x(.) = q_x(.)$, $y = Q_r(0) = a$. We use the following polynomial to represent the key of the leaf node:
$i = att(x)$,
if $i \in \gamma$, $D_x = g_1'^{Q_x(0)}(g_1'^{t_{ij}})^{r_x}$
$R_x = g_1'^{r_x}$, choose randomly $r_x \in Z_p$,
if $i \notin \gamma$, $g_3 = g_1'^{Q_x(0)} = g_1'^{q_x(0)}$
$D_x = g_3(g_1'^{t_{ij}})^{r_x}$
$R_x = g_1'^{r_x}$, choose randomly $r_x \in Z_p$.

Therefore, we have successfully constructed the leaf node's private key for the access structure tree Υ.

Next, the adversary can access the random oracle H, H_1, ask the trapdoor queries for keyword w and query execution similar paper (Boneh et al (2004); Rhee et al (2010)).

• **Challenge:** adversary chooses two keywords w_0 and w_1 to the simulator algorithm $\mathcal{B}$, the algorithm $\mathcal{B}$ randomly chooses a bit b, chooses randomly $c' \in Z_P$ and returns ciphertext of m_b:
$C = (E_0 = MY^c, E_1 = g_2'^c, C_{ij} = g_1'^{t_{ij}}, E_2 = JZ, E_3 = g_2'^{c'})$, Z is define an above phase.

More trapdoor queries: Adversary algorithm $\mathcal{A}$ continues making trapdoor queries of the form w and the attacker can not ask for the trapdoors w_0 and w_1. algorithm $\mathcal{B}$ acts exactly as it did in attributes key queries.

• **Output:** Adversary algorithm $\mathcal{A}$ outputs its guess b'. When $b = b'$, the algorithm $\mathcal{B}$ outputs $d' = 0$, else, outputs $d' = 1$. The advantage of the algorithm $\mathcal{B}$ in the DBDH game is:

$$\begin{aligned}&\Pr[Succes_B]\\&= 1/2\Pr[d = d'|d = 0] + 1/2\Pr[d \neq d'|d = 1] - 1/2\\&= \varepsilon/2.\end{aligned}$$

Since $E_1 = g_2'^c$, $J = H_2(e(H_1(W), PK_s^s))$, security proof similar paper Boneh et al (2004). Adversary algorithm $\mathcal{A}$ can analyze query to $H_2(t)$ and the pair (t, J) in an H_2 list.

$$t = e(H_1(w_b), g_2^{ac}) = e(g_1, g_2)^{ac(b+a_b)}.$$

At this point, Algorithm $\mathcal{B}$ can output $t/e(g_1, g_2)^{aca_b}$ as its guess for $e(g_1, g_2)^{abc}$, so the Algorithm $\mathcal{B}$ advantage is at least $\varepsilon' = \varepsilon/eq_T q_{H_2}$ as required.

So, q_{H_2} and q_{H_1} are hash function queries, if adversary $\mathcal{A}$ is an polynomial time attack algorithm can breake the muti-authority ABKS with the advantage ε. Then there is the attack algorithm can solve the DBDH game and have probability overall is $\varepsilon/2$. In the case of DBDH problem solvable, then there is the attack algorithm can solve the BDH game and have probability overall is $\varepsilon' = \varepsilon/eq_T q_{H_2}$.

Hence, the proof of **Theorem1** is completed. □

An adversary can obtain trapdoor for any keyword of his choice, even under this attack the adversary can not indistinguish the encryption of two challenge keywords for which he did not obtain the trapdoor.

Theorem 2: Multi-authority ABKS is an scheme satisfies the trapdoor indistinguishability against a chosen keyword attack, under assumption that Hash Diffie-Hellman (HDH) is intractable.

