5-2018

DOAS: Efficient data owner authorized search over encrypted cloud data

Yibin MIAO

Jianfeng MA

Ximeng LIU
Singapore Management University, xmliu@smu.edu.sg

Zhiquan LIU

Junwei ZHANG

See next page for additional authors

DOI: https://doi.org/10.1007/s12083-016-0523-7

Citation

MIAO, Yibin; MA, Jianfeng; LIU, Ximeng; LIU, Zhiquan; ZHANG, Junwei; and WEI, Fushan. DOAS: Efficient data owner authorized search over encrypted cloud data. (2018). *Peer-to-Peer Networking and Applications*. 11, (3), 349-360. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/3628

Author

Yibin MIAO, Jianfeng MA, Ximeng LIU, Zhiquan LIU, Junwei ZHANG, and Fushan WEI

Published in Peer-to-Peer Networking and Applications, 15 October 2016, Pages 1-12, Advance Online
http://doi.org/10.1007/s12083-016-0523-7

DOAS: Efficient data owner authorized search over encrypted cloud data

Yinbin Miao[1] · Jianfeng Ma[2] · Ximeng Liu[3] · Zhiquan Liu[2] · Junwei Zhang[2] · Fushan Wei[4]

Abstract Data outsourcing service can shift the local data storage and maintenance to cloud service provider (CSP) to ease the burden from data owner, but it brings the data security threats as CSP is always considered to honest-but-curious. Therefore, searchable encryption (SE) technique which allows cloud clients (including data owner and data user) to securely search over ciphertext through keywords and selectively retrieve files of interest is of prime importance. However, in practice, data user's access permission always dynamically varies with data owner's preferences. Moreover, existing SE schemes which are based on attribute-based encryption (ABE) incur heavy computational burden through attribution revocation and policy updating. To allow data owner to flexibly grant access permissions, we design a secure cryptographic primitive called as efficient data owner authorized search over encrypted data scheme through utilizing identity-based encryption (IBE) technique. The formal security analysis proves that our scheme is secure against chosen-plaintext attack (CPA) and chosen-keyword attack (CKA) without random oracle. Besides, empirical experiments over real-world dataset show that our scheme is efficient and feasible with regard to data access control.

Keywords Cloud computing · Searchable encryption · Data owner authorized search · Chosen-plaintext attack · Chosen-keyword attack

1 Introduction

Cloud computing [1, 2] develops rapidly due to its long list of unprecedented advantages (such as low maintenance cost, ubiquitous network access, on-demand service, etc), but it still brings new and challenging security issues [3, 4] over outsourced data due to the honest-but-curious CSP [5]. To reduce heavy local data maintenance and computation burden, both individuals and Information Technology (IT) enterprises prefer to backup their sensitive data on CSP. Encryption is trivial solution to keep data confidentiality, but it makes the retrieval over encrypted cloud data

Jianfeng Ma
jfma@mail.xidian.edu.cn

Yinbin Miao
ybmiao@stu.xidian.edu.cn

Ximeng Liu
snbnix@gmail.com

Zhiquan Liu
zqliu@xidian.org.cn

Junwei Zhang
jwzhang@xidian.edu.cn

Fushan Wei
weifs831020@163.com

[1] School of Telecommunication Engineering,
Xidian University, Xi'an, China

[2] School of Computer Science and Technology,
School of Cyber Engineering, Xidian University,
Xi'an, China

[3] School of Information Systems,
Singapore Management University,
80 Stamford Road, Singapore, Singapore

[4] State Key Laboratory of Mathematical Engineering
and Advanced Computing, The PLA Information Engineering
University, Zhengzhou, China

extremely difficult. Besides, simply downloading the whole ciphertext locally is a naive solution due to computation and bandwidth resources waste. Thus, search over encrypted data recently has been envisioned as a critical topic which incurs considerable amount of interest in both academia and industry fields, and the SE technique [6, 7] which allows the data users to securely search over ciphertext through keywords and selectively retrieve files of interest is of prime importance.

The SE scheme should not only protect data privacy from CSP and unauthorized data users, but also meet the search requirements in practice. Without the access control, the data confidentiality cannot be guaranteed as data owners directly lose the physical control over remote cloud data. To the best of our knowledge, there still exist some limitations in the traditional access control mechanisms due to high system overheads. The latest cryptographic primitive called as ABE [8, 9] is proposed and it can achieve fine-grained access control over encrypted data. However, the state-of-the-art SE schemes [10, 11] which are based on ABE can only identity certain category of data users according to descriptive attributes. Moreover, as data user's access permission always dynamically varies with data owner's preferences, these schemes ultimately incur heavy computational burden when updating access permissions. Therefore, the practical SE scheme should support flexible access permissions.

While the access permission which is based on specific identity instead of obscure attribute set remains to be addressed for the sake of preventing unauthorized accesses. As far as we know, existing SE schemes can allow authorized data user to issue search queries with the help of CSP. However, in practice, we should place the duty of access control enforcement on data owners due to semi-trusted CSP. Moreover, traditional SE schemes need to build secure channel between data user and CSP, thereby incurring much more communication overhead. Therefore, supporting free secure channel [12, 13] remains an overarching concern in this field. As data user's access privilege changes with the DO's preferences, our scheme should provide direct access control to prevent unauthorized access. Besides, as the existing SE schemes have inherent defects, our scheme should resist CKA without random oracle. To tackle aforementioned problems, we propose a **D**ata-**O**wner **A**uthorized **S**earch (**DOAS**) over encrypted data scheme based on IBE. The main contributions of this work can be summarized as follows:

1) ***Direct authorization***. In the semi-trusted cloud computing environment, our scheme can provide direct control access so that data owner dynamically decides data user's access permission rather than fuzzy attribute set.
2) ***Security without random oracle***. For data security concerns, our scheme is secure against CPA and CKA without random oracle.
3) ***Efficient and feasible***. Without secure channel, our scheme is more efficient and feasible in practice than existing SE schemes which are based on ABE.

The remainder of this paper is organized as follows. Section 2 presents the previous work associated with DOAS scheme. Section 3 gives the preliminaries used in DOAS scheme. The problem formulation (such as system model, threat model, design goals, scheme definition and security model) is shown in Section 4, followed by Section 5 which demonstrates the concrete construction of DOAS scheme. Section 6 analyzes the DOAS scheme in terms of correctness, security and performance. Section 7 draws some concluding remarks.

2 Related work

With the growing awareness of data privacy-preserving, a considerable number of data owners are motivated to outsource data on CSP to reduce heavy computational burden. Aiming at achieving secure and efficient search over encrypted data, SE technique [14–17] becomes increasingly popular among enterprises and individuals. Until now, abundant privacy-preserving SE schemes have achieved a variety of search functionalities, such as single keyword search [18–20], multiple keywords search [21, 22], ranked search [23–25]. However, a major defect is that these schemes need a secure channel between CSP and data users, which is costly and not feasible in actual applications. Moreover, in cloud computing environment data users are actually granted different access privileges specified by data owners. Although there exists a significant amount of work focusing on access control over encrypted data, providing the access control enforcement remains to be addressed.

To the best of our knowledge, existing SE schemes can be categorized into key-based access control (KBAC) and attribute-based access control (ABAC). The former usually directly assigns decryption keys to authorized data users, thus it may incur heavy key management burden on data owners. And the latter exploits attribute-based encryption (ABE) techniques [8, 9] to specify access policy for data or secret keys. In ABE the private key and ciphertext are associated with attributes or access policy, respectively, only there is a match between attributes and access policy data users can decrypt encrypted data. Existing ABE schemes can be roughly categorized into key-policy ABE [8] (KP-ABE, i.e. the key is associated with access policy and ciphertext is embedded with attribute) or ciphertext-policy ABE [9] (CP-ABE, contrary to KP-ABE).

