High Availability of Outsource Attribute-Based Encryption with Performance in Cloud Storage

Uma Devi.A[1], MD. Rafeeq[2]
([1]PG Student M.Tech (CSE), CMRTC, Medchal, TS, India)
([2]Associate Professor in CSE, CMRTC, Medchal, TS, India)

Abstract: Cloud computing becomes increasingly popular for data owners to outsource their data to public cloud servers while allowing intended data users to retrieve these data stored in cloud. This kind of computing model brings challenges to the security and privacy of data stored in cloud. Attribute-based encryption (ABE) technology has been used to design fine-grained access control system, which provides one good method to solve the security issues in cloud setting. However, the computation cost and cipher text size in most ABE schemes grow with the complexity of the access policy. Outsourced ABE (OABE) with fine grained access control system can largely reduce the computation cost for users who want to access encrypted data stored in cloud by outsourcing the heavy computation to cloud service provider (CSP). However, as the amount of encrypted files stored in cloud is becoming very huge, which will hinder efficient query processing. To deal with above problem, we present a new cryptographic primitive called attribute-based encryption scheme with outsourcing key-issuing and outsourcing decryption, which can implement keyword search function (KSF-OABE). The proposed KSF-OABE scheme is proved secure against chosen-plaintext attack (CPA). CSP performs partial decryption task delegated by data user without knowing anything about the plaintext. Moreover, the CSP can perform encrypted keyword search without knowing anything about the keywords embedded in trapdoor.

Keywords: attribute-based encryption, cloud computing, keyword search, outsourced key-issuing, outsourced decryption.

1. Introduction

Cloud computing is a new computation model in which computing resources is regarded as service to provide computing operations. This kind of computing paradigm enables us to obtain and release computing resources rapidly. So we can access resource-rich, various, and convenient computing resources on demand [1]. The computing paradigm also brings some challenges to the security and privacy of data when a user outsources sensitive data to cloud servers. Many applications use complex access control mechanisms to protect encrypted sensitive information. Sahai and Waters [2] addressed this problem by introducing the concept for ABE. This kind of new public-key cryptographic primitive enables us to implement access control over encrypted files by utilizing access policies associated with cipher texts or private keys. Two types of ABE schemes, namely key-policy ABE (KPABE) [3-8] and cipher text-policy ABE (CP-ABE) [9-15] are proposed. For KP-ABE scheme, each cipher text is related to a set of attributes, and each user's private key is associated with an access policy for attributes. A user is able to decrypt a cipher text if and only if the attribute set related to the cipher text satisfies the access policy associated with the user's private key. For CP-ABE scheme, the roles of an attribute set and an access policy are reversed. Bethencourt [9] et al. provided a CP-ABE scheme, which ensures encrypted data is kept confidential even if the storage server is un trusted. In order to withstand collusion attack and avoid sensitive information leakage from access structure, Qian et al. [11] proposed a privacy-preserving decentralized ABE scheme with fully hidden access structure. Deng et al. [12] constructed a cipher text-policy hierarchical attribute based encryption (CP-HABE) with short cipher texts, which enables a CP-HABE system to host many users from different organizations by delegating keys. In CPABE scheme, a malicious user maybe shares his attributes with other users, which might leak his decryption privilege as a decryption black box due to financial profits. In order to solve above problem, Cao et al. [13-15] presented some traceable CP-ABE schemes, which can find the malicious users who intentionally leak the partial or modified decryption keys to others. Some schemes [26-28] have been proposed to focus on the above problems. Qian et al. [26] provided a privacy preserving personal health record by utilizing multi-authority ABE.

2. Preliminary Knowledge

We give some definitions and review related cryptographic knowledge about bilinear pairing, complexity assumption, access structures, and secret sharing scheme that our scheme relies on.

2.1 Notations

Table1 lists some notations utilized in this paper.

TABLE 1 Notations

Acronym	Description
TA	Trusted Authority
KG-CSP	Key generation cloud service provider
D-CSP	Decryption cloud service provider
S-CSP	Storage cloud service provider
DO	Data owner
DU	Data user

2.2 Bilinear Pairing

Let G_1 and G_2 be multiplicative cyclic groups with prime order p. Suppose g is a generator of G_1 . e : $G_1 \times G_1 \rightarrow G_2$ is a bilinear map if it satisfies the following properties:

1) **Bilinearity**: For all u,v ϵ G_1, $e(u^a,v^b)$ = e (u,v)□ □ , where a,bϵ Z□ are selected randomly.
2) **Non degeneracy:** There exists u,v □ G_1 such that e(u,v) ≠1.
3) **Computability:** For all u,v ϵ G_1, there is an efficient algorithm to compute e(u,v).

