IEICE TRANS. FUNDAMENTALS, VOL.E103–A, NO.1 JANUARY 2020

PAPER Special Section on Cryptography and Information Security

Searchable Public Key Encryption Supporting Simple Boolean Keywords Search

Yu ZHANG[†a)], ***Member***, **Yansong ZHAO**[†], **Yifan WANG**[††], ***and*** **Yin LI**[†], ***Nonmembers***

SUMMARY Searchable encryption with advanced query function is an important technique in today's cloud environment. To date, in the public key setting, the best query function supported by the previous schemes are conjunctive or disjunctive keyword search, which are elementary but not enough to satisfy the user's query requirements. In this paper, we make a progress for constructing a searchable public key encryption scheme with advanced query function called simple Boolean keyword search. To create our scheme, we proposed a keywords conversion method that projects the index and query keywords into a group of vectors. Based on a combination of these obtained vectors and an adaptively secure inner product encryption scheme, a public key encryption with simple Boolean keyword search scheme is proposed. We also present both theoretical and experimental analysis to show the effectiveness of this scheme. To the best of our knowledge, it is the first time to give a searchable public key encryption scheme supporting queries like $q_1\ op_1\ q_2\ op_2\ \cdots\ op_{i-1}\ q_i\ op_i\ \cdots\ op_{n-1}\ q_n$, where op_i is a logical operator which can be *and*($\wedge$) or *or*($\vee$) and q_i is a keyword.

key words: *public key encryption, Boolean keywords search, searchable encryption*

1. Introduction

During recent years, cloud computing has received extensive attention from academic and industrial communities due to its efficiency and convenience. A large amount of organizations and users now are willing to outsource their data to the public cloud. Unfortunately, there are certain privacy risks as the service provider can access the data freely. The common method of preserving data privacy and security is encrypting the data before uploading it to the cloud. However, this simple approach brings a new issue of how to search these encrypted data since it is difficult to apply the search techniques of plaintext to ciphertext. To safely access the sensitive or private data, searchable encryption (SE) has been studied over the past few years. In a SE scheme, the stored records are normally identified by sets of keywords, and encrypted as a secure index; the query is also expressed by a set of keywords, and encrypted as a trapdoor which is used to make keywords search.

Considering the fact that current information retrieval systems support advanced keyword search, how to construct a public key encryption scheme supporting complex query conditions like Boolean keyword search is becoming an important issue [2]. A Boolean keyword query is represented by $Q = q_1\ op_1\ q_2\ op_2\ \cdots\ op_{i-1}\ q_i\ op_i\ \cdots\ op_{n-1}\ q_n$, where op_i is a logical operator which can be *and*($\wedge$) or *or*($\vee$) and q_i is a keyword [1]. To address this issue, the problems of conjunctive keyword search and disjunctive keyword search are two important issues needed to be researched first [20].

The model and security definitions of conjunctive keyword search on encrypted data in public key system were proposed in [9]. Based on these definitions, two concrete schemes are given. The first scheme needs lots of bilinear pairing computations and the second one needs private keys in proportion to the number of keywords. Then, Hwang and Lee [8] designed a more efficient scheme and introduced a new concept called a multi-user public key encryption with conjunctive keyword search (PECK) which can effectively manage the encrypted documents in a server for multiple users. Boneh and Waters [14] presented a hidden vector encryption (HVE) scheme which supports conjunctive search, comparison queries($x \geq a$) and subset queries ($x \in S$) on encrypted data.

Compared with PECK, the public key encryption with disjunctive keyword search (PEDK) scheme appeared relatively late. In order to support disjunctive keyword search, Katz et al. proposed an inner product encryption (IPE)* scheme in [10]. They presented a method of changing an IPE scheme into a PEDK scheme. To construct an encryption scheme supporting a Boolean query, a naive thought is that a Boolean query scheme can be obtained by expanding a PECK or PEDK scheme, i.e., by combining the query results of PECK or PEDK. Unfortunately, we argue that this method fails to achieve the goal.

To better illustrate our motivation, we briefly review two simple solutions and explain why they are unsatisfactory. Let a Boolean keyword search be $q_1 \vee q_2 \wedge q_3$, where q_1, q_2, and q_3 are three keywords. The first approach is that we execute the query q_1 and the query $q_2 \wedge q_3$ by making use of the PECK scheme, respectively, and obtain the union of the query results of q_1 and $q_2 \wedge q_3$. Nevertheless, the adversary can obtain trapdoors of q_1 and $q_2 \wedge q_3$ from the trapdoor of $q_1 \vee q_2 \wedge q_3$. By using these trapdoors, both the search results of q_1 and that of $q_2 \wedge q_3$ are leaked. Over time, the adversary may combine this information with knowledge of statistics to infer information about the user's docu-

Manuscript received March 13, 2019.
Manuscript revised June 20, 2019.

[†]The authors are with the School of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, P.R. China.
[††]The author is with Wayne State University, 42 W Warren Ave, Detroit, MI 48202, USA.
a) E-mail: willow1223@126.com

DOI: 10.1587/transfun.2019CIP0006

*This encryption system is also called predicate encryption supporting inner product.

ments. The second method is that we execute the query of $q_1 \vee q_2$ and q_3 by making use of the PEDK scheme, respectively, and then obtain the intersection of the search results of $q_1 \vee q_2$ and q_3. However, so far, the space and time complexity of the fully secure PEDK scheme is exponential. For example, if we make a query like $q_1 \vee q_2 \vee \ldots \vee q_n \wedge q_{n+1}$, the drawback is that the space and time complexity of the obtained scheme is $O(2^n)$. In order to build a practical scheme, we have to find a different way to ensure the security and efficiency of the scheme.

In this paper, we will combine a keyword conversion method and a predicate-only IPE (PO-IPE, a simplified version of IPE) scheme to realize a secure and efficient public-key encryption with Boolean keyword search (PEBKS) scheme. Concretely, we present a method that can convert the query Q and the index keyword set W into a set of predicate vectors and an attribute vector, respectively. After this, by applying the obtained vectors to a PO-IPE scheme introduced in [33], a secure PEBKS scheme is built.

Main Works. Concretely, the main works in this paper are listed as follows.

1) In order to support Boolean keywords search, we design a new keywords conversion method, which converts an index keyword set and a query into an attribute vector and a group of predicate vectors, respectively. Explicit comparison between our approach and the previous methods is also given, which demonstrates that the dimension of predicate and attribute vectors obtained by our method is much less than the previous methods.
2) We give a detailed framework and security definition of PEBKS according to the searchable encryption introduced in [9], [13]. According to the framework, by taking advantage of the conversion method mentioned in 1), we propose a construction of PEBKS based on an efficient PO-IPE scheme [33]; We also prove the security of our PEBKS scheme according to the given security definition. Moreover, we give an experiment to verify the efficiency of the proposed scheme.

As a result, our scheme can also be reckoned as the first searchable public-key encryption (SPE) scheme supporting simple Boolean keyword query.

Related work. There are two classes of searchable encryption schemes in terms of different cryptography primitives: public key system and symmetric key system.

Song et al. first proposed the definition of searchable symmetric encryption (SSE) and gave a concrete scheme [4]. After this, many works [5]–[7] aim to create SSE scheme supporting multi-keyword query. However, the search time in these schemes is linear to the number of documents. To improve the search efficiency, by taking advantage of tree structures, such as r-tree and kd-tree, SSE schemes with sub-linear search time were released in [11], [12]. Kuzu et al. and Raykova et al. [16], [17] proposed SSE schemes for fuzzy keywords query that relies on a concept called keyword distance. If the distance between the query and index is less than a preset threshold, it indicates that the query matches with index. In order to return the query results more accurate, sorting query results is also an important research point. Recently, SSE schemes that can quickly search top-k related documents were presented in [18], [19].

In the public key setting, the first SPE encryption solution is designed by Boneh et al. [13], and is called public-key encryption with keyword search (PEKS). Based on this, Abdalla et al. defined the computational and statistical consistency of PEKS, and gave a concrete scheme [20]. However, their works only support a single keyword search. The first PECK scheme was proposed by Park et al. [9], but the scheme used the keyword field as an additional information, which are not practical in many applications. In order to avoid using keyword field, Boneh and Waters presented the hidden vector encryption (HVE) that supports conjunctive keyword search, comparison queries and subset queries on encrypted data [14]. To realize disjunctive keyword search, Katz et al. proposed an IPE scheme [10].

