symmetry

MDPI

Article

Towards Secure Data Retrieval for Multi-Tenant Architecture Using Attribute-Based Key Word Search

Hanshu Hong [1], Yunhao Xia [1] and Zhixin Sun [1,2,*]

[1] Key Lab of Broadband Wireless Communication and Sensor Network Technology, Ministry Education, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; hhskaka@163.com (H.H.); 2015070108@njupt.edu.cn (Y.X.)

[2] Institute of Modern Posts, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

* Correspondence: sunzx@njupt.edu.cn; Tel.: +86-025-83535108

Academic Editor: Vladimir Shpilrain

Received: 29 March 2017; Accepted: 13 June 2017; Published: 16 June 2017

Abstract: Searchable encryption mechanism and attribute-based encryption (ABE) are two effective tools for providing fine-grained data access control in the cloud. Researchers have also taken their advantages to present searchable encryption schemes based on ABE and have achieved significant results. However, most of the existing key word search schemes based on ABE lack the properties of key exposure protection and highly efficient key updating when key leakage happens. To better tackle these problems, we present a key insulated attribute-based data retrieval scheme with key word search (KI-ABDR-KS) for multi-tenant architecture. In our scheme, a data owner can make a self-centric access policy of the encrypted data. Only when the possessing attributes match with the policy can a receiver generate a valid trapdoor and search the ciphertext. The proposed KI-ABDR-KS also provides full security protection when key exposure happens, which can minimize the damage brought by key exposure. Furthermore, the system public parameters remain unchanged during the process of key updating; this will reduce the considerable overheads brought by parameters synchronization. Finally, our KI-ABDR-KS is proven to be secure under chosen-keyword attack and achieves better efficiency compared to existing works.

Keywords: attribute-based data retrieval; key word search; key exposure protection; multi-tenant architecture

1. Introduction

With the rapid development of computer science and telecommunication, users can now enjoy various services via the Internet such as online shopping, remote medical monitoring, etc. These services produce massive data, which may contain a great amount of sensitive data like cellphone numbers, accommodation addresses, etc. Thus, the confidentiality of these data should be highly protected [1]. Encryption is a promising method to provide security protection for these sensitive data. Through encryption, these data are transformed into ciphertexts and stored securely in data clusters. However, traditional encryption techniques will prevent some common operations on ciphertexts—especially in terms of searching. For instance, a data owner wants to share some important data with some receivers in the multi-tenant data center, but data receivers do not know the exact location where these data have been stored. Since these data have been transferred into ciphertexts, it is inconvenient for them to search these encrypted data and determine the exact file they want. Thus, how to enable data owners to encrypt their data and make them searchable is a challenging and practical problem. The key word search mechanism is a promising tool to satisfy this demand. A keyword search protocol usually involves the participation of three parties: uploader, storage server, and receiver. The interaction process of a keyword search usually involves three steps:

Firstly, the uploader generates the search index for the corresponding keywords and uploads them with the ciphertexts to the storage server.

Secondly, the receiver computes the trapdoor for the desired keywords and sends the trapdoor to the storage server.

Thirdly, the storage server checks if the trapdoor generated by the receiver corresponds with the search index. The ciphertexts are returned to the receiver on the condition that the trapdoor and the search index are matched.

The first keyword search scheme based on PKC (Public Key Cryptography) was presented by Boneh et al. [2] in 2004. Afterwards, many studies [3–5] have been presented to provide better performances, higher security level, and more advanced functions. Aside from these properties, fine-grained access managements are also important because an uploader can take this advantage to make self-centric access policies on their private data [6–9]. To better satisfy this demand, Sahai et al. presented attribute-based encryption (ABE) [10–12] which efficiently brings flexible access control. Researchers have also taken the advantages of ABE [13,14] and keyword search to present attribute-based keyword search schemes [15–18]. Until now, several schemes have achieved keyword search based on ABE, but the performance can still be further optimized. To begin with, although the proposed scheme can provide flexible revocation, they cannot minimize the damage when key exposure occurs. In multi-tenant architecture environments, the number of users is very large and key exposure seems inevitable. If key leakage happens, the confidentiality of the whole system will no longer exist. Further, in most of the existing schemes, there exist additional transmission overheads of key updating. Consequently, an attribute-based keyword search with key exposure protection mechanism and efficient key refreshing [8] urgently needs to be proposed.

