Privacy-preserving Attribute Based Searchable Encryption

Payal Chaudhari[1] and Manik Lal Das[2]

[1] LDRP - ITR, Gandhinagar, payal.ldrp@gmail.com
[2] DA-IICT, Gandhinagar, maniklal_das@daiict.ac.in

Abstract. Attribute Based Encryption (ABE) is a promising public-key cryptographic primitive that can be used for cryptographically enforced access control in untrusted storage. Storing data on untrusted storage not only requires data security for data owners but also poses data protection from untrusted storage server. To address this important requirement, Anonymous Attribute Based Encryption (AABE) is a suitable primitive that provides users to access data from untrusted storage without revealing their identities. At the same time user data can be stored in untrusted storage in an encrypted form. While storing data in an encrypted form, keyword-based query search (and data retrieval) is a challenging research problem. In this paper we present an anonymous attribute based searchable encryption (A^2SBE) scheme which facilitates user to retrieve only a subset of documents pertaining to his chosen keyword(s). User can upload documents in public cloud in an encrypted form, search documents based on keyword(s) and retrieve documents without revealing his identity. The scheme is proven secure under the selective ciphertext-policy and chosen plaintext attack (IND-sCP-CPA) model and selective ciphertext-policy and chosen keyword attack (IND-sCP-CKA) model. The scheme requires small storage for user's decryption key and reduced computation for decryption in comparison to other schemes.

Keywords: Attribute Based Encryption, Anonymity, Searchable Encryption, Access Structure.

1 Introduction

Cloud Computing provides a powerful infrastructure which facilitates elastic and unlimited resources as services to cloud users. Among many cloud services, cloud storage service has found enormous utilization because of low-cost pay-per-use service, data availability to users, scalability and efficient data management services. Although data storage on public cloud provides an ease of accessibility, it poses concerns of data confidentiality and access control. Applications such as electronic health record storage system demands data confidentiality, fine grained access control, and concealing the identity of data user as the intended security requirements. Attribute Based Encryption (ABE) is a promising public-key primitive that has been used for cryptographically enforced access control

in untrusted storage. Sahai and Waters [1] first introduced attribute based encryption(ABE) scheme. In ABE, both the user secret key and the ciphertext are associated with a set of attributes. A user is able to decrypt the ciphertext if and only if at least a threshold number of attributes overlap between the ciphertext and user secret key. ABE is flexible than the traditional public-key encryption, as ABE is intended for one-to-many encryption in which ciphertexts are not necessarily encrypted to one particular user. There are two variants of ABE - Ciphertext Policy Attribute Based Encryption (CP-ABE)[2] and Key Policy Attribute Based Encryption (KP-ABE)[3]. In CP-ABE while encrypting a document, the access policy related to the document is incorporated in the ciphertext. Each User possess a key specifying the attributes he possesses. If the user's key satisfies the access policy related to a ciphertext, then only the user is able to decrypt the ciphertext. However, along with ciphertext its access policy specifying the attributes of the intended receiver transmitted in clear form. In KP-ABE scheme access policy is attached with user's key and attribute list is attached with ciphertext.

Using ABE, on one hand, data access policies can be enforced on storage servers; on the other hand, confidentiality of sensitive data can be well protected against unauthorized entities, including storage servers. In addition to data security, concealing the receiver's identity becomes an important security feature in some applications such as electronic health record (EHR) system. A patient can upload his data to EHR system, be accessible by authorized clinicians, where the patient's privacy should be preserved. Anonymous attribute based encryption (AABE) is a suitable primitive which supports data receiver anonymity in addition to data confidentiality and access control security features. While storing data in encrypted form in storage server for protecting data from storage server, it is required users to search and retrieve data over encrypted documents. In addition, in many instances a user requires only a subset of documents containing a particular keyword, which makes the data access mechanism efficient.

Over the years many schemes have been devised based on ABE [4–10]. However, these schemes have a limitation that the access policy (or the required attribute values) has to be attached with the ciphertexts in clear form which makes the identity of the receiver public. By identifying the receiver from the ciphertext, one could guess the purpose of the ciphertext that would leak important information. For example, if a teacher is sending a ciphertext to group of students then by seeing student category and course names as attributes one has enough information to determine whether the ciphertext is related to examination, grading etc. The problem becomes bigger for applications like electronic health record system and some electronic commerce applications. This motivates researchers to hide the identity of receiver while using ABE, that is, enabling the access policy hidden in ABE. Kapadia *et al* proposed a CP-ABE scheme [11] with receiver anonymity. Although the scheme in [11] realizes hidden ciphertext policies represented by AND of different attributes, it is not collusion-resistant and requires an online semi-trusted server. Boneh and Waters proposed a predicate encryption scheme based on Hidden Vector Encryption [12]. Katz *et al* proposed a predicate

encryption scheme supporting inner product predicates [13]. Relying on the Decisional Bilinear Diffie-Hellman assumption and Decisional Linear assumption, Nishid *et al* and Jin *et al* have proposed an efficient anonymous CP-ABE scheme [14] and [15], respectively. Subsequently, a few AABE schemes [16–18] have also proposed in literature. To make the decryption procedure efficient, Zhang *et al* proposed a scheme [19] with a technique called a *match-then-decrypt* procedure. The technique allows a receiver to check if his attribute private key matches the hidden access policy in ciphertexts without decryption. However, Zhang *et al*'s scheme is found insecure [20].
Search operation over ABE has been considered a challenging research problem. In recent times, a few schemes [21–25] discuss search operation over ABE, but did not support receiver anonymity. Koo *et al* [26] proposed a searchable anonymous ABE scheme where user is able to retrieve a subset of documents based on his search query, where the search query includes a cipher component published by the sender and known as pseudonym. The scheme [26] has the prerequisite that the receiver has to gain the pseudonyms of a data owner from cloud service provider, scramble his key with this pseudonym and send the reformed key to cloud service provider. Shi *et al* [27] recently proposed a scheme that provides a successful search only if each keyword field entry in the document matches with the searched words in user query. The scheme [27] has a bottleneck that each time generating a new predicate for search word the user requires to pass a request to token generator.

Our Contributions. We present an anonymous attribute based searchable encryption (A^2SBE) scheme, which facilitates user to retrieve only a subset of documents pertaining to his chosen keyword(s). User can upload documents in public cloud in an encrypted form, search documents based on keyword(s) and retrieve documents without revealing his identity. The scheme is proven secure under the selective ciphertext-policy and chosen plaintext attack (IND-sCP-CPA) model and selective ciphertext-policy and chosen keyword attack (IND-sCP-CKA) model. The scheme requires small storage for user's decryption key and reduced computation for decryption in comparison to other schemes.

Organization of the paper. The remaining of the paper is organized as follows. We discuss some preliminaries in Section 2. We present the A^2SBE scheme in Section 3. We present the analysis of the A^2BSE scheme in Section 4. We conclude the paper in Section 5.

2 Preliminaries

2.1 Bilinear Mapping

Let G_1 and G_2 be two multiplicative cyclic groups of a large prime order p. Let g be a generator of G_1 and e be a bilinear map, $e : G_0 \times G_0 \rightarrow G_1$. The bilinear map e has the following properties:

- Bilinearity: $e(g^a, g^b) = e(g, g)^{ab}$ for all $a, b, \in \mathbb{Z}_p$

- Non-degeneracy: There exists g_1, $g_2 \in G_0$ such that $e(g_1, g_2) \neq 1$.
- There exists an efficient computable algorithm to compute $e(g_1, g_2)$ for all g_1, $g_2 \in G_0$.

We say that G_0 is a bilinear group if it satisfies the above mentioned three properties.

2.2 Decisional Bilinear Diffie-Hellman (DBDH) Assumption

Let $a, b, c, z \in \mathbb{Z}_p$ be chosen at random and g be a generator of G_1. The decisional BDH assumption is that no probabilistic polynomial-time algorithm $\mathcal{P}$ can distinguish the tuple ($A = g^a$, $B = g^b$, $C = g^c$, $e(g,g)^{abc}$) from the tuple ($A = g^a$, $B = g^b$, $C = g^c$, $e(g,g)^z$) with more than a negligible advantage ϵ.
The advantage of $\mathcal{P}$ is $\Pr[\mathcal{P}(A, B, C, e(g,g)^{abc}) = 0]$ - $\Pr[\mathcal{P}(A, B, C, e(g,g)^z) = 0] = \epsilon$.