During recent years, the works of SPE can be classified into two groups. The works in the first group aim to improve the efficiency and functionality on the standard SPE scheme which do not add extra mechanisms to the original framework [13], such as the trusted third part. Improved PECK schemes [22], [23] were proposed to reduce computation and communication costs. The improved PEDK scheme created by using fully secure IPE scheme was proposed in [21]. In order to create a confidential search system, Matsuda et al. proposed a group of methods which support "AND", "OR" and inclusion relation tests over encrypted coded data [39]. The SPE scheme supporting conjunctive and disjunctive keyword search simultaneously was proposed in [24]. By applying the tree structure and fully secure IPE scheme, the improved SPE scheme with range search was introduced in [25]. In [38], Kawai et al. introduced a new IPE scheme called IPE with trapdoor conversion mechanism (IPE-TC), and gave a concrete scheme. Based on the IPE-TC scheme, a SPE scheme supporting partial keyword matching was proposed. Compared with the previous scheme, their scheme needs less computation cost. To improve the query speed, by using a special hidden structure, Xu et al. proposed two SPE schemes supporting single keyword search [29], [30] whose search performance are very close to a practical SSE scheme.

The studies in the second group add some special abilities, e.g., access control, by adopting some extra mechanisms such as proxy servers. The standard PECK schemes assume that the cloud sever is honest but curious. To ensure that the search results returned from the cloud are authentic, the works in [26], [27] introduced a trusted third party to verify the query results. In order to add access control ability to the standard SPE, Zhu et al. proposed a SPE scheme with access control by using the access tree [31]. The SPE scheme with the abilities of verifiable and access control was

Table 1 Comparison between previous searchable public key encryption schemes and ours.

Type	Ref.	Query Condition	Additional special abilities
Standard SPE	Ours	Boolean keyword search	-
Standard SPE	[22]	conjunctive keyword search	-
	[23]	conjunctive keyword search	-
	[21]	disjunctive keyword search	-
	[14]	Range, conjunctive keyword, subset search	-
	[24]	Conjunctive and disjunctive keyword search	-
	[25]	Range search	-
	[29]	Single keyword search	-
	[30]	Single keyword search	-
	[38]	Single keyword search with partial matching	-
SPE with extra mechanisms	[26]	conjunctive keyword search	Verifiable
	[27]	conjunctive keyword search	Verifiable
	[28]	Single keyword search	Verifiable and Access control
	[31]	Fuzzy keyword search	Access control

SPE with extra mechanisms means that this kind of SPE adds some extra mechanisms, e.g. the trusted third party and proxy servers, to the standard SPE; The query condition represents the search mode supported by the scheme; The additional special abilities are access control and the verification of query results.

Table 2 Notation.

W	The index keyword set, $W = \{w_1, w_2, \ldots, w_n\}$
n	The number of keywords in W
w_i	A keyword in W, $i \in [1, n]$
I_W	The encrypted index of W
Q	The Boolean keywords query, $Q = Q_1 \vee Q_2 \vee \ldots \vee Q_m = (\wedge_{i\in[1,n_1]} q_{1i}) \vee (\wedge_{i\in[1,n_2]} q_{2i}) \vee \ldots \vee (\wedge_{i\in[1,n_m]} q_{mi})$
m	The number of clauses in Q
Q_j	The j-th clause in Q, $j \in [1, m]$, $Q_j = \wedge_{i\in[1,n_j]} q_{ji}$
$\overline{Q_j}$	The corresponding keyword set for Q_j, $j \in [1, m]$, $\overline{Q_j} = \{q_{j1}, q_{j2}, \ldots, q_{jn_j}\}$
n_j	The number of keywords in $\overline{Q_j}$, $j \in [1, m]$
q_{ji}	A keyword in $\overline{Q_j}$, $j \in [1, m]$ and $i \in [1, n_j]$
N	The maximum number of keywords in the query Q
T_Q	The trapdoor of Q

proposed in [28].

Table 1 shows some SPE works that proposed in recent years. According to Table 1, we found that the SPE scheme supporting simple Boolean keyword search has not yet appeared. Thus, this paper is devoted to building the first standard SPE scheme supporting such function.

Organization This paper is organized as follows: In Sect. 2, the framework and security model of PEBKS are defined. Some backgrounds are also given in the section. Our scheme and its security proof are presented in Sect. 3. The theoretical and experimental analysis is given in Sect. 4. Finally, we conclude this work In Sect. 5.

2. Preliminary

In this section, we give a formal definition of the framework and security model of PEBKS. Besides, we also briefly introduce some basic ingredients used in our scheme, including bilinear pairing group and PO-IPE scheme. In order to formulate models mathematically, the notations used in this paper are introduced in Table 2.

2.1 The Proposed PEBKS Model

Let pk, sk be the receiver's public key and secret key respectively, where pk is open to the public and sk can be only obtained by the receiver. A sender can send an encrypted message M with an encrypted index generated by using keywords $w_1, w_2, \ldots, w_n$ of M and pk to a server. If the receiver wants to retrieve the messages including a specific list of keywords, she can use sk and query keywords to construct a trapdoor, and sends the trapdoor to the server. The server then tests each encrypted index against the trapdoor and returns the matched messages to the receiver. The system architecture is illustrated in Fig. 1. Based on this architecture, the definition of the PEBKS model is given as follows.

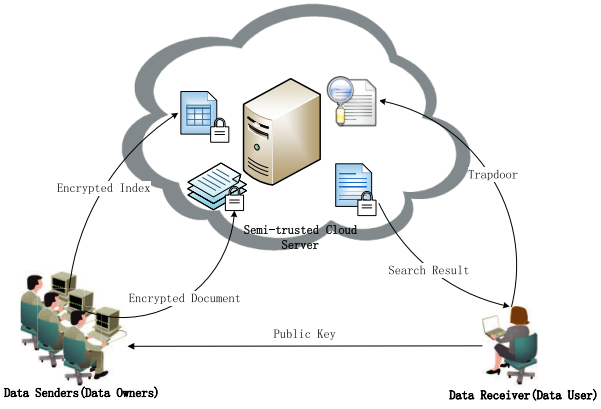

Fig. 1 Architecture of the search over encrypted cloud data.

Definition 1. *public key encryption with simple Boolean keyword search consists of four probabilistic polynomial time (PPT) algorithms, (**KeyGen**, **IndexBuild**, **Trapdoor**,*

***Test**) such that:*

1. ***KeyGen**(1^n): Given a security parameter 1^n, the algorithm outputs the system parameter (pk, sk), where pk is the public key and sk is the secret key.*
2. ***IndexBuild**(pk,W): The algorithm is executed by the sender to encrypt a keyword set $W = \{w_1, w_2, \ldots, w_n\}$. It produces a searchable encrypted index I_W of W by using the public key pk.*
3. ***Trapdoor**(sk,Q): The algorithm is executed by the receiver to construct a trapdoor. Given the secret key sk and a simple Boolean keyword query $Q = Q_1 \vee Q_2 \vee \ldots \vee Q_m = (\wedge_{i\in[1,n_1]} q_{1i}) \vee (\wedge_{i\in[1,n_2]} q_{2i}) \vee \ldots \vee (\wedge_{i\in[1,n_m]} q_{mi})$, where the keywords in each clause Q_j is $\{q_{j1}, q_{j2}, \ldots, q_{jn_j}\}$ and denoted by $\overline{Q_j}$, $n_j \leq n$ and $j \in [1, m]$, the algorithm generates a trapdoor T_Q.*
4. ***Test**(pk, T_Q, I_W): Suppose that a keyword query Q used in the trapdoor and a keyword set W used in the index are described as above, a function $f(W, Q)$ is defined as follows: if there exists some $i \in [1, m]$ such that $\overline{Q_i} \subseteq W$, then $f(W, Q) = 1$; Otherwise, $f(W, Q) = 0$. The test algorithm is executed by the server, and takes input as a trapdoor T_Q, a secure index I_W and the public key pk, then outputs 1 if $f(W, Q) = 1$, or 0 otherwise.*

Correctness property: for a simple Boolean keyword query Q and a keyword set W, for correctly generated $KeyGen(\gamma) \rightarrow \{pk, sk\}$, $IndexBuild(pk, W) \rightarrow I_W$ and $Trapdoor(sk, Q) \rightarrow T_Q$, it holds that $Test(pk, T_Q, I_W) = 1$ if $f(W, Q) = 1$. Otherwise, it holds with negligible probability.