Proof : Security proof similar paper Rhee et al (2010), suppose an polynomial time malicious outside adversary $\mathcal{A}$ is an an polynomial time attack algorithm can break the trapdoor indistinguishability about the muti-authority ABKS. We construct an algorithm $\mathcal{B}$ that solves the HDH problem with $\varepsilon' = \varepsilon/2$, q_T is hash function queries to trapdoor. Algorithm $\mathcal{B}$ is given $(g, g^a, g^b, \eta) \in G_1^4$ and $H : \{0,1\}^* \to G_1$ ia a hash function, where η is either $H(g^{ab})$ or a random HDH challenge of G_1. Challenger randomly chooses $d \in \{0,1\}$, output $(g, g^a, g^b, \eta) \in G_1^4$, When $d = 1$, we choose $\eta = H(g^{ab})$, else $d = 0$, choose a random element η. Algorithm $\mathcal{B}$ simulates the challenger and interact with adversary $\mathcal{A}$ as follows:

•**Setup:** Algorithm $\mathcal{B}$ randomly chooses $l \in Z_p^*$ and sets the server key pairs $(PK_s, SK_s) = (g^{al}, al)$. It randomly chooses $\partial' \in Z_p$ and sets the user secret key $SK_u = \partial'$. Here, We define an unknown value a such that $SK_S = a = al$.

•**Attributes key queries:** $\mathcal{A}$ makes trapdoor queries of the form w and can not ask for the trapdoors w_0 and w_1. Algorithm $\mathcal{B}$ acts as follows:

Algorithm $\mathcal{B}$ randomly chooses $r' \in Z_p$ and computes $T_1^* = H_1(w)H((g^{al})^{r'})$ and $T_2 = g^{r'}$, where l

is selected values in the setup phase and D_x, R_x are selected similar the **Theorem 1**.

Algorithm $\mathcal{B}$ responds to Algorithm $\mathcal{A}$ with the trapdoor, $T_{w_i} = [T_1, T_2, T_3, T_4]$ of w_i.

•**Challenge:** Adversary algorithm $\mathcal{A}$ chooses two keywords w_0 and w_1 to the simulator algorithm $\mathcal{B}$. The algorithm $\mathcal{B}$ randomly choose bit b, and returns challenge trapdoor $T^*_{w_b}$ as follows:

The algorithm sets $T_1^* = H_1(w)\eta$, $T_2^* = g^{r'}$, $T_3^* = D_x^{\partial'}$ and $T_4^* = R_x^{\partial'}$, where l, r' is a selected value in the setup phase and η is a component of the HDH challenge.

Algorithm $\mathcal{B}$ responds to Algorithm $\mathcal{A}$ with the trapdoor $T_{w_b} = [T_1^*, T_2^*, T_3^*, T_4^*]$ of w_b. When the $\eta = H(g^{ab}), T_{w_b}$ is a valid challenge trapdoor w_b.

More trapdoor queries: Adversary algorithm $\mathcal{A}$ can continue to issue trapdoor queries of the form w and the attacker can not ask for the trapdoors w_0 and w_1. algorithm $\mathcal{B}$ acts exactly as it did in attributes key queries.

• **Output:** Adversary algorithm $\mathcal{A}$ outputs its guess b'. When $b = b'$, the simulator algorithm $\mathcal{B}$ outputs $d' = 0$, meaning that $\eta = H(g^{ab})$, else, output $d' = 1$, meaning that $\eta \neq H(g^{ab})$. The advantage of the algorithm $\mathcal{B}$ in the HDH game is $\varepsilon/2$ similar the Theorem 1.

Hence, the proof of **Theorem 2** is completed. □

Theorem 3: If multi-authority ABE is a secure scheme, multi-authority ABEKS is an ABKS scheme derived from multi-authority ABE, then multi-authority ABKS is a secure ABKS under assumption that DBDH or q-DDHI problem is intractable.

Theorem 4: If multi-authority ABE is a secure authority unlinkable ABE when at most N-2 of authorities are corrupted, then multi-authority ABKS is a secure multi-authority ABKS under assumption that XDH problem is intractable.

Paper Chow (2010) have proved that the multi-authority ABE scheme is secure by Theorem 7.4 and 7.5. We can show that our scheme is also confidential similar to the paper, the full proofs can be found in the paper Chow (2010), so we do not prove the security here in detail.