In this paper, we aim to tackle the above problems and present a key insulated attribute-based data retrieval with key word search (KI-ABDR-KS) scheme for multi-tenant architecture. We achieve flexible self-centric search management by utilizing a CP-ABE (Ciphertext Policy Attribute Based Encryption) [12] mechanism. The data owner generates the index for ciphertext using a self-centric access policy, indicating what kinds of receivers are given the privileges to gain access to these encrypted data. The receiver generates the trapdoor for the desired keyword using the private key she owns [19,20]. The cloud server checks if the trapdoor generated by receiver corresponds with the search index. The ciphertexts are returned to the receiver on the condition that the trapdoor and the search index are matched. A key insulation mechanism [21] is introduced to guarantee full security if key leakage occurs and helps to realize highly efficient key updating [22].

The detailed contributions established in the article are as follows:

(1) We present a novel keyword search based on ABE with key exposure protection. In our scheme, a data owner can make self-centric access policy of the encrypted data. Only if the possessing attributes match with the policy can a receiver generate a valid trapdoor and search the ciphertext.
(2) The proposed scheme provides secure key exposure protection as well as both backward and forward security.
(3) In our scheme, the system lifespan is split up into several time periods. The public parameters of the cryptosystem remain unvaried during the whole lifespan, and users' private keys are refreshed termly. When key leakage occurs, a user's private keys shall be updated in a timely fashion to minimize the damage brought by key exposure.
(4) Our scheme achieves keyword semantic security under chosen keyword attack. Meanwhile, it is shown to be superior in terms of computation efficiency compared to existing works.

2. Related Works

2.1. Attribute-Based Cryptosystem

In a classical PKC mechanism, a user is given the right to make secure data shared with others in a private way based on their identities. However, it is not fully practical when data sharing is conducted

via a more expressive access policy. In some scenarios (e.g., cloud computing), the amount of users and private data may be enormous. Assuming that a data owner wants to share some sensitive data with certain users using traditional encryption methods, she may run encrypt algorithms many times, since each user's public key is unique and the encryption is inefficient.

ABE is a cryptographic notion supporting flexible data access control, and is equipped with many advantages. In ABE, the concept of "access policy" is introduced; only if the user's attributes suit with the policy can she complete decryption. A file owner may set a data-centric access policy without concern about the specific identity of each user in the system (note that the amount of users in the system may be very large). Consequently, ABE is a more effective tool for data protection in large data outsource platforms. Existing literatures related to ABE have achieved many results in terms of fine-grained access control [7,13], revocation [6], key abuse protection [9], etc. Researchers have also implemented ABE in several practical scenarios such as wireless communications, cloud computing [14], etc.

2.2. Attribute-Based Keyword Search

Attribute-based keyword search (ABKS) combines the advantages of ABE and searchable encryption and has been given attention from researchers all over the world. Han et al. in [15] proposed an attribute-based searchable encryption with key policy. Their scheme achieves flexible access control on the search indexes of ciphertext. However, the proposed scheme directly sends the users' private keys to the file server as the trapdoor. This results in key exposure to the file server. If the server becomes dishonest or is being attacked, all of the legal private keys will be obtained by the attackers, which will bring huge damage to the whole cryptosystem. Yang in [16] designed a keyword search scheme based on ABE and applied it to an electronic health system. The proposed scheme supports fine-grained authorization and flexible revocation in the semi-trusted cloud server. However, the scheme generates a unique additional key pair for each user in the system. The generation of a search index also involves the public key of each user; this will bring a considerable computation burden when the amount of users is large. Sun et al. in [17] presented a novel searchable encryption for cloud computing based on CP-ABE. Their scheme provides self-centric search authorization as well as authenticity check over the encrypted data. The proposed scheme also achieves selective confidentiality under chosen keyword attack and secure revocation. Zheng et al. in [18] proposed a verifiable keyword search scheme. Their scheme permits users with promising credentials to search the ciphertext using the generated trapdoor. Their scheme can also distinguish if a server has honestly carried out the tasks which are sent by users. Miao et al. in [23] applied ABKS to modern medical systems and demonstrated the high efficiency and security of their scheme. Zhou et al. in [24] presented a novel type of ABKS which supports both online and offline decryption; thus, it was equipped with better flexibility. Wang et al. in [25] did some path breaking work in terms of introducing the attribute and keywords vector to optimize the decryption efficiency. Dong et al. in [26] proposed a lightweight ABKS scheme, the application of which is very appropriate to networks with constrained computation resources (e.g., mobile networks). Li et al. in [27] tackled the search authorization issue in the cloud and designed a secure ABKS scheme which not only achieves trapdoor unlinkability and confidentiality, but also resists collusion attack. Vahid et al. in [28] combined attribute-based cryptography with fuzzy search token techniques and presented a novel ABKS scheme. They also proved it to be secure under keyword guessing attack.