Actually, in the PEBKS scheme, for a message M with keyword set W, an index of M is constructed by $IndexBuild(pk, W)$, and M is encrypted by $Enc(pk, M)$, where $Enc(\cdot)$ is a secure public key encryption function, e.g. RSA. As a result, a ciphertext of M has the form of $\{Enc(pk, M), IndexBuild(pk, W)\}$. Similar with other related works, the proposal only concentrate on searchable encryption part.

2.2 Security Definition of the PEBKS

We propose a new security definition which is similar to the definition presented in [9]. The security of our PEBKS scheme is followed by the definition below.

Definition 2. *A PEBKS scheme is adaptively index-hiding against chosen plaintext attacks if for all probabilistic polynomial-time adversaries* A*, the advantage of* A *in the following game is negligible in the security parameter.*

1. *Setup: the challenger* C *runs the KeyGen(1^n) algorithm to generate pk and sk, and gives pk to the adversary* A.
2. *Phase 1: the adversary* A *can adaptively ask the challenger* C *for the trapdoor T_Q for any query Q of his choice.*
3. *Challenge:* A *selects two keyword sets $W^{(0)}$ and $W^{(1)}$ and sends them to* C*. Suppose that $Q^{(1)}, Q^{(2)}, \ldots, Q^{(t)}$ are the keywords queries used to construct trapdoors in Phase 1, the only restriction is that $f(W^{(0)}, Q^{(i)}) = f(W^{(1)}, Q^{(i)})$ for each $i \in [1, t]$, where t is the number of trapdoors queried in the Phase* 1 *(The function $f(W, Q)$ is defined in* Definition 2.1*). Then, randomly choosing a bit $\beta \in \{0, 1\}$,* C *produces $I_\beta = IndexBuild(pk, W^{(\beta)})$ and sends $\{I_\beta, W^{(0)}, W^{(1)}\}$ to* A.
4. *Phase 2:* A *can continue to ask for trapdoor T_Q for any query* Q *of his choice. The only restriction is that $f(W^{(0)}, T_Q) = f(W^{(1)}, T_Q)$.*
5. *Response: the adversary* A *outputs $\beta' \in \{0, 1\}$ and wins the game if $\beta' = \beta$.*

According to the game mentioned above, we define A*'s advantage in the above game as:*

$$Adv^{\mathrm{A}}_{Game} = |Pr[\beta' = \beta] - \tfrac{1}{2}|$$

The essential of this security definition is to insure that the encrypted form of W^0 and that of W^1 are computationally indistinguishable to the adversary A.

2.3 Prime Order Bilinear Group and PO-IPE Scheme

2.3.1 Prime Order Bilinear Group

Let G_1, G_2 be two cyclic groups of prime order p. There are three properties in the bilinear pairings map $\hat{e} : G_1 \times G_1 \rightarrow G_2$.

1) Bilinear: $\hat{e}(a^u, b^v) = \hat{e}(a, b)^{uv}$,where $a, b \in G_1$ and $u, v \in Z_p^*$;
2) Non-degenerate: If $g \in G_1$ then $\hat{e}(g, g) \in G_2$;
3) Computable: For any $a, b \in G_1$, $\hat{e}(a, b)$ can be efficiently computable.

An efficient bilinear map can be obtained by applying the Weil pairing or the Tate pairing [3].

2.3.2 Framework of PO-IPE

To create our PEBKS scheme, we first convert the index keywords and query keywords into an attribute and predicate vectors, respectively, and then adopt the PO-IPE scheme to encrypt these vectors. For clarity, we introduce the framework of PO-IPE as follows.

The original definition of PO-IPE was presented in [10]. Specifically, for the class of inner-product predicate, an attribute can be expressed as a vector $\vec{x}$ and a predicate, associated with a vector $\vec{v}$, can be expressed as $f_{\vec{v}}$. We have $f_{\vec{v}}(\vec{x}) = 1$, if and only if $\vec{v} \cdot \vec{x} = 0$. We denote Σ as an arbitrary set of attributes and $\mathbb{F}$ as an arbitrary set of predicates over Σ.

Definition 3. *[10] An PO-IPE scheme with predicates $\mathbb{F}$ and attributes Σ consists of four probabilistic polynomial-time algorithms: **Setup**, **KeyGen**, **Enc** and **Dec**. They are given as follows:*

1. ***Setup** takes as input the security parameter 1^n, it outputs pk and master sk (msk).*

2. ***KeyGen*** *takes as input the master secret key msk and the predicate vector* $\vec{v} \in \mathbb{F}$. *It outputs the corresponding secret key* $sk_{\vec{v}}$.
3. ***Enc*** *takes as input the public key pk and the attribute vector* $\vec{x} \in \Sigma$. *It returns the ciphertext C.*
4. ***Dec*** *takes as input the public key pk, the secret key* $sk_{\vec{v}}$ *and the ciphertext C. It outputs either* 1 *or* 0.

Consistency in PO-IPE*: For all* $f_{\vec{v}} \in \mathbb{F}$ *and* $\vec{x} \in \Sigma$, *for correctly generated* $\textbf{\textit{Setup}}(1^n) \rightarrow \{pk, msk\}$, $\textbf{\textit{KeyGen}}(msk, \vec{v}) \rightarrow sk_{\vec{v}}$ *and* $Enc(pk, \vec{x}) \rightarrow C$, *it holds that* $\textbf{\textit{Dec}}(pk, sk_{\vec{v}}, C) = 1$ *if* $\vec{v} \cdot \vec{x} = 0$. *Otherwise, it outputs* 0.

In a standard IPE scheme, an encryption algorithm takes as input not only a vector $\vec{x}$ but also a message M, and its corresponding decryption algorithm outputs M if $\vec{v} \cdot \vec{x} = 0$. That is to say, IPE encrypts both the message and the attribute vector, while PO-IPE only considers to protect attribute vector and sets the message to be 1. According to these description, it can be seen that the key to achieve the goal of Boolean query is to transform the query and index keyword sets into the predicate and attribute vectors, respectively.

Because our scheme is based on the PO-IPE scheme, the security of our scheme relies on the security of the PO-IPE scheme. In order to clarify the security of our scheme in the next section, we first introduce the security definition of PO-IPE presented in [10].

Definition 4. *An PO-IPE scheme is adaptively attribute-hiding (AH) against chosen plaintext attacks if for all probabilistic polynomial-time adversaries A, the advantage of A in the following experiment is negligible in the security parameter.*

1) *Setup:* $\textbf{\textit{Setup}}(1^n)$ *is run to generate pk and msk, and pk is given to the adversary* A.
2) *Phase 1:* A *may adaptively make* q' *secret key queries for* q' *predicate vectors* $\vec{v_1}, \vec{v_2}, \ldots, \vec{v_{q'}}$. *In response,* A *is given the corresponding keys* $sk_{\vec{v_1}}, sk_{\vec{v_2}}, \ldots, sk_{\vec{v_{q'}}}$.
3) *Challenge:* A *randomly outputs two challenge attribute vectors* $\vec{x}^{(0)}, \vec{x}^{(1)}$, *subject to the following restrictions: for the secret key of the predicate vector* $\vec{v_l}$, *where* $l \in [1, q']$, *it satisfies one of the following conditions.*
   - $\vec{v_l} \cdot \vec{x}^{(0)} \neq 0$ *and* $\vec{v_l} \cdot \vec{x}^{(1)} \neq 0$;
   - $\vec{v_l} \cdot \vec{x}^{(0)} = 0$ *and* $\vec{v_l} \cdot \vec{x}^{(1)} = 0$.