2.3 Decisional Diffie Hellman (DDH) assumption

Let $a, b, z \in \mathbb{Z}_p$ be chosen at random and g be a generator of G_1. The decisional DH assumption is that no probabilistic polynomial-time algorithm $\mathcal{P}$ can distinguish the tuple ($A = g^a$, $B = g^b$, $C = g^{ab}$) from the tuple ($A = g^a$, $B = g^b$, $C = g^z$) with more than a negligible advantage ϵ.
The advantage of $\mathcal{P}$ is $\Pr[\mathcal{P}(A, B, C) = 0]$ - $\Pr[\mathcal{P}(A, B, g^z) = 0] = \epsilon$.

2.4 Decisional Linear (D-Linear) Assumption

Let z_1, z_2, z_3, z_4, $z \in \mathbb{Z}_p$ be chosen at random and g be a generator of G_1. We say that the D-Linear assumption holds in G if no probabilistic polynomial-time algorithm $\mathcal{P}$ can distinguish the tuple ($g, Z_1 = g^{z_1}$, $Z_2 = g^{z_2}$, $Z_3 = g^{z_1 z_3}$, $Z_4 = g^{z_2 z_4}$, $Z = g^{z_3 + z_4}$) from the tuple $e(g, Z_1 = g^{z_1}$, $Z_2 = g^{z_2}$, $Z_3 = g^{z_1 z_3}$, $Z_4 = g^{z_2 z_4}$, $Z = g^z$) with non-negligible advantage ϵ.
The advantage of $\mathcal{P}$ is $\Pr[\mathcal{P}(Z_1, Z_2, Z_3, Z_4, e(g,g)^{z_3+z_4}) = 0]$ - $\Pr[\mathcal{P}(Z_1, Z_2, Z_3, Z_4, e(g,g)^z) = 0] = \epsilon$.
For the proposed scheme we consider a variant of D-Linear assumption. It states that no probabilistic polynomial-time algorithm $\mathcal{P}$ can distinguish the tuple (g, $Z_1 = g^{z_1}$, $Z_2 = g^{z_2}$, $Z_3 = g^{z_2 z_4}$, $Z_4 = g^{z_3+z_4}$, $Z = g^{z_1 z_3}$) from the tuple (g, $Z_1 = g^{z_1}$, $Z_2 = g^{z_2}$, $Z_3 = g^{z_2 z_4}$, $Z_4 = g^{z_3+z_4}$, $Z = g^z$) with non-negligible advantage ϵ.

2.5 Access Structure

Let there be n attribute in the universe and each attribute i (for all $1 \leq i \leq n$) has value set $V_i = \{v_{i,1}, v_{i,2}, \cdots, v_{i,m_i}\}$. $L = [L_1, L_2, \cdots, L_n]$ is an attribute list, where each L_i represents one value from the value set of attribute i. A ciphertext policy is defined as $T = [T_1, T_2, \cdots, T_n]$, where each T_i represents the set of permissible values of an attribute i in order to decrypt the ciphertext. An attribute list L satisfies an access structure T, if $L_i \in T_i$ or $T_i = *$ for all $1 \leq i \leq n$.

3 The Proposed A^{2}BSE Scheme

3.1 Security Goals

The proposed scheme aims to achieve the following goals.

Key Privacy. A valid key can only be generated by the trusted third party using secret parameters of the system. A user himself will not be able to generate a key or modifying a key to include different attributes.

Ciphertext Privacy. Given an encrypted ciphertext, if the user's key does not include the required attributes then the user will not be able to decrypt the ciphertext. Even if users collude but do not possess the required attributes can not decrypt the ciphertext.

Receiver's Anonymity. Given an encrypted ciphertext, if the user is not the intended recipient then he will not be able to know the required attributes to decrypt the ciphertext. Even if a user is the intended recipient then he will only be able to decrypt the ciphertext but will not be able to know, who else are the intended recipients for the ciphertext.

Trapdoor Privacy. User's search query includes his search key masked with the keyword for which the search has to be performed. The adversary will not be able to gain any useful information from the search query. The only information an adversary is able to produce from a search result is the number of documents provided in result of a search query.

3.2 System Model

The system (Figure 1) comprises the following entities.

- Attribute Center: Attribute Center is a trusted third party and is responsible for generating system parameters and issuing keys to users.
- User: User of the system who encrypts and uploads documents in the cloud. User can issue search query and retrieve the intended documents from the cloud server in a protected form. For simplicity, we consider two types of users - Data owners and Data retrievers. Data owners are the users who encrypt and store the data on cloud. Data retrievers are the users who issue search query and obtain the documents from the cloud.
- Cloud Service Provider (CSP): CSP provides storage and computation services to the users of the system. On receiving a search query from user, CSP performs the search on its storage for a proper match and then, for all the matched documents, CSP performs a partial decryption procedure before returning the documents to the user in a protected form.

3.3 Definition of A^{2}BSE

We define a binary relation $\mathrm{F}(L, T)$, where L is the list of attributes included in a user's key and T is the hidden access policy with which the encryption

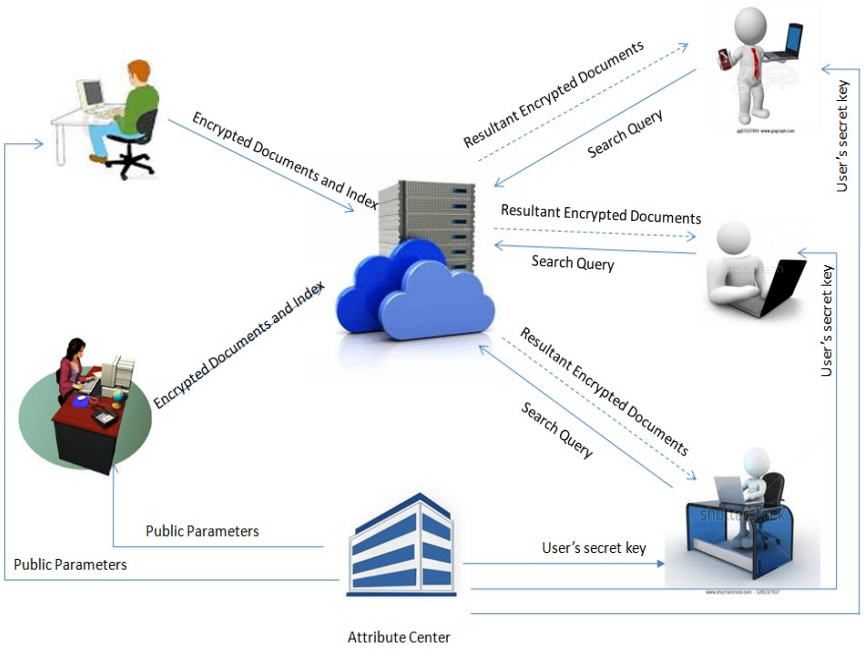

Fig. 1. The System Model of A^{2}BSE

has to be performed. F$(L,T) = 1$ if the attributes in L can satisfy the access policy specified in T; else, F$(L,T) = 0$. We also assume that the system has n categories of attributes and in each category i $(1 \leq i \leq n)$ permissible number of attributes are m_i.

Definition 1. *An A^{2}BSE system consists of seven algorithms Setup, KeyGen, Encryption, Encrypt_Index, QueryGen, Search, and Decryption, defined as follows.*

- Setup(1^l): The Setup(1^l) is run by Attribute Center. It takes as input parameter a security parameter l and outputs the master private key MK and public parameters PK.
- KeyGen(MK,L): The KeyGen(MK,L) algorithm takes as input the master key MK along with a set of attributes L and outputs two secret keys SK_s and SK_d comprising components for all attributes in L. The former key is used for searching anonymously over encrypted data in cloud storage and the later is for decryption of ciphertext.
- Encryption(PK,M,T): The Encryption(PK,M,T) algorithm is used for encrypting user's document M as per the access policy T using the system's public key PK. The algorithm outputs an encrypted document CT which is to be uploaded to the cloud.

