Authorized Keyword Search on Encrypted Data

Jie Shi[1,2], Junzuo Lai[2,⋆], Yingjiu Li[1], Robert H. Deng[1], and Jian Weng[2]

[1] Singapore Management University, Singapore
{jieshi,yjli,robertdeng}@smu.edu.sg
[2] Jinan University, China
{laijunzuo,cryptjweng}@gmail.com

Abstract. Cloud computing has drawn much attention from research and industry in recent years. Plenty of enterprises and individuals are outsourcing their data to cloud servers. As those data may contain sensitive information, it should be encrypted before outsourced to cloud servers. In order to ensure that only authorized users can search and further access the encrypted data, two important capabilities must be supported: *keyword search* and *access control*. Recently, rigorous efforts have been made on either keyword search or access control over encrypted data. However, to the best of our knowledge, there is no encryption scheme supporting both capabilities in a public-key scenario so far. In this paper, we propose an authorized searchable public-key encryption scheme supporting expressive search capability and prove it fully secure in the standard model.

Keywords: Authorized Searchable Public-Key Encryption, Attribute-Based Encryption, Public-Key Encryption with Keyword Search ,Public-Key Encryption.

1 Introduction

Recently, as a new commercial model, cloud computing has attracted much attention from both academia and industry. A major advantage of cloud computing is that it supplies virtually unlimited storage capabilities and elastic resource provisioning [1]. In order to reduce the capital and operational expenditures for hardware and software, plenty of IT enterprises and individuals are outsourcing their data to cloud servers instead of building and maintaining their own data centers [2].

Despite clear benefits provided by cloud computing, there are many impediments to its widespread adoption. Data security and privacy concerns are probably the biggest challenges. As outsourced data may contain much sensitive/private information, such as Personal Health Records (PHRs), personal photos and business documents, some cloud servers or unauthorized users are motivated to access and derive such sensitive/private information. Without addressing such concerns, users may hesitate to outsource their data to cloud servers. As it is shown in many

⋆ Corresponding author.

M. Kutyłowski and J. Vaidya (Eds.): ESORICS 2014, Part I, LNCS 8712, pp. 419–435, 2014.

recent works [3,2,4], data encryption is applied on users' data before outsourcing so as to address the security and privacy concerns.

While documents are encrypted and outsourced to cloud servers, two important capabilities should be supported: *keyword search* and *access control*. The keyword search capability facilitates data users to access encrypted data as it enables quick location of required data based on keywords. The access control capability allows data owners to share their information with restricted users according to the access control policies associated with their encrypted data. In the literature, much work has been done on either keyword search or access control over encrypted data. However, no rigorous effort has been dedicated on supporting both keyword search and access control at the same time, which means that only authorized users are allowed to process keyword search and further access encrypted data. We call it *authorized searchable encryption* if an encryption scheme enables authorized users only to perform keyword search. Many real-world applications demand such authorized searchable encryption. One example is the cloud storage system in healthcare as it is shown in Figure 1. In this system, any patient (i.e. data owner) outsources his/her medical records to a cloud server so as to share with authorized users such as hospital doctors. Assuming that the medical records are sensitive, they are encrypted before outsourced to the cloud. The encrypted data should support both keyword search and access control in this scenario. In particular, data owner 1 *John* outsources an encrypted medical record to the cloud with both *keywords* and an *access policy*. The *keywords* specify the features about the encrypted data which can be used in any authorized users' queries, while the *access policy* specifies who are the authorized users (i.e., a cardiologist in Hospital A or a patient with social security number 110-222-1234). Since both *keywords* and *access policy* associated with a medical record contain sensitive/private information, they should be hidden from the cloud service provider or any unauthorized users, just as the medical record itself. Every user in this system is associated with a set of attributes; for example, the attributes of user 1 in Figure 1 include her name, her social security number, her affiliation, and her occupation. When a user intends to obtain certain information from the cloud server, the user submits an authorized token constructed by an authority according to the user's keywords query and the user's attributes. The query token enables the cloud server to locate all medical records such that the *keywords* of the medical records satisfy the user's query and the attributes of the user meet the *access policy* of the medical records.