   A *random bit* β *is chosen, and* A *is given* $C^{(\beta)} \rightarrow \textbf{\textit{Enc}}(pk, \vec{x}^{(\beta)})$.
4) *Phase 2: The adversary* A *may continue to request keys corresponding to the additional predicates vectors,* $v_{\vec{q'+1}}, v_{\vec{q'+2}}, \ldots, \vec{v_q}$, *subject to the restriction given in Step 3).* A *is given the corresponding keys* $sk_{v_{\vec{q'+1}}}, sk_{v_{\vec{q'+2}}}, \ldots, sk_{\vec{v_q}}$.
5) *Response:* A *outputs a bit* β', *and succeeds if* $\beta' = \beta$.

Note that the security definition of PO-IPE has similar phases with the one of PECDK. Thus, we can utilize the security of PO-IPE to guarantee the security of our scheme.

In addition, for the public key setting, anyone holding the public key can generate the encrypted index, which makes trapdoors suffer from keywords guessing attack inherently [34]. To protect the security of trapdoor, a technique called dual-server can be used [35]. But, applying this method need change the standard security definition. Considering that our scheme is under a standard SPE model, the security of trapdoor is beyond our goal.

3. Proposed PEBKS Scheme

This section involves three parts: 1) presenting a method that can convert index and query keywords into attribute and predicate vectors, respectively; 2) applying these vectors to a PO-IPE scheme based on the framework and security model descried in the previous section; 3) giving a detailed security proof for our scheme.

3.1 Conversion Method

Suppose that any keyword w can be expressed as $\{0,1\}^*$, and define a function $H_1 : \{0,1\}^* \rightarrow Z_p^*$. Since p is a large prime and is larger than the number of the all words, H_1 can be collision-resistance. This means that, if $i \neq j$, then $H_1(w_i) \neq H_1(w_j)$, where w_i and w_j are two distinct keywords. The approach is described as follows:

1) For each keyword set $\overline{Q_j} = \{q_{j1}, q_{j2}, \ldots, q_{jn_j}\}$ for the j-th clause Q_j in the query $Q = Q_1 \vee Q_2 \vee \ldots \vee Q_m$, we can construct a vector $\vec{x_j} = \{x_{j0}, x_{j1}, \ldots, x_{jn}\}$, where $x_{ji} = H_1(q_{j1})^i + H_1(q_{j2})^i + \ldots + H_1(q_{jn_j})^i$, $i \in [0, n]$ and $j \in [1, m]$. By applying this method, we can create a group of vectors, $\{\vec{x_1}, \vec{x_2}, \ldots, \vec{x_m}\}$, for the query Q.
2) For the keyword set $W = \{w_1, w_2, \ldots, w_n\}$, we can construct a function:

$$\begin{aligned} f(x) &= (x - H_1(w_1))(x - H_1(w_2)) \ldots (x - H_1(w_n)) \\ &= a_n x^n + a_{n-1} x^{n-1} + \ldots + a_0 x^0 \end{aligned}$$

According to the coefficients of the $f(x)$, a vector $\vec{a} = \{a_0, a_1, \ldots, a_n\}$ of W can be obtained.
3) Note that if there exists an j such that $\overline{Q_j} \subseteq W$, it is not difficult to verify that $\vec{a} \cdot \vec{x_j} = 0$, where $j \in [1, m]$. If we only use this property, the knowledge of $\overline{Q_j} \subseteq W$ will be leaked. To prevent such leakage, we consider increasing randomness to the verification equation. For each clause Q_j, we randomly select a clause (denoted by $Q_{\pi(j)}$) from $\{Q_1, Q_2, \ldots, Q_m\}$, and combine the Q_j and $Q_{\pi(j)}$ into a pair $\{Q_j, Q_{\pi(j)}\}$, where $j \in [1, m]$.
4) For each pair $\{Q_j, Q_{\pi(j)}\}$, if $\overline{Q_j} \subseteq W$ or $\overline{Q_{\pi(j)}} \subseteq W$, the pair is contained in W. According to this, we will create a predicate vector of the pair and an attribute vector of W. According to the item 2), we can use the equation $(\vec{a} \cdot \vec{x_i}) \times (\vec{a} \cdot \vec{x_{\pi(i)}})$ to verify whether the pair is contained in W. Note that whether $\overline{Q_j} \subseteq W$ or $\overline{Q_{\pi(j)}} \subseteq W$, the equation $(\vec{a} \cdot \vec{x_i}) \times (\vec{a} \cdot \vec{x_{\pi(i)}})$ outputs 0. Based on this equation, we can build the following matrix.

$$\begin{pmatrix} x_{i0}a_0x_{\pi(i)0}a_0 & x_{i0}a_0x_{\pi(i)1}a_1 & \cdots & x_{i0}a_0x_{\pi(i)n}a_n \\ x_{i1}a_1x_{\pi(i)0}a_0 & x_{i1}a_1x_{\pi(i)1}a_1 & \cdots & x_{i1}a_1x_{\pi(i)n}a_n \\ \vdots & \vdots & \ddots & \vdots \\ x_{in}a_nx_{\pi(i)0}a_0 & x_{in}a_nx_{\pi(i)1}a_1 & \cdots & x_{in}a_nx_{\pi(i)n}a_n \end{pmatrix}$$

5) According to the matrix, we can build the predicate vector $\vec{X_i}$ of the pair $\{Q_i, Q_{\pi(i)}\}$, and attribute vector $\vec{A}$ of the keyword set W as follows.

$$\vec{X_i} = \{x_{i0}x_{\pi(i)0}, x_{i0}x_{\pi(i)1}, \ldots, x_{is}x_{\pi(i)t}, \ldots, x_{in}x_{\pi(i)n}\},$$

$$\vec{A} = \{a_0a_0, a_0a_1, \ldots, a_sa_t, \ldots, a_na_n\},$$

where $s \in [0, n]$ and $t \in [0, n]$.
It can be verified that if the pair $\{Q_i, Q_{\pi(i)}\}$ is included in W, then $\vec{X_i} \cdot \vec{A} = 0$.

As a result, by applying the method above, we can make Boolean keyword search without revealing the information of which clause Q_i matches W. Based on this, a concrete PEBKS scheme will be proposed in next section.

3.2 Construction Details

According to the definition of PO-IPE introduced in the Sect. 2.3, let $Setup_{IPE}$, $KeyGen_{IPE}(pk_{IPE}, msk_{IPE}, \vec{v})$, $Enc_{IPE}(pk_{IPE}, \vec{x})$, and $Dec_{IPE}(pk_{IPE}, c, sk_{\vec{v}})$ be the four algorithms in the PO-IPE scheme, where pk_{IPE} and msk_{IPE} are the public key and the master secret key generated by using $Setup_{IPE}$, $\vec{x}$ is the attribute vector, $\vec{v}$ is the predicate vector, c is the ciphertext generated by using Enc_{IPE} and $sk_{\vec{v}}$ is the secret key generated by using $KeyGen_{IPE}$. Based on the PO-IPE scheme [33], our PEBKS scheme can be built as follows.

- **KeyGen**: By using the $Setup_{IPE}$ algorithm, pk_{IPE} and msk_{IPE} can be obtained. The algorithm sets $pk = pk_{IPE}$ and $sk = msk_{IPE}$, and outputs pk and sk.
- **IndexBuild**: For a keyword set $W = \{w_1, w_2, \ldots, w_n\}$, the algorithm generates an attribute vector $\vec{A}$ based on the method described in Sect. 3.1. Then it generates $I_W = Enc_{IPE}(pk, \vec{A})$.
- **Trapdoor**: Given a keywords query $Q = Q_1 \vee Q_2 \vee \ldots \vee Q_m$, according to the approach given in Sec III.A, the algorithm firstly generates a group of keyword set pairs. Then it converts each pair into a predicate vector $\vec{X_i}$ and generates t_i by making use of $KeyGen_{IPE}(sk, pk, \vec{X_i})$ for each $i \in [1, m]$. Finally, it outputs a trapdoor $T_Q = \{t_1, t_2, \ldots, t_m\}$.
- **Test**: Given a T_Q, a I_W and the pk, the algorithm works as follows.
  1) Choosing a counter i, and setting $i = 1$;
  2) If $i > m$, then go to step 3), otherwise the algorithm computes: $R = Dec(pk, I_W, t_i)$. If $R = 1$, then the algorithm outputs 1 and ends. Otherwise, it sets $i = i + 1$ and goes to the step 2).
  3) The algorithm outputs 0 and ends.

