Received November 15, 2019, accepted December 18, 2019, date of publication December 23, 2019,
date of current version January 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2961633

Expressive Public-Key Encryption with Keyword Search: Generic Construction from KP-ABE and an Efficient Scheme Over Prime-Order Groups

CHEN SHEN [1], YANG LU [1,2], AND JIGUO LI [3,4]
[1]College of Computer and Information, Hohai University, Nanjing 211100, China
[2]School of Computer Science and Technology, Nanjing Normal University, Nanjing 210097, China
[3]College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350007, China
[4]State Key Laboratory of Cryptology, Beijing 100878, China

Corresponding author: Yang Lu (luyangnsd@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772009, Grant U173610004, and Grant 61972095, and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20181304 and Grant BK20161511.

ABSTRACT Public key encryption with keyword search (PEKS) allows a cloud server to retrieve particular ciphertexts without leaking the contents of the searched ciphertexts. This kind of cryptographic primitive gives users a special way to retrieve the encrypted documents they need while preserving privacy. Nevertheless, most existing PEKS schemes only offer single-keyword search or conjunctive-keyword search. The poorly expressive ability and constantly inaccurate search results make them hard to meet users' requirements. Although several expressive PEKS (EPEKS) schemes were proposed, they entail high computation and communication costs. An ideal EPEKS scheme should enable fast and accurate ciphertext retrieval, while lowering the storage server's load and reducing the amount of communication data. Drawing on the strongly expressive ability of key-policy attribute-based encryption (KP-ABE), we propose a generic construction of EPEKS from KP-ABE. We demonstrate that the derived EPEKS scheme is secure under the chosen keyword attack if the implicit KP-ABE scheme fulfills the anonymity under the chosen plaintext attack. Furthermore, we present a concrete EPEKS scheme over the prime-order groups. The comparison and experimental results indicate that our scheme is more efficient than the existing EPEKS schemes.

INDEX TERMS Searchable encryption, expressive keyword search, key-policy attribute-based encryption, prime-order group.

I. INTRODUCTION

With the prevalence of the Internet and the widespread application of cloud computing technology, personal privacy information often undergoes massive transmission via channels such as computer networks and public communication devices. These information transmission media are unsafe yet hardly replaceable. Asymmetric cryptosystem was developed to allow people to share secret information without transmitting decryption keys. But in some cases, people need to process the encrypted information. Imagining such a situation, a user uploads a large quantity of encrypted data files to an untrusted server. Later, the user wants to fetch back some certain files from the server. How could the server pick out the target documents from a large amount of ciphertexts?

The associate editor coordinating the review of this manuscript and approving it for publication was Aniello Castiglione.

In another case, to protect personal privacy, a user sends encrypted mails to the email sever. How could the receiver of the mails tell which mails contain important contents that need urgent processing and which ones could be directly ignored? One primitive way is to download and decrypt all received emails, before being able to get the wanted information. But this will result in large communication and computation cost, hence very inefficient. To address the problem, the paradigm of public key encryption with keyword search (PEKS) [1] was invented. PEKS allows a message sender to create a searchable ciphertext by attaching a keyword ciphertext to the encrypted file. To execute ciphertext search, the recipient makes use of his/her private key to produce a trapdoor of the search keyword (or keywords) and then sends it to the server. The server can search the ciphertexts using the trapdoor and returns all matching files. In this process, no information (neither the contents of the searched

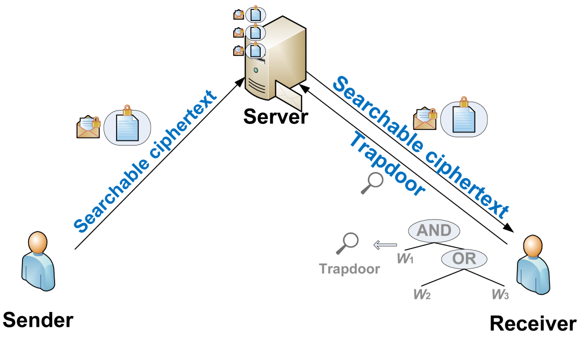


FIGURE 1. Framework of EPEKS.

ciphertexts nor the search keyword(s)) would be disclosed to the server.

A data file may be associated with multiple keywords. However, a PEKS scheme only enables the server to retrieve documents that contain a certain keyword. These schemes can't meet the user's needs because the single keyword search often results in coarse search results. Users are more likely to use multiple keywords in their daily searches. Therefore, Boolean combination of search keywords is necessary to make data retrieval effective. In [2], Park *et al.* proposed the first PEKS scheme that can execute multiple-keyword search, namely public key encryption with conjunctive keyword search (PECKS). PECKS enables recipients to seek encrypted files with more than one keyword. But, it can only support keyword conjunction, therefore does not have sufficient expressive power. If a user wants to get the documents marked by a keyword "important" or a keyword "urgent", he/she must search twice. To realize more expressive keyword search, Lai *et al.* [3] proposed the expressive PEKS (EPEKS) scheme that supports the logical expression of both "AND" and "OR". As illustrated in FIGURE 1, an EPEKS scheme includes three entities: the server, the sender and the receiver. The sender sends to the server ciphertexts attached with searchable encrypted labels. The searchable encrypted labels are associated with a keyword set. The receiver generates a trapdoor according to the logical expression of keywords (which, in FIGURE 1, is shown as a logic tree). When the server gets the trapdoor from the receiver, it runs a test algorithm and sends to the receiver particular ciphertexts that pass the test algorithm. In [3], [4], two EPEKS schemes were presented respectively but over the composite-order groups. These two schemes are unfriendly to PCs because in the composite-order groups, the elements are longer than elements in the prime-order groups and the computation cost is higher. How to build efficient EPEKS schemes over the prime-order groups with strong expressive ability remains a hotspot.

As is known to all, attribute-based encryption (ABE) has a very strong access control capability [5]. In ABE, attributes are usually administered by a single central trusted authority that awards private keys to users. Each user's private key contains information on user attributes. There are two types of ABE schemes: one is the key-policy ABE (KP-ABE), and the other is ciphertext-policy ABE (CP-ABE). In a KP-ABE, an access structure (AS) is implanted in the private key and the ciphertext has a bearing on a set of attributes. Opposite to that in KP-ABE, an access structure in a CP-ABE is implanted in the ciphertext and the private key has a bearing on a set of attributes. FIGURE 2 shows the framework of KP-ABE. In a KP-ABE scheme, the trusted center uses a logical expression of attributes (which, in FIGURE 2, is shown as a logic tree) to generate an access structure. One sound way to construct an access structure is using a linear secret-sharing scheme (LSSS). The ciphertext gets decrypted only when the access structure is met by the attribute set. An access structure built via LSSS could enable the KP-ABE scheme to realize access control in cases that the logical expressions of attributes contain "AND" and "OR".

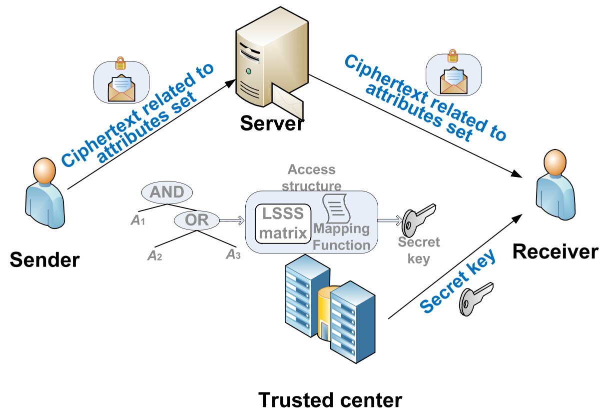


FIGURE 2. Framework of KP-ABE.

This paper proposes a generic construction of EPEKS from KP-ABE and gives an efficient EPEKS scheme over the prime-order groups.

A. RELATED WORKS

In [6], Song came up with the concept of searchable encryption and exhibited a specific scheme under symmetric key system. Boneh *et al.* [1] gave the first PEKS scheme in 2004 and proposed a generic construction of PEKS from identity-based encryption (IBE). Since then, many scholars have proposed lots of improved PEKS schemes to enhance the scheme performance or security [7]–[20].