2.3 Access Structures

Definition2 (Access Structure) [16]. Suppose {p₁,...p□} are a set of parties. A collection A⊆$2^{\{p_1,\ldots,p_n\}}$ is monotone if, BϵA and B⊆C then CϵA. A monotone access structure is a monotone collection A which is a nonempty subset for {p₁,...,p□}. The set in A is called an authorized set, and the set out of A is called an unauthorized set. Let ω and A be an attribute set and access policy. A predicate ϒ(ω,A) is defined as follows: ϒ(ω,A)ϵ{0,1}, if ωϵA , the value of ϒ(ω,A) equals to 1, else the value is 0.

2.4 Secret Sharing Scheme

The secret sharing scheme [33] used in our paper is based on Lagrange interpolation method. The Lagrange interpolation formula is as follows:

P□(x) =$\sum_{k=0}^{n} l_k(x) y_k = \sum_{k=0}^{n} (\prod_{j=0}^{n} \frac{x-x_j}{x_k-x_j}) y_k$, where $l_\kappa(x)$ is known as Lagrange coefficient and called basic function, the Y_κ is known as the interpolated function of k^{th} insertion point used for sharing the secret. The n-degree polynomial P□ (x) can be reconstructed through n+1 insertion points. The value ofP□ (0) will be kept as secret. In this paper, we select a random d-1 degree polynomial $P_{d-1}(x)$. The k insertion points are known as k attributes. We restore the (d-1) degree polynomial $P_{d-1}(x)$ through d shares y□ and the lagrange coefficient which is denoted as

Δ_i, □ = $\prod_{j \epsilon s} \frac{x-j}{i-j}$ where □ s□ =d. We obtain the secret $P_{d-1}(0)$ by computing $P_{d-1}(0) = \sum_{i \epsilon s} \Delta_i$, □$(0). y_i$.

3 KSF-OABE Scheme

Our scheme is based on the OABE proposed in [16]. We use tree-based access structure described as in [16]. A is an tree-based access policy bound up with user private key, ω is an attribute set embedded in cipher text, U is the attribute universe, and d is a threshold value set in advance. If ϒ (ω , A) = 1, S is a attribute set which satisfies S ⊆ { ω ∩ A} ∧ | S |=d. Based on the structure of the above, we add a function called keyword search function [28].

Set up (λ): TA chooses multiplicative cyclic groups G_1, G_2 with prime order p, g is a generator of G_1. TA selects a bilinear map e : $G_1 \times G_1 \rightarrow G_2$ and defines the attributes in U as values in Z_p . For simplicity, we set n = | U | and take the first n values in Z_p to be the attribute universe. TA randomly selects an integer x ϵ Z_p , computes $g_1 = g^x$, and chooses $g_2, h, h_1, .., h_n$ ϵ G_1 randomly where n is the number of attributes in universe. $H_1 : \{0,1\}^* \rightarrow G_1$ and $H_2 : G_2 \rightarrow \{0,1\}^{\log p}$ are two secure hash functions. TA publishes PK = (G_1, G_2, g, g_1, g_2, h, h_1,..h_n, H_1, H_2) as system public parameter, and keeps the master secret key MSK = x secret.

OABE-KeyGen$_{init}$ (A,MSK): Upon receiving a private key request on access policy A, TA selects x_1□ Z_p randomly and computes $x_2 = x - x_1$ mod p. $OK_{KGCSP} = x_1$ is sent to KG-CSP to generate outsourcing private key SK_{KGCSP} . $OK_{TA} = x_2$ is used to generate local private key SK_{TA} at TA side.

OABE-KeyGen$_{out}$(A,OK_{KGCSP}): TA sends OK_{KGCSP} to KGCSP for generating outsourcing private key SK_{KGCSP}. Upon receiving the request on (A,OK_{KGCSP}), KG-CSP chooses a (d-1)degree polynomial q(.) randomly such that q(0) = x_1. For i ϵ A, KG-CSP chooses r_i □ Z_p randomly, and computes $d_{i0} = g_2^{q(i)} (g_1 h_i)^{r_i}$ and $d_{i1} = g^{r_i}$. KG-CSP sends outsourcing private key SK_{KGCSP} = $\{d_{i0}, d_{i1}\}_{i□\omega}$ to TA.