3.3 Security Proof

The proposed PEBKS scheme is constructed by making use of the fully secure PO-IPE scheme [33]. Inspired by the method of security proof in [36], [40], [41], we give the following proposition to prove the security of our scheme.

Proposition 1. If the PO-IPE scheme is secure, then our PEBKS scheme is secure.

Proof Sketch. If there is a PPT algorithm $\mathbb{A}$ which can break the PEBKS scheme, we can say that $\mathbb{A}$ can break the PO-IPE scheme. The proof process is listed as follows.

1) Setup: To create pk and sk in the PEBKS scheme, the challenger $\mathbb{C}$ uses the $Setup_{IPE}$ algorithm to generate pk_{IPE}, msk_{IPE} and sets $pk = pk_{IPE}$, $sk = msk_{IPE}$.
2) Phase 1: $\mathbb{A}$ can adaptively ask trapdoors of queries $\{Q^{(1)}, Q^{(2)}, \ldots, Q^{(t)}\}$. The challenger $\mathbb{C}$ uses the $KeyGen_{IPE}$ algorithm to generate a group of trapdoors $\{T_{Q^{(1)}}, T_{Q^{(2)}}, \ldots, T_{Q^{(t)}}\}$, where each trapdoor can be seen as a set of decryption keys for the PO-IPE scheme.
3) Challenge: After phase 1, $\mathbb{A}$ outputs two challenge keyword sets W^0 and W^1, under a constraint that $f(W^{(0)}, Q^{(i)}) = f(W^{(1)}, Q^{(i)})$, where $i \in [1, t]$. $\mathbb{C}$ flips a coin $\beta \in \{0, 1\}$ and generate an index $I_{W^{(\beta)}}$ by using the algorithm Enc_{IPE}. Then, $\mathbb{C}$ gives an index $I_{W^{(\beta)}}$ to $\mathbb{A}$. Note that this index can be seen as a challenge ciphertext of PO-IPE.
4) Phase 2: $\mathbb{A}$ can continue querying trapdoors which subject to the restriction described above. In order to construct trapdoors which can meet the above constraint, we will give the steps for creating the trapdoors. For each query Q containing keyword sets $\{\overline{Q_1}, \overline{Q_2}, \ldots, \overline{Q_m}\}$, these keyword sets can be divided into three parts.

$$\begin{cases} A = \{Q_i | \overline{Q_i} \subseteq W^0\}, \\ B = \{Q_i | \overline{Q_i} \not\subset W^0 \text{ and } \overline{Q_i} \subseteq W^1\}, \\ C = \{Q_i | \overline{Q_i} \not\subset W^0 \text{ and } \overline{Q_i} \not\subset W^1\}, \end{cases} \quad 1 \le i \le m.$$

In order to build the keyword set pair, there are three situations:

1) For each $Q_i \in A$, we randomly select a keyword set (denoted by $Q_{\pi(i)}$) from set B, and combine the Q_i and $Q_{\pi(i)}$ into a pair $\{Q_i, Q_{\pi(i)}\}$;
2) For each $Q_i \in B$, we randomly select a keyword set (denoted by $Q_{\pi(i)}$) from set A, and combine the Q_i and $Q_{\pi(i)}$ into a pair $\{Q_i, Q_{\pi(i)}\}$;
3) For each $Q_i \in C$, we randomly select a keyword set (denoted by $Q_{\pi(i)}$) from set C, and combine the Q_i and $Q_{\pi(i)}$ into a pair $\{Q_i, Q_{\pi(i)}\}$;

The pairs obtained through the above method either matches both W^0 and W^1, or matches neither W^0 nor W^1. Thus, the trapdoor constructed by using these pairs fails to distinguish between W^0 and W^1. For each key pair, $\mathbb{C}$ uses $KeyGen_{IPE}$ algorithm to generate a key of $PO - IPE$. All these keys are seen as a trapdoor of

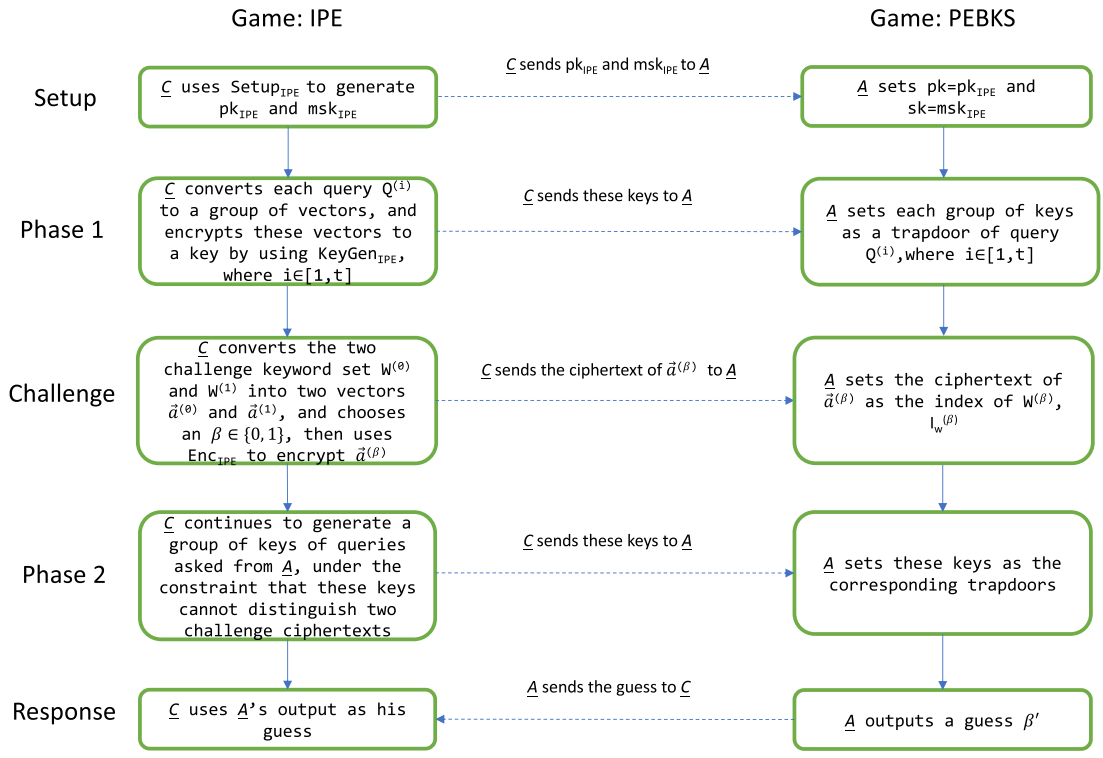

Fig. 2 An illustration of security proof (A and C are adversary and challenger, respectively; the left and right part are security game of PO-IPE and PEBKS, respectively).

PEBKS. Therefore, the obtained trapdoors can still be seen as a group of decryption keys for PO-IPE.

5) Response: Finally, $\mathbb{A}$ gives a guess β'. This guess can also be regarded as a guess of security game of $PO-IPE$.

Note that if $\mathbb{A}$ can break the PEBKS scheme, the value of $|Pr[\beta' = \beta] - \frac{1}{2}|$ is not negligible. It means that the two challenge indices can be distinguished. Because the challenge indices in the PEBKS scheme is equal to the challenge ciphertexts in the PO-IPE scheme, according to the security definition for PO-IPE, it means that $\mathbb{A}$ can break the PO-IPE scheme. In conclusion, we give a figure to further explain the proof process. From Fig. 2, we can find that the adversary $\mathbb{A}$ uses the algorithm $\mathbb{C}$ to generate pk, sk, trapdoors, and challenge indexes. This means that the security game of PO-IPE is identical with that of PEBKS in the view of $\mathbb{A}$. If $\mathbb{A}$ can break the security of PEBKS, then the probability that $\mathbb{A}$'s guess is correct should not be neglected, which means that the advantage of $\mathbb{A}$ breaking the security of PO-IPE is non-negligible. Thus, we reckon that the proposition is right.