About the security keyword guessing attack, we add public key and private key for the cloud server and the data owner can re-encrypt the keyword ciphertext and trapdoor in public channel. This guarantees only the cloud server matches the keyword ciphertext and trapdoor. Our scheme can effectively prevent the guessing attack by this approach.

5 Security and Performance Analysis

Basic operations are recorded as: Let U denotes number of attributes, E denotes an exponentiation operation, P is the basic operation of hash operations, e denotes a pairing operation, k represents the maximum number of trapdoor, q denote number of leaf node, f denotes a polynomial operation, n denote number of node, M denotes a multiplication operation in the group. Table 1 and Table 2 give us the comparison between our scheme and the previous searchable encryption scheme. We use Trap Ind, Ciph Ind, Anonymity to denote Trapdoor indistinguishability, Ciphertext indistinguishability, User anonymity, Keyword guessing attack.

6 Conlusions

In order to overcome the disadvantages of traditional attribute-based keyword search scheme in multi-authority environment, this paper proposes a multi-authority attribute searchable encryption scheme by using bilinear on technique. By defining the data security and user privacy, we have proved that the scheme achieves attribute ciphertext indistinguishability, trapdoor indistinguishability, user anonymity, keyword guessing attack. Comparing with existing schemes, our scheme supports the access control on the searching result based upon fuzzy identity and elimates the central authority in multi authority environment. It is a suitable method for solving the practical problem which is described in the introduction. For the encrypted personal health record system, the universe of attributes can be partitioned into four disjoint sets as {Medical Association Membership, Chief Physician, Medical Researcher, Police}. Patients encrypt keywords and messages specifying from the four disjoint sets, such that only an authorized person who has adequate attribute keys from four authorities {Medical Association, Hospital, Scientific Research Institution and Public Security Bureau} can search and decrypt the message in cloud environment. Through the aforementioned content, we can get that this proposed multi-authority attribute searchable encryption scheme is a secure and wide applicable protocol, and has a certain practical value.

Compliance with Ethical Standards

Conflict of Interest: The authors declare that they have no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent: N/A.

Table 1 Security comparison

	Boneh et al	Baek et al	Yang et al	Our.
Trap Ind	NO	NO	YES	YES
Ciph Ind	YES	YES	YES	YES
Anonymity	YES	YES	YES	YES
KG	NO	NO	YES	YES

Table 2 Performance comparison

	Boneh et al	Baek et al	Yang et al	Our.
KeyGenServer	-	M	-	E
KeyGenReceiver	E	M	6E	-
PEKS	2E+2P+e	E+M+P+2e	(2k+6)E	M+(U+4)E+2P+e
Trapdoor	E+P	P+M	4f	M+(n+2)E+2P
Test	e+P	M+e	2E+4f	(q+1)e+2P+2E

References

Abdalla M, Bellare M, Catalano D, et al (2008) Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions. Journal of Cryptology, 21(3):350-391.

Baek J, Safavi-Naini R, Susilo W (2008) Public Key Encryption with Keyword Search Revisited. Computational Science and Its Applications C ICCSA 2008, pp:1249-1259.

Bethencourt J, Sahai A, Waters B (2007) Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, pp:321-334

Boneh D, Crescenzo G D, Ostrovsky R, et al (2004) Public Key Encryption with Keyword Search. Advances in Cryptology-EUROCRYPT 2004. Springer Berlin Heidelberg, pp:506-522.

Chang Y C, Mitzenmacher M(2005) Privacy preserving keyword searches on remote encrypted data.Applied Cryptography and Network Security. Springer Berlin Heidelberg, pp:442-455.

Chase M, Chow S S M(2009) Improving privacy and security in multi-authority attribute-based encryption. ACM Conference on Computer and Communications Security. ACM, pp:121-130.

Chow S M (2010). New privacy-preserving architectures for identity-attribute-based encryption. New York University.

Curtmola R, Garay J, Kamara S, Ostrovsky R (2006) Searchable symmetric encryption: improved definition and efficient constructions. Proceedings of the 13th ACM conference on Computer and communications security. ACM, pp: 79-88.