In this paper, we focus on constructing an *authorized searchable encryption* scheme in a public-key scenario, which we call *authorized searchable public-key encryption (AS-PKE)*. It is challenging to design an AS-PKE scheme supporting both expressive search capability and being fully secure in the standard model. In the literature, there exist two kinds of encryption schemes close to AS-PKE, which are the attribute-based encryption and the public-key encryption with keyword search. First, the attribute-based encryption (ABE) was introduced by Sahai and Waters [5] and further developed into two complimentary forms: KP-ABE [6,7] and CP-ABE [8,9]. There also exist many solutions in ABE with hidden access structures, including predicate encryption [10] and CP-ABE

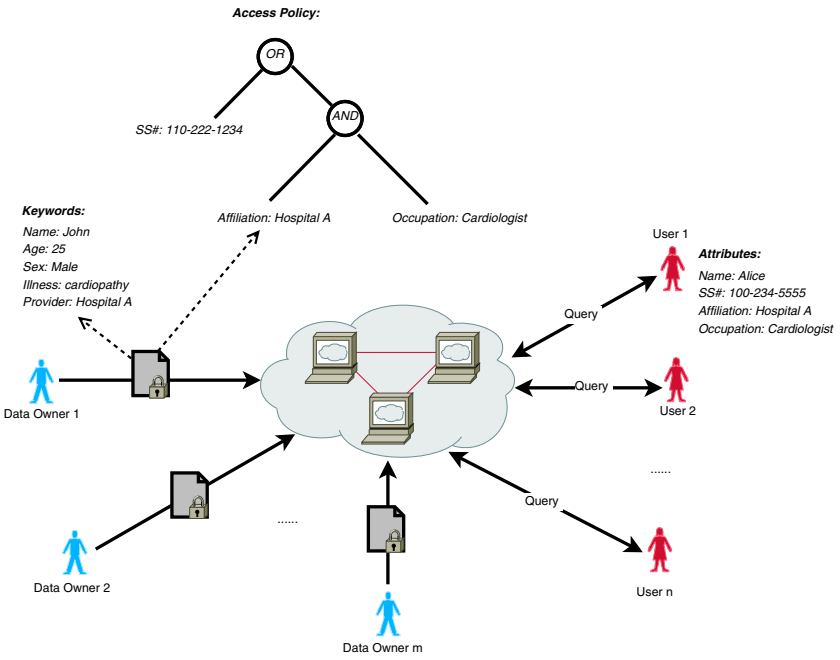

Fig. 1. An example of cloud storage system architecture

with hidden access structures [11]. Second, the public-key encryption with keyword search (PEKS) was proposed by Boneh et al. [12], which supports equality queries only. Later, Park et al. proposed the notion of public key encryption with conjunctive keyword search [13] and Katz et al. proposed the notion of inner-product predicate encryption [10], which can be extended to construct public key encryption with disjunctive keyword search. Neither ABE nor PEKS satisfies the requirements of AS-PKE; in other words, they do not support keyword search and access control at the same time. Simply combining ABE and PEKS schemes cannot achieve AS-PKE too, as in AS-PKE both keywords and access control policies are required to be hidden and expressive search on encrypted data is required to be supported.

1.1 Our Contribution

In [11], Lai et al. proposed a new model of CP-ABE with partially hidden access structures. In this model, each attribute consists of two parts: attribute name and its value. In the access policy associated with a ciphertext, all attribute values are hidden, while the other information, such as attribute names, about the access structure is public. Taking the access policy in Figure 1 as an example, the policy is published in the following format in Lai et al.'s model:

$$\text{SS\#} : \star \text{ OR } (\text{Affiliation}: \star \text{ AND Occupation}:\star)$$

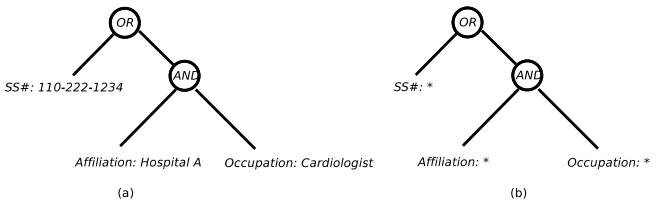

Fig. 2. An access structure (a) and the corresponding partially hidden access structure (b)

Note that all attribute values, such as "110-222-1234", "Hospital A" and "Cardiologist", are hidden. Figure 2 shows graphically this example of a partially hidden access structure.

Based on the CP-ABE scheme with partially hidden access structure given in [11] and the KP-ABE scheme proposed in [7], we design a flexible and expressive construction as an AS-PKE scheme, and prove that it is fully secure in the standard model.

The proposed AS-PKE scheme can be considered as a variant of dual-policy ABE [14] in which the object attributes and the subject access policy are both hidden and the scheme is fully secure in the standard model. In other words, the proposed AS-PKE implies a fully secure dual-policy ABE scheme.