4. Performance Evaluation

4.1 Comparison with Previous Conversion Methods

In [10], Katz et al. gave two methods that convert keywords into vectors, which can be used to support conjunctive or disjunctive keywords search. Let $X = \{x_1, x_2, \ldots, x_n\}$ be an index keyword set. For the conjunctive keywords search, e.g., $q_1 \wedge q_2 \wedge \ldots \wedge q_t$, a polynomial $p(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{t} \prod_{j=1}^{n} (H_1(q_i) - H_1(x_j))$ is constructed. If $\{q_1, q_2, \ldots, q_t\} \subseteq X$, then $p(x_1, x_2, \ldots, x_n) = 0$. The number of terms in $p(x_1, x_2, \ldots, x_n)$ is n. For the disjunctive keywords search, e.g., $q_1 \vee q_2 \vee \ldots \vee q_t$, a polynomial $p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{t} \prod_{j=1}^{n} (H_1(q_i) - H_1(x_j))$ is constructed. If there is a $q_j \in X$, where $j \in [1, t]$, then $p(x_1, x_2, \ldots, x_n) = 0$. The number of terms in $p(x_1, x_2, \ldots, x_n)$ is n^t. Note that each term in $p(x_1, x_2, \ldots, x_n)$ consists of two parts: the attribute part which is the product of some elements in $X \cup \{1\}$ and the predicate part which is the product of some elements in $\{q_1, q_2, \ldots, q_t, 1\}$. By using the attribute part and the predicate part in each term, an attribute vector $\vec{x}$ and a predicate vector $\vec{q}$ are created. By applying the attribute and predicate vectors to an IPE scheme, the conjunctive or disjunctive keywords search over encrypted data can be realized.

Through merging the previous methods for supporting conjunctive or disjunctive keywords search, Okamoto and Takashima proposed a keyword conversion method that supports Boolean keywords search [37]. For a Boolean keywords query, e.g., $(q_{11} \vee q_{12} \vee \ldots \vee q_{1t_1}) \wedge (q_{21} \vee q_{22} \vee \ldots \vee q_{2t_2})$, a polynomial $p(x_1, x_2, \ldots, x_n) = r_1 \prod_{j=1}^{n} \prod_{i=1}^{t_1} (H_1(q_{1i}) - H_1(x_j)) + r_2 \prod_{j=1}^{n} \prod_{i=1}^{t_2} (H_1(q_{2i}) - H_1(x_j))$ was built. Because the number of terms in this polynomial $p(x_1, x_2, \ldots, x_n)$ is $O(n^t)$, the dimension of the predicate and attribute vectors are both $O(n^t)$, where t is the larger number between t_1 and t_2.

As shown in Sect. 3.1, our conversion method is to support Boolean keywords search, and the dimension of the vector obtained by utilizing our method is linear with n^2. In order to illustrate the advantage of our method, we give a table to show the comparison between our approach and the previous methods. According to the Table 3, we can argue that our method is more suitable in building the PEBKS scheme since the time and cost complexities of the PEBKS scheme are linear with the dimension of predicate and attribute vectors.

4.2 Theoretical Analysis

Our scheme is based on an efficient PO-IPE scheme [33]. So, we first need to show the performance analysis for this PO-IPE scheme. Based on this analysis, we show the performance of our scheme.

Our scheme has two important parameters: one is the number of keywords in an index, denoted by n; the other is the number of clauses in a query, represented by m. The key ideal of the proposal is converting the keyword set W into an attribute vector and the query Q into a group of predicate vectors, respectively. The dimension of both attribute vector and predicate vector is n^2, and the number of predicate vectors for Q is m. That is because the vector is constructed by using cartesian product of two keyword sets, and each clause has its own predicate vector. Based on this analysis, the Table 4 and Table 5 are presented to show the storage and time overheads for PO-IPE and the proposed scheme. In these tables, E_1 and E_2 mean exponentiation computation in G_1 and G_2, respectively; P indicates pairing computation; $|G_1|$ and $|G_2|$ represent the bit sizes of G_1 and G_2, respectively.

4.3 Experimental Results

Specifically, we implement our construction in JAVA with Java Pairing Based Cryptography (JPBC) library [15]. In our implementation, the bilinear map is instantiated as Type A pairing (base field size is 128-bit), which offers a level of security equivalent to 1024-bit DLOG [15]. Our experiment was run on Intel(R) Core(TM) i7-4570 CPU at 3.60GHz processor and 8GB memory size. Such a experiment is based on a group of artificial keyword sets with different number of keywords in each set (i.e., $n = 2; 4; 6; 8; 10; 12; 14; 16; 18; 20$), where each keyword set can be seen as an index of a document. In each keyword set, we denote each keyword as a unique integer in the range of $[0, 5000]$, where 5000 can be regarded as the number of different words in the artificial keyword sets. We encrypt each keyword set with the proposed PEBKS scheme, and the encrypted indices were stored on our machine. We then execute random queries over these encrypted indices (The number of documents is denoted by D).

Table 3 Comparison with the previous conversion methods.

Method	Functionality	Vector dimension
[10]	Conjunctive keywords search	$O(n)$
	Disjunctive keywords search	$O(n^t)$
[37]	Boolean keywords search	$O(n^t)$
Our method	Boolean keywords search	$O(n^2)$

4.3.1 Impact of the Keywords Size (n)

For a query with 6 clauses ($m = 6$), Fig. 3 and Fig. 4 show that:

1) Fig. 3a and Fig 4a show that the execution time of key generation, index building, trapdoor generation and testing is linear with $O(n^2)$; and
2) Fig. 3b and Fig 4b show that the storage cost of pk and sk, indices and trapdoors is linear with $O(n^2)$. According to the analysis in Sect. 4.1, we know that the keywords in each index and query are converted into vectors whose dimension are n^2. By applying an efficient PO-IPE scheme, the time and space complexities for our PEBKS scheme is linear with $O(n^2)$. According to this analysis, we argue that the experimental result is consistent with our theoretical analysis.

4.3.2 Impact of the Number of Clauses in a Query (m)

According to the analysis in Sect. 4.1, it is clear that parameter m only affects algorithms of trapdoor generation and testing. For an index with 10 keywords ($n = 10$), Fig. 5 shows that the time consuming in trapdoor generation and testing are linearly with $O(m)$, and the storage cost of trapdoors is

Table 4 Time overhead in PO-IPE and our proposal.

PO-IPE		Proposed	
Setup	$(n+2)E_1 + (n+1)E_2$	KeyGen	$((n+1)^2+2)E_1 + ((n+1)^2+1)E_2$
Enc	$(2n+2)E_1$	IndexBuild	$(2(n+1)^2+2)E_1$
KeyGen	$(n+3)E_2$	Trapdoor	$m((n+1)^2+3)E_2$
Dec	$nE_1 + 3P$	Test	$m(n+1)^2E_1 + 3mP$

Table 5 Storage overhead in PO-IPE and our proposal.

PO-IPE		Proposed	
pk	$(n+3)\|G_1\|$	pk	$((n+1)^2+3)\|G_1\|$
msk	$(n+2)\|G_2\|$	sk	$((n+1)^2+2)\|G_2\|$
Ciphertexts Size	$(n+2)\|G_1\|$	Index Size	$m((n+1)^2+2)\|G_1\|$
Key Size	$3\|G_2\| + n\|Z_p^*\|$	Trapdoor Size	$3m\|G_2\| + m(n+1)^2\|Z_p^*\|$

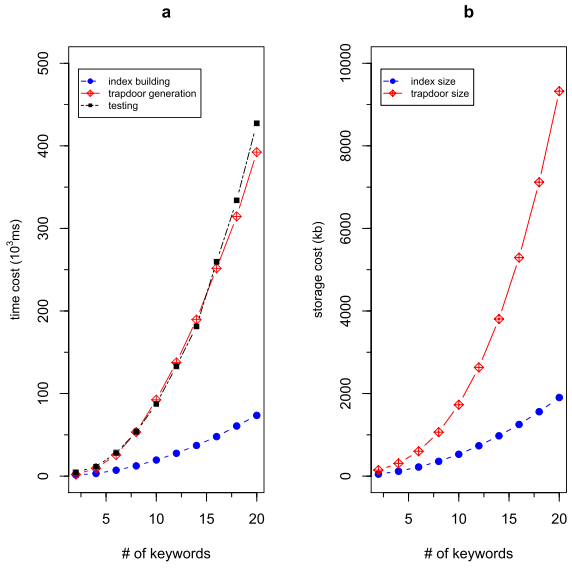

Fig. 3 Impact of n on the time cost of index building, trapdoor generation and testing (a); and impact of n on the storage overhead of indices and trapdoors. ($D = 100, m = 6, n = \{2; 4; 6; 8; 10; 12; 14; 16; 18; 20\}$).