The existing works mentioned above have achieved significant progress in attribute-based cryptosystems and keyword search mechanisms. However, these schemes lack the security protection mechanism when key exposure happens. In a large data outsourcing system with multiple users, key exposure seems unavoidable. Once it is leaked, any user obtaining the private key can generate a legal trapdoor and the confidentiality of the whole system will no longer exist. Thus, it is essential to carryout key exposure protection for attribute-based keyword search schemes.

3. Models and Definitions

3.1. Framework of KI-ABDR-KS

The system framework of our scheme is illustrated in Figure 1. It contains four entities: attribute authority (AA), multi-tenant server, data owner, and data receiver. AA manages universal attributes and distributes users' private keys. It is also responsible for updating users' temporal private keys when the cryptosystem enters into a new time period. The data owner generates a secure index for each ciphertext using a self-centric policy, while the data receiver generates a trapdoor for the required ciphertext according to the desired keywords. The multi-tenant server provides secure storage services for the encrypted data and responses to receivers' requests if the trapdoors are valid.

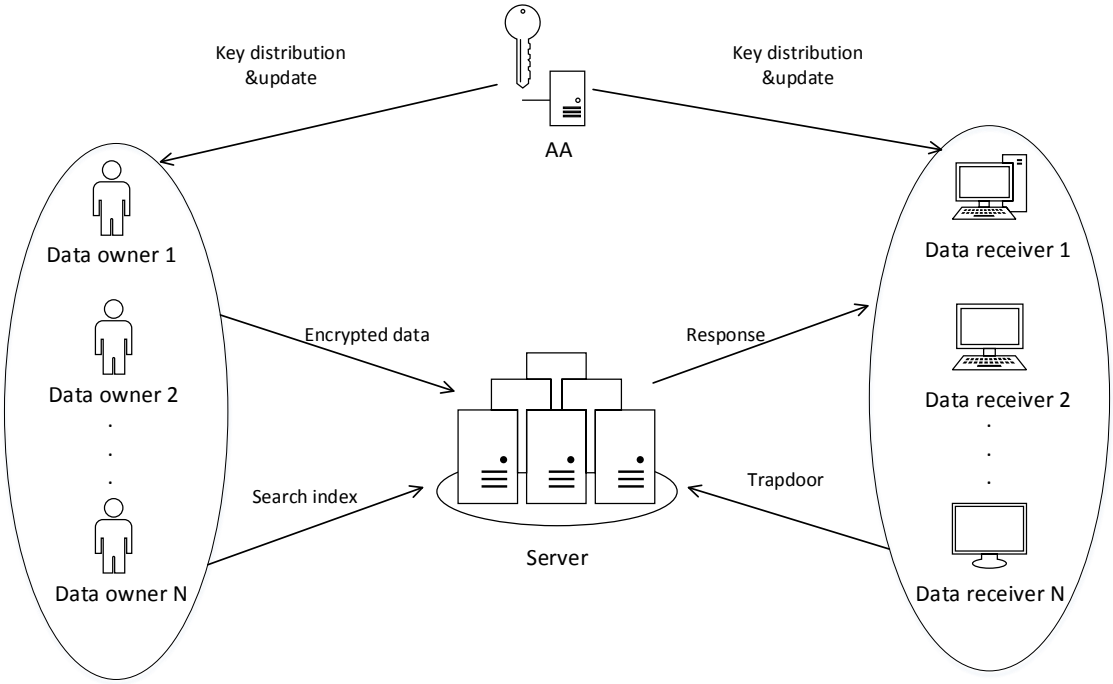

Figure 1. Framework of our scheme. AA: attribute authority.

3.2. Formulized Definitions of KI-ABDR-KS

In this section, we will give the interactions between entitites illustrated in Figure 1 and the formulized definitions of the algorithms. The proposed scheme contains seven algorithms, as below:

Setup: This algorithm is run by AA. It takes a security number as input and outputs system public parameters as well as master keys.

Key generation: This algorithm is run by AA. It takes system parameters, the initial time period, and the attribute set a user owns as input; it outputs the master key of key helper and the initial private key for a user.

Key update: This algorithm is run by AA. It takes system parameters and the newest time period as input. It outputs the key updating component for a user.

User update: This algorithm is run by the users. It takes the temporal private key of the previous period and key updating component as input, and it outputs the temporal private key at the latest version.

Search index generation : This algorithm is run by the data owner. It takes system parameters, an access structure, and key words as input; it outputs an index for a ciphertext.

Trapdoor : This algorithm is run by the users. It takes users' private keys and key word as input; it outputs a trapdoor.

Test : This algorithm is run by the server. It takes users' trapdoor as input and outputs the corresponding ciphertext.