However, there still exist some concerns when completely placing search operation and access control enforcement on CSP. For the sake of security, we should separate the search operation from access control enforcement. The most practical SE schemes [10, 11] can only identity certain category of data users according to descriptive attributes. Moreover, updating access permissions (such as attribute revocation and policy updating) will result in heavy computational burden. Thus in actual applications, data owners should be personally involved in privilege authentication to guarantee data confidentiality and avoid unauthorized accesses.

Aiming at supporting exact access authorization, we utilize IBE technique to achieve this goal. Since the first IBE cryptographic primitive was proposed, vast follow-up works [26–28] have focused on secure distributed storage model to conquer data privacy disclosure threats. In IBE, public key can be denoted as an arbitrary string and the secret key is launched by the third trusted server. Moreover, two parties who have direct interaction can eliminate the public key certificates. Through extending the IBE scheme [26] to SE scheme, we propose an efficient DOAS scheme to tackle the problems of access control enforcement and illegal accesses.

3 Preliminaries

In this section, we present some necessary cryptographic background associated with DOAS scheme.

Definition 1 (**Bilinear Map**) Let G_1, G_2 be two multiplicative cyclic groups of prime order q, g be a generator of G_1. Then $e : G_1 \times G_1 \rightarrow G_2$ is a bilinear map such that for all $u, v \in G_1$ and $a, b \in \mathcal{Z}_q^*$, $e(u^a, v^b) = e(u, v)^{ab}$. This bilinearity implies that for any $a, b, c \in G_1$, $e(ab, c) = e(a, c) \cdot e(b, c)$. Besides, there exists an efficiently computable algorithm for computing e and the bilinear map should be nontrivial, i.e., e is nondegenerate: $e(g, g) \neq 1$, where "1" is the identity element in G_1.

Definition 2 (**Decisional Bilinear Diffie-Hellman (DBDH) Assumption**) Given the bilinear map parameters (G_1, G_2, e, q, g) and random elements $a, b, c, d \in \mathcal{Z}_q^*$, if there exists no probabilistic polynomial-time adversary $\mathcal{A}$ can distinguish $(g^a, g^b, g^c, e(g, g)^{abc})$ from $(g^a, g^b, g^c, e(g, g)^d)$ with the advantage $Adv_{\mathcal{A}}^{DBDH}(k) = |Pr[\mathcal{A}(g^a, g^b, g^c, e(g, g)^{abc})] - Pr[\mathcal{A}(g^a, g^b, g^c, e(g, g)^d)]| \geq \epsilon(k)$, then we say the DBDH assumption relative to G_1 holds.

Definition 3 (**Truncated Decisional p-Augmented Bilinear Diffie-Hellman Exponent (p-ABDHE) Assumption**) Given the bilinear map parameters (G_1, G_2, e, q, g) and random elements $a, b, c, d \in \mathcal{Z}_q^*$, if there exists no probabilistic polynomial-time adversary $\mathcal{A}$ can solve truncated decisional p-ABDHE problem with the advantage $Adv_{\mathcal{A}}(k) = |Pr[\mathcal{A}(g, g^a, ..., g^{a^p}, g^b, g^{ba^{p+2}}, e(g, g)^{ba^{p+1}}) = 1] - Pr[\mathcal{A}(g, g^a, ..., g^{a^p}, g^b, g^{ba^{p+2}}, e(g, g)^c) = 1]| \geq \epsilon(k)$, then we say truncated decisional p-ABDHE assumption relative to G_1 holds.

4 Problem formulation

4.1 System model & threat model

In this section, we consider a cloud storage system involving with four main entities (namely data owner, data user, key generation server and CSP) which are illustrated in Fig. 1. Data owner (DO) encrypts data and builds indexes before outsourcing ciphertext to CSP, CSP stores data and conducts search operations, key generation server (KGS) is responsible for distributing keys for cloud clients and CSP, and authorized data user (DU) can issue search queries according to specified keyword. Once DU is proved to be legitimate by the DO, then CSP returns the relevant ciphertext to DU. One thing to note is that the cloud clients include DO and DU.

Like most of previous SE schemes, CSP is assumed to be honest-but-curious. Specifically, CSP honestly follows the designated protocols in general, but it is still anxious to deduce as much as sensitive information. Besides, KGS and authorized DU are believed to be fully trusted entities, and CSP cannot collude with DU.

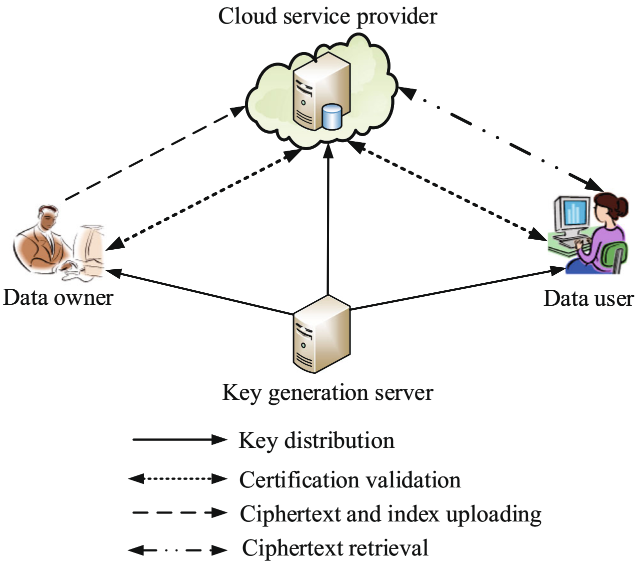

Fig. 1 The framework of our proposed scheme

4.2 Design goals

To achieve data-owner authorized search over encrypted data, DOAS scheme should meet the following design goals:

1) ***Data security and privacy***. The unauthorized access should be impeded by CSP and DO to guarantee data security and privacy. Besides, the privacy of search query should be protected against chosen-keyword attack.
2) ***Direct access control***. Independent of the third-party server, DO should be able to directly decide DU's access privilege so as to avoid unauthorized accesses.
3) ***Secure channel-free***. As building secure channel will incur heavy computational costs, DOAS scheme should remove the high-cost secure channel.
4) ***Efficiency and feasibility***. Comparing with the state-of-the-art SE schemes, DOAS scheme should be efficient and feasible in practical applications.

4.3 Definition of DOAS scheme

In this paper the symbol $x \in X$ is denoted as randomly selecting an element x from the set X. For the integer S, the symbol $[S]$ is denoted as an integer set $\{1, 2, \cdots, S\}$. Let an integer k be the security level, id be an identity which is an n-bit string, W, M be keyword and message space, respectively. DOAS scheme includes seven algorithms which are demonstrated in Definition 1.