- Encrypt_Index(PK, W, T): The set of words W are extracted from M is given as input to the Encrypt_Index(PK, W, T) algorithm along with PK and T. The algorithm outputs an encrypted index CT_W.
- QueryGen(PK, SK_s, w): A user runs the QueryGen(PK, SK_s, w) algorithm to make a search query for the documents containing the word w and for which his key SK satisfies the decryption policy. The algorithm outputs a trapdoor tw generated for w.
- Search(tw, CT_W): The Search(tw, CT_W) algorithm takes as input the trapdoor sent by the user in query and encrypted index associated with a document. The algorithm returns true if the word in the user query matches with the word in the index of document and user's key satisfies the access policy of the document. The receiver (data retriever) gets the encrypted documents in result of his query for which the search procedure returns true.
- Decryption(CT,SK_d): With Decryption(CT,SK_d) algorithm the user performs the decryption for the resultant documents CT which he has retrieved from the cloud in result of his search query.

3.4 Detailed Construction of A^{2}BSE

Setup(1^l). Attribute Center chooses a security parameter l which determines key length, and follows the following steps to generate system keys and public parameters.

(i) choose two multiplicative cyclic groups G_0 and G_1 with a prime order p where length of p is determined by the security parameter l.
(ii) select g_1, g_2 as two generators of group G_0 and define a bilinear mapping $e : G_0 \times G_0 \rightarrow G_1$.
(iii) define two hash functions $H_0 : \{0,1\}^* \rightarrow Z_p$ and $H_1 : \{0,1\}^* \rightarrow G_0$
(iv) choose m+4 random exponents $\{\alpha, \beta, y, \vartheta, r_1, r_2, \cdots, r_m\}$. Here, m is maximum of m_i for $(1 \leq i \leq n)$. These elements serves as private key of the system.
(v) publish the public key as $\langle\ g_1, g_2, g_1^y, g_2^{\vartheta}, e(g_1,g_2)^y, e(g_1,g_2)^{\alpha}, \{g_2^{\frac{\alpha}{\beta}}, g_2^{r_1}, g_2^{r_2}, \cdots, g_2^{r_m}\}\ \rangle$.

Key Generation(MK,L). Each user in the system will get two keys representing the attributes the user possesses. One key, known as the search key, is used to generate the search query and the second key, known as the decryption key, is used for decrypting the ciphertexts. Let the user possess an j^{th} value $v_{i,j}$ for an attribute i, $1 \leq i \leq n$. The attribute center chooses two random r and ρ, and generates user's keys as follows.

Search Key:

- $D_{s0} = g_1^{r\beta}$
- $\{D_{s,i1} = g_1^{(H_0(i\|v_{i,j})+r)\frac{\alpha}{r_1}}, D_{s,i2} = g_1^{(H_0(i\|v_{i,j})^2+r)\frac{\alpha}{r_2}}, \cdots, D_{s,im_i} = g_1^{(H_0(i\|v_{i,j})^{m_i}+r)\frac{\alpha}{r_{m_i}}}\}$ $_{1\leq i\leq n}$ $(v_{i,j} \in L)$

Decryption Key :

- $D_d = (g_2 \prod_{i=1}^{n} H_1(i\|v_{i,j}))^y \cdot g_2^{\rho}$, where $v_{i,j} \in L$.
- $\bar{D}_d = g_1^{\frac{\rho}{\vartheta}}$

Encryption(PK,M,T). Let T =$\{T_1, T_2, \cdots, T_n\}$ where T_i $\{1 \leq i \leq n\}$ is the set of values for an attribute i, which are permissible for decryption. When a sender wants to send a document M in encrypted form to a set of users with specific set of attributes, he encrypts the document by following steps.

(i) choose two random secret values s, $t \in \mathbb{Z}_p$.
(ii) make n portions of t as t_i such that $\sum_{i=1}^{n} t_i = t$. For attribute values included in T_i, he creates the cipher components as

$\tilde{C_{i,j}} = g_2^{t_i} H_1(i\|v_{i,j})^{st}$, if $v_{i,j} \in T_i$
$\tilde{C_{i,j}}$ is a random value, if $v_{i,j} \notin T_i$.

(iii) $\hat{C} = g_1^{st}$,$\bar{C} = g_2^{st\vartheta}$ are calculated.
(iv) The document M is encrypted as $C' = M \cdot e(g_1, g_2)^{y(s-1)t}$.

The encrypted document is a tuple $\langle$ C', $\hat{C}$, $\bar{C}$,$\{\{\tilde{C_{i,j}}\}_{1\leq j\leq m_i}\}_{1\leq i\leq n}$ $\rangle$.

Encrypt_Index(PK,W,T). The set of words W extracted from M is given as input to this algorithm along with PK and T. The output of this algorithm is an encrypted index CT_W. The sender

(i) chooses a random secret values s' from Z_p.
(ii) generates s'_1,s'_2,$\cdots$,s'_{n-1} for 1$\leq$ i $\leq$n and calculates s'_n= s' - $\sum_{i=1}^{n-1} s'_i$
(iii) For every attribute field i, the sender chooses random values a'_i for 1$\leq$i$\leq$ n and then computes:

$$f(x_i) = a'_i(x_i - H_0(i\|v_{i,1}))(x_i - H_0(i\|v_{i,2})) \cdots (x_i - H_0(i\|v_{i,m_i})) + s'_i \quad (1)$$

where $v_{i,j}$=jth value of attribute i if it is included in T_i, else it will be a random value. The resultant equation is

$$f(x_i) = a'_{i0} + a'_{i1}x + a'_{i2}x^2 + \cdots + a'_{im_i}x^{m_i}. \quad (2)$$

The matched attribute value when supplied to the equation (2), it gives the value s'_i, else it gives any random value. Summation of all coefficients except a'_{i0} from all equations will be calculated as $A_w = \sum_{i=1}^{n} (\sum_{j=1}^{m_i} a'_{ij})$.
The sender now computes and sends the following values to cloud server : C_{w1}=$g_2^{\frac{A_w \alpha}{\beta}}$, $\{C_{wi1} = g_2^{a'_{i1}r_1}, C_{wi2} = g_2^{a'_{i2}r_2}, \cdots, C_{wim_i}$=$g_2^{a'_{im_i}r_{m_i}}\}$ for $1 \leq i \leq n$.
(iv) For each of the word w from set W an entry in the index is formed as $C_w = e(g_1, g_2)^{(s' - \sum a'_{i0})\alpha H_0(w_p)}$

The algorithm returns $\langle$ $\{C_w \mid, \forall\ w \in W\ \}$, $C_{w1}, \{\{C_{wij}\}_{1\leq j\leq m_i}\}_{1\leq i\leq n}$ $\rangle$ as the output.

QueryGen(PK,SK_s,w). In order to search for the documents containing word w, a user prepares his search query with the following two parameters.

- $D'_{s0} = g_1^{r\beta H_0(w)}$
- $\{D'_{s,i1} = g_1^{(H_0(i||v_{i,j})+r)\frac{\alpha}{r_1}H_0(w)}, D'_{s,i2} = g_1^{(H_0(i||v_{i,j})^2+r)\frac{\alpha}{r_2}H_0(w)}, \cdots, D'_{s,im_i} = g_1^{(H_0(i||v_{i,j})^{m_i}+r)\frac{\alpha}{r_{m_i}}H_0(w)}\}_{1\leq i\leq n}$

The outputs of QueryGen(PK,SK_s,w) is the trapdoor tw, which will be used in the Search algorithm.

Search(tw,CT_W). After receiving a search query, the cloud server initiates following procedure to find a match for the word and access policy.

$$R_{s1} = \prod_{i=1}^{n}(\prod_{j=1}^{n_i} e(C_{wij}, D'_{s,ij})) \tag{3}$$

$$\begin{aligned} R_{s2} &= e(C_{1w}, D'_0) = e(g_1, g_2)^{A_w \alpha r H_0(w)} \\ R_s &= \frac{R_{s1}}{R_{s2}} = e(g_1, g_2)^{(s' - \sum a'_{i0})\alpha H_0(w)} \end{aligned} \tag{4}$$

We note that the simplification of equation (3), as illustrated below, makes it easy to understand why the cloud server computes R_{s2} and R_s:

$$\begin{aligned} R_{s1} &= \prod_{i=1}^{n}(\prod_{j=1}^{n_i} e(g_1^{(H_0(v_i)^j + r_1)\frac{\alpha H_0(w)}{r_j}}, g_2^{a'_{ij} r_j})) \\ &= \prod_{i=1}^{n}(e(g_1, g_2)^{(\sum_{j=1}^{m_i} a'_{ij} H_0(v_i)^j)\alpha H_0(w)} \cdot e(g_1, g_2)^{\sum_{j=1}^{m_i} a'_{ij} r \alpha H_0(w)}) \\ &= (g_1, g_2)^{\sum_{i=1}^{n}(s'_i - a'_{i0})\alpha H_0(w)} \cdot e(g_1, g_2)^{A_w \alpha r H_0(w)} \\ &= e(g_1, g_2)^{(s' - \sum(a'_{i0}))\alpha H_0(w)} \cdot e(g_1, g_2)^{A_w \alpha r H_0(w)} \end{aligned}$$

From the encrypted index of the corresponding document the server searches for a match with result of equation (4). If the match is found then the algorithm returns true and sends the document's ciphertext for further decryption procedure Decryption(CT,SK_d).