1.2 Related Work

In this section, we briefly review the related works in the areas of ABE, KP-ABE, CP-ABE, PEKS, and PE (Predicate Encryption).

Attribute-Based Encryption (ABE). The concept of ABE was first proposed by Sahai and Waters as an application of *fuzzy identity-based encryption (IBE)* scheme [5], where both ciphertext and secret key are labeled with sets of descriptive attributes. The decryption of a ciphertext is enabled if and only if the cardinality of the intersection of these labeled attributes exceeds a certain threshold.

Key Policy Attribute-Based Encryption (KP-ABE). Two complimentary forms of ABE — KP-ABE and CP-ABE — were formulated by Goyal et al. [6]. In a CP-ABE scheme, each ciphertext is associated with an access structure while each decryption key is associated with a set of attributes. Reversely, in a KP-ABE scheme, each decryption key is associated with an access structure while each ciphertext is associated with a set of attributes. Generally, a KP-ABE scheme can be transformed into a CP-ABE using the method proposed in [15]. While the KP-ABE scheme proposed by Goyal et al. [6] supports monotonic access structures only, Ostrovsky et al. [16] presented a KP-ABE system supporting more flexible access control policies — non-monotone access structures.

Ciphertext Policy Attribute-Based Encryption (CP-ABE). Bethencourt et al. proposed the first CP-ABE scheme [8], which was proven to be secure under the

generic group mode. Later, Cheung and Newport presented a CP-ABE scheme that is secure under the standard model [17]. However, the access structures in this scheme are restricted to conjunctions of different attributes. Recently, secure and expressive CP-ABE schemes were proposed in [9,7]. In order to hide access structures, Nishide et al. introduced the concept of CP-ABE with partially hidden access structures [18]. Recently, Lai et al. proposed a fully secure (cf. selectively secure) CP-ABE scheme with partially hidden access structures [19]; however, the scheme only supports restricted access structure as in [18]. Later, Lai et al. proposed a fully secure CP-ABE scheme with partially hidden access structures [11] that can be expressed as an LSSS which is more flexible and expressive than the previous work [18].

Predicate Encryption (PE). Predicate encryption can be considered as attribute-based encryption supporting attribute-hiding. Katz et al. introduced the concept of PE and designed the first inner-product PE [10]. Shi and Waters presented a delegation mechanism for a class of PE [20]; later, Okamota and Takashima presented a (hierarchical) delegation mechanism for an inner-product PE scheme [21]. Shen et al. introduced a new security notion of PE called predicate privacy and also proposed a symmetric-key inner-product PE, which achieves both plaintext privacy and predicate privacy [22]. However, these schemes were proven to be selectively secure only. The first fully secure inner-product PE was proposed by Lweko et al. [7]. Okamota and Takashima presented a fully secure PE for a wide class of admissible predicates, which are specified using non-monotone access structures combined with inner-product predicates [23].

Public-key Encryption with Keyword Search (PEKS). Boneh et al. initiated the research on PEKS and provided a specific scheme, which supports equality query only [12]. Park et al. proposed the notion of public key encryption with conjunctive keyword search [13]; Hwan and Lee made an improvement on the sizes of ciphertext and private key, and extended the scheme in a multi-user setting [24]. Boneh and Waters presented a general framework for analyzing and constructing several schemes that support arbitrary conjunctions [25]. Katz et al. proposed the notion of inner-product predicate encryption (IPE), which can be extended to construct public key encryption with disjunctive keyword search [10]. However, as shown in [10], the resulting solution suffers from a super polynomial blowup in ciphertext size and search-token key size.

Others. Recently, Li et al. [2] presented a framework for authorized private keyword search (APKS) over encrypted cloud data and proposed two schemes for APKS. In their proposed framework, every data owner's trust is delegated to a trusted authority and/or several local trusted authorities who are in charge of determining users' search privileges. Based on this framework, they employed the hierarchical predicate encryption to construct APKS. However, there exists a significant difference between the APKS and our AS-PKE: the access control policies are defined and maintained by trusted authorities in APKS scheme; however, in our AS-PKE scheme, the access control policies are defined by data owners themselves. Therefore, our AS-PKE scheme is more general and can be

used in many applications which require access control policies to be defined by data owners. In [26], Sun et al. proposed an attribute-based keyword search with fine-grained owner-enforced search authorization scheme, which supports limited authorization policies with "AND" gates and limited keyword queries with conjunctive keywords only. Our AS-PKE scheme supports more expressive authorization policies and keyword queries supporting arbitrary Boolean formulas. In [27], Narayan et al. combined PEKS and ABE to create a secure electronic health record system providing both keyword search and access control functionalities; however, it does not address the privacy of access control policies as in our work.