To improve search accuracy when using search engines, users are more likely to search several keywords rather than a single keyword. Multi-keyword search is also needed for retrieving ciphertext. Golle *et al.* [21] constructed a searchable symmetric encryption scheme with conjunctive-keyword search. In the scheme, every document has several keyword domains and each keyword domain has a keyword to represent a feature. The communication cost changes linearly with the number of keyword domains and the feature representation is not flexible enough due to constraints by keyword domains. Park *et al.* [2] gave the first PEKS scheme supporting conjunctive-keyword search. Based on Park *et al.*'s works, further efforts were made to reduce computation cost and trapdoor size [22]–[25].

EPEKS has attracted widespread concern in the domain of searchable encryption because of its strong search function. Lai *et al*. [3] put forward the first EPEKS scheme on the basis of a completely secure KP-ABE scheme [26]. Lai *et al*.'s scheme is established over the composite-order groups. Hence, its computation cost is high and the length of the ciphertext and that of the trapdoor are both linear to the keyword number. Lv *et al*. [4] proposed the first expressive PEKS scheme supporting "AND", "OR" and "NOT". This scheme is also over the composite-order groups and hence inefficient. In 2016, Cui *et al*. [27] embedded the LSSS structure into keyword search and, for the first time, implemented an EPEKS scheme over the prime-order groups. However, both the communication cost and the computation cost of the scheme remain high.

In 1984, Shamir published a paper to describe the concept of identity-based cryptography [28]. His core idea is to directly use some inherent identity information as users' public keys, while the private keys are distributed to users by a trusted third party. In 2001, Boneh and Franklin successfully constructed the first pragmatic identity-based encryption scheme which is provably secure [29]. Their scheme makes use of the bilinear mapping technology. In [30], Sahai and Waters proposed a fuzzy identity-based encryption (FIBE) scheme which is regarded as the embryonic form of ABE. FIBE extends IBE by lablling each user with a set of idenities. In an FIBE scheme, a ciphertext could be decrypted only when intersection of the identity set for encryption and the identity set for decryption is greater than a threshold. But, the threshold access structure limits the scope of scheme application. In [31], Goval *et al*. published a paper and exhibited the first KP-ABE scheme. This scheme is not applicable in large attribute universe environment because its public parameter is linear to the attribute number in the universe. Lewko and Waters [32] proposed the first large universe KP-ABE scheme but over the composite-order groups. Lemko [33] proposed a KP-ABE scheme in large universe and this scheme was constructed over the prime-order groups. By now, many efficient KP-ABE schemes have been given [34]–[37]. In [38], Wang *et al*. gave an ABE scheme with keyword search. This scheme combines ABE with PEKS, and makes it possible that only users complying with the access control strategy could search the ciphertexts. In [39], Zheng *et al*. designed a verifiable attribute-based keyword search scheme. This scheme could verify whether the server has performed retrieval operations as required, therefore supports the monitoring of malicious servers. In 2017, Li *et al*. also proposed schemes of this type [40], [41]. In addition, Zhang *et al*. [42] and Jung *et al*. [43] respectively gave anonymous ABE schemes to protect the privacy of attributes.

B. MOTIVATION AND CONTRIBUTIONS

This paper focuses on the efficient construction of EPEKS from KP-ABE. KP-ABE has strong access control capacity and efficient operation performance. In a KP-ABE scheme, every user is marked by an attribute set and only users with specific attributes are authorized to decrypt a specific ciphertext. Clearly, KP-ABE makes user screening possible. Implementing such a screening process on a cloud storage sever, users can only retrieve specific files, which is exactly what EPEKS could do. This inspires us to devise a generic transformation from KP-ABE to EPEKS.

In a KP-ABE scheme, a trusted center authority generates users' private keys according to the user attributes. If the user attributes are regarded as the search keywords, then the private key generation algorithm in the KP-ABE scheme could be used to generate the trapdoors of search keywords in the EPEKS scheme. Correspondingly, the keyword ciphertexts in EPEKS could be generated by using the KP-ABE encryption algorithm to encrypt a random message. The test algorithm in the EPEKS scheme could be executed by decrypting the random-message ciphertext and checking whether the decrypted message is the same as that in the original ciphertext. In so doing, the strong access control ability of KP-ABE on user screening could be inherited by the derived EPEKS scheme to screen files. However, such transformation is unsuitable to most existing KP-ABE schemes, because these schemes should attach an attribute set behind the generated ciphertext and thus don't provide any protection to the user attributes. Privacy protection of the keywords is a very important issue in the construction of EPEKS. Therefore, these KP-ABE schemes cannot be directly exploited to construct the EPEKS schemes.

To protect the privacy of attributes, some anonymous ABE schemes were proposed, *e.g*. [34], [35]. This kind of schemes can be transformed to EPEKS directly, but they are quite inefficient. After a close examination of existing KP-ABE schemes, we find that most KP-ABE schemes could turn anonymous if the attribute sets get removed from the ciphertexts. But such removal makes the ciphertext decryption a challenging task, which also makes the test algorithm in the post-transformation EPEKS scheme ineffective. In [27], Cui *et al*. provided a solution to this problem, which exposes the keyword attribute names while hiding the keyword values. For example, during the production of a ciphertext with a keyword set {"job = teacher", "gender = male"}, the attribute names ("job", "gender") are attached to the ciphertext without displaying the keyword values. In this way, the privacy of keywords is preserved. Actually, in many practical retrieval systems, the search keywords are input in certain orders according to the attributes of the generic names. After inputting the search keywords, users could search for their expected documents accurately. In such context, the number and order of keywords are both pre-defined. Therefore, if the attributes (including the number and the order) of the keywords encrypted in ciphertexts are pre-defined, the keyword attribute names need not be attached to the ciphertexts.

In this paper, we provide a generic construction of EPEKS from anonymous KP-ABE. Then, a concrete EPEKS scheme is derived from an anonymous KP-ABE scheme to show the application of the generic construction. Below are the concrete contributions:

1) We present an efficiently generic EPEKS construction that provides a general way to build the EPEKS schemes from the anonymous KP-ABE schemes directly. The derived EPEKS scheme is indistinguishable secure against chosen keyword attacks if the underlying KP-ABE scheme fulfills anonymity against chosen plaintext attacks. We formally show the proving process.
2) We construct an efficient EPEKS scheme and formally prove that it achieves indistinguishability against chosen keyword attacks. As shown in Table 1, our EPEKS scheme enjoys many merits. It is established over the prime-order groups so that it has significant advantages in performance over the EPEKS schemes over the composite-order groups [3], [4]. The comparison and the experimental results show that it also outperforms Cui *et al.*'s scheme [27] which is the only EPEKS scheme over the prime-order groups before ours. Moreover, it supports unbounded keywords and expressive search by the logical expression "AND" and "OR" of the search keywords.

TABLE 1. Properties of the epeks schemes.

Schemes	Group type	Unbounded keywords	Expressiveness
[3]	Composite-order	no	AND, OR
[4]	Composite-order	no	AND, OR, NOT
[27]	Prime-order	yes	AND, OR
Ours	Prime-order	yes	AND, OR

C. PAPER ORGANIZATION

Section II briefly lists some background notions and definitions. In section III, we give the generic construction from an anonymous KP-ABE scheme to an EPEKS scheme and then demonstrate its security. In the ensuing section IV, we propose an anonymous KP-ABE scheme over the prime-order groups and formally prove its security. Then we convert the proposed KP-ABE scheme into a concrete EPEKS scheme. In Section V, we implement the derived EPEKS scheme and compare it with Cui *et al.*'s EPEKS scheme. In Section VI, we make a summary and present suggestions for further research efforts.

II. PRELIMINARIES

This section reviews some essential background knowledge briefly.

A. BILINEAR MAP AND COMPLEXITY ASSUMPTION

Define G as a group of prime order p. A bilinear map e: $G \times G \rightarrow G_T$ between group G and group G_T must be with properties as follows:

1) Bilinear: For all $g \in G$ and all $a,b \in Z_p$, $e(g^a, g^b) = e(g, g)^{ab}$;
2) Non-degenerate: $e(g, g) \neq 1$.
3) Computable: For any $g_1,g_2 \in G$, $e(g_1,g_2)$ can be computed efficiently.

The security of our proposed EPEKS scheme is on the basis of decisional (q - 2) assumption [5].