3.3. Security Requirements

(1) Keyword semantic security: This security property guarantees that an *Adversary* cannot obtain the ciphertext without the valid trapdoor. In this paper, the requirement of key semantic security can be proved by a game described as follows:

Step 1 *Setup* :

Challenger runs *Setup* to obtain the related parameters in the game.

Adversary claims an access structure γ_{ic} and $\{A_{ic}\}$ is the attribute set involved.

Step 2 *Trapdoor queries* :

Trapdoor queries : query: *Challenger* can obtain the trapdoor of several keywords for attribute set S by running *Trapdoor* algorithm and sends the results back to *Adversary*. Note that $|S \cap \{A_{ic}\}| < thr_x$.

Note that the trapdoor queries contain the implication of private key generation query.

Step 3 *Challenge* :

At the current time period TP_n, *Adversary* picks w_0 and w_1, which have not been queried before. *Challenger* picks $\sigma \in \{0,1\}$ and runs *Search index generation* algorithm to obtain SI_σ.

Adversary outputs σ^* as a guess of σ. If $\sigma^* = \sigma$, then *Adversary* wins the game.

The advantage of *Adversary* can be denoted by $Adv(A) = \left|\Pr[\sigma^* = \sigma] - \frac{1}{2}\right|$.

(2) Backward and forward security: This security property guarantees the system's security and confidentiality when key exposure happens.

4. Concrete Constructions

In this section, we will provide the concrete algorithms from the system level viewpoint. These algorithms are the concrete and detailed expansions of the formulized definitions in Section 3.2 based on the above defined algorithms. *Setup* : Define two p order groups G_1, G_2. Let $\hat{e} : G_1 \times G_1 \to G_2$ be a bilinear pairing and g is a generator of G_1. Define a global attribute set $\{A_i\}$. Define hash functions: $H_1 : \{0,1\}^* \to G_1$, $H_2 : \{0,1\}^* \to Z_p$. AA randomly chooses secret numbers $y, h \in Z_p^*$ and computes $Y = \hat{e}(g,g)^y, g^h$. The system masker keys are $\{g^y, h\}$ while system public parameters are $\left\{g, p,\ G_1, G_2, \hat{e}, H_1,\ H_2, Y, g^h\right\}$.

Key generation : At the initial time period l_0, for a user possessing attribute set $\{A_i\}$, AA picks $r \in Z_p^*$ and calculates $D_1 = g^{\frac{y-r}{h}}, D_{i,0} = g^r H_1(A_i, l_0)^h$. The initial private key of a user is denoted by $\{D_1, D_{i,0}\}$. Note that D_1 remains unchanged throughout the whole system lifetime, while $D_{i,m}$ updates when system enters a new time period.

Key update : When the system arrives in a new period from l_m to l_{m+1}, AA computes the key updating component $UP_{m+1} = \left(\frac{H_1(A_i, l_{m+1})}{H_1(A_i, l_m)}\right)^h$ and sends the result to the user. Then, a user updates her temporal private key by calculating $D_{i,m+1} = D_{i,m} \cdot UP_{m+1} = g^r H_1(A_i, l_m)^h \cdot \left(\frac{H_1(A_i, l_{m+1})}{H_1(A_i, l_m)}\right)^h = g^r H_1(A_i, l_{m+1})^h$ (D_1 remains unchanged).

Search index generation : Data owner picks $s \in Z_p^*$ and chooses a polynomial q_x for each node x in the access control structure γ. Let the threshold value of the node be one more than the degree of q_x. For the root node, the data owner sets $q_{root}(0) = s$. For others, let $q_x(0) = q_{parent(x)} index(x)$. Denote $\{i\}$ to be the leaf nodes in γ, then the search index SI is constructed as:

$$IN_0 = Y^{sH_2(w)}, IN_1 = g^{hs},\ IN_{2,i} = g^{q_i(0)},\ IN_{3,i} = H_1(A_i, l_m)^{q_i(0)H_2(w)} SI : \{IN_0, IN_1, IN_{2,i}, IN_{3,i}\} \quad (1)$$

Trapdoor : For the desired keyword w, the data receiver picks a random number $x \in Z_p^*$ and calculates the trapdoor TR as Equation (2):

$$TR_1 = (D_1 \cdot g^{-x})^{H_2(w)} = g^{(\frac{y-r}{h}-x)H_2(w)} TR_{2,i} = \left(D_{i,m} \cdot g^{hx}\right)^{H_2(w)} = g^{(r+hx)H_2(w)} H_1(A_i, l_m)^{hH_2(w)} \quad (2)$$

Then, the data receiver sends $TR = \{TR_1, TR_{2,i}\}$ to the cloud server.