Goh E (2003) Secure indexes. Technical Report, 2003/216, IACR ePrint Cryptography Archive. http://eprint.iacr.org/2003/216.

Golle P, Staddon J, Waters B (2004) Secure Conjunctive Keyword Search over Encrypted Data. Lecture Notes in Computer Science, 3089:31-45.

Goyal V, Pandey O, Sahai A, et al (2006) Attribute-based encryption for fine-grained access control of encrypted data. ACM Conference on Computer and Communications Security. ACM, pp:89-98.

Han F, Qin J, Zhao H, et al (2014) A general transformation from KP-ABE to searchable encryption. Future Generation Computer Systems, 30(C):107-115.

Koo D, Hur J, Yoon H(2013) Secure and efficient data retrieval over encrypted data using attribute-based encryption in cloud storage. Computers & Electrical Engineering, 39(1):34-46.

Li J, Li X, Wang L, et al (2017) Fuzzy encryption in cloud computation: efficient verifiable outsourced attribute-based encryption. Soft Computing, pp:1-8.

Li J, Zhang L (2014) Attribute-Based Keyword Search and Data Access Control in Cloud. Tenth International Conference on Computational Intelligence and Security. IEEE, pp:382-386.

Lin H, Cao Z, Liang X, et al (2008) Secure Threshold Multi Authority Attribute Based Encryption without a Central Authority. International Conference on Cryptology in India: Progress in Cryptology. Springer-Verlag, pp:426-436.

Liu P, Wang J, Ma H, et al (2014) Efficient Verifiable Public Key Encryption with Keyword Search Based on KP-ABE, 2014 Ninth International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), pp: 584-589.

Liu Z, Weng J, Li J, et al (2016) Cloud-based electronic health record system supporting fuzzy keyword search. Soft Computing, 20(8):3243-3255.

Ma S, Lai J, Deng R H, et al(2016) Adaptable key-policy attribute-based encryption with time interval. Soft Computing, pp:1-10.

Park D J, Kim K, Lee P J (2004) Public key encryption with conjunctive field keyword search. International Conference on Information Security Applications. Springer-Verlag, pp:73-86.

Rhee H S, Park J H, Susilo W, et al (2010) Trapdoor security in a searchable public-key encryption scheme with a designated tester. Journal of Systems & Software, 83(5):763-771.

Sahai A, Waters B (2005) Fuzzy Identity-Based Encryption. Lecture Notes in Computer Science, 3494:457-473.

Song D X, Wagner D, Perrig A(2000) Practical Techniques for Searches on Encrypted Data. IEEE Symposium on Security and Privacy. IEEE Computer Society, pp:44-55.

Shi Y, Liu J, Zhen H, et al (2014) Attribute-Based Proxy Re-Encryption with Keyword Search. Plos One, 9(12):e116325.

Wang C, Li W, Li Y, et al (2013) A Ciphertext-Policy Attribute-Based Encryption Scheme Supporting Keyword Search Function. Cyberspace Safety and Security, pp:377-386.

Wang H, He D, Shen J, et al(2016) Verifiable outsourced ciphertext-policy attribute-based encryption in cloud computing. Soft Computing, pp:1-11.

Wang H, He D, Shen J, et al (2017) Fuzzy matching and direct revocation: a new CP-ABE scheme from multilinear maps. Soft Computing, pp:1-8.

Xhafa F, Wang J, Chen X, et al (2014) An efficient PHR service system supporting fuzzy keyword search and fine-grained access control. Soft Computing, 18(9):1795-1802.

Xu J, Wen Q, Li W, et al (2016) Succinct multi-authority attribute-based access control for circuits with authenticated outsourcing. Soft Computing, pp:1-15.

Zheng Q, Xu S, Ateniese G (2014). VABKS: Verifiable Attribute-based Keyword Search over Outsourced Encrypted Data. IEEE INFOCOM 2014 - IEEE Conference on Computer Communications. IEEE, pp:522-530.

Zhong H, Zhu W, Xu Y, et al(2016) Multi-authority attribute-based encryption access control scheme with policy hidden for cloud storage. Soft Computing, pp:1-9.