Definition 1: Define q as an integer and let there be a bilinear group environment (p, G, G_T, e). The decisional (q-2) assumption is: given elements

$$\begin{array}{ll} g, g^x, g^y, g^z, g^{(xz)^2} & \\ g^{b_i}, g^{xzb_i}, g^{xz/b_i}, g^{x^2zb_i}, g^{y/b_i^2}, g^{y^2/b_i^2} & \forall i \in [q] \\ g^{xzb_i/b_j}, g^{yb_i/b_j^2}, g^{xyzb_i/b_j}, g^{(xz)^2b_i/b_j} & \forall i,j \in [q], i \neq j \end{array}$$

in G, it is hard to differentiate $e(g, g)^{xyz}$ from a random element T in G_T for any polynomial-time (PT) adversary. Here $g \in G$ and $x, y, z, b_1, \ldots, b_q$ are chosen randomly from Z_p.

The decisional (q-2) assumption declares that for any PT adversary A, the advantage Adv_A in figuring out the decisional (q-2) problem is negligible. Here Adv_A is defined to be $|\Pr[A(S, e(g, g)^{xyz}) = 1]$ - $\Pr[A(S, T) = 1 \,|T \in G_T]|$, where S denotes the set of given elements as shown above.

B. ACCESS STRUCTURE AND LINEAR SECRET SHARING SCHEME

We describe the concepts of access structure and linear secret sharing technique following the definitions in [5].

Definition 2: Define U as the attribute universe. An access structure AS on U is a collection of nonempty attribute sets, *i.e.* $AS \subseteq 2^U/\{\emptyset\}$. The sets in AS are named the authorized sets and the sets not in AS are named the unauthorized sets.

If an access structure satisfies that $C \in AS$ can be deduced from $\forall B$, $C \in AS$ and $B \subseteq C$, this access structure is monotone.

Definition 3: Define p as a prime and U as the universe of attributes. A secret-sharing scheme with domain of secrets Z_p realizing access structures on U is linear over Z_p if:

1) For each attribute form a vector over Z_p, the shares of a secret $s \in Z_p$.
2) For each access structure AS on U, there is a share-generating matrix $MA \in Z_p^{l\times n}$.
3) There exists a mapping ρ, that connects each row of MA with an attribute from U, i.e. $\rho \in F([l] \rightarrow U)$, which conform to the following rules: In the course of the construction of the shares, we construct the column vector $\vec{v} = (s,r_2, \ldots, r_n)^{\perp}$, where $r_2, \ldots, r_n \in_R Z_p$. Then the vector of l shares of the secret s is equal to $MA\vec{v} \in Z_p^{l\times n}$. The share $(MA\overrightarrow{v})_j$ is related to attribute $\rho(j)$, where $j \in [l]$. Here $[l] = \{i \in Z | i < l\}$. The pair (MA, ρ) is the policy of the access structure AS.

C. ANONYMOUS KP-ABE AND SECURITY DEFINITION

A KP-ABE scheme is formed by four algorithms:

1) *Setup*(f). This algorithm is executed by a trusted central authority (TCA) and requires a security parameter f

as input. It generates the public parameters PP and a master key MK. MK is maintained secret by the TCA and the PP are made public.
2) *KeyGen*(PP, MK, AS). This algorithm is executed by the TCA and requires PP, MK, and an access structure AS as input. It generates a private key SK_{AS} according to the access structure AS.
3) *Encrypt*(PP, M, ATS). This algorithm is executed by the sender and requires PP, a message M and an attribute set ATS as input. It generates a ciphertext CT_{ATS} and outputs it. Only users with access structure AS that is met by ATS can decrypt CT_{ATS}.
4) *Decrypt*(PP, SK_{AS}, CT_{ATS}). This algorithm is executed by the receiver and demands PP, SK_{AS} and CT_{ATS} as input. It outputs a message M if the attribute set ATS corresponding to the ciphertext CT_{ATS} meets the access structure AS embedded in SK_{AS}. Otherwise, the algorithm will fail.

The following adversarial game defines the security of an anonymous KP-ABE scheme [34]. This game is carried out between an adversary A and a challenger Ch:
1) Init. A declares two challenge attribute sets ATS_0, ATS_1 with the same length.
2) Setup. Ch executes the *Setup* algorithm to get PP and MK. It then publishes PP and keeps the MK secret.
3) Phase 1. A can adaptively make private key queries for some access structures AS. If none of ATS_0, ATS_1 meets the queried access structure, the challenger executes *KeyGen* algorithm and returns the relevant private key SK_{AS} to A. Otherwise, it outputs $\perp$. The private key queries can be asked for a finite number of times.
4) Challenge. A sends Ch a message M. Ch picks a random number $b \in \{0, 1\}$ and executes algorithm *Encrypt*(PP, ATS_b, M) to get a ciphertext which is returned to A afterwards.
5) Phase 2. Proceed as in Phase 1.
6) Guess. The adversary A outputs a guess bit $b' \in \{0, 1\}$ and wins the game if $b = b'$. The adversary's advantage in the adversarial game is $Adv_A = |\Pr[b = b'] - 1/2|$.

Definition 4: A KP-ABE scheme satisfies the anonymity under the chosen plaintext attack (ANO-IND-CPA) if no polynomial-time adversary can break the above adversarial game with a non-negligible advantage.

D. EPEKS AND SECURITY DEFINITION

An EPEKS scheme is formed by four randomized algorithms below:
1) *KeyGen*(f). This algorithm is performed by the receiver and requires a security parameter f as input. It outputs user's public key PK and private key SK.
2) *Trapdoor*(PK, SK, P). This algorithm is executed by the receiver and requires PK,SK and a search predicate P as input. It generates T_P as the trapdoor of the predicate P.
3) *Encrypt*(PK, WS). This algorithm is executed by the sender and requires PK and a keyword set WS as input. It produces a searchable encryption SE_{WS} of the keyword set WS.
4) *Test*(PK, T_P, SE_{WS}). This algorithm is executed by the server and requires PK, T_P and SE_{WS} as input. It outputs 1 if the keyword set WS corresponding to the searchable encryption SE_{WS} meets the predicate P embedded in trapdoor T_P or 0 otherwise.

An EPEKS scheme should not leak any information about the WS encoded in SE_{WS}. It should guarantee that the adversary can't distinguish two encryptions of WS_0 and WS_1 as long as the adversary has never gained the corresponding trapdoor. In this paper, we adopt the security model provided by Cui *et al.* [27], where the security of an EPEKS scheme is defined through the following adversarial game:
1) Init. The adversary A declares two challenge keyword sets WS_0, WS_1 with the same length.
2) Setup. The challenger Ch executes the *KeyGen* algorithm to generate PK and SK. It publishes PK and keeps the SK secret.
3) Phase 1. The adversary can request the trapdoor T_P for any predicate P as long as WS_0 and WS_1 do not meet P. The Ch then performs the Trapdoor algorithm and returns the result to the A. This procedure can be executed for a finite number of times.
4) Challenge. The challenger tosses a coin and gets a random number $b \in \{0, 1\}$. It sends the adversary $S_{WS_b} = Encrypt(PK, WS_b)$ as the challenge ciphertext.
5) Phase 2. Proceed as in Phase 1.
6) Guess. The adversary outputs its answer bit $b' \in \{0, 1\}$ and wins the adversarial game if $b = b'$. The adversary's advantage in the game is $Adv_A = |\Pr[b = b'] - 1/2|$.

Definition 5: An EPEKS scheme satisfies the indistinguishability under the chosen keyword attack (IND-CKA) if no PT adversary can break the above adversarial game with a non-negligible advantage.

III. FROM KP-ABE TO EPEKS

In this section, we propose a generic construction of EPEKS from anonymous KP-ABE and demonstrate its security.

A. GENERIC CONSTRUCTION

Let $KP\text{-}ABE$ = (*Setup*, *KeyGen*, *Encrypt*, *Decrypt*) be an anonymous KP-ABE scheme with message space *MSpace*. Then, an EPEKS scheme *EPEKS* = (*KeyGen*, *Encrypt*, *Trapdoor*, *Test*) can be constructed in the following steps:
1) *EPEKS.KeyGen*(f). Inputting f, this algorithm executes as follows:
   - Run (PP, MK) $\leftarrow$ *KP-ABE.Setup*(f);
   - Set $PK \leftarrow PP$ and $SK \leftarrow MK$;
   - Output (PK, SK).
2) *EPEKS.Trapdoor*(PK, SK, P). Inputting PK, SK and P, this algorithm executes as follows:
   - Generate an AS from P;
   - Run SK_{AS} $\leftarrow$ *KP-ABE.KeyGen*(PK, SK, AS);

- Set $T_P \leftarrow SK_{AS}$;
- Output T_P.