OABE-KeyGen$_{in}$(OK$_{TA}$): TA takes OK$_{TA}$ as input and computes d$_{□ 0}$= $g_2^{x_2}$ $(g_1 h)^{r\theta}$ and d$_{\theta 1}$=$g^{r\theta}$, where r$_\theta$ □ Z$_p$ is selected randomly, θ is the default attribute. TA sets private key SK = (SK$_{KGCSP}$, K$_{TA}$), where SK$_{TA}$ = { d$_{\theta 0}$, d$_{\theta 1}$} TA responses the user with SK by secure channel.

KSF-KeyGen(PK,MSK,A,q$_{BF}$): To get a query private key of DU with access policy A, DU and TA interacts as follow:

—DU chooses a blinding factor BF = $u \in Z_p^*$ randomly, and provides a commitment q$_{BF}$ = $g_2^{\frac{1}{u}}$ and an access policy A to TA. DU keeps *u* secret.

—TA retrieves $(g_1 h)^{r\theta}$ corresponding to A, and computes a query private key QK = $g_2^{\frac{x}{u}}(g_1 h)^{r\theta}$ for the DU.
—TA sends the query private QK to DU by secure channel.

Encrypt(M,PK,ω): It takes as input a message M □ G$_2$, the public parameters PK and an attribute set ω associated with cipher text. DO randomly selects s □ Z$_p$ and calculates C$_0$ = Me(g$_1$,g$_2$)^s,C$_1$=g^s,C$_i$=(g$_1$,h$_i$)^s for each i□ ω, C$_\theta$=(g$_1$,h)^s. DO outputs the cipher text with attribute set ω, where CT=(ω ∪ {θ},C$_0$,C$_1$, {C$_i$}$_{i□ ω}$,C$_\theta$).

Index(PK,CT,KW): DO selects r □ Z$_p$ randomly and runs the index generation algorithm to compute k_i = e(g1,g$_2$)^s.e(g,H$_1$(kw$_i$))^s □ G$_2$ for each kw$_i$ □ KW where i = 1,..,m. DO outputs the indexes of keywords set as IX(KW)=(K$_1$,K$_2$,K$_i$) for kw$_i$ □ KW where K$_1$=C$_1$=g^s,
K$_2$=C$_\theta$ = (g$_1$h)^s, K$_i$ =H$_2$(k$_i$) . DO uploads the tuple (CT, IX(KW)) to the S-CSP.

Trapdoor (PK,QK,BF,kw) : In order to generate a trapdoor for a keyword kw , DU computes T$_q$(kw) = H$_1$(kw)QK^u , and sets $I = (I_{i0} = d_{i0}, I_{i1} = d_{i1})$ for all i □ A , D$_1$ = $d_{\theta 1}^u$. DU sets trapdoor for the keyword kw as T$_{kw}$ = (T$_q$(kw),I,D$_1$)

Test(IX(KW),T$_{kw}$,CT): DU submits a keyword search request by sending a trapdoor T$_{kw}$ for keyword kw along with an access policy A which is bound up with private key for DU. If the attribute set embedded cipher text satisfies the access policy A, D-CSP downloads all those cipher texts and executes partial decryption for them.

D-CSP computes: Q$_{CT}$ = $\frac{\prod_{i\epsilon s} e(c_1, I_{i0})^{\Delta_{i,s}(0)}}{\prod_{i\epsilon s} e(I_{i1}, c_i)^{\Delta_{i,s}(0)}} = e(g, g_2)^{s x_1}$

D-CSP searches for the corresponding cipher text CT related to the appointed index of keywords through submitted trapdoor T$_{kw}$. D-CSP computes:

K$_{kw}$ = $\frac{e(K_1, T_q(kw))}{e(D_1, K_2)}$ = e (g$_1$,g$_2$)^s.e(g,H$_1$(kw))^s, and H$_2$(k$_{kw}$).

D-CSP obtains the matching cipher text by comparing H$_2$(k$_{kw}$) with each tuple(CT,IX(KW)) stored in S-CSP. D-CSP tests whether H$_2$(k$_i$) = H$_2$(k$_{kw}$) for each kw$_i$ □ KW. D-CSP outputs ⊥ if does not find matched tuple, otherwise D-CSP sends the search result that includes the tuple (CT, IX(KW)) and partial decryption data Q$_{CT}$ to DU.