1.3 Organization

The rest of the paper is organized as follows. In Section 2, we review necessary standard notations and cryptographic definitions. In Section 3, we define the security model of AS-PKE, and propose a concrete construction of AS-PKE. In Section 4, we conclude our paper.

2 Preliminaries

In this paper, we use $s \xleftarrow{\$} S$ to denote the operation of picking an element s uniformly at random from a set S. Let $\mathbb{N}$ be the set of natural numbers, and 1^λ denote the string of λ ones if $\lambda \in \mathbb{N}$. Let $z \leftarrow \mathsf{A}(x, y, \ldots)$ denote the operation of running an algorithm A with inputs $(x, y, \ldots)$ and output z. A function $f(\lambda)$ is *negligible* if for every $c > 0$ there exists a λ_c such that $f(\lambda) < 1/\lambda^c$ for all $\lambda > \lambda_c$.

2.1 Access Structures

Definition 1 (Access Structure [28]). *Let $\{P_1, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,\ldots,P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) $\mathbb{A}$ of non-empty subsets of $\{P_1, \ldots, P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1,\ldots,P_n\}} \backslash \{\emptyset\}$. The sets in $\mathbb{A}$ are called authorized sets, and the sets not in $\mathbb{A}$ are called unauthorized sets.*

In our context, attributes play the role of parties. We focus on the monotone access structures in this paper. However, it is possible to (inefficiently) realize general access structures using the proposed technique by taking the negation of an attribute as a separate attribute. In what follows, unless stated otherwise, the access structures are monotone access structures.

2.2 Linear Secret Sharing Schemes

We will make use of linear secret sharing schemes in our design of AS-PKE. The following definition is adapted from those given in [28].

Definition 2. *[Linear Secret-Sharing Schemes (LSSS)] A secret sharing scheme* Π *over a set of parties* $\mathcal{P}$ *is called linear (over* $\mathbb{Z}_p$*) if*

1. *The shares for each party form a vector over* $\mathbb{Z}_p$.
2. *There exists a matrix* $\mathbf{A}$ *with* ℓ *rows and* n *columns called the share-generating matrix for* Π. *For all* $i = 1, \ldots, \ell$, *the* i^{th} *row of* $\mathbf{A}$ *is labeled by a party* $\rho(i)$ *(*ρ *is a function from* $\{1, \ldots, \ell\}$ *to* $\mathcal{P}$*). When we consider the column vector* $v = (s, r_2, \ldots, r_n)$, *where* $s \in \mathbb{Z}_p$ *is the secret to be shared, and* $r_2, \ldots, r_n \in \mathbb{Z}_p$ *are randomly chosen, then* $\mathbf{A}v$ *is the vector of* ℓ *shares of the secret* s *according to* Π. *The share* $(\mathbf{A}v)_i$ *belongs to party* $\rho(i)$.

It is shown in [28] that every linear secret-sharing scheme according to the above definition enjoys the linear reconstruction property, defined as follows. Suppose that Π is an LSSS for an access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be any authorized set, and $I \subset \{1, \ldots, \ell\}$ be defined as $I = \{i | \rho(i) \in S\}$. Then there exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that, if $\{\lambda_i\}$ are valid shares of any secret s according to Π, then $\sum_{i \in I} \omega_i \lambda_i = s$. Let A_i denote the i^{th} row of $\mathbf{A}$, we have $\sum_{i \in I} \omega_i A_i = (1, 0, \ldots, 0)$. These constants $\{\omega_i\}$ can be found in time polynomial in the size of the share-generation matrix $\mathbf{A}$ [28]. Note that, for unauthorized sets, no such constants $\{\omega_i\}$ exist.

Boolean Formulas. Access structures might also be described in terms of monotonic boolean formulas. Using standard techniques [28] one can convert any monotonic boolean formula into an LSSS representation. When a boolean formula is represented as an access tree with ℓ leaf nodes, it will result in an LSSS matrix of ℓ rows. Details on how to perform this conversion refer to the appendix of [29].

2.3 Composite Order Bilinear Groups

We construct our scheme in composite order bilinear groups whose order is the product of four distinct primes. Composite order bilinear groups were first introduced in [30].