Test : The cloud server tests:

$$\hat{e}(IN_1, TR_1) \cdot \prod_{i \in \gamma} \frac{\hat{e}(TR_{2,i}, IN_{2,i})}{\hat{e}\left(IN_{3,i}, g^h\right)} = IN_0 \quad (3)$$

If Equation (3) is set up, the cloud server sends the corresponding ciphertext to the data receiver.

Correctness proof:

$$\begin{aligned} & \hat{e}(IN_1, TR_1) \cdot \prod_{i \in \gamma} \frac{\hat{e}\left(TR_{2,i}, IN_{2,i}\right)}{\hat{e}\left(IN_{3,i}, g^h\right)} \\ = & \hat{e}\left(g^{hs}, g^{(\frac{y-r}{h}-x)H_2(w)}\right) \cdot \prod_{i \in \gamma} \frac{\hat{e}\left(g^{(r+hx)H_2(w)} H_1(A_i, l_m)^{hH_2(w)}, g^{q_i(0)}\right)}{\hat{e}\left(H_1(A_i, l_m)^{q_i(0)H_2(w)}, g^h\right)} \\ = & \hat{e}(g,g)^{syH_2(w)} \hat{e}(g,g)^{-(sr+shx)H_2(w)} \cdot \hat{e}(g,g)^{srH_2(w)+shxH_2(w)} \\ = & \hat{e}(g,g)^{syH_2(w)} \\ = & IN_0 \end{aligned} \quad (4)$$

5. Discussion

5.1. Keyword Semantic Security

Before giving our proof, we first give the hardness assumption [17] that our scheme relies on:

Decision bilinear Diffie–Hellman assumption (DBDH): Picks random numbers $a, b, c, z \in Z_q^*$, assuming that the value of (g, g^a, g^b, g^c, z) are given, no probabilistic polynomial-time algorithm can distinguish the tuples $(A = g^a, B = g^b, C = g^c, \hat{e}(g,g)^{abc})$ and $(A = g^a, B = g^b, C = g^c, \hat{e}(g,g)^z)$ with a non-negligible probability.

Theorem 1. *Our KI-ABDR-KS is keyword semantic secure if the DBDH hardness assumption holds.*

Proof. If our scheme can be broken by an *Adversary* with advantage of ε, then a simulator can be constructed to break the DBDH hardness assumption with an advantage of $\frac{\varepsilon}{2}$. The challenge game is described as follows:

Setup :

Let G_1 and G_2 be two cyclic groups with prime order p. Denote g as the generator of G_1. Let $\hat{e} : G_1 \times G_1 \to G_2$ be a bilinear pairing. Define a global attribute set $\{A_i\}$. Define hash functions $H_1 : \{0,1\}^* \to G_1$, $H_2 : \{0,1\}^* \to Z_p$. *Challenger* picks $\sigma \in \{0,1\}$, $a, b, c \in Z_p^*$ and sets:

$$\begin{cases} (A, B, C, Z) = \left(g^a, g^b, g^c, \hat{e}(g,g)^{abc}\right) \; if \; \sigma = 0 \\ (A, B, C, Z) = \left(g^a, g^b, g^c, \hat{e}(g,g)^z\right) \; if \; \sigma = 1 \end{cases} \quad (5)$$

The aim of the simulator is to guess the value of σ.

Adversary claims a challenging access structure γ (containing attribute set S) and plays the game on it.

Trapdoor queries :

When *Adversary* makes a trapdoor query for keyword w_q on attribute set $\{A_q\}$, the simulator responds as follows:

Simulator picks $h, r, u \in Z_p^*$, sets:

$$\begin{cases} D_1{}' = g^{\frac{ab-r}{h}}, D_{i,m}{}' = g^r H_1(A_i, l_m)^h, if\ A_q \in S \\ D_1{}' = g^{\frac{u-r}{h}}, D_{i,m}{}' = g^r H_1(A_i, l_m)^h, if\ A_q \notin S \end{cases} \tag{6}$$

Then, the trapdoor is constructed as:

$$\begin{cases} TR_1{}' = g^{(\frac{ab-r}{h}-x)H_2(w_q)}, TR_{2,i}{}' = \left(D_{i,m}{}' \cdot g^{hx}\right)^{H_2(w_q)}, \text{if } A_q \in S \\ TR_1{}' = g^{(\frac{u-r}{h}-x)H_2(w_q)}, TR_{2,i}{}' = \left(D_{i,m}{}' \cdot g^{hx}\right)^{H_2(w_q)}, \text{if } A_q \notin S \end{cases} \tag{7}$$

Note that the trapdoor queries contain the implication of private key generation query.