Definition 4 (**DOAS scheme**) DOAS scheme is a tuple (**Setup**, **KeyGen**, **Enc**, **Trap**, **Search**, **Verify**, **Dec**) of seven polynomial-time algorithms which are shown as follows:

(1) **Setup**$(1^k) \rightarrow (pk, msk)$. Given a security parameter k, KGS runs this probabilistic algorithm to generate public key pk and master key msk, where msk is owned by itself.
(2) **KeyGen**$(pk, msk) \rightarrow (pk_s, sk_s, pk_c, sk_c)$. KGS performs this probabilistic algorithm to return public/secret key pairs $\{(pk_s, sk_s), (pk_c, sk_c)\}$ for CSP and cloud client, respectively.
(3) **Enc**$(m, w, id, pk, pk_c, pk_s) \rightarrow (C, I)$. DO whose identity is id first extracts keyword $w \in W$ from the message $m \in M$, then he runs this probabilistic algorithm to encrypt message m as C and build index I for it, finally he sends the tuple (C, I) to CSP.
(4) **Trap**$(w', sk_c, pk) \rightarrow (T_{w'})$. DU runs this probabilistic algorithm to generate a search token (trapdoor) $T_{w'}$ based on his submitted keyword w' and send it to CSP.
(5) **Search**$(T_{w'}, I, C, pk, sk_s) \rightarrow C'$. Once gaining the trapdoor $T_{w'}$, CSP runs this deterministic algorithm to check whether the trapdoor $T_{w'}$ matches with the index I. If $T_{w'}$ matches with I, CSP outputs the relevant results C'. Otherwise, CSP outputs $\perp$. Before returning C' to DU, CSP asks DU to submit his $Proof$ information to check his legality.
(6) **Verify**$(sk_c, Proof) \rightarrow \{0, 1\}$. If $T_{w'}$ matches with I in **Search** algorithm, then DO continues to run this probabilistic algorithm; otherwise, it abolishes. If the $Proof$ information submitted by DU is legitimate, then CSP returns the relevant Ciphertext C' to DU. Otherwise, CSP rejects it. Where "1" means that DU passes the DO's verification and gains the relevant ciphertext C', "0" means that DU is not an authorized entity and gains $\perp$.
(7) **Dec**$(C', sk_c, pk_c) \rightarrow m$. DU runs this deterministic algorithm to decrypt ciphertext C' and gain the plaintext m through his secret key sk_c.

Different from previous schemes, DOAS scheme not only supports secure channel-free during search procedure, but also it can accurately grant DU's access permission.

4.4 Security model

In this section, we formalize the security models of DOAS scheme through the indistinguishability chosen-keyword attack (IND-DOAS-CKA) and indistinguishability chosen-plaintext attack (IND-DOAS-CPA). Informally, IND-CKA security can prevent internal malicious CSP to gain the trapdoor for the specified keyword. Moreover, CSP cannot distinguish which ciphertext corresponds to which keyword. The IND-DOAS-CKA game is presented as follows:

Definition 5 (**IND-DOAS-CKA game**) Let k be the security parameter, then the IND-DOAS-CKA is defined by the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

– **Setup**. Give the public key pk and master key msk, $\mathcal{C}$ first runs **KeyGen** algorithm to generate public and secret key pairs $\{(pk_s, sk_s), (pk_c, sk_c)\}$ for cloud client and CSP, respectively. Then $\mathcal{C}$ sends the tuple $\{pk_s, sk_s), pk_c\}$ to $\mathcal{A}$.
– **Phase 1**. $\mathcal{A}$ issues search queries for a set of keywords $W = \{w_1, ..., w_t\}$, then each keyword $w_i \in W$ is handled by the following algorithm.

  $-Trap(w_i)$. $\mathcal{A}$ can adaptively issue trapdoor T_{w_i} for any keyword $w_i \in W$, then $\mathcal{C}$ performs this algorithm and returns trapdoor T_{w_i} to $\mathcal{A}$.
– **Challenge**. After **Phase 1** is over, $\mathcal{A}$ outputs a target keyword pair (w_0, w_1). There exists a restriction that none of keywords (w_0, w_1) has been queried in **Phase 1**. $\mathcal{C}$ responses it through selecting a random bit $b \in \{0, 1\}$, then $\mathcal{C}$ returns the target index I_b to $\mathcal{A}$.

- **Phase 2**. $\mathcal{A}$ again issues a number of search queries $W' = \{w'_1, ..., w'_t\}$ as in **Phase 1** on the condition that $w_0, w_1 \notin W'$.
- **Guess**. $\mathcal{A}$ outputs his guess b', and he wins the game if $b = b'$.

We consider $\mathcal{A}$ as an IND-DOAS-CKA adversary, and the advantage in attacking DOAS scheme is set as $Adv_{\mathcal{A}}^{IND-DOAS-CKA}(k) = |Pr[b = b'] - \frac{1}{2}|$.

Definition 6 (**IND-DOAS-CPA game**) For **Setup**,**KeyGen**,**Enc** algorithms, DOAS scheme is said to be IND-CPA secure if there exist no polynomial-time adversaries $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ can win the following IND-CPA game, which as shown in the Table 1 with an non-negligible advantage $Adv_{\mathcal{A}}^{IND-DOAS-CPA}(k) = |Pr[b = b'] - \frac{1}{2}|$.

5 Proposed DOAS scheme

To avoid fuzzy access authorization we design an efficient DOAS scheme via IBE scheme to enhance the access control enforcement. To have a better understanding of DOAS scheme, we first demonstrate some notations in Table 2.

5.1 Concrete construction of DOAS scheme

In this section, we detail the construction of DOAS scheme. By efficiency, we explore to reduce the heavy computational burden as much as possible through removing the secure channel. By security, we aim that the legal access permission should be authorized by DO instead of CSP. Before presenting the specific construction of DOAS scheme we first show its basic protocol in Fig. 2.

Setup(1^k) Given a security parameter k, KGS first runs this algorithm to output the bilinear map parameters (e, q, g, G_1, G_2). Let h be the hash function $h : \{0, 1\}^* \to \mathcal{Z}_q^*$, the elements g_0, g_1, g_2 be three generators of G_1, $\mathcal{U} = \{u_0, u_1, ..., u_n\}$ be the random element set in G_1, then KGS randomly selects $\alpha \in \mathcal{Z}_q^*$ and sets $X_1 = g^{\alpha}$, $X_2 = g_0^{\alpha}$. The public key pk and master key msk are denoted by Eq. 1.

$$pk = \{G_1, G_2, e, h, g, g_0, g_1, g_2, \mathcal{U}\}, msk = \alpha. \quad (1)$$

Table 1 IND-DOAS-CPA game

$Game_{DOAS,\mathcal{A}}^{IND-CPA}(k)$
$(pk_c, sk_c) \leftarrow Setup(1^k)$; $b \in \{0, 1\}$;
$(m_0, m_1) \leftarrow \mathcal{A}_0(pk_c)$; $C_b \leftarrow Enc(pk_c, m_b)$;
$b' \in \{0, 1\} \leftarrow \mathcal{A}_1(\{C_b\}_{pk_c})$; If $b' = b$, $\mathcal{A}$ wins the game;

Table 2 Notation descriptions

Notations	Descriptions
$\{pk, msk\}$	Public/Master keys
$\{pk_s, sk_s\}$	Public/secret key pair of CSP
$\{pk_c, sk_c\}$	Public/secret key pair of cloud client
$id = \{id_1, \cdots, id_n\}$	Identity of cloud client
$\mathcal{V} = \{i\}$	Integer set which satisfies $id_i = 1$
m	Selected message or data file
$C = \{c_0, c_1, c_2\}$	Ciphertext for m
$I = \{I_0, I_1\}$	Index for keyword in m
$T_{w'}$	Trapdoor for queried keyword w'
C'	Returned search results
$Proof$	DU's proof information

KeyGen(pk, msk) For the CSP, KGS first chooses a random element $\lambda \in \mathcal{Z}_q^*$ and computes $\varphi = g^{\lambda}$, then KGS randomly selects $\theta \in G_1$ and sets the public/private key pair of CSP as $pk_s = \{\varphi, \theta\}$, $sk_s = \{\lambda\}$. While for each cloud client, let his identity be a n-bit string, id_i be the i-th bit of id, and each index i in the set $\mathcal{V}$ should satisfy $id_i = 1$. KGS first randomly chooses $\mu, \pi_{id} \in \mathcal{Z}_q^*$, $\tau \in G_1$ and computes $\psi = g^{\mu}$, then he sets the public/private key pair of cloud client as $pk_c = \{\psi, \tau, g^{\pi_{id}}, g_0^{\pi_{id}}, X_1, X_2\}$, $sk_c = \{\mu, k_{id} = g_1^{\alpha}(u_0 \prod_{i \in \mathcal{V}} u_i)^{\pi_{id}}\}$.