3) *EPEKS.Encrypt*(*PK*, *WS*). Inputting *PK* and *WS*, this algorithm executes as follows:
   - Pick a random message $R \in MSpace$;
   - Set $ATS \leftarrow WS$;
   - Run $CT_{ATS} \leftarrow$ *KP-ABE.Encrypt*(*PK*, *R*, *ATS*);
   - Set $SE_{WS} \leftarrow (CT_{ATS}, R)$;
   - Output SE_{WS}.
4) *EPEKS.Test*(*PK*, T_P, SE_{WS}). Inputting *PK*, T_P and SE_{WS}, this algorithm executes as follows:
   - Parse SE_{WS} as (CT_{ATS}, R);
   - Set $SK_{AS} \leftarrow T_P$;
   - Run $R' \leftarrow$ *KP-ABE.Decrypt*(*PK*, SK_{AS}, CT_{ATS});
   - If $R' = R$, output 1; else, output 0.

Theorem 1: If the scheme KP-ABE is ANO-IND-CPA secure, then the derived scheme EPEKS is IND-SCP-CKA secure.

Proof: Assuming that there is an adversary *A* who can break the IND-CKA security of the scheme *EPEKS* with a non-negligible advantage ε, we show that an adversary *B* can be built to break the ANO-IND-CPA security of the *KP-ABE* scheme with the same advantage. Here *Ch* is the challenger of the ANO-IND-CPA game. The adversary *B* imitates the challenger of the IND-CKA game and interacts with the adversary *A* as follows.

1) Init. *A* sends two different keyword sets WS_0 and WS_1 of the same length to the adversary *B*. Then, *B* sends them to *Ch* as two attribute sets ATS_0 and ATS_1 in the ANO-IND-CPA game.
2) Setup. *Ch* runs the algorithm *KP-ABE.Setup* to generate (*PP*, *MK*) and gives *B* the parameters *PP*. After getting *PP*, the adversary *B* sends it to *A* as the challenge public key *PK* in the IND-CKA game.
3) Phase 1. Adversary *A* adaptively makes a polynomial number of trapdoor queries. When *A* requests for the trapdoor of a predicate *P*, the adversary *B* performs in the following way:
   - If none of WS_0, WS_1 meets the predicate *P*, the adversary *B* builds an access structure *AS* corresponding to the logical expression of the predicate *P*, and then requests for the private key corresponding to the access structure *AS* from the challenger *Ch* in the ANO-IND-CPA game. *Ch* runs the algorithm *KP-ABE.KeyGen* to produce a private key SK_{AS} and feeds it back to *B*. The adversary *B* sends SK_{AS} as the trapdoor of the predicate *P* to *A*.
   - Otherwise, the adversary *B* rejects the query.
4) Challenge. *B* sends a random message *R* to *Ch*. The challenger *Ch* tosses a coin and gets a random number $b \in \{0, 1\}$. *Ch* executes algorithm *KP-ABE.Encrypt*(*PP*, ATS_b, *R*) to produce a challenge ciphertext CT_{ATS_b} and feeds it back to the adversary *B*. Once getting CT_{ATS_b}, *B* sends (CT_{ATS_b}, R) to *A* as the challenge ciphertext in the IND-CKA game.
5) Phase 2. Proceed as in Phase 1.
6) Guess. The adversary *A* outputs its answer $b' \in \{0,1\}$. Then, the adversary *B* sends b' to *Ch* as its guess in the ANO-IND-CPA game.

According to the above simulation, we clearly have that the adversaries *A* and *B* have the same success probability in guessing *b*. Therefore, if *A* can break the IND-CKA security of the scheme *EPEKS* with advantage ε, then the adversary *B* can break the ANO-IND-CPA security of the scheme *KP-ABE* with the same advantage.

This proves Theorem 1.

IV. A CONCRETE EPEKS SCHEME

In this section, we first propose an efficient KP-ABE scheme and demonstrate it to be ANO-IND-CPA secure. Then, we transform the proposed KP-ABE scheme into an EPEKS scheme by using the generic construction presented above.

A. AN ANONYMOUS KP-ABE SCHEME

The proposed anonymous KP-ABE scheme is constructed as follows:

1) *Setup*(*f*). This algorithm is executed by the TCA and requires inputting a security parameter *f*. It generates a bilinear group (G, G_T) of prime order *p* and a bilinear map e: $G \times G \rightarrow G_T$. Then it picks a random generator $g \in G$ and three random elements $u, h, w \in G$ and a random number $\alpha \in Z_p$. Finally, it outputs the public parameters $PP = (p, G, G_T, e, g, u, h, w, e(g, g)^{\alpha})$ and maintains the master key $MK = \alpha$ secret.
2) *KenGen*(*PP*, *MK*, *AS*). This algorithm is executed by the TCA. It first picks a vector $\vec{y} = (\alpha, y_2, \ldots, y_n)^{\perp}$ where $y_2, \ldots, y_n \in Z_p$. Then it computes $\vec{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_l)^{\perp} = MA\vec{y}$, where *MA* is the share-generating matrix in the access structure *AS*. After this, it picks *l* random numbers $t_1, t_2, \ldots, t_l \in Z_p$. For every $\tau \in [l]$, it calculates $K_{\tau,0} = g^{\lambda_\tau} w^{t_\tau}$, $K_{\tau,1} = \left(u^{\rho(\tau)} h\right)^{-t_\tau}$ and $K_{\tau,2} = g^{t_\tau}$. Finally, it outputs the private key $SK_{AS} = (MA, \{K_{\tau,0}, K_{\tau,1}, K_{\tau,2}\}_{\tau \in [l]})$.
3) *Encrypt*(*PP*, *M*, *ATS*). This algorithm is executed by the sender. It chooses $k+1$ random numbers $s, r_1, r_2, \ldots, r_k \in Z_p$, calculates $C = M \cdot e(g, g)^{\alpha s}$, $C_0 = g^s$, and for every $\tau \in [k]$ it computes $C_{\tau,1} = g^{r_\tau}$ and $C_{\tau,2} = \left(u^{W_r} h\right)^{r_\tau} w^{-s}$, where $[k] = \{i \in Z | i < k\}$. Finally, it generates the ciphertext $CT_{ATS} = (C, C_0, \{C_{\tau,1}, C_{\tau,2}\}_{\tau \in [k]})$.
4) *Decrypt*(*PP*, SK_{AS}, CT_{ATS}). This algorithm is executed by the server. Let $\boldsymbol{I}_{AS}$ be the minimum subset meeting *AS*. The sever calculates $\boldsymbol{I}_{AS}$ from the access structure *MA* and checks whether there is an $\boldsymbol{I} \in \boldsymbol{I}_{AS}$ satisfying

$$M = \frac{C}{\prod_{i \in \boldsymbol{I}} \left(e\left(C_0, K_{i,0}\right) e\left(C_{\tau,1}, K_{i,1}\right) e\left(C_{\tau,2}, K_{i,2}\right)\right)^{\omega_i}},$$

where $\{\omega_i \in Z_p\}_{i \in I}$. Note that $\sum_{i \in I} \omega_i MA_i = (1, 0, \ldots, 0)$ where MA_i is the i^{th} row of the matrix *MA*. It outputs $\perp$ if no element in $\boldsymbol{I}_{AS}$ satisfies the above equation or *M* otherwise.

Correctness: If the attribute set ATS is authorized, then we have the equation $\sum_{i\in \boldsymbol{I}} \omega_i\lambda_i = \alpha$.According to the above description, we have

$$\begin{aligned}&\prod\nolimits_{i\in \boldsymbol{I}} \left(e\left(C_0, K_{i,0}\right) e\left(C_{\tau,1}, K_{i,1}\right) e\left(C_{\tau,2}, K_{i,2}\right)\right)^{\omega_i}\\ &= \prod_{i\in \boldsymbol{I}} e\left(g,g\right)^{s\omega_i\lambda_i} e\left(g,w\right)^{st_i\omega_i} e\left(g, u^{\rho(i)}h\right)^{-r_\tau t_i \omega_i}\\ &\cdot \prod_{i\in \boldsymbol{I}} e\left(g, u^{\rho(i)}h\right)^{r_\tau t_i\omega_i} e\left(g,w\right)^{-st_i\omega_i}\\ &= e\left(g,g\right)^{s\sum\limits_{i\in \boldsymbol{I}}\omega_i\lambda_i}\\ &= e\left(g,g\right)^{\alpha s}.\end{aligned}$$

Therefore, the proposed scheme is correct.