Challenge:

Adversary picks key words w_0, w_1. Simulator chooses $\in \{0, 1\}$, picks $s \in Z_p^*$, and calculates the following information:

If $\sigma = 0$, sets:

$$IN_{0,\sigma} = \hat{e}(g,g)^{absH_2(w_\sigma)}, IN_1 = g^s,\ IN_{2,i} = g^{q_i(0)},\ IN_{3,i} = H_1(A_i, l_m)^{q_i(0)H_2(w)} \tag{8}$$

If $\sigma = 1$, sets:

$$IN_{0,\sigma} = \hat{e}(g,g)^{zH_2(w_\sigma)}, IN_1 = g^s,\ IN_{2,i} = g^{q_i(0)},\ IN_{3,i} = H_1(A_i, l_m)^{q_i(0)H_2(w)} \tag{9}$$

Simulator sends the above indexes $IN_{0,\sigma}$ to *Adversary*.

Let $g^s = g^c$, so we have:

$$IN_{0,\sigma} = \begin{cases} \hat{e}(g,g)^{abcH_2(w_\sigma)}\ if\ \sigma = 0 \\ \hat{e}(g,g)^{zH_2(w_\sigma)}\ if\ \sigma = 1 \end{cases} \tag{10}$$

Adversary outputs a value σ^*. If $\sigma^* = \sigma$, *Adversary* wins the game.

Next, we will analyze the simulator's advantage in distinguishing the tuples in DBDH assumption.

If $\sigma = 1$, E is an invalid search index and *Adversary* guesses randomly,

$$Pr(\sigma^* \neq \sigma | \sigma = 1) = \frac{1}{2} \tag{11}$$

If $\sigma = 0$, E is a valid index. According to the definition, *Adversary* has an advantage ε.

$$Pr(\sigma^* = \sigma | \sigma = 0) = \frac{1}{2} + \varepsilon \tag{12}$$

From what has been discussed, the simulator's advantage can be denoted by:

$$\begin{array}{c} \frac{1}{2}Pr(\sigma^* = \sigma | \sigma = 0) + \frac{1}{2}Pr(\sigma^* = \sigma | \sigma = 1) - \frac{1}{2} \\ = \frac{1}{2}\left(\frac{1}{2} + \varepsilon\right) + \frac{1}{2} \times \frac{1}{2} - \frac{1}{2} \\ = \frac{\varepsilon}{2} \end{array} \tag{13}$$

□

5.2. Users' Privacy and Trapdoor Unlinkability

In our scheme, the users' privacy can be highly protected. According to the *Trapdoor* algorithm in our KI-ABDR-KS, a secret component x is embedded into the trapdoor. Thus, the service provider cannot obtain any sensitive information of the private key. Besides, since the secret component

x is chosen by different users at random, it is computationally infeasible for cloud severs to distinguish different trapdoors containing the same key words, which meets the security demand of trapdoor unlinkability.

5.3. Forward and Backward Security

Our scheme can provide protection when key exposure happens. When key exposure happens at period l_{m-1}, the system can still maintain its security by updating users' temporal private keys to l_m version. A user's private key leakage during l_m will not harm the security in the rest time periods. Our scheme also supports random access key updating, since attribute authority is capable of updating users' temporal private keys from any previous time periods (denote these time periods by l_f) to the last version in just one step by calculating $D_{i,m} = g^r H_1\left(A_i, l_f\right)^h \cdot \left(\frac{H_1(A_i,l_m)}{H_1(A_i,l_f)}\right)^h = g^r H_1(A_i, l_m)^h$.

5.4. Efficient Key Updating with Constant Size of Parameters

The process of key updating in the proposed KI-ABDR-KS is very efficient because when a new time period arrives, only partial key components have to be refreshed. According to the *Key update* algorithm, the calculation of key updating component UP_m only takes one exponentiation. More importantly, though users' private keys are updated periodically, the system public parameters remain the same throughout the whole lifetime. This will reduce the considerable computation cost which parameter synchronization brings about.

5.5. Performance Evaluation

We compare our scheme with schemes in [17,23,28], which also implement attribute-based cryptosystem to achieve flexible key word search. The comparison is conducted with regard to the computation cost of each algorithm. Denote "Pair", "Exp" to be the bilinear pairing and exponential operations, respectively, and "n" is the amount of attributes involved. The results are listed in Table 1.

Table 1. Comparison results. KI-ABDR-KS: key insulated attribute-based data retrieval with key word search.