Decrypt (PK,CT,Q$_{CT}$,SK$_{TA}$) : Upon receiving the Q$_{CT}$ and the CT from D-CSP, DU can completely decrypt the cipher text and obtain the message M = $\frac{C_0 . e(d_{\theta 1}, C_\theta)}{Q_{CT} . e(C_1, d_{\theta 0})}$. The proposed KSF-OABE construction is correct as the following equations hold Q$_{CT}$ = $\frac{\prod_{i\epsilon s} e(C_1, I_{i0})^{\Delta_{i,s}(0)}}{\prod_{i\epsilon s} e(I_{i1}, C_i)^{\Delta_{i,s}(0)}}$ = $\frac{\prod_{i\epsilon s} e(g^s, g_2^{q(i)}(g_1 h_i)^{r i})^{\Delta_{i,s}(0)}}{\prod_{i\epsilon s} e(g^{r i}, (g_1 h_i)^s)^{\Delta_{i,s}(0)}}$

$$\frac{e(g, g_2)^{s \sum_{i\epsilon s} q(i)\Delta_{i,s}(0)} \prod_{i\epsilon s} e(g^s, (g_1 h_i)^{r i})^{\Delta_{i,s}(0)}}{\prod_{i\epsilon s} e(g^{r i}, (g_1 h_i)^s)^{\Delta_{i,s}(0)}} = e(g, g_2)^{s x_1}$$

$$K_{kw} = \frac{e((K_1, T_q(kw))}{e(D_1, K_2)} = \frac{e(g^s, H_1(KW)QK^u)}{e(d_{\theta 1}^u, (g_1 h)^s)} = \frac{e(g^s, H_1(kw) g_2^x (g_1 h)^{r\theta u})}{e(g^{r\theta u}, (g_1 h)^s)}$$

$$= \frac{e(g^s, (g_1 h)^{r\theta u} e(g, H_1(kw))^s e(g, g_2)^{sx}}{e(g^{r\theta u}, (g_1 h)^s)} = e(g_1, g_2)^s . e(g, H_1(kw))^s$$

$$M = \frac{C_0 . e(d_{\theta 1}, C_\theta)}{Q_{CT} . e(C_1, d_{\theta 0})} = \frac{Me(g_1, g_2)^s e(g^{r\theta}, (g_1 h)^s)}{e(g, g_2)^{s x_1} e(g^s, g_2^{x_2}(g_1 h)^{r\theta})}$$

$$= \frac{Me(g_1, g_2)^s e(g^{r\theta}, (g_1 h)^s)}{e(g_1, g_2)^s e(g^s, (g_1 h)^{r\theta})} = M$$

4 Security Proof

The first challenge of our construction on security and privacy is to defend the conspiracy attack from dishonest users and D-CSP. The conspiracy attack can be resisted because the master key x is randomly divided. We assume that there are two users who submit request on the generation of private keys. The master key x are randomly divided into two parts twice, and we get the splits (x$_1$, x$_2$) and (x$_1$', x$_2$'). We have x$_1$ + x$_2$ = x mod p and x$_1$' + x$_2$' = x mod p. We use x$_1$, x$_1$' to generate the outsourcing private key SK$_{KGCSP}$, SK$_{KGCSP}$'. X$_2$, x$_2$' are used to generate corresponding local private key SK$_{TA}$, SK$_{TA}$' respectively. If and only if SK$_{KGCSP}$ matches SK$_{TA}$, or SK$_{KGCSP}$' matches SK$_{TA}$', the cipher text will be fully decrypted. Although the dishonest users collude with S-

CSP and D-CSP to get all SK_{KGCSP} of users, they still aren't able to forge a valid private key SK of any user. We consider the security against Type I adversary here.

Theorem 1. The proposed KSF-OABE scheme with keyword search function is secure against chosen-plaintext attack launched by Type I adversary in selective model under DBDH assumption.
Proof. Assume that A is an Type I adversary that can break the proposed scheme, we can build an algorithm B that uses A as a sub-algorithm to solve the DBDH problem as follows. $H_1(.)$, $H_2(.)$ are defined as random oracles.
Setup: Algorithm B receives a challenge attribute set ω*
From A,B sends ω* to challenger C as its challenge attribute set. Challenger C gives B the public parameter ($X=g^x$,$Y=g^y$,$Z=g^z$). Algorithm B sets $g_1=X$,$g_2=Y$ and h= $g_1^{-1}g^{-\alpha}$ where α is selected from Z_p randomly. For each i ϵ ω*, B selects α_i ϵ Z_p randomly and sets h_i = $g_1^{-1}g^{-\alpha i}$, For each i ϵ ω*. B also chooses two collision resistant hash functions where
$H_1 : G_2 \rightarrow \{0,1\}^{\log p}$. B sends the public parameter (g, g_1, g_2, h, h_1, H_1, H_2) to A .
Query phase 1: The algorithm B initializes an empty table T. The adversary A makes any of the following queries adaptively:
H_1, H_2 -query. The adversary A can ask the random oracles H_1or H_2 at any time. In order to answer H_1 queries, the algorithm B maintains a list <kw_i,β_i,b_i,c_i> called H_1-list. The list is initially empty. When the adversary A asks the random oracle H_1 at the point of kw_i ϵ $\{0,1\}^*$, the algorithm B returns as follows:

a. If the kw_i has already existed in the H_1-list <kw_i, β_i,b_i,c_i>, algorithm B responds with $H_1(kw_i) = \beta_i$ where β_i □ G_1.
b. Otherwise, algorithm B generates c_i ϵ {0,1} by flipping a coin and picks a b_i □ Z_p randomly. If c_i = 0, B computes $\beta_i \leftarrow g_2 g^{b_i}$ ϵ G_1. Else, B computes $\beta_i \leftarrow g^{b_i}$ □ G_1.
c. The algorithm B adds the tuple <kw_i,β_i,b_i,c_i> to the H_1 -list and sets $H_1(kw_i) = \beta_i$ sends β_i to A.

Similar to H_1-queries, the adversary A can issue a query on H_2 at any time. In order to respond to the query $H_2(k_i)$ on k_i , the algorithm B maintains a list for the tuple <k_i, I_i> called H_2-list. The list is also initially empty. If the query on k_i exists in H_2-list. B responds I_i to A. Otherwise, B randomly picks for every I_i □ $\{0,1\}^{\log p}$ for every k_i and sets $H_2(k_i) = I_i$. Algorithm B adds the pair (k_i, I_i) to H_2-list. B sends I_i to A.
OABE-$KeyGen_{out}$ query. Upon receiving the query of private key SK_{KGCSP} on A, B checks if the tuple (A, SK_{KGCSP},.,.,.) exists in T. If so, B returns SK_{KGCSP} to A. Else, if □ A ∩ ω* □ < d, the algorithm B picks x_2 □ Z_p randomly and defines three sets Γ, Γ' and S, where Γ = A ∩ ω*, I Γ' I = d-1, , Γ ⊆ Γ' ⊆ A and S = Γ' ∪ {0} . Then, for each i □ Γ', B computes $d_0 = g_2^{Ti}$ $(g_1h_i)^{ri}$ and $d_{i1} = g^{ri}$ where *Ti* , r_i □ Z_p are selected at random. For each I □ A\Γ', We have $r_i = -y\,\Delta_{0,s}(i)+r_i'$ implicitly. B sets
$d_{i0} = g_2^{\sum_{j\epsilon\Gamma'} \Delta_{j,s}(i)\Gamma_j-(x_2+\alpha_i)\Delta_{0,s}(i)}(g_1 h_i)^{r'_i}$ and
$d_{i1} = g_2^{-\Delta_{0,s}(i)}g^{r'_i}$ by choosing r_i' □ Z_p randomly. B stores the partial private key in table T. Otherwise, B outputs ⊥ and terminates the game.
OABE-$KeyGen_{in}$ query. Upon receiving a private key query for A with □ A ∩ ω* □ < d, the algorithm B checks if the tuple (A,.,SK,.,.) exists in T. If so, B returns SK to A. Else, if x_2 for such tuple has not been selected in OABE-$KeyGen_{out}$ query, B picks x_2 □ Z_p randomly and obtains SK_{KGCSP} which is similar to out OABE-$KeyGen_{out}$ query (i.e., □ A ∩ ω* □ < d). Algorithm B computes $SK_{TA} = \{d_{\theta 0} = g_2^{x_2}(g_1 h)^{r_\theta}, d_{\theta 1}=g^{r_\theta}\}$, where r_θ □ Z_p is selected at random. B adds (A,SK_{KGCSP},K,.,.) into T and returns SK = (SK_{KGCSP}, SK_{TA}) to A.

5 Performance Analysis

5.1 Complexity Analysis

In Table 2 and Table 3, we briefly compare our scheme with the work in [16,17,27]. PK,MK,SK,CT,TK,RK,CT' represent the size of public key, master key, private key, cipher text length, transform key, retrieving key and transformed cipher text excluding the access structure respectively. Additionally, Encrypt, Transform, $Decrypt_{out}$, Decrypt denote the computational costs of the algorithms encryption, transformation, outsourcing decryption, decryption respectively. □ G_1□ , □ G_T□ , □ Z_p□ denote the bit length of the elements belong to G_1, G_T, Z_p, kG_1, kC_e, kH denote the k -times calculation over the group G_1, pairing and hash function. Let U = {att_1,…, att_n} be the attribute universe. N_1 and N_2 are amount of the attributes associated with cipher text and private key respectively. K is the number of keywords associated with a cipher text. As the operation cost over Z_p is much less than group and pairing operation, we ignore the computational time over Z_p.