B. SECURITY OF THE PROPOSED KP-ABE SCHEME

Theorem 2: If the q-2 decisional assumption holds, then the proposed KP-ABE scheme conforms to the ANO-IND-CPA security in the standard model.

Proof: If there is a PT adversary A who can break the ANO-IND-CPA security of the proposed KP-ABE scheme with a non-negligible advantage ε, then we can build an algorithm B to solve the decisional (q-2) problem with a non-negligible advantage ε.

Assuming that the algorithm B gets a random instance of the decisional (q-2) problem

$$\left\{\begin{array}{l} p, G, G_T, e, g, g^x, g^y, g^z, g^{(xz)^2}\\ g^{b_i}, g^{xzb_i}, g^{xz/b_i}, g^{x^2zb_i}, g^{y/b_i^2}, g^{y^2/b_i^2}\ \forall i\in[q]\\ g^{xzb_i/b_j}, g^{yb_i/b_j^2}, g^{xyzb_i/b_j}, g^{xyzb_i/b_j}\ \forall i,j\in[q], i\neq j\\ T\end{array}\right\},$$

where $g \in G$, $x, y, z, b_1, \ldots, b_q \in Z_p^*$ and $T \in G_T$. The aim of the algorithm B is to ascertain that whether $T = e(g, g)^{xyz}$. To do so, the algorithm B simulates the challenger of the ANO-IND-CPA game and interacts with A as follows.

1) Init. The adversary A gives the algorithm B two attribute sets ATS_0 and ATS_1. We assume that both ATS_0 and ATS_1 include k ($k \leq q$) different attributes.
2) Setup. The algorithm B randomly chooses $\beta \in \{0, 1\}$. It then picks two random integers $\tilde{u}, \tilde{h} \in Z_p$ and sets $w = g^x$, $u = g^{\tilde{u}} \cdot \prod_{i\in[k]} g^{y/b_i^2}$, $h = g^{\tilde{h}} \cdot \prod_{i\in[k]} g^{xy/b_i} \cdot \prod_{i\in[k]} (g^{y/b_i^2})^{-A_i^*}$ and $e(g, g)^\alpha = e(g^x, g^y)$. Finally, it outputs PP= (p, G, G_T, e, g, u,h, w, $e(g, g)^\alpha$) as the PP to the adversary A. Here the master key is set as $\alpha = xy$ implicitly which is not known to the algorithm B.
3) Phase 1. In this phase, the algorithm B is required to create a private key for each access structure (MA, ρ) queried by the adversary A. The restriction is that the access structure is not met by either ATS_0 or ATS_1. Since ATS_β is not authorized by (MA, ρ), there exists a vector $\vec{\omega} = (\omega_1, \ldots, \omega_{\mathrm{n}})^\perp \in Z_p^n$ such that $\omega_1 = 1$ and $MA_i \cdot \vec{\omega} = 0$ for all ($i \in [l]$, $\rho(i)$ $\in ATS_\beta$). The vector $\vec{y}$ that will be shared is $\vec{y} = xy\vec{\omega} + (0, \vec{y}_2, \vec{y}_3, \ldots, \vec{y}_n)^\perp$ (this vector is set implicitly), where $\vec{y}_2, \vec{y}_3, \ldots, \vec{y}_n$ are random elements in Z_p. For each row $\tau \in [l]$, the share is $\lambda_\tau = MA_\tau \cdot \vec{y} = xy(MA_\tau \cdot \vec{\omega})+ (MA_\tau \cdot (0, \vec{y}_2, \vec{y}_3, \ldots, \vec{y}_n)^\perp) = xy(MA_\tau \cdot \vec{\omega}) + \vec{\lambda}_\tau$.

For each row in MA, if $\rho(\tau) \in ATS_\beta$, then $MA_\tau \cdot \vec{\omega} = 0$. In this case $\lambda_\tau = \vec{\lambda}_\tau$, the algorithm B selects a random element $t_\tau \in Z_p$ and outputs $K_{\tau,0}$, $K_{\tau,1}$, $K_{\tau,2}$ as in the algorithm *KeyGen*.

In another case, if $\rho(\tau) \notin ATS_\beta$, the algorithm B selects a random element $t_\tau \in Z_p$ and implicitly sets

$$t_\tau = -y(MA_\tau \cdot \vec{\omega}) + \sum_{i\in[k]} \frac{xzb_i(MA_i \cdot \vec{\omega})}{\rho(\tau) - ATS_{\beta,i}} + \tilde{t}_\tau.$$

Then, it produces a private key in the following way:

$$\begin{aligned}K_{\tau,0} &= g^{\lambda_\tau} w^{t_\tau}\\ &= g^{xy(MA_\tau\cdot\vec{\omega})+\vec{\lambda}_\tau} \cdot g^{-xy(MA_\tau\cdot\vec{\omega})+\sum_{i\in[k]}\frac{x^2zb_i(MA_\tau\cdot\vec{\omega})}{\rho(\tau)-ATS_{\beta,j}}} \cdot \omega^{\tilde{t}_\tau}\\ &= g^{\tilde{\lambda}_\tau} \cdot \prod_{i\in[n]} (g^{x^2zb_i})^{(MA_\tau\cdot\vec{\omega})/(\rho(\tau)-ATS_{\beta,i})} \cdot \omega^{\tilde{t}_\tau},\end{aligned}$$

$$\begin{aligned}K_{\tau,1} &= \left(u^{\rho(\tau)}h\right)^{-t_\tau}\\ &= (g^{\rho(\tau)\tilde{u}+\tilde{h}} \cdot \prod_{i\in[n]} g^{xz/b_i}\\ &\quad\cdot \prod_{i\in k} g^{y(\rho(\tau)-ATS_i)/b_i^2})^{y(MA_\tau\cdot\vec{\omega})-\sum_{i\in[k]}\frac{xzb_i(MA_\tau\cdot\vec{\omega})}{\rho(\tau)-ATS_i}}\\ &\quad\cdot(u^{\rho(\tau)}h)^{-\tilde{t}_\tau}\\ &= g^{y(MA_\tau\cdot\vec{\omega})(\rho(\tau)\tilde{u}+\tilde{h})}\\ &\quad\cdot \prod_{i\in[k]} g^{-xzb_i(\rho(\tau)\tilde{u}+\tilde{h})(MA_\tau\cdot\vec{\omega})/(\rho(\tau)-ATS_i)}\\ &\quad\cdot \prod_{i\in[k]} g^{xyz(MA_\tau\cdot\vec{\omega})/b_i}\\ &\quad\cdot \prod_{(i,j)\in[k,k]} g^{-(xz)^2b_j(MA_\tau\cdot\vec{\omega})/b_i(\rho(\tau)-ATS_i)}\\ &\quad\cdot \prod_{i\in[k]} g^{y^2(MA_\tau\cdot\vec{\omega})(\rho(\tau)-ATS_i)/b_i^2}\ \cdot\\ &\quad \prod_{(i,j)\in[k,k]} g^{-xyz(MA_\tau\cdot\vec{\omega})b_j(\rho(\tau)-ATS_i)/b_i^2(\rho(\tau)-ATS_i)}\\ &\quad\cdot(u^{\rho(\tau)}h)^{-\tilde{t}_\tau}\\ &= (g^y)^{(MA_\tau\cdot\vec{\omega})(\rho(\tau)\tilde{u}+\tilde{h})}\\ &\quad\cdot \prod_{i\in[k]} (g^{xzb_i})^{-(\rho(\tau)\tilde{u}+\tilde{h})(MA_\tau\cdot\vec{\omega})/(\rho(\tau)-ATS_i)}\\ &\quad\cdot \prod_{(i,j)\in[k,k]} (g^{(xz)^2b_j/b_i})^{-(MA_\tau\cdot\vec{\omega})/(\rho(\tau)-ATS_i)}\\ &\quad\cdot \prod_{i\in[k]} (g^{y^2/b_i^2})^{(MA_\tau\cdot\vec{\omega})(\rho(\tau)-ATS_i)}\ \cdot\\ &\quad \prod_{\substack{(i,j)\in[k,k]\\ i\neq j}} (g^{xyzb_j/b_i^2})^{-(MA_\tau\cdot\vec{\omega})(\rho(\tau)-ATS_i)/(\rho(\tau)-ATS_i)}\\ &\quad\cdot(u^{\rho(\tau)}h)^{-\tilde{t}_\tau},\end{aligned}$$

$$K_{\tau,2} = g^{t_\tau} = (g^y)^{-(MA_\tau\cdot\vec{\omega})} \cdot \prod_{i\in[k]} (g^{xzb_i})^{(MA_\tau\cdot\vec{\omega})/(\rho(\tau)-ATS_i)} \cdot g^{\tilde{t}_\tau}.$$

Therefore, the algorithm B can answer to the adversary A's private key queries correctly.