Decryption(CT,SK_d). The user performs the decryption procedure for the encrypted documents which he received as a result of his search query from the cloud server with the following computation.
User selects $\tilde{C}_{i,j}$ elements, each related to $v_{i,j}$ value which is included in user's secret key.

$$\begin{aligned} &\frac{C' \cdot e(\prod_{i=1}^{n} \tilde{C}_{i,j}, g_1^{\alpha}) e(\bar{C}, \bar{D})}{e(\hat{C}, D)} \\ &= \frac{C' \cdot e(\prod_{i=1}^{n} (g_2^{t_i} H_1(i||v_{i,j})^{st}), g_1^{y}) e(g_2^{st\vartheta}, g_1^{\frac{\rho}{\vartheta}})}{e(g_1^{st}, (g_2 \prod_{i=1}^{n} H_1(i||v_{i,j}))^{y} \cdot g_2^{\rho})} \end{aligned}$$

$$= \frac{M \cdot e(g_1, g_2)^{(y(s-1))t}}{e(g_1, g_2)^{(y(s-1))t}}$$
$$= M$$

4 Security and Performance Analysis

4.1 Adversary's Goal and Assumptions

Adversary's goal of the A^{2}BSE comprises the following:

- Retrieve the plaintext from ciphertext using a single or combination of keys such that none of them alone can decrypt the ciphertext.
- Retrieve the information about underlying access policy.
- Gain the information about the word being search for.
- Know who are the target recipients.

The security of A^{2}BSE relies on strength of Decisional Linear(D-Linear), Decisional Bilinear Diffie-Hellman (DBDH) and Decisional Diffie-Hellman(DDH) assumptions. We divide the functionality of A^{2}BSE in two parts. The first part is for encryption of index and performing search operation over the encrypted index. The second part is for encryption and decryption of a document. It is required that the encrypted message does not reveal any information about the ciphertext and underlying access policy to an adversary. We prove this claim by the security of *Indistinguishability against selective ciphertext-policy and chosen plaintext attack (IND-sCP-CPA)* model. We show that unless a correct decryption key is available, the ciphertext is indistinguishable from any other group element. Then, we prove that given an encrypted index and a search token the adversary can only gain information whether the search is successful by *Indistinguishability against selective ciphertext-policy and chosen keyword attack (IND-sCP-CKA)* model.

IND-sCP-CPA Model. We consider this security model as a game between a Challenger and an Adversary as follows.

Init: The adversary commits two ciphertext access policies T_0^* and T_1^* for which he wishes to be challenged upon.

Setup: The challenger gives l as the security parameter to run the Setup algorithm and retrieves the master secret key MK and public parameters PK. The public parameters PK is sent to the adversary.

Phase 1: The adversary is allowed to request for polynomially bounded number of secret keys on attributes L with the restriction that $\mathrm{F}(L, T_0^*)=\mathrm{F}(L, T_1^*)=0$ or $\mathrm{F}(L, T_0^*)=\mathrm{F}(L, T_1^*)=1$.

Challenge: The adversary outputs two messages M_0 and M_1 on which he wishes to be challenged upon with respect to the challenge access policy T_0^* and T_1^*. Here, the restriction is that if for any key generated in Phase 1 on an attribute list L $\mathrm{F}(L,T_0^*)= \mathrm{F}(L,T_1^*)=1$, then $M_0 = M_1$. The challenger randomly chooses

$b \in \{0,1\}$ and computes CT_b as output of Encryption(PK,M_b,T_b^*). The CT_b is sent to the adversary.

Phase 2: Same as in Phase 1. The adversary issues polynomially bounded number of queries for generating secret keys with the restriction as specified in Phase 1.

Guess: The adversary outputs a guess b' of b. The adversary wins the game if $b' = b$. The advantage of an adversary $\mathcal{A}$ in the game is defined as $\text{Adv}_{\mathcal{A}}(l)= |Pr[b' = b] - 1/2|$.

Theorem 1. *The A^2BSE scheme is IND-sCP-CPA secure under the D-Linear assumption if there is no polynomial time adversary $\mathcal{A}$ who can win the above mentioned game with non-negligible advantage $Adv_{\mathcal{A}}(l)$ in security parameter l.*

Proof. We consider a challenger $\mathcal{C}$, a simulator $\mathcal{S}$ and a polynomial-time adversary $\mathcal{A}$. Suppose that the adversary is able to distinguish a valid ciphertext from a random element with advantage $\epsilon_1(l)$. We build a simulator $\mathcal{S}$ that can play the D-Linear game with advantage $\frac{\epsilon_1(l)}{2}$. The simulation proceeds as follows.
Let the challenger set the groups G_0 and G_1 with an efficient bilinear map, e and generator g. The challenger flips a fair binary coin b, outside of $\mathcal{S}$'s view. If $b = 0$, the challenger sets $(g, Z_1, Z_2, Z_3, Z_4, Z) = (g, g^{z_1}, g^{z_2}, g^{z_2 z_4}, g^{z_3+z_4}, g^{z_1 z_3})$, otherwise it sets $(g, Z_1, Z_2, Z_3, Z_4, Z) = (g, g^{z_1}, g^{z_2}, g^{z_2 z_4}, g^{z_3+z_4}, g^{z})$ for values z_1, z_2, z_3, z_4 and z chosen randomly from $\mathbb{Z}_p$.

Init: The simulator $\mathcal{S}$ runs $\mathcal{A}$. $\mathcal{A}$ commits two access policies T_0^* and T_1^* for which he wishes to be challenged upon.

Setup: $\mathcal{S}$ takes the following values: $g_1 = g^a$, $g_2 = g$, $g_1^y = g^{ay}$ with the assumption that $y = z_1$. Here a is chosen randomly from $\mathbb{Z}_p$. With the selection of random value ϑ from $\mathbb{Z}_p$, g_2^{ϑ} is calculated as g^{ϑ}. $H_1(i\|v_{i,j})$ is assumed as $g^{\frac{1}{nz_1}+H'(i\|v_{i,j})}$. Here n denotes the number of attribute categories and $H'(i||v_{i,j})$ is computed from random oracle. The simulator $\mathcal{S}$ announces the public key as $g_1 = g^a$, $g_2 = g$, $g_1^y = g^{az_1}$, $g_2^{\vartheta} = g^{\vartheta}$, $\text{e}(\text{g}_1,\text{g}_2)^y = \text{e}(\text{g},\text{g})^{az_1}$.

Preprocessing Phase: The attacker $\mathcal{A}$ collects following results in response of his queries made to the simulator.