Enc(m, w, id, pk, pk_c, pk_s) Given the message m, DO whose identity is id extracts keyword w from m and builds index for it. He first selects random elements $s_{id}, r_{id} \in \mathcal{Z}_q^*$ and computes $\beta = h(e(\varphi, \theta)^{s_{id}})$, then he sets ciphertext C and index I as $C = \{c_0 = g^{s_{id}}, c_1 = m \cdot e(X_1, g_1)^{s_{id}}, c_2 =$

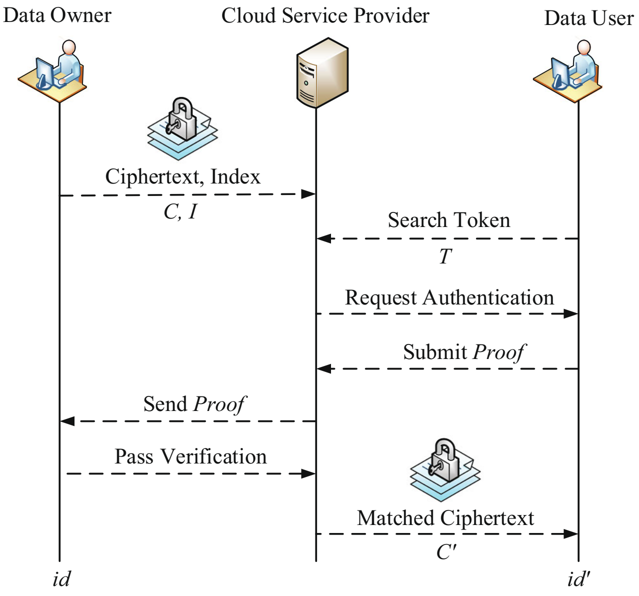

Fig. 2 The protocol of DOAS scheme

$(u_0 \prod_{i \in \mathcal{V}} u_i)^{s_{id}}\}$, $I = \{I_0 = g^{r_{id}}, I_1 = (\psi g^{-w})^{r_{id}/\beta}\}$, respectively, finally he sends the tuple $\{I, C\}$ to CSP.

Trap($\boldsymbol{w', sk_c, pk}$) When DU whose identity is id' wants to retrieve the message including keyword w', he first chooses random element $\gamma_{w'} \in \mathcal{Z}_q^*$ and computes $\delta_{w'} = (\tau g^{-\gamma_{w'}})^{1/\mu - w'}$, then he returns the trapdoor $T_{w'} = \{\gamma_{w'}, \delta_{w'}\}$ to CSP.

Search($\boldsymbol{T_{w'}, I, C, pk, sk_s}$) After gaining the trapdoor $T_{w'}$, CSP first computes $\beta' = h(e(c_0, \theta)^\lambda)$, then he verifies whether the Eq. 2 holds when $\beta' = \beta, w' = w$. If Eq. 2 holds, DU needs to submit his *Proof* information to CSP. Otherwise, CSP returns $\perp$.

$$e(I_1^{\beta'}, \delta_{w'}) e(I_0, g)^{\gamma_{w'}} = e(I_0, \tau). \tag{2}$$

Verify($\boldsymbol{sk_c, Proof}$) If Eq. 2 holds, the CSP asks the DU to submit his *Proof* information. DU first chooses a random element $\epsilon \in \mathcal{Z}_q^*$ and computes $k'_{id'} = k_{id'} g_2^\epsilon$, $\Gamma = g_0^\epsilon$, then DU sends $Proof = \{id', k'_{id'}, \Gamma\}$ to CSP. Finally CSP returns *Proof* to DO and DO checks whether the Eq. 3 holds. If Eq. 3 holds, DO asks CSP to return relevant ciphertext C' to DU. Otherwise, DU gains $\perp$.

$$e(k'_{id'}, g_0) = e(g_1, X_2) e(u_0 \textstyle\prod_{i \in \mathcal{V}} u_i, g_0^{\pi_{id'}}) e(g_2, \Gamma). \tag{3}$$

Dec($\boldsymbol{C', sk_c, pk_c}$) DU decrypts the ciphertext C' through Eq. 4.

$$m = c_1 \cdot e(g^{\pi_{id}}, c_2) / e(k_{id}, c_0). \tag{4}$$

Remark To guarantee data confidentiality, unauthorized DU should not be able to access sensitive information, only DU passes the CSP's search operation and DO's verification he can gain the relevant search results. Furthermore, through removing the secure channel DOAS scheme can reduce the computational costs. Therefore, in the efficient DOAS scheme only authorized DU can pass the CSP's search operation and DO's verification. Unfortunately, in DOAS scheme DO needs to remain online, while DOAS scheme can effectively reduce the computational costs. Therefore, it is a tradeoff between computational burden and communication costs.

6 The analysis of DOAS scheme

6.1 Correctness

Only the DU's search token is correctly tested by CSP and his *Proof* information passes the DO's authorization he can gain the relevant ciphertext. So the correctness of **Search** and **Verify** algorithms can be checked as follow:

According to $\beta = h(e(\varphi, \theta)^{s_{id}}) = h(e(g^\lambda, \theta)^{s_{id}})$ and $\beta' = h(e(c_0, \theta)^\lambda) = h(e(g^{s_{id}}, \theta)^\lambda)$, we have $\beta' = \beta$. First, we show Eq. 2 holds when $w = w'$.

$$\begin{aligned} e(I_1^{\beta'}, \delta_{w'}) &= e(g^{(\mu - w) r_{id}}, (\tau g^{-\gamma_{w'}})^{1/\mu - w'}) \\ &= e(g, \tau)^{r_{id}} e(g, g^{-\gamma_{w'}})^{r_{id}}; \\ e(I_0, g)^{\gamma_{w'}} &= e(g^{r_{id}}, g)^{\gamma_{w'}}; \\ e(I_1^{\beta'}, \delta_{w'}) e(I_0, g)^{\gamma_{w'}} &= e(g, \tau)^{r_{id}} = e(g^{r_{id}}, \tau) = e(I_0, \tau). \end{aligned}$$

Second, We verify the legitimacy of DU if Eq. 3 holds.

$$\begin{aligned} e(k'_{id'}, g_0) &= e(k_{id'} g_2^\epsilon, g_0) = e(g_1^\alpha (u_0 \prod_{i \in \mathcal{V}} u_i)^{\pi_{id'}} g_2^\epsilon, g_0) \\ &= e(g_1, g_0^\alpha) e(u_0 \prod_{i \in \mathcal{V}} u_i, g_0^{\pi_{id'}}) e(g_2, g_0^\epsilon) \\ &= e(g_1, X_2) e(u_0 \prod_{i \in \mathcal{V}} u_i, g_0^{\pi_{id'}}) e(g_2, \Gamma). \end{aligned}$$

Finally, DU can decrypt the returned ciphertext through (4).

$$\begin{aligned} c_1 \cdot \frac{e(g^{\pi_{id}}, c_2)}{e(k_{id}, c_0)} &= m \cdot \frac{e(X_1, g_1)^{s_{id}} \cdot e(g^{\pi_{id}}, (u_0 \prod_{i \in \mathcal{V}} u_i)^{s_{id}})}{e(g_1^\alpha (u_0 \prod_{i \in \mathcal{V}} u_i)^{\pi_{id}}\}, g^{s_{id}})} \\ &= m \cdot \frac{e(g^\alpha, g_1)^{s_{id}}}{e(g_1^\alpha, g^{s_{id}})} = m. \end{aligned}$$

6.2 Security

In this section, we first analyze the security of our proposed DOAS scheme and show it coincides with the design goals presented in Section 4.