4) Challenge. A determines a message M and sends it to the algorithm B. B implicitly sets $s = z$ and $r_\tau = b_\tau$ for each $\tau \in [k]$. Then it sets $C = M \cdot T$, $C_0 = g^s = g^z$, $C_{\tau,1} = g^{r_\tau} = g^{b_\tau}$ and

$$\begin{aligned} C_{\tau,2} &= (u^{ATS_{\beta,\tau}}h)^{r_\tau} \cdot \omega^{-s} \\ &= g^{b_\tau(uATS_{\beta,\tau}+\tilde{h})} \cdot \prod_{i\in[k]} g^{xzb_\tau/b_i} \\ &\quad \prod_{i\in[k]} g^{yb_\tau(ATS_{\beta,k}-ATS_{\beta,i})/b_i^2} \cdot g^{-xz} \\ &= (g^{b_\tau})^{\tilde{u}ATS_{\beta,\tau}+\tilde{h}} \cdot \prod_{\substack{i\in[k] \\ i\neq\tau}} g^{xzb_\tau/b_i} \\ &\quad \prod_{\substack{i\in[k] \\ i\neq\tau}} (g^{yb_\tau/b_i^2})^{ATS_{\beta,\tau}-ATS_{\beta,i}}. \end{aligned}$$

Finally, the algorithm B sends $CT_{ATS} = (C, C_0, \{C_{\tau,1}, C_{\tau,2}\}_{\tau\in[k]})$ to A as a challenge ciphertext.

5) Phase 2. Proceed as in Phase 1.

6) Guess. A outputs its answer β' for β. If $\beta' = \beta$, the algorithm B outputs 1 which means that T is equal to $e(g, g)^{xyz}$. Otherwise, it outputs 0.

If $T = e(g, g)^{xyz}$, the algorithm B provides a legal challenge ciphertext to A. Therefore, $\Pr[\beta' = \beta] = 1/2 \pm \varepsilon$. Otherwise, the ciphertext is invalid and thus $\Pr[\beta' = \beta] = 1/2$. Therefore, the advantage of the algorithm B in dealing with the given decisional (q-2) problem is $|1/2 \pm\varepsilon - 1/2| = \varepsilon$.

This proves Theorem 2.

C. AN EFFICIENT EPEKS SCHEME

Based on the above anonymous KP-ABE scheme, an EPEKS scheme can be derived as follows:

1) *KeyGen*(f). This algorithm generates the environment including bilinear groups (G, G_T) of prime order p and a bilinear map e: $G \times G \rightarrow G_T$. Then it picks a random generator $g \in G$, three random elements u,h, $w \in G$ and a random number $\alpha \in Z_p$. Finally, it outputs $PK = (p, G, G_T, e, g, u, h, w, e(g, g)^\alpha)$ and $SK = \alpha$.

2) *Trapdoor*(PK, SK, P). This algorithm is executed by the receiver. It first generates an access structure AS from P. Then it picks a vector $\vec{y} = (\alpha, y_2, \ldots, y_n)^\perp$ where $y_2, \ldots, y_n \in Z_p$ and computes $\vec{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_l)^\perp = MA\vec{y}$, where MA is the share-generating matrix in the access structure AS. Finally, it picks l random numbers $(t_1, t_2, \ldots, t_l) \in Z_p$ and computes $K_{\tau,0} = g^{\lambda_\tau} w^{t_\tau}$, $K_{\tau,1} = \left(u^{\rho(\tau)}h\right)^{-t_\tau}$, $K_{\tau,2} = g^{t_\tau}$ for every $\tau \in [l]$. The trapdoor is $T_P = (MA, K_{\tau,0}, K_{\tau,1}, K_{\tau,2})_{\tau\in[l]}$.

3) *Encrypt*(PK,WS). This algorithm is performed by the sender and requires a set of attributes $WS = \{W_1, W_2, \ldots, W_k\} \subseteq Z_p$ and the receiver's PK as input. It chooses $k+1$ random numbers $(s, r_1, r_2, \ldots, r_k) \in_R Z_p$ and calculates $C = e(g, g)^{\alpha s}$, $C_0 = g^s$, and for every $\tau \in [k]$ it calculates $C_{\tau,1} = g^{r_\tau}$ and $C_{\tau,2} = \left(u^{W_r}h\right) w^{-s}$. The searchable encryption is $SE_{WS} = (C, C_0, \{C_{\tau,1}, C_{\tau,2}\}_{\tau\in[k]})$.

4) *Test*(PK, SE_{WS}, T_P). This algorithm is executed by the server. Let $\boldsymbol{I}_{AS}$ be the minimum subset meeting AS generated from P. The sever calculates $\boldsymbol{I}_{AS}$ from MA and checks whether there is an $\boldsymbol{I} \in \boldsymbol{I}_{AS}$ satisfying

$$C = \prod\nolimits_{i\in\boldsymbol{I}} \left(e\left(C_0, K_{i,0}\right) e\left(C_{\tau,1}, K_{i,1}\right) e\left(C_{\tau,2}, K_{i,2}\right)\right)^{\omega_i},$$

where $\sum_{i\in\boldsymbol{I}} \omega_i MA_i = (1, 0, \ldots, 0)$ and MA_i is the i^{th} row of MA. It outputs 0 if no element in $\boldsymbol{I}_{AS}$ meets this equation, and 1 otherwise.

Theorem 3: If the q-2 decisional assumption holds, then the above EPEKS scheme conforms to the IND-CKA security in the standard model.

Proof: This theorem can be proved by combining Theorem 1 and Theorem 2.

V. PERFORMANCE ANALYSIS

In this section, we compare our EPEKS scheme with Cui *et al.*'s scheme [27] in the aspects of the computation cost and the communication cost. Considering that the EPEKS schemes in [3, 4] are over the composite-order groups and hence inefficient, we do not involve them into the comparison.

A. COMPARISON

Let l be the row number of the matrix in AS, k be the number of keywords encrypted in a ciphertext, $|MA|$ be the size of an access structure, $|G|$ be the element length in the group G, $|G_T|$ be the element length in the group G_T, Ex be an exponentiation computation, Pa be a pairing computation, X_1 be the of element number in $\boldsymbol{I}_{\boldsymbol{M},\rho} = \{I_1, \ldots, I_{X_1}\}$ (the number of authorized sets), X_2 be $|I_1| + \ldots + |I_{X_1}|$ and X_3 be the number of keywords in a search predicate. The computation cost and the communication cost of the compared schemes are respectively shown in TABLE 2 and TABLE 3. It is obvious that our scheme outperforms Cui *et al.*'s scheme on both the computation cost and the communication cost.

TABLE 2. Comparison of communication cost.

Schemes	Public parameters	Trapdoor	Ciphertext
[27]	$8\|G\| + \|G_T\|$	$(6l+2)\|G\| + \|MA\|$	$(5k+1)\|G\| + \|G_T\|$
Ours	$4\|G\| + \|G_T\|$	$3l\|G\| + \|MA\|$	$(2k+1)\|G\| + \|G_T\|$

TABLE 3. Comparison of computation cost.

Schemes	*Trapdoor*	*Encrypt*	*Test*
[27]	$(16l+2)Ex + lPa$	$(7k+2)Ex$	$(X_2+1)Ex + (6X_2+1)Pa$
Ours	$5lEx$	$(4k+2)Ex$	$X_2Ex + 3X_2Pa$

B. EXPERIMENTAL RESULTS

We test two schemes on a Lenovo L440 Laptop equipped with Intel Core i7 CPU (2.3GHz) and 8GB RAM. Our operate system is Win 7 (64 bit). The PBC (Pairing-Based Cryptography)-0.5.14 library [44] is installed for cryptographic operation. The bilinear map is established on Type A pairing over the elliptic curve with 512-bit group size.