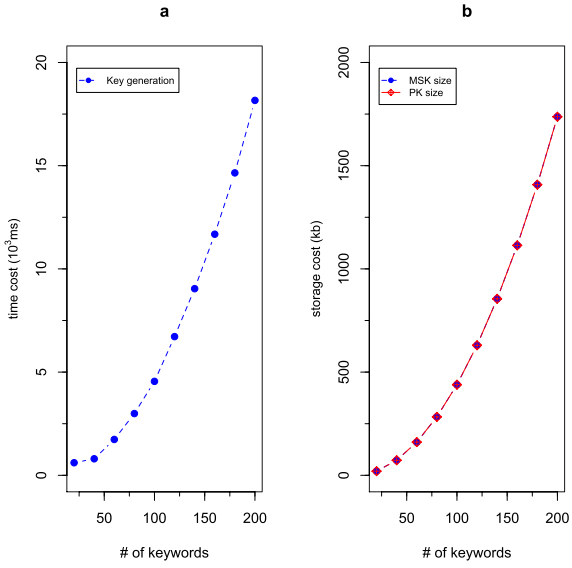

Fig. 4 Impact of n on the time cost of key generation (a); and impact of n on the storage overhead of PK and MSK. ($D = 100, n = \{20; 40; 60; 80; 100; 120; 140; 160; 180; 200\}$).

also linearly with $O(m)$. m is the number of clauses in a query, and can be seen as the number of predicate vectors obtained by using the keywords conversion method. When the number of clauses increases linearly, the number of corresponding vectors also increases linearly. Therefore, we know that the time and space complexities for our PEBKS scheme is linear with the parameter m. As expected, we argue that the experimental result is consistent with our theoretical analysis.

4.3.3 More Comments

For the experiment result with $n = 10$ and $m = 10$, the generation time of a single index and a single trapdoor is 200 ms and 1220 ms, respectively, and the test time of a single document is 1170 ms. In general, the number of keywords in a document (n) is usually only $3 \sim 5$ (e.g. the scientific paper), and the number of keywords in a query is often less than 10 [32]. According to above results, we can reckon that both n and m are less than 10 in the actual process of retrieval. Moreover, the search process is preformed by the cloud server, which has strong computing power. Considering this actual situation and the experimental result, we think that our scheme is practical.

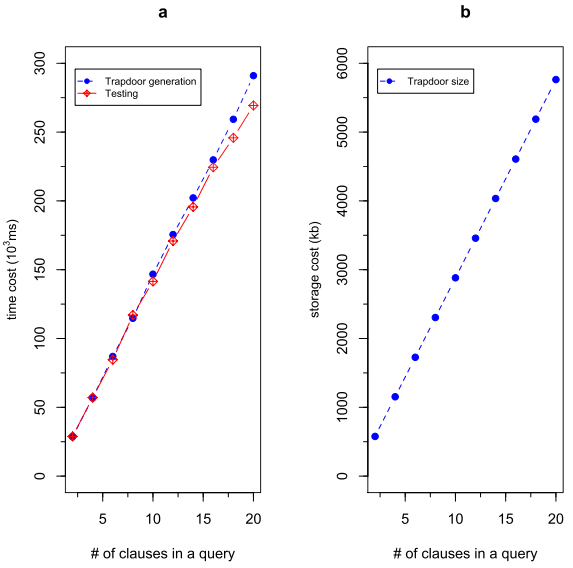

Fig. 5 Impact of m on the time cost of trapdoor generation and testing (a); impact of m on the storage cost of trapdoors (b) ($D = 100, n = 10, m = \{2; 4; 6; 8; 10; 12; 14; 16; 18; 20\}$).

Because each documents has its own encrypted index, we can easily accelerate the search process by utilizing the technique of parallel computation. The related method is introduced in [18]. Thus, we argue that our scheme is more practical in the cloud platform. The illustration of how to use the parallel method for improving the search speed is given in Fig. 6. As shown in Fig. 6, when the receiver make a Boolean keywords search, he or she sends a trapdoor of this query to the cloud server. The cloud server then distributes the encrypted files in the dataset to three tasks. Each task performs the keywords search independently, and adds the result to the result set. After this, the result set will be returned to the receiver. Since there is no communication between the tasks, there is no additional communication overhead. According to this, the search efficiency can be significantly improved. Moreover, the experiment results listed in the Table 6 also shows that the parallel method is efficient.

5. Conclusion

In this paper, we proposed a new approach that can convert the operation of simple Boolean keyword search into inner product operations among vectors. By combining this approach and an efficient PO-IPE scheme, we give a concrete

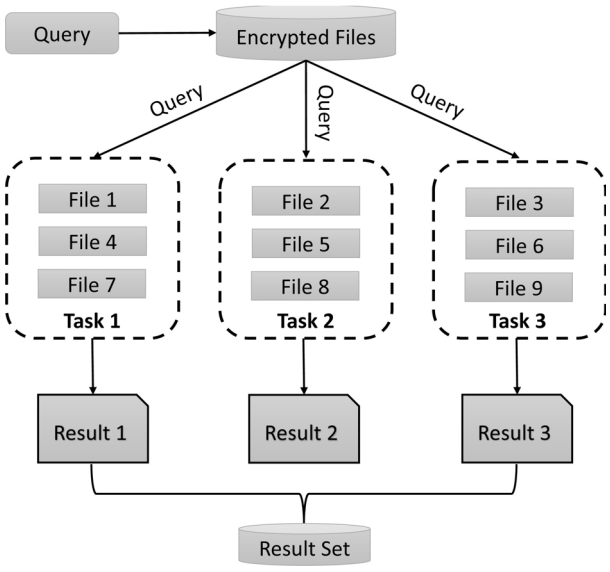

Fig. 6 How to use the parallel computation method for improving the search efficiency.

Table 6 The efficiency of a search by utilizing the method of parallel computation (n=10, m=6, D=100).

Number of parallel threads	Time consumption (s)
1	91.32
2	49.19
4	28.57
8	14.78

scheme, which is proven to be secure under an adaptive security model. To the best of our knowledge, this is the first searchable public key encryption scheme supporting simple Boolean keyword query.

To justify the efficiency of the proposed scheme, we present detailed theoretical analysis and experimental results. These results show that our scheme is practical in the cloud setting. Note that the time and storage consumption in the proposed scheme increases with n^2. Thus, in the future, it is necessary to build a PEBKS scheme with a better time and space complexities.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 61402393, 61601396), by Shanghai Key Laboratory of Integrated Administration Technologies for Information Security (No. AGK201607) and by Nanhu Scholars Program for Young Scholars of XYNU.