Theorem 1 *DOAS scheme is $(t, q_1, q_2, \epsilon(k))$ secure against indistinguishability chosen-plaintext attack (IND-CPA) if the $(t', \epsilon(k)')$-DBDH assumption holds in the bilinear map (e, p, G_1, G_2). Where $t' = t + \mathcal{O}(t), \epsilon(k)' = \frac{\epsilon(k)}{32(n+1)(q_1 + 2q_2)}$.*

Proof Similar to the proof process in IBE scheme [26], we assume that there exists an $(t, q_1, q_2, \epsilon(k))$-adversary $\mathcal{A}$ can break the IND-CPA security of DOAS scheme, then we construct a simulator $\mathcal{B}$ which can utilize $\mathcal{A}$ to break the DBDH assumption. Given security parameter k, the challenger $\mathcal{C}$ first generates the bilinear map parameters $(e, p, g, G_1, G_2) \leftarrow \mathcal{G}(1^k)$, then it flips a coin with $\varrho \in \{0, 1\}$. If $\varrho = 0$, it sends the tuple $(g^a, g^b, g^c, e(g, g)^{abc})$ to $\mathcal{B}$; if $\varrho = 1$, it returns the tuple $(g^a, g^b, g^c, e(g, g)^d)$ to $\mathcal{B}$, where $d \in \mathcal{Z}_q^*$. Finally, $\mathcal{B}$ outputs his guess ϱ' on ϱ.

Setup $\mathcal{B}$ first sets $\varpi = 4(q_1 + 2q_2)$ and selects a random integer $Int \in [n]$. Then he chooses two integrity vectors $\overrightarrow{\nu'} = \{\nu_1, ..., \nu_n\}$, $\overrightarrow{\mu'} = \{\mu_1, ..., \mu_n\}$, where $\nu_i \in [\varpi - 1]$, $\mu_i \in \mathcal{Z}_q^*$ for $i = 1, 2, ..., n$. Finally, he randomly selects

$\nu_0 \in [\varpi - 1]$, $\mu_0 \in \mathcal{Z}_q^*$ and defines three functions through Eq. 5.

$$f_1(id) = (q - \varpi Int) + \nu_0 + \sum_{i \in \mathcal{V}} \nu_i;\\ f_2(id) = \mu_0 + \sum_{i \in \mathcal{V}} \mu_i;\\ f_3(id) = \begin{cases} 0, \; if \, \nu_0 + \sum_{i \in \mathcal{V}} \nu_i \equiv 0 (mod \, \varpi) \\ 1, \; if \, \nu_0 + \sum_{i \in \mathcal{V}} \nu_i \not\equiv 0 (mod \, \varpi) \end{cases} . \tag{5}$$

Besides, $\mathcal{B}$ randomly selects $\vartheta \in \mathcal{Z}_q^*$, $g_2 \in G_1$, and sets $g^a = g^\alpha = X_1$, $g^b = g_1$, $g_0 = g^\vartheta$, $X_2 = g_0^\alpha = g^{\alpha\vartheta}$, $u_0 = g_1^{(q-\varpi Int)+\nu_0} g^{\mu_0}$, $u_i = g_1^{\nu_i} g^{\mu_i}$. Then $\mathcal{B}$ publishes the tuple $(e, q, G_1, G_2, g, g_0, g_1, g_2, X_1, X_2, \mathcal{U})$ and computes $g_1^\alpha = g^{b\alpha}$.

Phase 1 This phase includes the following two kind of queries:

Key generation query.: When $\mathcal{A}$ issues the key query for the DU with an identity id, $\mathcal{B}$ first verifies the equation $f_3(id) \overset{?}{=} 1$.

- If $f_3(id) = 1$ holds, $\mathcal{B}$ first randomly chooses $\pi \in \mathcal{Z}_q^*$, then he generates the DU's key as follows:

$$k_{id} = X_1^{\frac{-f_2(id)}{f_1(id)}} (\nu_0 \prod_{i \in \mathcal{V}} \nu_i)^\pi;\\ g^\pi = X_1^{\frac{-1}{f_1(id)}} g_0^\pi, g_0^\pi = g^{\vartheta\pi}. \tag{6}$$

Finally, $\mathcal{B}$ sends the tuple (k_{id}, g^π, g_0^π) to $\mathcal{A}$.

- If $f_3(id) = 0$ holds, $\mathcal{B}$ aborts this process and randomly returns his guess ϱ'. And we claim the key for DU with an identity id is generated correctly, which is shown by Eq. 7.

$$\begin{aligned} k_{id} &= X_1^{\frac{-f_2(id)}{f_1(id)}} (\mu_0 \prod_{i \in \mathcal{V}} \mu_i)^\pi \\ &= g^{\frac{-af_2(id)}{f_1(id)}} (g^{bf_1(id)+f_2(id)})^\pi \\ &= (g^{bf_1(id)+f_2(id)})^{\frac{-a}{f_1(id)}} g^{ab} (g^{bf_1(id)+f_2(id)})^\pi \\ &= (g^{bf_1(id)+f_2(id)})^{\pi - \frac{a}{f_1(id)}} g^{ab} \\ &= g_1^a (\mu_0 \prod_{i \in \mathcal{V}} \mu_i)^{\pi - \frac{a}{f_1(id)}}. \end{aligned} \tag{7}$$

Set $\widehat{\pi} = \pi - \frac{a}{f_1(id)}$ and gain Eq. 8.

$$k_{id} = g_1^a (\mu_0 \prod_{i \in \mathcal{V}} \mu_i)^{\pi - \frac{a}{f_1(id)}};\\ g^\pi = X_1^{\frac{-1}{f_1(id)}} g_0^\pi = g^{\pi - \frac{a}{f_1(id)}} = g^{\widehat{\pi}};\\ g_0^\pi = g^{\vartheta\pi} = g^{\vartheta\widehat{\pi}} = g_0^{\widehat{\pi}}. \tag{8}$$

Thus, the key for DU is correctly generated according to above analysis.

Verify query: $\mathcal{A}$ issues verify queries on (id, id'), then $\mathcal{B}$ first confirms whether he has generated keys for DUs with identities id and id'. If not, he needs to verify whether the equations $f_1(id) = 1$, $f_1(id') = 1$ hold.

If above two equations hold, $\mathcal{B}$ first computes the keys $k_{id}, k_{id'}$ for identities id, id', respectively. Then he selects a random element $\epsilon \in \mathcal{Z}_q^*$ and computes $k'_{id'} = k_{id'} g_2^\epsilon$, $\Gamma = g_0^\epsilon$. Finally, he returns $(k'_{id'}, \Gamma)$ to $\mathcal{A}$. Otherwise, he terminates this process and randomly outputs his guess ϱ'.

Challenge $\mathcal{A}$ issues an identity id^* and two messages m_0, m_1 which have equal length. Then $\mathcal{B}$ verifies $f_3(id^*) \overset{?}{=} 0$.

- If $f_3(id^*) = 1$, $\mathcal{B}$ terminates the subsequent processes and randomly outputs his guess ϱ'.
- If $f_3(id^*) = 0$, $\mathcal{B}$ first flips a coin with $b' \in \{0, 1\}$. Then he generates the ciphertext through Eq. 9,

$$c_0^* = g^c, c_1^* = m_{b'} e(g, g)^d,\\ c_2^* = g^{cf_2(id^*)} = u_0 \prod_{i \in \mathcal{V}^*} u_i^c. \tag{9}$$

Finally, $\mathcal{B}$ sends the ciphertext $C^* = \{c_0^*, c_1^*, c_2^*\}$ for the message $m_{b'}$ to $\mathcal{A}$.

Phase 2 The process of this phase is the same as **Phase 1** except for some restrictions as follows:

Key generation query. The key for the identity id^* cannot be issued by $\mathcal{A}$.