FIGURE 3, 4, 5 and 6 show the experimental results. We randomly choose 2-10 keywords to generate a predicate P and get trapdoor from the P. Actually, the number of keywords in a searching query is no more than 10 in practical application. As shown in FIGURE 3, Trapdoor generation for 2, 4, 6, 8, 10 keywords in our scheme costs about 32.485ms, 59.693ms, 83.046ms, 125.338ms and 178.189ms, respectively, while that in scheme [27] is about 93.265ms, 179.731ms, 258.124ms, 349.251ms and 452.572ms, respectively. To check the time cost of the encryption algorithm, we generate different random keyword sets containing 10-50 keywords to generate the ciphertexts. As shown in FIGURE 4, our scheme costs about half of the time required by Cui *et al.*'s scheme [27]. The computation cost of Test algorithm is related to predicate P and the keywords used to generate SE_{WS}. The computation time will increase as the number of keywords in both the trapdoor and the ciphertext increases. The experimental results of two compared schemes are respectively given in FIGURE 5 and 6.

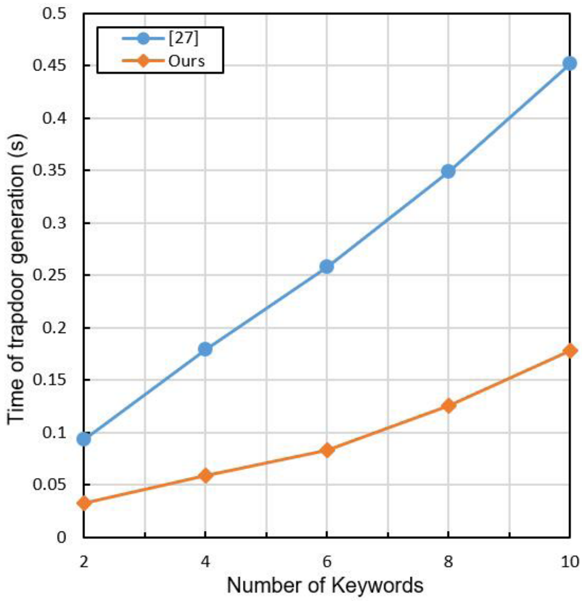


FIGURE 3. Computational cost of the Trapdoor algorithm.

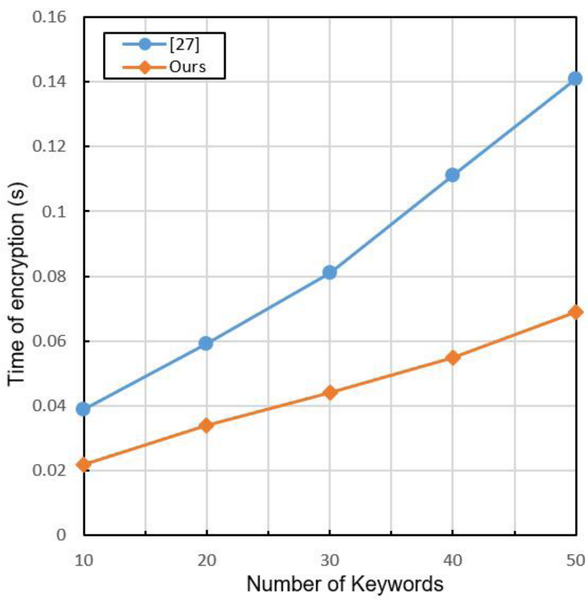


FIGURE 4. Computational cost of the Encryption algorithm.

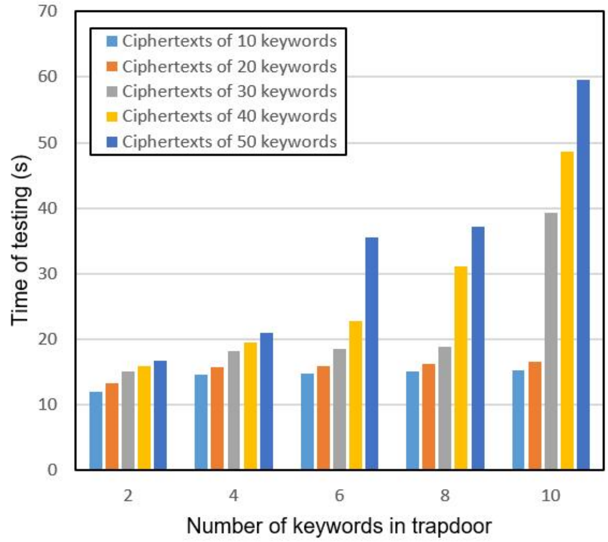


FIGURE 5. Computational cost of the Test algorithm in [27].

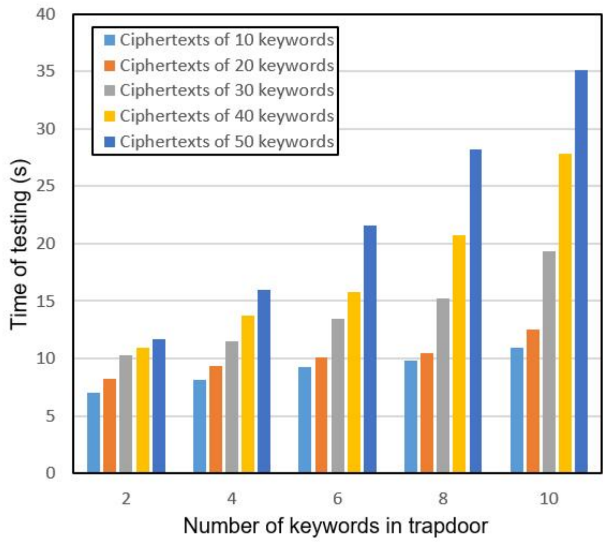


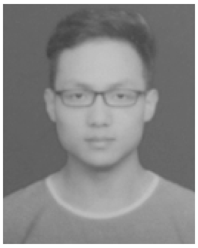
FIGURE 6. Computational cost of the Test algorithm in our scheme.

VI. CONCLUSION AND PROSPECT

In this paper, we propose a new generic construction of EPEKS from anonymous KP-ABE and formally prove its security. An efficient concrete EPEKS scheme over the prime-order groups is given and its performance is analyzed. Yet, the EPEKS proposed in this paper only supports the logical expression of "AND" and "OR", excluding "NOT". And existing schemes that support the logical expression of "AND", "OR" and "NOT" are all based on composite-order groups, hence not quite efficient. Therefore, to propose an efficient EPEKS scheme over the prime-order groups that supports the "AND", "OR" and "NOT" operations of search keywords deserves further research efforts.