– Whenever $\mathcal{A}$ makes its k^{th} key generation query for the set L_k of attributes such that $\text{F}(L_k,T_0^*)= \text{F}(L_k,T_1^*)=0$. The simulator $\mathcal{S}$ selects a random value $\rho \in \mathbb{Z}_p$ and the key components are generated as.

$$
\begin{aligned}
D &= (g_2 \prod_{i=1}^{n} H_1(i||v_{i,j_i}))^y \cdot g_2^{\rho} \\
&= (g \prod_{i=1}^{n} g^{\frac{1}{nz_1}+H'(i\|v_{i,j_i})})^{z_1} \cdot g^{\rho} \\
&= g^{z_1} \cdot g^{1+\sum_{i=1}^{n}(H'(i\|v_{i,j_i})z_1)} \cdot g^{\rho} \\
&= Z_1 \cdot g \cdot Z_1^{\sum_{i=1}^{n}(H'(i\|v_{i,j_i}))} \cdot g^{\rho} \\
\bar{D} &= g_1^{\frac{\rho}{\vartheta}} = g^{\frac{a\rho}{\vartheta}}
\end{aligned}
$$

– In response to the query for ciphertext of messages M^* as per the access policies T^* (where $T^* \neq T_0^* \neq T_1^*$), submitted by $\mathcal{A}$, the simulator $\mathcal{S}$ computes the ciphertext with the selection of random numbers s and t from $\mathbb{Z}_p$. The cipher components are generated with the public key parameters set by the simulator $\mathcal{S}$

Challenge: Let the two challenge ciphertext policies submitted by the adversary $\mathcal{A}$ are $T_0^* = [v_{0,1}, v_{0,2}, \cdots, v_{0,n}]$ and $T_1^* = [v_{1,1}, v_{1,2}, \cdots, v_{1,n}]$. The adversary outputs two messages M_0 and M_1 on which he wishes to be challenged upon with respect to the challenge access policy T_0^* and T_1^*. If for any key generated in Phase 1 on an attribute list L F(L,T_0^*)= F(L,T_1^*)=1, then $M_0 = M_1$.
Now $\mathcal{S}$ flips a random coin v, and encrypts M_v as per access policy T_v^*. $\mathcal{S}$ assumes st=$z_1 z_3$, with the assumptions that $s = \frac{z_1 z_3}{z_4(z_2+1)}$ and $t = z_4(z_2 + 1)$. For the values which are included in T_v^*, $\mathcal{C}$ calculates $\tilde{C}_{i,j} = g_2^{t_i} H_1(i||v_{i,j_i})^{st} = g^{\frac{z_2 z_4 + z_4}{n}} g^{(\frac{1}{n z_1} + H'(i||v_{i,j_i}))z_1 z_3} = Z_4^{\frac{1}{n}} Z_3^{\frac{1}{n}} Z^{H'(i||v_{i,j_i})}$. For other attribute values which are not included in T_v^*, $\tilde{C}_{i,j}$ are random values. Cipher components are calculated by $\mathcal{S}$ as $\hat{C} = g_1^{st} = Z^a$, $\bar{C} = g_2^{st\beta} = Z^\beta$ and $C' = M_v \cdot \frac{e(Z_1^a, Z) e(g, Z)}{e(Z_1^a, Z_4) e(Z_1, Z_3)}$. C' is the correct cipher component only if $Z = g^{z_1 z_3}$. Else, C' is a random element. The ciphertext components $\{\{\tilde{C}_{i,j}\}_{1 \leq j \leq m_i}\}_{1 \leq i \leq n}, \bar{C}, \hat{C}, C'$ are given to $\mathcal{A}$.

Post Processing Phase: $\mathcal{A}$ is allowed to run a number of queries for attribute keys and matching phase components with the same conditions as imposed in the preprocessing phase.

Guess: $\mathcal{A}$ submits a guess v' of v. If $v' = v$, then $\mathcal{S}$ outputs $b = 1$ to indicate that it was given a valid D-Linear tuple, else, it outputs $b = 0$ to indicate that the ciphertext is a random element. Therefore, $\mathcal{A}$ gains no information about v, in turn, $Pr[v \neq v' | b = 0] = \frac{1}{2}$. As the simulator guesses b'=0 when $v \neq v'$, $Pr[b = b' | b = 0] = \frac{1}{2}$. If $b = 1$, then $\mathcal{A}$ is able to view the valid encryption components with advantage $\epsilon_1(l)$, a negligible quantity in security parameter in l. Therefore, $Pr[v = v' | b = 1] = \frac{1}{2} + \epsilon_1(l)$. Similarly, the simulator $\mathcal{S}$ guesses b'=1 when $v = v'$, in turn, $Pr[b' = b | b = 1] = \frac{1}{2} + \epsilon_1(l)$. The overall advantage of the simulator in D-Linear game is $\frac{1}{2} \times Pr[b = b' | b = 0] + \frac{1}{2} \times Pr[b = b' | b = 1]$ - $\frac{1}{2} = \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times (\frac{1}{2} + \epsilon_1(l))$ - $\frac{1}{2} = \frac{\epsilon_1(l)}{2}$.
Therefore, if the $\mathcal{A}$ has a non-negligible advantage $\epsilon_1(l)$ in the above game then we can build a simulator (S) which can break the D-Linear problem with non-negligible quantity $\frac{\epsilon_1(l)}{2}$, which is an intractable problem.
Hence, the theorem. □

Now we prove that the cipher components provides receiver anonymity. We prove this even if an adversary gains a valid decryption key, he will be able to decrypt the message, but can not figure out the underlying access policy. This implies, even if any user decrypts a message successfully with his key, he is not able to find out who else are the other intended recipients for the same message.

Theorem 2. *The A^2BSE provides receiver anonymity in IND-sCP-CPA game under the DDH assumption if there is no polynomial time adversary $\mathcal{A}$ who can distinguish a valid ciphertext and a random element with non-negligible advantage $Adv_{\mathcal{A}}(l)$ in security parameter l.*

Proof. The challenger flips a binary coin μ outside of $\mathcal{S}$ view and assigns a tuple (g,$A = g^a$,$B = g^b$, Z) to $\mathcal{A}$. If $\mu = 1$ then the challenger sets Z as g^{ab} else a random value with equal probability.

Init : The simulator $\mathcal{S}$ runs $\mathcal{A}$. $\mathcal{A}$ commits two access policies T_0^* and T_1^* for which he wishes to be challenged upon. In T_0^* and T_1^* for one attribute λ, $T_{0,\lambda}^* \neq T_{1,\lambda}^*$. There is at least one value $v_{\lambda,r}$ is from value set of attribute λ, such that $v_{\lambda,r} \notin T_{0,\lambda}^*$ and $v_{\lambda,r} \in T_{1,\lambda}^*$. Here, $1 \leq r \leq m_\lambda$. For rest of the attributes $T_{0,i}^* = T_{1,i}^*$ where $1 \leq i \leq n$ and $i \neq \lambda$

Setup: $\mathcal{S}$ takes the following values: $g_1 = g$, $g_2 = g^{x_2}$. Here x_2 is chosen randomly from Z_p. $H(i\|v_{i,j})$ is computed as an output of an random oracle. Another two random secret values are chosen as y and ϑ. The simulator $\mathcal{S}$ announces the public key as : $g_1 = g$, $g_2 = g^{x_2}$, $g_1^y = g^{x_1 y}$, $g_2^\vartheta = g^{x_2 \vartheta}$, $e(g_1,g_2)^y = e(g,g)^{x_2 y}$.

Phase 1 : The attacker $\mathcal{A}$ collects following results in response of his queries made to simulator.

- Whenever $\mathcal{A}$ makes its k^{th} key generation query for the set L_k of attributes such that $F(L_k , T_0^*) = F(L_k,T_1^*)$. That is, $\mathcal{A}$ is allowed to issue a valid decryption key which can decrypt the challenge ciphertext, with the restriction that the key should be able to satisfy both the challenge access structure T_0^* and T_1^*. The simulator $\mathcal{S}$ selects a random value ρ and the key components are generated as follows.

$$D = (g_2 \prod_{i=1}^{n} H_1(i\|v_{i,j_i}))^y \cdot g_2^\rho = (g^{x_2} \cdot g^{H_1'(i\|v_{i,j})})^y \cdot g^{x_2 \rho}$$
$$\bar{D} = g_1^{\frac{\rho}{\vartheta}} = g^{\frac{\rho}{\vartheta}}$$

- In response to the query for ciphertext of messages M^* as per the access policies T^* (where $T^* \neq T_0^* \neq T_1^*$), submitted by $\mathcal{A}$, the simulator $\mathcal{S}$ computes the ciphertext with the selection of random numbers s and t from Z_p. The cipher components are generated with the public key parameters set by the simulator $\mathcal{S}$
- The attacker issues the query for $H_1(i\|v_(i,j))$. For all the attribute values except $v_{\lambda,r}$, The simulator S runs the random oracle function and provides as output an element from G_0 . The simulator S records the queries and its outputs. So that if any query is repeated by attacker then the same result as given in previous query is given in output. For an query for $H_1(\lambda||v_{\lambda,r})$ the output returned is $B = g^b$

Challenge : Then $\mathcal{S}$ flips a random coin ν.