Verify query. $\mathcal{A}$ cannot issue verify query for the tuple (id, id', c_0^*) and key query for identity id^*.

Guess $\mathcal{A}$ first outputs his guess b'' on b'. If $b'' = b'$, $\mathcal{B}$ returns $\varrho' = 0$. Otherwise, $\varrho' = 1$.

From aforementioned processes we notice that the simulation of key generation verify queries is analogous to that in real protocol. And the $\mathcal{B}$ will not abort the simulation if and only if the key for DU is correctly generated and $f_3(id^*) = 0$. For q_1 key generation queries and q_2 verify queries, $\mathcal{B}$ is required to create at most $q_1 + 2q_2$. Therefore, $\mathcal{A}$ has at least $\epsilon(k)' = \frac{\epsilon(k)}{32(n+1)(q_1+2q_2)}$ advantage to break the DBDH assumption. This completes the proof of Theorem 1. □

Theorem 2 *DOAS scheme is secure against indistinguishability chosen-keyword attack (IND-CKA) without random oracle if truncated decisional p-ABDHE assumption is intractable.*

Proof Assume that the number of trapdoor queries p_k satisfies $p \geq p_k + 1$, and there exists a polynomial-time adversary $\mathcal{A}$ who can attack DOAS scheme without random oracle, then we construct a simulator $\mathcal{B}$ that can simulate the p-ABDHE game. The challenger $\mathcal{C}$ generates the bilinear map parameters (G_1, G_2, q, g, e), and a p-ABDHE instance $(g, g^a, g^{a^2}, ..., g^{a^p}, g^b, g^{ba^{p+2}}, \mathcal{D})$ is returned by $\mathcal{B}$, then $\mathcal{B}$'s goal is to distinguish $\mathcal{D} = e(g, g)^{ba^{p+1}}$ from a random element in G_2. Finally, the simulation proceeds are shown as follows:

Setup Given the security parameter k and public parameters (G_1, G_2, q, g, e, h), $\mathcal{B}$ first chooses random elements $\lambda \in \mathcal{Z}_q^*, \theta \in G_1$ and computes $\varphi = g^{\lambda}$, then he sets the public/secret key pair of CSP as $pk_s = (\varphi, \theta), sk_s = (\lambda)$, respectively. In the same way, $\mathcal{B}$ first chooses a random element μ and a p-degree polynomial $p(a)$; then he computes $\psi' = g^a, \tau = g^{p(a)}$; finally, he sends partial public key $pk_c = (\psi, \tau)$ of cloud client to $\mathcal{A}$.

Phase 1 $\mathcal{A}$ adaptively makes p_k trapdoor generation queries. When $\mathcal{A}$ issues the trapdoor query for the keyword w', $\mathcal{B}$ first computes $\gamma_{w'} = p(w'), \delta_{w'} = (\tau g^{-\gamma_{w'}})^{1/\mu - w'}$, then he sends $T_{w'} = \{\gamma_{w'}, \delta_{w'}\}$ to $\mathcal{A}$. If $p \geq p_k + 1$, then $\gamma_{w'}$ is a random element from the view of $\mathcal{A}$ as $p(x)$ is a random p-degree polynomial.

Challenge When **Phase 1** is over, $\mathcal{A}$ outputs two keywords w'_0, w'_1. Then $\mathcal{B}$ first selects a random element $b \in \{0, 1\}$, then he sets $w^* = w'_b, \gamma_{w^*} = p(w^*)$ and computes $\delta_{w^*} = g^{(p(a) - p(w^*))/(a - w^*)}$. Finally, he selects random elements $s_{id}^*, r_{id}^* \in \mathcal{Z}_q^*$ and computes $I_0^* = g^{s_{id}^*}, \beta^* = h(e(\varphi, \theta)^{s_{id}^*})$, he also defines a $(p+1)$-degree random polynomial $p^*(a) = (a^{p+2} - (w^*)^{p+2})/(a - w^*) = \sum_{i=0}^{p+1} p_i^*(a^i)$. Finally, he needs to compute Eq. 10.

$$\begin{aligned} I_1^* &= (g^{ba^{p+2}}(g^b)^{-(w')^{p+2}})^{1/\beta^*}, \\ e(g,g)^{r_{id}^*} &= \mathcal{D}^{p^*_{p+1}} e(g^b, \textstyle\prod_{i=0}^{p}(g^{a^i})^{p_i^*}), \\ e(g,\tau)^{r_{id}^*} &= e((I_1^*)^{\beta^*}, \delta_{w^*})(e(g,g)^{r_{id}^*})^{\gamma_{w^*}}. \end{aligned} \tag{10}$$

Besides, he sends the index (I_0^*, I_1^*) to $\mathcal{A}$. Set $r_{id}^* = bp^*(a)$, if $\mathcal{D} = e(g,g)^{ba^{p+1}}$, then Eq. 11 holds.

$$\begin{aligned} I_1^* &= (g^{ba^{p+2}}(g^b)^{-(w^*)^{p+2}})^{1/\beta^*} \\ &= g^{(a-w^*)(b(a^{p+1}-(w^*)^{p+2})/(a-w^*))^{1/\beta^*}} \\ &= g^{a-w^* r_{id}^*/\beta^*} = (\psi' g^{-w^*})^{r_{id}^*/\beta^*}. \end{aligned} \tag{11}$$

Phase 2 $\mathcal{A}$ repeatedly issues the trapdoor generation queries as **Phase 1** except for the keywords (w'_0, w'_1).

Guess $\mathcal{A}$ returns the guess b'. If $b' = b$, set $\mathcal{D} = e(g,g)^{ba^{p+1}}$; otherwise, set $\mathcal{D} = e(g,g)^c$.

If $\mathcal{D} = e(g,g)^{ba^{p+1}}$, we argue that the simulation is perfect, and $\mathcal{A}$ can correctly guess the bit b with probability $1/2 + \epsilon(k)$. Otherwise, the element $\mathcal{D}$ is uniformly distributed, and $(I_1^*, e(g,g)^{r_{id}^*})$ is a uniformly random and independent tuple. In this situation, the inequality $e(g,g)^{r_{id}^*} \neq (e(g, ((I_1^*)^{\beta^*}))^{1/a-w^*}$ holds with probability $1-1/q$. If the above inequality holds, the uniformly random element can be set through Eq. 12.

$$\begin{aligned} e(g,\tau)^{r_{id}^*} &= e((I_1^*)^{\beta^*}, \delta_{w^*})(e(g,g)^{r_{id}^*})^{\gamma_{w^*}} \\ &= \frac{e((I_1^*)^{\beta^*}, \tau^{1/a-w^*})(e(g,g)^{r_{id}^*})}{(e(g,(I_1^*)^{\beta^*}))^{1/a-w^*})^{\gamma_{w^*}}}. \end{aligned} \tag{12}$$

And this element is also independent from the view of $\mathcal{A}$ as the element γ_{w^*} is uniformly random. Especially, $\gamma_{w^*} = p(w^*)$ (except for $e(g,g)^{r_{id}^*}$) is random and independent from the view of $\mathcal{A}$ when $p \geq p_k + 1$. Therefore, $e(g,\tau)^{r_{id}^*}$ is a uniformly random and independent element. Additionally, I_0^* is also uniformly random and independent from the tuple $(I_1^*), e(g,g)^{r_{id}^*}, e(g,\tau)^{r_{id}^*})$ as $s_{id}^* \in \mathcal{Z}_q^*$ is randomly selected. Thus the index cannot reveal any valuable information regarding on the bit b. This completes the proof of Theorem 2. □

6.3 Performance

In this section, we evaluate the performance of DOAS scheme in terms of asymptotic computation complexity and its actual execution time through exploiting the Type A curves within the Paring Based Cryptography (PBC) library. The experiments are implemented on an Ubuntu 15.04 Server with Intel Core i5 Processor 2.3 GHz through using C and PBC Library. In PBC Library, the Type A is denoted as $E(F_q) : y^2 = x^3 + x$, G_1 is a subgroup of $E(F_q)$, and the cyclic group is a subgroup of $E(F_q)^2$, where q is a large prime number. The group order of G_1 is 160-bit, and the base field is 512-bit. Before showing the asymptotic computational complexity of DOAS scheme, we consider several computational operations (e.g. exponentiation operation e_1 in G_1, exponentiation operation e_2 in G_2, hash operation h_1 which maps a bit-string to an element of G_1, pairing operation p, and hash operation h_2 which maps a bit-string to an element of $\mathcal{Z}_q$, where h_2 is much more efficient than other operations).