Table 2 Size of each value

	PK	MK	SK	CT	TK	RK	CT'
LCL13[16]	$(n+4)\| G_1\|$	$\| Z_p \|$	$2N_2 \|G_1\|$	$(N_1+2) \|G_1\| + \|G_T\|$	$2(N_2-1)\|G_1\|$	$2 \|G_1\|$	$2\|G_1\|+2\|G_T\|$
LHL14[17]	$(n+4)\| G_1\|$	$\| Z_p \|$	$2N_2 \|G_1\|+$ $\|Z_p\|$	$(N_1+2) \|G_1\| + \|G_T\|$	$2N_2 \|G_1\|$	$\| Z_p \|$	$\|G_T\|$
GHW13[27]	$4\| G_1\|$	$\| Z_p \|$	none	$N_1+1)\|G_1\|+\|G_T\|+n\|Z_p\|$	$2N_2 \|G_1\|$	$\| Z_p \|$	$2\|G_T\|$
Our scheme	$(n+4)\| G_1\|$	$\| Z_p \|$	$2N_2 \|G_1\|$	$(N_1+4) \|G_1\|+(1+K) \|G_T\|$	$2N_2 \|G_1\|$	$\| Z_p \|$	$2\|G_T\|$

Table 3 Computational cost

	Encrypt	Transform	Decrypt_{out}	Decrypt
LCL 13[16]	$C_e+2G_T+(3+2N_1)G_1$	None	$2N_1C_e+(2N_1+1)G_T$	$2C_e+3G_T$
LHL 14[17]	$C_e+2G_T+(3+2N_1)G_1$	$2N_2G_1$	$2(N_1+1)C_e+(2N_1+3)G_T$	$3G_T$
GHW 13[27]	$C_e+(N_1+1)G_1+3G_T+N_1H$	$2N_2G_1$	$(N_1+1)C_e+2N_1G_1+N_1G_T$	$2G_T$
Our scheme	$(1+K)C_e+2(1+K)G_T+(3+2N_1)G_1+KH$	$4G_1$	$2(N_1+1)(C_e + G_T)$	$2C_e+3G_T$

5.2 Efficiency Analysis

We compared the performance of the four stages in our scheme with the scheme [16] as figures below. Our experiment is simulated with the java pairing-based cryptography (JPBC) library version 2.0.0 [35], which is a port of the pairing-based cryptography (PBC) library [36] in C. When selecting a secure elliptic curve, two factors should be considered: the group size l of the elliptic curve and the embedding degree d. To achieve the 1024-bit RSA security, these two factors should satisfy $l \times d \geq 1024$. We implement our scheme on Type A curve $y^2 = x^3 + x$, where p is 160 bits, $l = 512$. We select SHA as the hash function. We implement our scheme and the scheme [16] on a Windows machine with Intel Core 2 processor running at 2.13 GHz and 4G memory. The running environment of our experiment is Java Runtime Environment 1.7 (JRE1.7), and the Java Virtual Machine (JVM) used to compile our programming is 32 bit (x86) which brings into correspondence with our operation system. For simplicity, we assume that DU submits one keyword and obtains one partial decryption data to be decrypted fully in our system.

From Fig. 2(a) and Fig. 2(c), we see that the computation costs at the stages of Setup and Encryption grow linearly with the amount of the attribute in both systems and the computation costs in our scheme which is similar to the scheme [16]. Fig. 2(b) shows that the computation cost at the stage of KeyGen for KG-CSP grows linearly with the amount of the attributes in the system, but the computational cost for TA just keeps in a low level. The computation costs in our scheme are similar to the scheme [16] on both TA and KG-CSP side. Fig. 2(d) shows that the computation cost at the stage of Decryption for DU grows linearly with the amount of data belong to the DU in the system for scheme [16], but the computational cost in our system keeps in a low level.

6 Conclusion

In this article, we propose a CP-ABE scheme that provides outsourcing key-issuing, decryption and keyword search function. Our scheme is efficient since we only need to download the partial decryption cipher text corresponding to a specific keyword. In our scheme, the time-consuming pairing operation can be outsourced to the cloud service provider, while the slight operations can be done by users. Thus, the computation cost at both users and trusted authority sides is minimized. Furthermore, the proposed scheme supports the function of keywords search which can greatly improve communication efficiency and further protect the security and privacy of users. Actually, we are easy to extend our KSF-OABE scheme to support access structure represented by tree in [9].