- $\mathcal{A}$ submits two messages M_0 and M_1 . If the adversary has retrieved a key for any queried L=[L_1, L_2, $\cdots$, L_n], such that $F(L,T_0^*) = F(L,T_1^*) = 1$, then $M_0 = M_1$
- $\mathcal{S}$ sets st as a and t is selected as any random value chosen from Z_p. This results in $A = g^a = g_1^{st}$ = and $A^{x_2\vartheta}$= $g^{x_2 a\vartheta} = g_2^{st\vartheta}$.
- The simulator $\mathcal{S}$ generates n shares of value t and use each share t_i for encrypting the values for attribute i. For all the cipher components $T_{\nu,i}^*$ where $1 \leq i \leq n$ and $i \neq \lambda$ the cipher components $\tilde{C_{i,j}}$ are generated as $g_2^{t_i} H_1(i||v_{i,j})^{st} = g^{x_2 t_i} g^{H_1'(i||v_{i,j})a}$.
- For values $v_{\lambda,j}$ of attribute $T_{\nu,\lambda}^*$ which are included in both $T_{0,\lambda}$ and $T_{1,\lambda}$,, the cipher components $\tilde{C_{\lambda,j}}$ are generated as in the real scheme using the value of t_λ.
- For the value $v_{\lambda,r}$ which makes a differentiation between $T_{0,\lambda}$ and $T_{1,\lambda}$, the cipher component $\tilde{C_{\lambda,r}}$ is calculated as follows.
  - If $\nu = 0$ then $\tilde{C_{\lambda,r}}$ will be a random value. This is valid because $\tilde{C_{\lambda,r}}$ is not in $T_{0,\lambda}$ as per definition.
  - If $\nu = 1$ then $\tilde{C_{\lambda,r}}$ is set as $g^{x_2 t_\lambda} Z$. Here we have taken the output of $H_1(\lambda||v(\lambda,j))$ from random oracle as g^b. If Z is a valid element with value g^{ab} then $\tilde{C_{\lambda,r}}$ will be a correct element else it will be a random element.
- C' is calculated as e(g,g)^{ax_2y}.

The adversary is given ciphertext components $\langle$ C', $\hat{C}$, $\bar{C}$,$\{\{\tilde{C_{i,j}}\}_{1\leq j\leq m_i}\}_{1\leq i\leq n}$ $\rangle$. At the end of the challenge phase the adversary uses following values $\langle$ g_1^{st} = g^a, $H_1(\lambda||v(\lambda,r))$= g^b, $C_{\lambda,r}$ $\rangle$ to find out the underlying access policy. If $C_{\lambda,r}$ is a correct cipher component then it represents the value $g_2^{t_\lambda} \cdot g^{ab}$ else it is a random value.

Phase 2: $\mathcal{A}$ is allowed to run a number of queries for attribute keys and ciphertext phase components with the same conditions as imposed in the preprocessing phase.

Guess:$\mathcal{A}$ submits a guess ν' of ν. If $\nu' = \nu$, then $\mathcal{S}$ outputs μ=1 to indicate that it was given a valid DDH-tuple, else it outputs μ=0 to indicate that the ciphertext is a random element. Therefore, $\mathcal{A}$ gains no information about ν, in turn, Pr[$\nu \neq \nu'$ — μ=0] = $\frac{1}{2}$. As the simulator guesses μ'=0 when $\nu \neq \nu'$, Pr[μ = μ' — μ=0] = $\frac{1}{2}$. If $\mu = 1$, then the adversary $\mathcal{A}$ is able to view a valid cipher components with advantage $\epsilon_2(l)$, a negligible quantity in security parameter in l. Therefore, $Pr[\nu = \nu'|\mu = 1] = \frac{1}{2} + \epsilon_1(l)$. Similarly, the simulator $\mathcal{S}$ guesses μ'=1 when $\nu = \nu'$, in turn, $Pr[\mu' = \mu|\mu = 1] = \frac{1}{2} + \epsilon_2(l)$. The overall advantage of the simulator in DDH game is $\frac{1}{2}$ × Pr[$\mu = \mu'$ — μ=0] + $\frac{1}{2}$ × Pr[$\mu = \mu'$ — μ=1] - $\frac{1}{2}$ = $\frac{1}{2}$ × $\frac{1}{2}$ + $\frac{1}{2}$ ×($\frac{1}{2}$ + $\epsilon_2(l)$) - $\frac{1}{2}$ = $\frac{\epsilon_2(l)}{2}$.
Therefore, if the $\mathcal{A}$ has a non-negligible advantage $\epsilon_2(l)$ in the above game then we can build a simulator (S) which can break the DDH problem with non-negligible quantity $\frac{\epsilon_2(l)}{2}$.
Hence, the theorem. □

Theorem 3. *The A^{2}BSE is IND-sCP-CKA secure under DBDH assumption if there is no polynomial time adversary can win the game, stated in Appendix, with nonnegligible advantage $Adv_A(l)$ in terms of security parameter l.*

Proof. The proof of the Theorem 3 is provided in Appendix.

4.2 Performance Analysis

The Table 1 shows a comparison of A^{2}BSE scheme with two related schemes [15], [19]. The comparison shows that A^{2}BSE scheme is efficient than [15], [19].

Schemes / Parameters	Li *et al* [15]	Zhang *et al* [19]	A^{2}BSE
No. of User key components	O(n)	O(n)	O(1)
No. of Cipher components	O(mn)	O(mn)	O(mn)
N. of Bilinear Mapping Operations	O(n)	O(n)	O(1)

Table 1. Efficiency of A^{2}BSE Scheme

Here, n represents the number of attributes in the system and m is the maximum value of m_i, where m_i denotes the number of values for i^{th} attribute ($1 \leq i \leq$ n).

5 Conclusion

The anonymous ABE provides an interesting security feature receiver anonymity in addition to data confidentiality and fine-grained access control of ABE. While storing encrypted documents in public cloud, an efficient search functionality facilitates user to retrieve a subset of documents for which the user has access rights on stored documents. We proposed an anonymous attribute based searchable encryption (A^{2}SBE) scheme which facilitates user to retrieve only a subset of documents pertaining to his chosen keyword(s). User can upload documents in public cloud in an encrypted form, search documents based on keyword(s) and retrieve documents without revealing his identity. The scheme is proven secure under the standard adversarial model. The scheme is efficient, as it requires small storage for user's decryption key and reduced computation for decryption in comparison to other schemes.

References

1. A. Sahai, B.Waters. Fuzzy Identity Based Encryption. In Proceedings of Advances in Cryptology-EUROCRYPT, LNCS 3494, Springer, pp. 457–473,2005.
2. J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In Proceedings of IEEE Symposium on Security and Privacy,pp. 321–334, 2007.

3. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the ACM Conference on Computer and Communications Security, pp. 89–98, 2006.
4. L. Cheung, C. Newport. Provably secure ciphertext policy ABE. In Proceedings of the ACM Conference on Computer and Communications Security, pp. 456–465, 2007.
5. V. Goyal, A. Jain, O. Pandey, A. Sahai. Bounded ciphertext policy attribute based encryption. In Proceedings of Automata, Languages and Programming, pp. 579–591, 2008.
6. A. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In Proceedings of Advances in Cryptology-EUROCRYPT, LNCS 6110, Springer, pp. 62–91, 2010.
7. T. Okamoto, K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In Proceedings of Advances in Cryptology-CRYPTO, LNCS 6223, Springer, pp. 191–208, 2010.
8. R. Ostrovsky, A. Sahai, B. Waters. Attribute-based encryption with non-monotonic access structures. In Proceedings of the ACM Conference on Computer and Communications Security, pp. 195–203, 2007.
9. B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Proceedings of Public Key Cryptography-PKC, LNCS 6571, Springer, pp. 53–70, 2011.
10. S. Yamada, N. Attrapadung, G. Hanaoka, N. Kunihiro. Generic constructions for chosen-ciphertext secure attribute based encryption. In Proceedings of Public Key Cryptography-PKC, LNCS 6571, Springer, pp. 71–89, 2011.
11. A. Kapadia, P. P. Tsang, and S. W. Smith. Attribute-based publishing with hidden credentials and hidden policies. In Proceedings of Network and Distributed System Security Symposium, pp. 179–192, 2007.
12. D. Boneh, B. Waters. Conjunctive, subset, and range queries on encrypted data. In Proceedings of Theory of Cryptography, LNCS 4392, Springer, pp. 535–554, 2007.
13. J. Katz, A. Sahai, B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Journal of Cryptology, 26(2): 191–224, 2008.
14. T. Nishide, K. Yoneyama, and K. Ohta. Attribute-based encryption with partially hidden encryptor-specified access structures. In Proceedings of Applied Cryptography and Network Security, LNCS 5037, Springer, pp. 111–129, 2008.
15. J. Li, K. Ren, B. Zhu, and Z. Wan. Privacy-aware attribute-based encryption with user accountability. In Proceedings of Information Security, LNCS 5735, Springer, pp. 347–362, 2009.
16. M. Chase, S. S. Chow. Improving privacy and security in multi-authority attribute-based encryption. In Proceedings of the ACM Conference on Computer and Communications Security, pp. 121–130, 2009.
17. S. Yu, K. Ren, W. Lou, J. Li. Defending against key abuse attacks in KP-ABE enabled broadcast systems. In Proceedings of Security and Privacy in Communication Networks, pp. 311–329, 2009.
18. S. Yu, C. Wang, K. Ren, W. Lou. Attribute based data sharing with attribute revocation. In Proceedings of the ACM Symposium on Information, Computer and Communications Security, pp. 261–270, 2010.
19. Y. Zhang, X. Chen, J. Li, D. S. Wong, and H. Li. Anonymous attribute-based encryption supporting efficient decryption test. In Proceedings of the ACM SIGSAC