Let $\mathcal{G}$ be a group generator, an algorithm taking a security parameter 1^λ as input and outputting a tuple $(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e)$, where p_1, p_2, p_3, p_4 are distinct primes, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $N = p_1 p_2 p_3 p_4$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a map such that

1. Bilinear: For all $g, h \in \mathbb{G}$, and $a, b \in \mathbb{Z}_N$, we have $e(g^a, h^b) = e(g, h)^{ab}$;
2. Non-degeneracy: $\exists g \in \mathbb{G}$ such that $e(g, g)$ has order N in $\mathbb{G}_T$.

It further requires that the group operation in $\mathbb{G}$ and $\mathbb{G}_T$ and the bilinear map e are both efficiently computable in time polynomial in λ. Let $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}$, and $\mathbb{G}_{p_4}$ be the subgroups of $\mathbb{G}$ having order p_1, p_2, p_3, and p_4 respectively. Thus, $\mathbb{G} = \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3} \times \mathbb{G}_{p_4}$. Note that if $g_1 \in \mathbb{G}_{p_1}$ and $g_2 \in \mathbb{G}_{p_2}$, then $e(g_1, g_2) = 1$. Similar rules hold whenever e is applied to elements in distinct subgroups.

We adopt the following four complexity assumptions in this paper, which were also used in [11,31].

Assumption 1. *Given a group generator $\mathcal{G}$, we define the following distribution:*

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda),\ N = p_1p_2p_3p_4,$$
$$g \xleftarrow{\$} \mathbb{G}_{p_1},\ X_3 \xleftarrow{\$} \mathbb{G}_{p_3},\ X_4 \xleftarrow{\$} \mathbb{G}_{p_4},$$
$$D = (\mathbb{G}, \mathbb{G}_T, N, e, g, X_3, X_4),$$
$$T_1 \xleftarrow{\$} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2},\ T_2 \xleftarrow{\$} \mathbb{G}_{p_1}.$$

The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 1 is defined as

$$Adv^1_{\mathcal{A}} = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 3. *we say $\mathcal{G}$ satisfies Assumption 1 if for any polynomial time algorithm $\mathcal{A}$, $Adv^1_{\mathcal{A}}$ is negligible.*

Assumption 2. *Given a group generator $\mathcal{G}$, we define the following distribution:*

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda),\ N = p_1p_2p_3p_4,$$
$$g, X_1 \xleftarrow{\$} \mathbb{G}_{p_1},\ X_2, Y_2 \xleftarrow{\$} \mathbb{G}_{p_2},\ X_3, Y_3 \xleftarrow{\$} \mathbb{G}_{p_3},\ X_4 \xleftarrow{\$} \mathbb{G}_{p_4},$$
$$D = (\mathbb{G}, \mathbb{G}_T, N, e, g, X_1X_2, Y_2Y_3, X_3, X_4),$$
$$T_1 \xleftarrow{\$} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3},\ T_2 \xleftarrow{\$} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}.$$

The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 2 is defined as

$$Adv^2_{\mathcal{A}} = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 4. *we say $\mathcal{G}$ satisfies Assumption 2 if for any polynomial time algorithm $\mathcal{A}$, $Adv^2_{\mathcal{A}}$ is negligible.*

Assumption 3. *Given a group generator $\mathcal{G}$, we define the following distribution:*

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{\$} \mathcal{G}(1^\lambda),\ N = p_1p_2p_3p_4,$$
$$s \xleftarrow{\$} \mathbb{Z}_N,\ g, h \xleftarrow{\$} \mathbb{G}_{p_1},\ g_2, X_2, B_2, D_2 \xleftarrow{\$} \mathbb{G}_{p_2},$$
$$X_3 \xleftarrow{\$} \mathbb{G}_{p_3},\ B_4, D_4, X_4, Z' \xleftarrow{\$} \mathbb{G}_{p_4},$$
$$D = (\mathbb{G}, \mathbb{G}_T, N, e, g, g_2, hX_2, hZ', g^sB_2B_4, X_3, X_4),$$
$$T_1 = h^sD_2D_4, T_2 \xleftarrow{\$} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}.$$

The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 3 is defined as

$$Adv^3_{\mathcal{A}} = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 5. *we say $\mathcal{G}$ satisfies Assumption 3 if for any polynomial time algorithm $\mathcal{A}$, $Adv^3_{\mathcal{A}}$ is negligible.*

Assumption 4. *Given a group generator $\mathcal{G}$, we define the following distribution:*