Algorithm	Scheme in [17]	Scheme in [23]	Scheme in [28]	Our KI-ABDR-KS
Setup	$(3n + 1)$ Exp + 1 Pair	3 Exp	3 Exp	2 Exp + 1 Pair
Key generation	$(2n + 3)$ Exp	$(2n + 2)$ Exp	$(2n + 1)$ Exp	$(n + 2)$ Exp
Search index generation	$(n + 2)$ Exp	$(2n + 3)$ Exp	$(2n + 6)$ Exp	$(2n + 2)$ Exp
Trapdoor	$(2n + 1)$ Exp	$(2n + 2)$ Exp	$(2n + 6)$ Exp	$(n + 3)$ Exp
Test	$(n + 1)$ Pair + 1 Exp	$(2n + 2)$ Pair	$(2n + 2)$ Pair	$(2n + 1)$ Pair
Key update	$2n$ Exp	-	-	n Exp

From comparison, it can be seen that efficiency of *Setup*, *Key generation*, *Trapdoor*, and *Key update* are higher in our scheme. The *Test* algorithm takes more exponential operations in our scheme, but it is run by the cloud server which has large computation capacity. Thus, this will not add a computation burden on the user side. In the scheme found in Reference [17], the access structure only supports AND gate, but our scheme provides a more flexible access structure which supports AND along with OR gate; thus, the *Search Index generation* algorithm in our scheme takes more exponential operations. Furthermore, unlike [23,28], our scheme is equipped with the function of highly efficient key updating. The system public parameters remain constant regardless of the number of attributes in the system and do not need to be changed during the process of key updating; this will reduce the considerable overheads brought by parameters synchronization. Consequently, our scheme has a better performance from the prospective of the overall efficiency.

6. Conclusions

In this paper, we propose a novel key insulated attribute-based data retrieval with keyword search mechanism. The proposed scheme can provide self-centric search indexes for the encrypted data. The proposed scheme also provides secure key exposure protection and full security when key exposure happens. By performance analysis, our scheme is of high-level security and is superior with respect to computation efficiency.

Acknowledgments: This research is supported by the National Natural Science Foundation of China (61373135 and 61672299).

Author Contributions: Hanshu Hong carries out the research of this paper. Yunhao Xia and Zhixin Sun check and revise the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kim, S.-H.; Lee, I.-Y. Study on user authority management for safe data protection in cloud computing environments. *Symmetry* **2015**, *7*, 269–283. [CrossRef]
2. Boneh, D.; Di Crescenzo, G.; Ostrovsky, R.; Persiano, G. Public key encryption with keyword search. In Proceedings of the 23rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004; pp. 506–522.
3. Sun, W.H.; Wang, B.; Cao, N.; Li, M.; Lou, W.; Hou, Y.T.; Li, H. Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking. In Proceedings of the ACM 8th Symposium on Information, Computer and Communications Security, Hangzhou, China, 8–10 May 2013; pp. 71–82.
4. Li, M.; Yu, S.C.; Cao, N.; Lou, W. Authorized private keyword search over encrypted data in cloud computing. In Proceedings of the IEEE 31th International Conference on Distributed Computing Systems, Minneapolis, MN, USA, 20–24 June 2011; pp. 383–392.
5. Li, J.; Liu, C.; Zhou, R.; Wang, W. Top-k keyword search over probabilistic XML data. In Proceedings of the IEEE 27th International Conference on Data Engineering, Hannover, Germany, 11–16 April 2011; pp. 673–684.
6. Fu, X.B.; Nie, X.Y.; Li, F.G. Black box traceable ciphertext policy attribute-based encryption scheme. *Information* **2015**, *6*, 481–493. [CrossRef]
7. Ying, Z.B.; Li, H.; Ma, J.F.; Zhang, J.; Cui, J. Adaptively secure ciphertext-policy attribute-based encryption with dynamic policy updating. *Sci. China Inform. Sci.* **2016**, *59*, 1–16. [CrossRef]
8. Hong, H.S.; Sun, Z. High efficient key-insulated attribute based encryption scheme without bilinear pairing operations. *Springerplus* **2016**, *5*, 131. [CrossRef] [PubMed]
9. Wang, Y.T.; Chen, K.F.; Long, Y. Accountable authority key policy attribute-based encryption. *Sci. China Inform. Sci.* **2012**, *55*, 1631–1638. [CrossRef]
10. Jiang, S.R.; Zhu, X.Y.; Wang, L.M. EPPS: Efficient and privacy-preserving personal health information sharing in mobile healthcare social networks. *Sensors* **2015**, *15*, 22419–22438. [CrossRef] [PubMed]
11. Goyal, V.; Pandey, O.; Sahai, A.; Waters, B. Attribute based encryption for fine-grained access control of encrypted data. In Proceedings of the ACM 13th conference on Computer and Communications Security, Alexandria, VA, USA, 30 October–3 November 2006; pp. 89–98.
12. Waters, B. Ciphertext policy attribute based encryption: An expressive, efficient, and provably secure realization. In Proceedings of the 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, 6–9 March 2011; pp. 53–70.
13. Lewko, A.; Okamoto, T.; Sahai, A.; Takashima, K.; Waters, B. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Riviera, French, 30 May–3 June 2010; pp. 62–91.
14. Yu, S.C.; Wang, C.; Ren, K.; Lou, W. Achieving secure, scalable, and fine-grained data access control in cloud computing. In Proceedings of the IEEE 29th International Conference on Infocom, San Diego, CA, USA, 14–19 March 2010; pp. 1–9.
15. Han, F.; Qin, J.; Zhao, H.W.; Hu, J. A general transformation from KP-ABE to searchable encryption. *Future Gener. Comput. Syst.* **2014**, *30*, 107–115. [CrossRef]