References

[1] V.I. Frants, J. Shapiro, I. Taksa, and V.G. Voiskunskii, "Boolean search: Current state and perspectives," J. American Society for Information Science, vol.50, no.1, pp.86–95, 1999.
[2] Y. Zhu, D. Ma, and S. Wang, "Secure data retrieval of outsourced data with complex query support," 2012 32nd International Conference on Distributed Computing Systems Workshops, IEEE, pp.481–490, 2012.
[3] A. Joux, "The Weil and Tate pairings as building blocks for public key cryptosystems," International Algorithmic Number Theory Symposium, pp.20–32, Springer, Berlin, Heidelberg, 2002.
[4] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searching on encrypted data," IEEE Symposium on Research in Security and Privacy, pp.44–55, 2000.
[5] E.J. Goh, "Secure indexes," IACR Cryptology ePrint Archive, 2003: 216, 2003.
[6] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," IEEE Trans. Inf. Forensics Security, vol.11, no.12, pp.2706–2716, 2017.
[7] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," IEEE Trans. Parallel Distrib. Syst., vol.27, no.9, pp.2546–2559, 2016.
[8] Y.H. Hwang and P.J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," International Conference on Pairing-Based Cryptography, pp.2–22, Springer, Berlin, Heidelberg, 2007.
[9] D.J. Park, K. Kim, and P.J. Lee, "Public key encryption with conjunctive field keyword search," International Workshop on Information Security Applications, pp.73–86, Springer, Berlin, Heidelberg, 2004.
[10] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp.146–162, Springer, Berlin, Heidelberg, 2008.
[11] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Roşu, and M. Steiner, "Highly-scalable searchable symmetric encryption with support for boolean queries," Annual Cryptology Conference, pp.353–373, Springer, Berlin, Heidelberg, 2013.
[12] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Roşu, and M. Steiner, "Dynamic searchable encryption in very-large databases: Data structures and implementation," Network and Distributed System Security Symposium, vol.14, pp.23–26, 2014.
[13] D. Boneh, G.D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," International Conference on the Theory and Applications of Cryptographic Techniques, pp.506–522, Springer, Berlin, Heidelberg, 2004.
[14] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," Theory of Cryptography Conference, pp.535–554, Springer, Berlin, Heidelberg, 2007.
[15] A.D. Caro, The Java pairing based cryptography library (JPBC), URL: http://gas.dia.unisa.it/projects/jpbc/, laatstnagekeken op, 2013: 02-24.
[16] M. Kuzu, M.S. Islam, and M. Kantarcioglu, "Efficient similarity search over encrypted data," IEEE, International Conference on Data Engineering, pp.1156–1167, IEEE, 2012.
[17] M. Raykova, A. Cui, B. Vo, B. Liu, T. Malkin, S.M. Bellovin, and S.J. Stolfo, "Usable, secure, private search," IEEE Security Privacy, vol.10, no.5, pp.53–60, 2012.
[18] Z. Fu, X. Sun, Q. Liu, L. Zhou, and J. Shu, "Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing," IEICE Trans. Commun., vol.E98-B, no.1, pp.190–200, Jan. 2015.
[19] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," IEEE Trans. Parallel Distrib. Syst., vol.27, no.2, pp.340–352, 2016.
[20] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions," Proc. CRYPTO 2005, Lecture Notes in Computer Science, vol.3621, pp.205–222, Springer, 2005.
[21] T. Okamoto and K. Takashima, "Achieving short ciphertexts or short

secret-keys for adaptively secure general inner-product encryption," Des. Codes Cryptogr., vol.77, no.2-3, pp.725–771, 2015.
[22] B. Zhang and F. Zhang, "An efficient public key encryption with conjunctive-subset keywords search," J. Netw. Comput. Appl., vol.34, no.1, pp.262–267, 2011.
[23] C. Song, X. Liu, and Y. Yan, "Efficient public key encryption with field-free conjunctive keywords search," International Conference on Trusted Systems, pp.394–406, Springer, Cham, 2014.
[24] Y. Zhang and S. Lu, "POSTER: Efficient method for disjunctive and conjunctive keyword search over encrypted data," Proc. 2014 ACM SIGSAC Conference on Computer and Communications Security, pp.1535–1537, ACM, 2014.
[25] B. Wang, Y. Hou, M. Li, H. Wang, and H. Li, "Maple: Scalable multidimensional range search over encrypted cloud data with tree-based index," Proc. ACM Symposium on Information, Computer and Communications Security, pp.111–122, 2014.
[26] M. Ding, F. Gao, Z. Jin, and H. Zhang, "An efficient public key encryption with conjunctive keyword search scheme based on pairings," IEEE International Conference on Network Infrastructure and Digital Content, pp.526–530, IEEE, 2013.
[27] M.S. Hwang, S.T. Hsu, and C.C. Lee, "A new public key encryption with conjunctive field keyword search scheme," Information Technology and Control, vol.43, no.3, pp.277–288, 2014.
[28] Y. Miao, M.A. Jianfeng, X. Liu, J. Zhang, and Z. Liuet, "VKSE-MO: Verifiable keyword search over encrypted data in multi-owner settings," Science China Information Sciences, vol.60, no.12, p.122105, 2017.
[29] P. Xu, Q. Wu, W. Wang, W. Susilo, J. Domingo-Ferrer, and H. Jin, "Generating searchable public-key ciphertexts with hidden structures for fast keyword search," IEEE Trans. Inf. Forensics Security, vol.10, no.9, pp.1993–2006, 2017.
[30] P. Xu, S. He, W. Wang, W. Susilo, and H. Jin, "Lightweight searchable public-key encryption for cloud-assisted wireless sensor networks," IEEE Trans. Ind. Informat., vol.14, no.8, pp.3712–3723, 2018.
[31] H. Zhu, Z. Mei, B. Wu, H. Li, and Z. Cui, "Fuzzy keyword search and access control over ciphertexts in cloud computing," Information Security and Privacy, Lecture Notes in Computer Science, vol.10342, pp.248–265, Springer, Cham, 2017.
[32] H. Cui, Z. Wan, R.H. Deng, G. Wang, and Y. Li, "Efficient and expressive keyword search over encrypted data in the cloud," IEEE Trans. Dependable and Secure Comput., vol.15, no.3, pp.409–422, 2018.
[33] I. Kim, S.O. Hwang, J.H. Park, and C. Park, "An efficient predicate encryption with constant pairing computations and minimum costs," IEEE Trans. Comput., vol.65, no.10, pp.2947–2958, 2016.
[34] W.C. Yau, S.H. Heng, and B.M. Goi, "Off-line keyword guessing attacks on recent public key encryption with keyword search schemes," International Conference on Autonomic and Trusted Computing, pp.100–105, Springer, Berlin, Heidelberg, 2008.
[35] Q. Huang and H. Li, "An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks," Inform. Sciences, vol.403-404, pp.1–14, 2017.
[36] B. Wang, Y. Hou, M. Li, H. Wang, and H. Li, "Maple: Scalable multi-dimensional range search over encrypted cloud data with tree-based index," Proc. 9th ACM symposium on Information, computer and communications security, pp.111–122, ACM, 2014.
[37] T. Okamoto and K. Takashima, "Fully secure unbounded inner-product and attribute-based encryption," International Conference on the Theory and Application of Cryptology and Information Security, pp.349–366, Springer, Berlin, Heidelberg, 2012.
[38] Y. Kawai, T. Hirano, Y. Koseki, and T. Munaka, "SEPM: Efficient partial keyword search on encrypted data," International Conference on Cryptology and Network Security, pp.75–91, Springer, Cham, 2015.
[39] N. Matsuda, M. Hattori, T. Ito, et al. Confidential search system and cryptographic processing system: U.S. Patent 8,615,668, Dec. 24, 2013.
[40] S. Xu and F. Ye, "A predicate encryption based anomaly detection scheme for e-Health communications network," 2018 IEEE International Conference on Communications (ICC), pp.1–6, IEEE, 2018.
[41] J. Sun, Y. Bao, X. Nie, and H. Xiong, "Attribute-hiding predicate encryption with equality test in cloud computing," IEEE Access, vol.6, pp.31621–31629, 2018.

Yu Zhang received the BSc degree in Henan university of economics and law in 2008, and the Ph.D. degree in Huazhong university of science and technology in 2015. Since 2016, he has worked in the Department of computer science and information technology, Xinyang Normal University, where he is now a lecturer. His major interests include information security, cryptography and information retrieval.

Yansong Zhao received his BS in computer science and technology from Xinyang Normal University, China, in 2007, admitted to the Institute of Computer and information, Xinyang Normal University, China, in 2016. He currently works in the laboratory of the School of Foreign Languages, Xinyang Normal University, China. His research interests include network and information security, language laboratory security, digital watermarking, digital forensic, database security and convolutional neural networks.

Yifan Wang received her BSc degree of Electrical Engineering in Zhengzhou University (China), and the MSc degree of Electrical Engineering in the University of Minnesota, Twin Cities (USA) in 2014 and 2016, respectively. She is now pursuing her Ph.D. in computer science at Wayne State University (USA) from 2016. Her current research interests include deep learning for 3D models, natural language processing and information retrieval.

Yin Li received his BSc degree in Information Engineering, and the MSc degree in Cryptography from Information Engineering University, Zhenzhou, in 2004 and 2007, and the Ph.D. in Computer Science from Shanghai Jiaotong University (SJTU), Shanghai (2011). He was a postdoc in Department of Computer Science, Ben-Gurion University of the Negev, Israel. Now he is a lecturer in Department of Computer Science and Technology, Xinyang Normal University, Henan, China. His current research interests include algorithm and architectures for computation in finite field, computer arithmetic, secure cloud computing.