Symposium on Information, Computer and Communications Security, pp. 511–516, 2013.
20. P. Chaudhari, M. L. Das, A. Mathuria. Security Weaknesses of an "Anonymous Attribute Based Encryption" appeared in ASIACCS'13. Cryptology ePrint Archive: Report 2014/1028, https://eprint.iacr.org/2014/1028.
21. Y. Shi, J. Liu, Z. Han, Q. Zheng, R. Zhang, S. Qiu. Attribute-based proxy re-encryption with keyword search. In: PloS One 9, no. 12: e116325, 2014.
22. J. Li, L. Zhang. Attribute-based keyword search and data access control in cloud. In Proceedings of International Conference on Computational Intelligence and Security, pp. 382–386, 2014.
23. P. Liu, W. Jianfneg, M. Hua, N. Haixin. Efficient verifiable public key encryption with keyword search based on KP-ABE. In Proceedings of International Conference on Broadband and Wireless Computing, Communication and Applications, pp. 584–589, 2014.
24. Q. Zheng, X. Shouhuai , G. Ateniese. VABKS: Verifiable attribute-based keyword search over outsourced encrypted data. In Proceedings of IEEE Conference on Computer Communications (INFOCOM), pp. 522–530, 2014.
25. C. Wang, W. Li, Y. Li, X. Xu. A ciphertext-policy attribute-based encryption scheme supporting keyword search function. In Proceedings of Cyberspace Safety and Security, LNCS 8300, Springer, pp. 377–386, 2013.
26. D. Koo, J. Hur, H. Yoon. Secure and efficient data retrieval over encrypted data using attribute-based encryption in cloud storage. In: Computers & Electrical Engineering, Elsevier, 39:34–46, 2013,
27. J. Shi, J. Lai, Y. Li, R. H. Deng, J. Weng. Authorized keyword search on encrypted data. In Proceedings of Computer Security-ESORICS, LNCS 8712, Springer, pp. 419-435, 2014.

Appendix

IND-sCP-CKA Game

Init : The adversary commits to two ciphertext access policies T_0^* and T_1^* for which he wishes to be challenged upon.

Setup : The challenger gives l as security parameter to run the setup algorithm and retrieve master secret key MK and public parameters PK. The public parameters PK is sent to the adversary.

Phase 1: The adversary is allowed to request for polynomially bounded number of secret keys on attributes L with the restriction that $\mathrm{F}(L, T_0^*) = \mathrm{F}(L, T_1^*) = 0$ or $\mathrm{F}(L, T_0^*) = \mathrm{F}(L, T_1^*) = 1$.

Challenge : The Adversary outputs two words w_0 and w_1 on which he wishes to be challenged upon with respect to challenge access policy T_0^* and T_1^*. Here, the restriction is that if for any key generated in phase 1 on an attribute list L $\mathrm{F}(L, T_0^*) = \mathrm{F}(L, T_1^*) = 1$ then $\mathrm{w}_0 = \mathrm{w}_1$. The challenger randomly chooses $\mathrm{b} \in \{0,1\}$ and computes CT_b as output of Encryption(PK, w_b, T_b^*). The CT_b is sent to the adversary

Phase 2: Same as in Phase1 the adversary issue polynomially bounded number of queries for generating secret keys with the restriction as specified in Phase 1.

Guess: The adversary outputs a guess b' of b. The adversary wins the game if $b' = b$. The advantage of an adversary A in this game is defined as $\mathrm{Adv}_A(\mathrm{l})= |Pr[b' = b] - 1/2|$.

The A^2BSE is IND-sCP-CKA secure under the DBDH assumption.

Proof. The adversary commits to the challenge ciphertext policies $T_0^* = [T_{0,1}, T_{0,2}, \cdots, T_{0,n}]$ and $T_1^* = [T_{1,1}, T_{1,2}, \cdots, T_{1,n}]$. In first section the adversary obtains the sk_L with respect to attribute list L satisfying the condition of $F(L, T_0^*) = F(L, T_1^*) = 0$.

We show that if an adversary is able to distinguish an encrypted word from a random element with advantage $\epsilon_1(l)$, then we can build a simulator $\mathcal{S}$ which can break the DBDH problem with advantage $\frac{\epsilon_1(l)}{2}$

First the challenger sets the group G_0 and G_1. Then the challenger flips a binary coin μ outside of $\mathcal{S}$ view. If $\mu = 0$ then the challenger sets $(g, A, B, C, Z) = (g, g_a, g_b, e(g,g)^{abc})$. Else the challenger sets $(g, A, B, C, Z) = (g, g_a, g_b, g_c, e(g,g)^z)$ for some random value $a \in \mathbb{Z}_p$

Init: $\mathcal{S}$ initiates $\mathcal{A}$. The $\mathcal{A}$ gives $\mathcal{S}$ two challenge access policies T_0^* and T_1^* before setup algorithm.

Setup: $\mathcal{S}$ assumes $g_2 = B$ and $g_1 = A$. Also $\mathcal{S}$ will choose following random values $\alpha, \beta, r, r_1, r_2, \cdots, r_m \in \mathbb{Z}_p$.

Phase 1: The adversary requests for polynomially bounded number of secret keys on attributes $L = \{L_1, L_2, \cdots, L_n\} = \{v_{1,p1}, v_{2,p2}, \cdots, v_{n,pn}\}$ with the restriction that $\mathrm{F}(L, T_0^*) = \mathrm{F}(L, T_1^*) = 0$. $\mathcal{S}$ provide keys to $\mathcal{A}$ with following computation $\{D_0 = g_1^{r\beta} = \mathrm{A}^{r\beta}, \{ \{ \mathrm{D}_{ij} = \mathrm{g}_1^{(H(i||v_{i,j})^j + r)\frac{\alpha}{r_j}} = \mathrm{AP}^{(H(i||v_{i,j})^j + r)\frac{a}{r_j}} \}_{1 \le j \le m_i} \}_{1 \le i \le n}\}$.

Challenge: $\mathcal{A}$ submits two challenge keywords w_0 and w_1. Consider z as the secret value used to encrypt the keyword. $\mathcal{S}$ flips a random coin $b \in \{0,1\}$. The simulator $\mathcal{S}$ computes the challenge ciphertext with following values.

- For $1 \le i \le n$-1 select a_i, z_i and build the equations for each attribute category as follows :

$$f(x_i) = a_i(x - H_0(\hat{v}_{ii}))(x - H_0(\hat{v}_{i2})) \cdots (x - H_0(\hat{v}_{im_i})) + z_i \tag{5}$$

$$f(x_i) = a_{i0} + a_{i1}x + a_{i2}x^2 + \cdots a_{im_i}x^{m_i} \tag{6}$$

  where in equation 5 $\hat{v_{ij}} = v_{ij}$ if $v_{ij} \in T_b$ else if $v_{ij} \notin T_b$ then $\hat{v_{ij}}$ is some random value chosen from Z_p for $1 \le j \le m_i$.