7 Future Work

Verifiability is an important feature of KSF-OABE, so one of our future works is to construct KSF-OABE which can provide verifiability. Furthermore, our scheme was only RCCA secure in the random oracle model, hence constructing KSF-OABE which is CCA secure in the standard model is another future work.

8 Acknowledgment

I would like to thank my guide Mr. Md. Rafeeq, Assoc.prof in CSE department. As he having good teaching experience as well as research in the cloud computing. My sincere thanks to coordinator Mr. N. Bhaskar and HOD Dr. K. Srujan Raju for continuo's support and encouragement.

9 References

[1]. S. Pearson, Y. Shen and M. Mowbray, "A Privacy Manager for Cloud Computing," Proc. First International Conference Cloud Computing (CloudCom '09), M. Gilje-Jaatun, G. Zhao and C. Rong, eds., LNCS 5931, Berlin: Springer-Verlag, pp. 90-106, 2009.

[2]. A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," EUROCRYPT '05, LNCS, vol. 3494, pp. 457-473, 2005.

[3]. V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," Proc. 13th ACM Conference on Computer and Communications Security (CCS '06), pp. 89-98, 2006, doi:10.1145/ 1180405.1180418.

[4]. J.W. Li, C.F. Jia, J. Li and X.F. Chen, "Outsourcing Encryption of Attribute-Based Encryption with Mapreduce," Proc. 14th International Conference on Information and Communications Security (ICICS '12), LNCS 7618, Berlin: Springer-Verlag, pp. 191-201, 2012. doi: 10.1007/ 978-3-64234129-8_17

[5]. A. Lewko, T. Okamoto, A. Sahai, K. Takashima and B. Waters, "Attribute-Based Encryption and (Hierarchical) Inner Product Encryption," EUROCRYPT '10, H. Gilbert, ed., LNCS 6110, Berlin: Springer-Verlag, pp. 62-91, 2010.

[6]. J.G. Han, W. Susilo, Y. Mu and J. Yan, "Privacy-Preserving Decentralized Key-Policy Attribute-Based Encryption," IEEE Transactions on Parallel and Distributed Systems, vol. 23, no.11, pp. 21502162, Nov. 2012, doi: 10.1109/ TPDS.2012.50.

[7]. T. Okamoto and K. Takashima, "Fully Secure Functional Encryption with General Relations from the Decisional Linear Assumption," CRYPTO '10, T. Rabin, ed., LNCS 6223, Berlin: Springer-Verlag, pp. 191208, 2010.

[8]. W.R. Liu, J.W. Liu, Q.H. Wu, B. Qin, and Y.Y. Zhou, "Practical Direct Chosen Ciphertext Secure Key-Policy Attribute-Based Encryption with Public Ciphertext Test," ESORICS '14, LNCS 8713, Berlin: SpringerVerlag, pp. 91-108, 2014.

[9]. J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy Attribute-Based Encryption," Proc. IEEE Symp. Security and Privacy (SP '07), pp. 321-334, May. 2007, doi:10.1109/ SP.2007.11.

[10]. L. Cheung and C. Newport, "Provably Secure Ciphertext Policy ABE," Proc. 14th ACM Conference on Computer and Communications Security (CCS '07), pp. 456-465, 2007, doi:10.1145/ 1180405.1180418.

[11]. H.L. Qian, J.G. Li and Y.C. Zhang, "Privacy-Preserving Decentralized Ciphertext-Policy Attribute-Based Encryption with Fully Hidden Access Structure," Proc. 15th International Conference on Information and Communications Security (ICICS '13), LNCS 8233, Berlin: Springer-Verlag, pp. 363-372, 2013.

[12]. H. Deng, Q.H. Wu, B. Qin, J. Domingo-Ferrer, L. Zhang, J.W. Liu, and W.C. Shi, "Ciphertext-Policy Hierarchical AttributeBased Encryption with Short Ciphertexts," Information Sciences, vol. 275, no. 8, pp. 370-384, Aug. 2014.

[13]. J.T. Ning, Z.F. Cao, X.L. Dong, L.F. Wei and X.D. Lin, "Large Universe Ciphertext-Policy Attribute-Based Encryption with White-Box Traceability," ESORICS '14, LNCS 8713, Berlin: SpringerVerlag, pp. 55-72, 2014.