REFERENCES

[1] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. EUROCRYPT*, Interlaken, Switzerland, 2004, pp. 506–522.
[2] D. J. Park, K. Kim, and P. J. Lee, "Public key encryption with conjunctive field keyword search," in *Proc. WISA*, Wuhan, China, 2005, pp. 73–86.
[3] J. Lai, X. Zhou, R. H. Deng, Y. Li, and K. Chen, "Expressive search on encrypted data," in *Proc. ASIA CCS*, Hangzhou, China, 2013, pp. 243–252.
[4] Z. Lv, C. Hong, M. Zhang, and D. Feng, "Expressive and secure searchable encryption in the public key setting," in *Proc. ISC*, Hong Kong, 2014, pp. 364–376.
[5] Y. Rouselakis and B. Waters, "New constructions and proof methods for large universe attribute-based encryption," in *Proc. ACM CCS*, Berlin, Germany, 2013, pp. 463–474.
[6] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE Symp. Secur. Privacy*, Berkeley, CA, USA, May 2000, pp. 44–55.
[7] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "Dual-server public-key encryption with keyword search for secure cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 4, pp. 789–798, Apr. 2016, doi: 10.1109/TIFS.2015.2510822.
[8] F.-K. Tseng, R.-J. Chen, and B.-S. P. Lin, "iPEKS: Fast and secure cloud data retrieval from the public-key encryption with keyword search," in *Proc. 12th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Melbourne, VIC, Australia, Jul. 2013, pp. 452–458.
[9] Y. Lu, J. Li, and Y. Zhang, "Secure channel free certificate-based searchable encryption withstanding outside and inside keyword guessing attacks," *IEEE Trans. Services Comput.*, to be published. [Online]. Available: https://ieeexplore.ieee.org/document/8685201, doi: 10.1109/TSC.2019.2910113.
[10] Y. Lu, G. Wang, and J. Li, "Keyword guessing attacks on a public key encryption with keyword search scheme without random oracle and its improvement," *Inf. Sci.*, vol. 479, pp. 270–276, Apr. 2019, doi: 10.1016/j.ins.2018.12.004.
[11] Y. Lu, J. Li, and Y. Zhang, "SCF-PEPCKS: Secure channel free public key encryption with privacy-conserving keyword search," *IEEE Access*, vol. 7, no. 1, pp. 40878–40892, Mar. 2019, doi: 10.1109/ACCESS.2019.2905554.
[12] 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, Sep. 2015, doi: 10.1109/TIFS.2015.2442220.
[13] K. Emura, L. Phong, and Y. Watanabe, "Keyword revocable searchable encryption with trapdoor exposure resistance and re-generateability," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Helsinki, Finland, Aug. 2015, pp. 167–174.
[14] H. S. Rhee and D. H. Lee, "Keyword Updatable PEKS," in *Proc. WISA*, Jeju Island, South Korea, 2016, pp. 96–109.
[15] L. Wu, B. Chen, K.-K. R. Choo, and D. He, "Efficient and secure searchable encryption protocol for cloud-based Internet of Things," *J. Parallel Distrib. Comput.*, vol. 111, pp. 152–161, Jan. 2018, doi: 10.1016/j.jpdc.2017.08.007.
[16] M. Ma, D. He, N. Kumar, K.-K. R. Choo, and J. Chen, "Certificateless searchable public key encryption scheme for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 759–767, Feb. 2018, doi: 10.1109/TII.2017.2703922.
[17] C. Gu, Y. Zhu, and H. Pan, "Efficient public key encryption with keyword search schemes from pairings," in *Proc. INSCRYPT*, Xining, China, 2007, pp. 372–383.
[18] 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, doi: 10.1016/j.jnca.2010.07.007.
[19] J. Baek, R. Safavi-Naini, and W. Susilo, "On the integration of public key data encryption and public key encryption with keyword search," in *Proc. ISC*, Samos Island, Greece, 2006, pp. 217–232.
[20] Q. Tang and L. Q. Chen, "Public key encryption with registered keyword search," in *Proc. EuroPKI*, Pisa, Italy, 2010, pp. 163–178.
[21] P. Golle, J. Staddon, and B. R. Waters, "Secure conjunctive keyword search over encrypted data," in *Proc. ACNS*, Yellow Mountain, China, 2004, pp. 31–45.
[22] Z. Chen, C. Wu, and D. Wang, "Conjunctive keywords searchable encryption with efficient pairing, constant ciphertext and short trapdoor," in *Proc. PAISI*, Kuala Lumpur, Malaysia, 2012, pp. 176–189.
[23] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in *Proc. Pairing*, Tokyo, Japan, 2007, pp. 2–22.
[24] L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient conjunctive keyword searches over encrypted data," in *Proc. ICICS*, Beijing, China, 2005, pp. 414–426.
[25] N. Cao, C. Wang, K. Ren, W. Lou, and M. Li, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," in *Proc. Proc. IEEE INFOCOM*, Shanghai, China, 2011, pp. 829–837.
[26] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Proc. EUROCRYPT*, Riviera, French, 2010, pp. 62–91.
[27] H. Cui, Z. Wan, R. Deng, G. Wang, and Y. Li, "Efficient and expressive keyword search over encrypted data in the cloud," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 3, pp. 409–422, May/Jun. 2018, doi: 10.1109/TDSC.2016.2599883.
[28] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. Adv. Cryptol. (CRYPTO)* (Lecture Notes in Computer Science), vol. 196. Berlin, Germany: Springer, 1984, pp. 47–53.
[29] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. CRYPTO*, Santa Barbara, CA, USA, 2001, pp. 213–229.
[30] A. Sahai and B. Waters, "Fuzzy identity based encryption," in *Proc. EUROCRYPT*, Aarhus, Denmark, 2005, pp. 457–473.
[31] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. ACM CCS*, Alexandria, VA, USA, 2006, pp. 89–98.
[32] A. B. Lewko and B. Waters, "Unbounded HIBE and attribute based encryption," in *Proc. EUROCRYPT*, Tallinn, Estonia, 2011, pp. 547–567.
[33] A. B. Lewko, "Tools for simulating features of composite order bilineat groups in the prime order setting," in *Proc. EUROCRYPT*, Cambridge, U.K., 2012, pp. 318–335.
[34] N. Attrapadung, B. Libert, and E. D. Panafieu, "Expressive key-policy attribute-based encryption with constant-size ciphertexts," in *Proc PKC*, Taormina, Italy, 2011, pp. 90–108.
[35] J. Lai, R. H. Deng, Y. Li, and J. Wang, "Fully secure key-policy attribute-based encryption with constant-size ciphertexts and fast decryption," in *Proc. ACM ASIA CCS*, Kyoto, Japan, 2014, pp. 248–389.
[36] Y. S. Rao and R. Dutta, "Computationally efficient expressive key-policy attribute based encryption schemes with constant-size ciphertext," in *Proc ICICS*, Beijing, China, 2013, pp. 246–362.
[37] J. Kim, W. Susilo, and F. Guo, "An efficient KP-ABE with short ciphertexts in prime order groups under standard assumption," in *Proc. ASIA CCS*, Abu Dhabi, UAE, 2017, pp. 823–834.
[38] C. Wang, W. Li, and L. Yuan, "A ciphertext-policy attribute-based encryption scheme supporting keyword search function," in *Proc. Cyberspace Saf. Secur.*, Zhangjiajie, China, 2013, pp. 377–386.
[39] Q. Zheng, S. Xu, and G. Ateniese, "VABKS: Verifiable attribute-based keyword search over outsourced encrypted data," in *Proc. IEEE INFOCOM*, Toronto, ON, Canada, Apr./May 2014, pp. 522–530.
[40] J. Li, Y. Shi, and Y. Zhang, "Searchable ciphertext-policy attribute-based encryption with revocation in cloud storage," *Int. J. Commun. Syst.*, vol. 30, no. 1, Jan. 2017, Art. no. e2942, doi: 10.1002/dac.2942.
[41] J. Li, X. Lin, Y. Zhang, and J. Han, "KSF-OABE: Outsourced attribute-based encryption with keyword search function for cloud storage," *IEEE Trans. Services Comput.*, vol. 10, no. 5, pp. 715–725, Sep./Oct. 2017, doi: 10.1109/TSC.2016.2542813.
[42] Y. Zhang, X. Chen, and J. Li, "Anonymous attribute-based encryption supporting efficient decryption test," in *Proc. ASIA CCS*, Hangzhou, China, 2013, pp. 511–516.
[43] T. Jung, X.-Y. Li, Z. Wan, and M. Wan, "Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 1, pp. 190–199, Jan. 2015, doi: 10.1109/TIFS.2014.2368352.
[44] B. Lynn. *PBC library: The Pairing-Based Cryptography Library, Version 0.5.14*. Accessed: Jun. 2013. [Online]. Available: http://crypto.stanford.edu/pbc



CHEN SHEN is currently pursuing the master's degree with the College of Computer and Information Engineering, Hohai University. His research interest is cryptography.

YANG LU was born in Yang Zhou, Jiangsu, China, in 1977. He received the B.S. degree in mathematics and the M.S. degree in computer science from Nanjing Normal University, Nanjing, China, in 2000 and 2003, respectively, and the Ph.D. degree in computer science from the PLA University of Science and Technology, Nanjing, China, in 2009.

He is currently a Professor with the College of Computer and Information, Hohai University, Nanjing, China, and the School of Computer Science and Technology, Nanjing Normal University. He has published more than 60 scientific articles in international conferences and journals. His major research interests include information security and cryptography, network security, and cloud security.

JIGUO LI received the B.S. degree in mathematics from Heilongjiang University, Harbin, China, in 1996, the M.S. degree in mathematics and the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, in 2000 and 2003, respectively.

From September 2006 to March 2007, he was a Visiting Scholar with the Centre for Computer and Information Security Research, School of Computer Science and Software Engineering, University of Wollongong, Australia. He was a Professor with the College of Computer and Information, Hohai University, Nanjing, China, from November 2003 to June 2018. From February 2013 to January 2014, he was a Visiting Scholar with the Institute for Cyber Security, University of Texas at San Antonio. He is currently a Professor with the College of Mathematics and Informatics, Fujian Normal University, Fuzhou, China, and the State Key Laboratory of Cryptology, Beijing, China. His research interests include cryptography and information security, cloud computing, wireless security, and trusted computing. He has published more than 150 research articles in refereed international conferences and journals. His work has been cited more than 3000 times at Google Scholar. He has served as a Program Committee Member in more than 20 international conferences and served as a Reviewer in more than 90 international journals and conferences.