To show the performance of our scheme, we compare it with the state-of-the-art schemes which are based on ABE, such as ABKS-UR scheme [10], CP-ABKS scheme [11],

Table 3 Asymptotic performance analysis

Scheme	**KeyGen**	**Enc**	**Trap**	**Search**
DOAS	$6e_1$	$5e_1 + 2e_2 + 2p + h_2$	$2e_1$	$4p + e_1 + 2e_2 + h_2$
CP-ABKS	$(2m+1)e_1 + mh_1$	$(2n+5)e_1 + nh_1 + h_2$	$(2m+4)e_1 + h_2$	$4p + me_1$
ABKS-UR	$(2n+1)e_1 + e_2$	$(n+1)e_1 + e_2$	$(2n+1)e_1$	$(n+1)p + e_2$

n: number of attributes in system; m: number of attributes submitted by DU etc. CP-ABKS scheme is selectively secure against chosen-keyword attack (CKA) in the generic bilinear group model, but its security is not based on the well-studied complexity-theoretic problem. Though ABKS-UR scheme can achieve stronger security in the standard model and provider data owner-enforced search authorization, the computational costs of its algorithms are affected by the number of attributes in system, which leads to heavy computational burden. Besides, the access permissions of these two schemes depend on the obscure attribute set rather than the specific identity. Thus, our scheme which can resist chosen-plaintext attack (CPA) and CKA without random oracle is feasible in practical applications. As a further contribution, we show that the actual performance of our scheme is more efficient than those of aforementioned two schemes through conducting empirical experiments over real-world dataset. Next, we give the asymptotic computational complexities of CP-ABKS scheme, ABKS-UR scheme and our scheme, as illustrated in Table 3.

In Table 3, we mainly take four algorithms into consideration, namely **KeyGen**, **Enc**, **Trap** and **Search** algorithms. For convenience, we set $n = 50, m = 10$. We notice that our scheme is more efficient than other two schemes as the computational costs of these two schemes are affected by the factors n, m, respectively, where $m \ll n$. However, ABKS-UR scheme will outperform other two schemes in **Trap** and **Search** algorithms when the number of submitted keywords is over a certain threshold, as our scheme and CP-ABKS scheme cannot support multi-keyword search. Therefore, our scheme is efficient and feasible in practical applications to some extent.

Next we conduct empirical study over a real-world dataset, namely email dataset,[1] to demonstrate the performance evaluation of aforementioned three schemes. For comparison, we set $m = 10, n = 50$. Besides, we randomly choose 10000 files (in which each file has 1000 distinct keyword fields) and run experiments for 100 times.

In Fig. 3, we demonstrate that the computational cost of **KeyGen** algorithm increases near linearly with the number of DUs, which ranges from 1 to 50. For comparison, we set $m = 10, n = 50$. As the key generation time of ABKS-UR scheme is affected by the number of attributes in access structure ($n = 50$) and that of CP-ABKS scheme depends the number of attributes submitted by DU ($m = 10$), our scheme is much more efficient than these two schemes.

In Fig. 4, we show the computational overhead of **Enc** algorithm (including ciphertext and index generation operations), and it increases with the number of data files, which ranges from 1 to 10000. As CP-ABKS scheme needs additional $(2n + 5)e_1$ and ABKS-UR scheme needs to conduct $(n + 1)$ exponentiation operations, our scheme has less computational burden than other two schemes, but ABKS-UR scheme outperforms CP-ABKS scheme. Though the **Enc** algorithm has more computational overhead than other algorithms, it will not affect user search experience as its computational burden exposed on DO is one-time cost. Therefore, our scheme is feasible and efficient in practice, especially for the resource-limited DO, such as sensor nodes and mobile terminals.

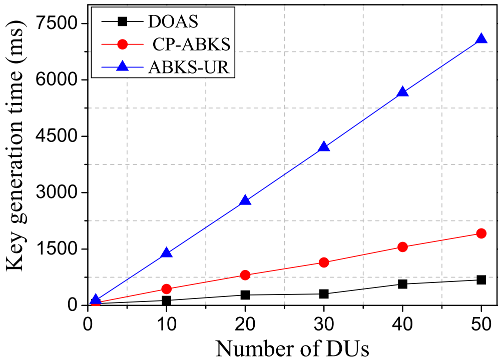

Fig. 3 Computational overhead of **KeyGen** algorithm

We present the computational overhead of **Trap** algorithm in the aforementioned three schemes in the Fig. 5. As ABKS-UR scheme can support conjunctive keyword search and h_2 is much more efficient than other operations, its computational cost remains unchanged. However, the computational costs of CP-ABKS scheme and our scheme increase with increasing the number of submitted keywords. When the number of submitted keywords is over 10, the computational burden of CP-ABKS scheme is heavier than that of ABKS-UR scheme, but our scheme is still less than ABKS-UR scheme. And when the number of queried keywords is close to 50, the trapdoor generation time of our scheme is the same as that ABKS-UR scheme. However, in practice, the number of queried keywords is

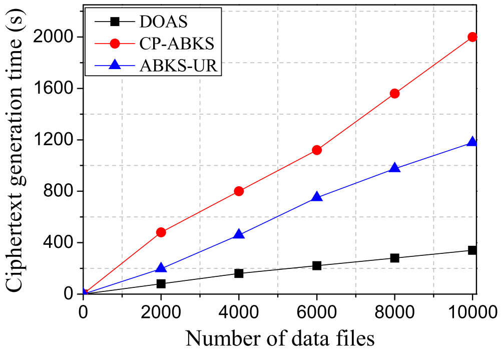

Fig. 4 Computational overhead of **Enc** algorithm

[1] http://www.cs.cmu.edu/~enron/

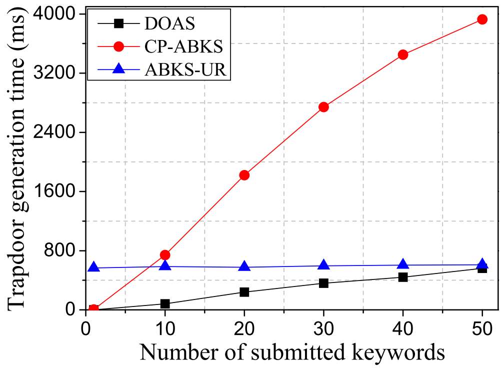

Fig. 5 Computational overhead of **Trap** algorithm

significantly smaller than 50. Thus, our scheme is efficient and acceptable in practice.

The actual execution time of **Search** algorithm in above three schemes is presented in Fig. 6, and we notice that the ciphertext search time of CP-ABKS scheme and our scheme vary with the number of submitted keywords, but the computational overhead of CP-ABKS scheme is slightly more than that of our scheme. When the number of queried keywords is over than 20, the computational burden of ABKS-UR scheme is less than that of CP-ABKS scheme and our scheme. As the search time of ABKS-UR scheme in Search algorithm is affected by the number of attributes in access structure rather than the number of submitted keywords, the computational cost of ABKS-UR scheme keeps unchanged. However, in reality, our scheme is still more efficient than ABKS-UR scheme when the number of submitted keywords is less than 20.