16. Yang, Y. Attribute-based data retrieval with semantic keyword search for e-health cloud. *J. Cloud Comput.* **2015**, *4*, 16. [CrossRef]
17. Sun, W.H.; Yu, S.C.; Lou, W.J.; Hou, Y.T.; Li, H. Protecting your right: Verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud. *IEEE Trans. Parallel Distrib. Systm.* **2016**, *27*, 1187–1198. [CrossRef]
18. Zheng, Q.J.; Xu, S.H.; Ateniese, G. Vabks: Verifiable attribute-based keyword search over outsourced encrypted data. In Proceedings of the IEEE 33rd International Conference on Infocom, Toronto, ON, Canada, 27 April–2 May 2014; pp. 522–530.
19. Gao, N.; Deng, Z.H.; Lü, S.L. XDist: An effective XML keyword search system with re-ranking model based on keyword distribution. *Sci. China Inform. Sci.* **2014**, *57*, 1–17. [CrossRef]
20. Li, Q.; Liu, X.M.; Ma, J.F.; Li, R.; Xiong, J. Provably secure unbounded multi-authority ciphertext-policy attribute-based encryption. *Secur. Commun. Netw.* **2015**, *8*, 4098–4109. [CrossRef]
21. Wen, J.; Li, X.X.; Chen, K.F.; Ma, C. Identity-based parallel key-insulated signature without random oracles. *J. Inform. Sci. Eng.* **2008**, *24*, 1143–1157.
22. Li, J.Z.; Zhang, L. Attribute-based keyword search and data access control in cloud. In Proceedings of the IEEE 10th International Conference on Computational Intelligence and Security, Kunming, China, 15–16 November 2014; pp. 382–386.
23. Miao, Y.; Ma, J.F.; Liu, X.M.; Wei, F.; Liu, Z.; Wang, X.A. m2-ABKS: Attribute-based multi-keyword search over encrypted personal health records in multi-owner setting. *J. Med. Syst.* **2016**, *40*, 246. [CrossRef] [PubMed]
24. Zhou, P.L.; Liu, Z.H.; Duan, S.H. Flexible attribute-based keyword search via two access policies. In Proceedings of the BWCCA 2016, Advances on Broad-Band Wireless Computing, Communication and Applications, Asan, Korea, 5–7 November 2016; pp. 815–822.
25. Wang, H.W.; Li, J.Q.; Yang, Y.L.; Ming, Z. Attribute-based and keywords vector searchable public key encryption. In Proceedings of the Smart Computing and Communication, SmartCom, Shenzhen, China, 17–19 December 2016; pp. 317–326.
26. Dong, Q.X.; Guan, Z.; Chen, Z. Attribute-based keyword search efficiency enhancement via an online/offline approach. In Proceedings of the IEEE 21st International Conference on Parallel and Distributed Systems, Melbourne, VIC, Australia, 14–17 December 2015; pp. 298–305.
27. Li, H.W.; Liu, D.X.; Jia, K.; Lin, X. Achieving authorized and ranked multi-keyword search over encrypted cloud data. In Proceedings of the IEEE International Conference on Communications, London, UK, 8–12 June 2015; pp. 7450–7455.
28. Yousefipoor, V.; Ameri, M.H.; Mohajeri, J.; Eghlidos, T. A secure attribute based keyword search scheme against keyword guessing attack. In Proceedings of the IEEE Communication and Information Systems Security Symposium, Tehran, Iran, 27–28 September 2016; pp. 124–128.