[14]. Z. Liu, Z.F. Cao, and D. S. Wong, "Traceable CP-ABE: How to Trace Decryption Devices Found in the Wild," IEEE Transactions on Information Forensics and Security, vol. 10, no.1, pp. 55-68, Jan. 2015.

[15]. J.T. Ning, X.L. Dong, Z.F. Cao, L.F. Wei and X.D. Lin, "WhiteBox Traceable Cphertext-Policy Attribute-Based Encryption Supporting Flexible Attributes," IEEE Transactions on Information Forensics and Security, vol. 10, no.6, pp. 1274-1288, Jun.

[16]. J. Li, X.F. Chen, J.W. Li, C.F. Jia, J.F. Ma and W.J. Lou, "Fine-Grained Access Control System Based on Outsourced Attribute-Based Encryption," Proc. 18th European Symposium on Research in Computer Security (ESORICS '13), LNCS 8134, Berlin: Springer-Verlag, pp. 592-609, 2013.

[17]. J. Li, X. Huang, J. Li and X. Chen, "Securely Outsourcing Attribute-Based Encryption with Checkability," IEEE Trans. Parallel and Distributed Systems, vol. 25, no. 8, pp. 22012210, Oct 2013/ Jul 2014, doi:10.1109/ TPDS.2013.271.

[18]. S. Hohenberger and A. Lysyanskaya, "How to Securely Outsource Cryptographic Computations," Proc. Second Theory of Cryptography Conference (TCC'05), J. Kilian, ed., LNCS 3378, Berlin: Springer-Verlag, pp. 264-282, 2005.

[19]. M. Yang, F. Liu, J.L. Han and Z.L. Wang, “An Efficient Attribute Based Encryption Scheme with Revocation for Outsourced Data Sharing Control,” Proc. 2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC ’11), pp. 516-520, Oct. 2011, doi:10.1109/ IMCCC.2011.134.

[20]. J.Z. Lai, R.H. Deng and C. Guan, “Attribute-Based Encryption with Verifiable Outsourced Decryption,” IEEE Transactions on Information Forensics and Security, vol. 8, no. 8, pp. 1343-1354, 2013, doi: 10.1109/ TIFS.2013.2271848.

[21]. F. Zhao, T. Nishide and K. Sakurai, “Realizing Fine-Grained and Flexible Access Control to Outsourced Data with Attribute-Based Cryptosystems,” Proc. 7th International Conference. Information Security Practice and Experience (ISPEC ’11), F. Bao and J. Weng, eds., LNCS 6672, Berlin: Springer-Verlag, pp. 83-97, 2011.

[22]. X. Chen, J. Li, X. Huang and J. Li, “Secure Outsourced Attribute-Based Signatures,” IEEE Trans. Parallel and Distributed Systems, vol. 25, no. 12, pp. 3285-3294, Jan/ Nov 2014, doi:10.1109/ TPDS.2013.229580.

[23]. G. Ateniese, K. Fu, M. Green and S. Hohenberger, “Improved Proxy Re-encryption Scheme with Application to Secure Distributed Storage,” J. ACM Transactions on Information and System Security (TISSEC), vol. 9, no. 1, pp. 1-30, Feb. 2006, doi:10.1145/1127345.1127346.

[24]. B. Libert and D. Vergnaud, “Unidirectional Chosen-Ciphertext Secure Proxy Re-encryption,” Proc. 11th International Workshop on Practice and Theory in Public Key Cryptography (PKC ‘08), R. Cramer, ed., LNCS 4939, Berlin: Springer-Verlag, pp. 360-379, 2008.

[25]. J. Hur and D.K. Noh, “Attribute-based Access Control with Efficient Revocation in Data Outsourcing Systems,” IEEE Transactions on Parallel and Distributed Systems, vol. 22, no.7, pp. 1214-1221, Nov 2010, doi: 10.1109/ TPDS.2010.203.

[26]. M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou,“Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption, ” IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 1, pp. 131-143, Jan. 2012, doi:10.1109/ TPDS.2012.97.

[27]. M. Green, S. Hohenberger and B. Waters, “Outsourcing the Decryption of ABE Ciphertexts,” Proc. 20th USENIX Conference on Security (SEC ’11), pp. 34, 2011.

[28]. D. Boneh, G.D. Cirescenzo, R. Ostrovsky and G. Persiano, “Public Key Encryption with Keyword Search,” EUROCRYPT ’04 , C. Cachin and J.L. Camenisch, eds., LNCS 3027, Berlin: Springer-Verlag, pp. 506-522, 2004.