From above figures we notice that the actual performance evaluation is in complete accord with theoretical computation complexity which is shown in Table 3. Except for the above four algorithms, there exists **Verify** algorithm in DOAS scheme, and its computational overhead is $2e_1 + 4p$, which is acceptable in a broad range of applications. In other words, our scheme is efficient and feasible in practice.

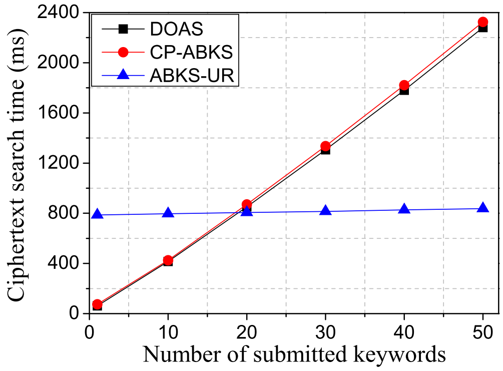

Fig. 6 Computational overhead of **Search** algorithm

7 Conclusions

In this paper we propose an efficient DOAS scheme in which DO can grant access permission according to his preference rather than the participation of third-party server, thus the unauthorized accesses can be prevented. Besides, DOAS scheme removes the secure channel to reduce the computational costs. Moreover, formal security analysis proves that DOAS scheme is secure against CPA and CKA without random oracle. As a further contribution, the empirical experiments over real-world dataset show that DOAS scheme is much more efficient than the state-of-the-art SE scheme which is based on ABE. Therefore, DOAS scheme is efficient and feasible in practice. As part of our future work, we need to eliminate the defect that DO needs to remain online, and further improve the search efficiency with supporting conjunctive keyword search.

Acknowledgments This work was supported by the National High Technology Research and Development Program (863 Program) (No. 2015AA016007, No. 2015AA017203), the Key Program of NSFC (No. U1405255, No. U1135002), the Changjiang Scholars and Innovation Research Team in University (No. IRT1078), the Fundamental Research Funds for the Center Universities (No. JY10000903001) and the Major Nature Science Foundation of China (No. 61370078, No. 61309016).

References

1. Jiang Q, Ma JF, Wei FS (2016) On the Security of a Privacy-Aware Authentication Scheme for Distributed Mobile Cloud Computing Services. IEEE Systems Journal. doi:10.1109/JSYST.2016.2574719
2. Jiang Q, Khan MK, Lu X, Ma JF, He DB (2016) A privacy preserving three-factor authentication protocol for e-health clouds. Journal of Supercomputing. doi:10.1007/s11227-015-1610-x
3. Ren YJ, Shen J, Wang J, Han J, Lee SY (2015) Mutual verifiable provable data auditing in public cloud storage. J Internet Technol 16(2):317–323
4. Wei LF, Zhu HJ, Cao ZF, Dong XL, Jia WW, Chen YL, Vasilakos A (2014) Security and privacy for storage and computation in cloud computing. Inf Sci 258:371–386
5. Yang B, Zhang MW, Du JQ (2016) An error-tolerant keyword search scheme based on public-key encryption in secure cloud computing. Concurr Comput Pract Exper 28(4):1083–1093
6. Boneh D, Crescenzo GD, Ostrovsky R, Persiano G (2004) Public key encryption with keyword search. In: International conference on the theory and applications of cryptographic techniques. Springer, pp 506–522
7. Dai SG, Li HG, Zhang FG (2016) Memory leakage-resilient searchable symmetric encryption. Fut Gen Comput Syst 62:76–84
8. Goyal V, Pandey O, Sahai A, Waters B (2006) Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on computer and communications security. ACM, pp 89–98

9. Bethencourt J, Sahai A, Waters B (2007) Ciphertext-policy attribute-based encryption. In: IEEE symposium on security and privacy. IEEE, pp 321–334
10. Sun WH, Yu SC, Lou WJ, Hou YT, Li H (2016) Protecting your right: Verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud. IEEE Trans Parallel Distrib Syst 27(4):1187–1198
11. Zheng Q, Xu SH, Ateniese EG (2014) Vabks: Verifiable attribute-based keyword search over outsourced encrypted data. In: IEEE International Conference on Computer Communications. IEEE, pp 522–530
12. Fang LM, Susilo W, Ge CP, Wang JD (2009) A secure channel free public key encryption with keyword search scheme without random oracle. In: International conference on cryptology and network security. Springer, pp 248–258
13. Miao YB, Ma JF, Wei FS, Liu ZQ, Wang XA, Lu CB (2016) VCSE: Verifiable conjunctive keywords search over encrypted data without secure-channel. Peer-to-Peer Network Appli: 1–13
14. Li HW, Liu DX, Dai YS, Luan TH, Shen XM (2015) Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage. IEEE Trans Emerg Topics Comput 3(1):127–138
15. Li HW, Liu DX, Dai YS, Luan TH (2015) Engineering searchable encryption of mobile cloud networks: When QoE meets QoP. IEEE Wireless Commun 22(4):74–80
16. Wen M, Lu RX, Zhang K, Lei JS, Liang XH, Shen XM (2013) PaRQ: A privacy-preserving range query scheme over encrypted metering data for smart grid. IEEE Trans Emerg Topics Comput 1(1):178–191
17. Wen M, Lu RX, Lei JS, Liang XH, Li HW, Shen XM (2013) ECQ: An efficient conjunctive query scheme over encrypted multidimensional data in smart grid. In: IEEE global communications conference. IEEE, pp 796–801
18. Fu ZJ, Ren K, Shu JG, Sun XM, Huang FX (2015) Enabling personalized search over encrypted outsourced data with efficiency improvement. IEEE Trans Parallel Distrib Syst. doi:10.1109/TPDS.2015.2506573
19. Wen M, Lei JS, Bi ZQ (2013) SSE: A secure searchable encryption scheme for urban sensing and querying. Int J Distrib Sensor Netw:2013
20. Miao YB, Ma JF, Liu ZQ (2016) Revocable and anonymous searchable encryption in multi-user setting. Concurr Comput Pract Exper 28(4):1204–1218
21. Yang Y, Ma MD (2016) Conjunctive keyword search with designated tester and timing enabled proxy re-encryption function for e-health clouds. IEEE Trans Inf Forens Secur 11(4):746–759
22. Li HW, Yang Y, Luan TH, Liang XH, Zhou L, Shen XM (2016) Enabling fine-grained multi-keyword search supporting classified subdictionaries over encrypted cloud data. IEEE Trans Depend Secur Comput 13(3):312–325
23. Fu ZJ, Sun XM, Liu Q, Zhou L, Shu JG (2015) Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing. IEICE Trans 98-B(1):190–200
24. Xia ZH, Wang XH, Sun XM, Wang Q (2016) A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE Trans Parallel Distrib Syst 27(2):340–352
25. Zhang W, Lin YP, Xiao S, Wu J, Zhou SW (2016) Privacy preserving ranked multi-keyword search for multiple data owners in cloud computing. IEEE Trans Comput 65(5):1566–1577
26. Waters B (2005) Efficient identity-based encryption without random oracles. In: Annual international conference on the theory and applications of cryptographic techniques. Springer, pp 114–127
27. Khader D (2007) Public key encryption with keyword search based on k-resilient IBE. In: International conference on computational science and its applications. Springer, pp 1086–1095
28. Tomida K, Doi H, Mohri M, Shiraishi Y (2015) Ciphertext divided anonymous HIBE and its transformation to identity-based encryption with keyword search. J Inf Process 23(5):562–569