  $\mathcal{S}$ computes $C_{i1} = B^{\frac{a_{i1}}{r_1}} = g_2^{\frac{a_{i1}}{r_1}}$, $C_{i2} = B^{\frac{a_{i2}}{r_2}} = g_2^{\frac{a_{i2}}{r_2}}$, $\cdots$, $C_{im_i} = B^{\frac{a_{im_i}}{r_{m_i}}} = g_2^{\frac{a_{im_i}}{r_{m_i}}}$ for $1 \le i \le n$-1

- For the n^{th} attribute category choose random value $a_n \in Z_p$ and compute the following equation

$$f(x_i) = a_n(x - H_0(\hat{v}_{ni}))(x - H_0(\hat{v}_{n2})) \cdots (x - H_0(\hat{v}_{nm_n})) \tag{7}$$

  Again $\hat{v_{nj}} = v_{nj}$ if $v_{nj} \in T_b$, else $\hat{v_{nj}}$ is some random value chosen from Z_p for $1 \le j \le m_n$. $\mathcal{S}$ computes $e(g_1,g_2)^{\acute{a}_{n0}}$, $C_{n1} = B^{\frac{a_{n1}}{r_1}} = g_2^{\frac{a_{n1}}{r_1}}$, $C_{n2} = B^{\frac{a_{n2}}{r_2}} = g_2^{\frac{a_{n2}}{r_2}}$, $\cdots$, $C_{nm_n} = B^{\frac{a_{nm_n}}{r_{m_n}}} = g_2^{\frac{a_{nm_n}}{r_{m_n}}}$.

– Finally $e(g_1,g_2)^{a_{n0}}$ is computed as follows :

$$\begin{aligned} e(g_1,g_2)^{a_{n0}\alpha} &= e(g_1,g_2)^{\acute{a}_{n0}\alpha} e(g_1,g_2)^{z_n\alpha} \\ &= e(g_1,g_2)^{\acute{a}_{n0}\alpha} \frac{Z^{\alpha}}{\prod_{i=1}^{n-1} e(g_1,g_2)^{z_i\alpha}} \end{aligned} \tag{8}$$

– Compute $C_{w1} = B^{\frac{A_w\alpha}{\beta}} = g_2^{\frac{A_w\alpha}{\beta}}$. Here, $A_w = \sum_{i=1}^{n} (\sum_{j=1}^{m_i} a_{ij})$

– Compute $C_{w_b} = \frac{Z^{H_0(w_b)}}{e(A,B)^{\sum a_{i0}\alpha H_0(w_b)}}$

Now $\mathcal{S}$ gives ciphertext as C_{w_b},C_{w1}, $\{C_{i1}$,C_{i2} , $\cdots$,$C_{im_i}\}$*for* $1 \leq i \leq n$ to $\mathcal{A}$.
Phase 2: $\mathcal{A}$ may repeat the private key queries as it did in Phase 1.
Guess : $\mathcal{A}$ outputs a guess b' of b. If $b' = b$, then $\mathcal{S}$ outputs μ=1 to indicate that it was given a valid DBDH-tuple, else it outputs μ=0 to indicate that the ciphertext is a random element. Therefore, $\mathcal{A}$ gains no information about b, in turn, $Pr[b \neq b'|\mu = 0] = \frac{1}{2}$. As the simulator guesses μ'=0 when $b \neq b'$, $Pr[\mu = \mu'|\mu = 0] = \frac{1}{2}$. If $\mu = 1$, then the adversary $\mathcal{A}$ is able to view a valid encryption of message with advantage $\epsilon_1(l)$, a negligible quantity in security parameter l. Therefore, $Pr[b = b'|\mu = 1] = \frac{1}{2} + \epsilon_1(l)$. Similarly, the simulator $\mathcal{S}$ guesses μ'=1 when $b = b'$, in turn, $Pr[\mu' = \mu|\mu = 1] = \frac{1}{2} + \epsilon_1(l)$. The overall advantage of the simulator in DBDH game is $\frac{1}{2} \times Pr[\mu = \mu'|\mu = 0]$ $+\frac{1}{2} \times Pr[\mu = \mu'|\mu = 1]$ - $\frac{1}{2} = \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times (\frac{1}{2} + \epsilon_1(l))$ - $\frac{1}{2} = \frac{\epsilon_1(l)}{2}$. Therefore, if the $\mathcal{A}$ has advantage $\epsilon_1(l)$ in the above game instance, then we can build a simulator (S) which can break the DBDH problem with negligible quantity $\frac{\epsilon(l)}{2}$.
Now we show that the cipher components generated with Encrypt_Index algorithm obscured the underlying access policy correctly and the advantage of an attacker in computing any function of the underlying access policy is nonnegligible.

Init : The Challenger $\mathcal{C}$ initiated the game. $\mathcal{A}$ commits two challenge ciphertext policies $T_0^* = [T_{0,1}, T_{0,2}, \cdots, T_{0,n}]$ and $T_1^* = [T_{1,1}, T_{1,2}, \cdots, T_{1,n}]$. Assume that for some value λ $1 \leq \lambda \leq n$ $T_{0,\lambda} \neq T_{1,\lambda}$. We assume here that $T_{1,\lambda}$ includes all the values for attribute i, while in $T_{0,\lambda}$ except only one value $v_{\lambda,r}$ $(1 \leq r \leq m_\lambda)$ all other values for attribute i are included.

Setup : $\mathcal{C}$ initiates the setup procedure as in the real scheme. The public parameters as specified in the real scheme are given to adversary.

Phase 1 : For polynomially bounded number of times $\mathcal{A}$ submits attribute lists L=$[L_1,L_2,\cdots,L_n] = [v_{1,j1}, v_{2,j2}, \cdots, v_{n,jn}]$ for requesting secret key. If $\mathcal{A}$ submits an attribute List L such that $F(L,T_0^*) \neq F(L,T_1^*)$ then $\mathcal{C}$ will abort. Else it computes the private key for the requested attribute list as $D_{s0} = g_1^{r\beta}$ = and $\{\{ D_{i,j} = g_1^{\frac{(H(i||v_{i,j})^j + r)/\alpha}{r_j}} \}_{1 \leq j \leq m_i}\}_1 \leq i \leq n$.

The attacker can issue the secret keys SK which satisfies both the challenge access policies T_0^* and T_1^*. The attacker can use these keys to generate search queries tw.

Challenge : $\mathcal{A}$ submits two words w_0 and w_1 . If the adversary has retrieved a key for any queried L=[L_1, L_2, $\cdots$, L_n], such that $F(L,T_0^*) = F(L,T_1^*) = 1$, then $w_0 = w_1$. The challenger $\mathcal{C}$ flips a random coin b and performs the encryption algorithm for message M_b as per the access policy T_b^*. The encryption components are submitted to adversary.

Phase 2: It facilitates same type of queries as in phase1 to $\mathcal{A}$ with the restrictions defined in the game.

Guess : $\mathcal{A}$ outputs a guess ν' of ν. If $\nu' = \nu$ then $\mathcal{A}$ wins the game.

Now with following arguments we prove that advantage of adversary $\mathcal{A}$ in winning this game is negligible. During encryption all the coefficients of equations are masked by a secret random value r_j $1 \leq j \leq m$. The way for an adversary to find out the underlying access policy is if he can calculate any function from cipher components of attribute λ. This is because for every other attribute i $1 \leq i \leq n$ and $i \neq \lambda$, the cipher components are same for both the challenge access policy T_0^* and T_1^*. To fulfill the challenge and disclose the access policy, the adversary needs to compute value $a'_{\lambda m_\lambda}$, that is generated as an coefficient from equation for attribute λ during encryption procedure 2. This is because to recalculate the equation coefficients for attribute λ one needs to retrieve the value of a'_λ which is used at the time of generating the equation 1. The value a'_λ for the attribute λ is hidden in the value $C_{w\lambda,m_\lambda} = g_2^{a'_{\lambda m_\lambda} r_{m_\lambda}} = g_2^{a_\lambda r_{m_\lambda}}$. The adversary knows the public parameter $g_2^{r_{m_\lambda}}$. If he gets success in getting the value of a'_λ from $C_{w\lambda,m_\lambda}$ then he can recalculate the equation for attribute λ and able to find out the cipher components $\{ C_{w\lambda,j}\}_{1\leq j\leq m_\lambda}$.

Based on the hardness of discrete logarithm problem we say that the adversary can gain a non-negligible advantage in calculating the coefficients of an equation from the given cipher component. Therefore, even if the search token is available, the adversary will not be able to calculate as which access policy has been used to encrypt the index. Hence, the theorem. □