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda),\ N = p_1p_2p_3p_4,$$

$$a, s \xleftarrow{\$} \mathbb{Z}_N,\ g \xleftarrow{\$} \mathbb{G}_{P_1},\ g_2, X_2, Y_2, D_2 \xleftarrow{\$} \mathbb{G}_{p_2}$$

$$X_3 \xleftarrow{\$} \mathbb{G}_{p_3},\ X_4, Z', Y_4, D_4 \xleftarrow{\$} \mathbb{G}_{p_4}$$

$$D = (\mathbb{G}, \mathbb{G}_T, N, e, g, g_2, g^a X_2, g^a Z', g^s Y_2 Y_4, X_3, X_4),$$

$$T_1 = g^{as} D_2 D_4,\ T_2 \xleftarrow{\$} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}.$$

The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 4 is defined as

$$Adv^4_{\mathcal{A}} = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 6. *we say $\mathcal{G}$ satisfies Assumption 4 if for any polynomial time algorithm $\mathcal{A}$, $Adv^4_{\mathcal{A}}$ is negligible.*

3 Authorized Searchable Public Key Encryption

In authorized searchable public key encryption (AS-PKE), a document is identified by a vector of m keywords $(o_1, \ldots, o_m)$, where o_x is the keyword of the document in the x-th keyword field. For notational purpose, let x be the x-th keyword field. Similarly, a user has n attributes $(s_1, \ldots, s_n)$ with each attribute belonging to a different category. Let i be the attribute name of the i-th category attribute. Our AS-PKE scheme supports arbitrary monotone boolean predicate for both access policy and user query. We express an access policy by an LSSS $(A, \rho, \mathcal{T})$ over user attributes, where A is an $l_s \times n$ matrix, ρ is a map from each row of A to an attribute field (i.e., ρ is a function from $\{1, \ldots, l_s\}$ to $\{1, \ldots, n\}$), $\mathcal{T}$ can be parsed into $(t_{\rho(1)}, \ldots, t_{\rho(l_s)})$ and $t_{\rho(i)}$ is the value of attribute field $\rho(i)$. Similarly, we express a user query by an LSSS $(\hat{A}, \hat{\rho}, \hat{\mathcal{T}})$ over document keywords, where $\hat{A}$ is an $l_o \times m$ matrix, $\hat{\rho}$ is a map from each row of $\hat{A}$ to a keyword field (i.e., $\hat{\rho}$ is a function from $\{1, \ldots, l_o\}$ to $\{1, \ldots, m\}$), $\hat{\mathcal{T}}$ can be parsed into $(\hat{t}_{\hat{\rho}(1)}, \ldots, \hat{t}_{\hat{\rho}(l_o)})$ and $\hat{t}_{\hat{\rho}(x)}$ is the value of keyword field $\hat{\rho}(x)$.

Before presenting our AS-PKE scheme, we give some intuitions of our construction. Suppose that a document is encrypted with a set of keywords $\mathbb{O} = (o_1, \ldots, o_m)$ and an access policy $(A, \rho, \mathcal{T})$, a query token key $TK_{\mathcal{P},\mathbb{S}}$ is embedded with a set of user attributes $\mathbb{S} = (s_1, \ldots, s_n)$ and a user query $\mathcal{P} = (\hat{A}, \hat{\rho}, \hat{\mathcal{T}})$. The encrypted document D will be returned if and only if there exist $\mathcal{I} \subseteq \{1, \ldots, l_s\}$, $\hat{\mathcal{I}} \subseteq \{1, \ldots, l_o\}$ and constants $\{w_i\}_{i \in \mathcal{I}}$, $\{\hat{w}_x\}_{x \in \hat{\mathcal{I}}}$ such that

$$\sum_{i \in \mathcal{I}} w_i A_i = (1, 0, \ldots, 0) \text{ and } s_{\rho(i)} = t_{\rho(i)} \text{ for } \forall i \in \mathcal{I},$$

$$\sum_{x \in \hat{\mathcal{I}}} \hat{w}_x \hat{A}_x = (1, 0, \ldots, 0) \text{ and } o_{\hat{\rho}(x)} = \hat{t}_{\hat{\rho}(x)} \text{ for } \forall x \in \hat{\mathcal{I}},$$

where A_i and $\hat{A}_x$ denote the i-th row of A and the x-th row of $\hat{A}$, respectively. We also say that $\mathcal{I} \subseteq \{1, \ldots, l_s\}$ satisfies $(A, \rho, \mathcal{T})$ if there exist constants $\{w_i\}_{i \in \mathcal{I}}$ such that $\sum_{i \in \mathcal{I}} w_i A_i = (1, 0, \ldots, 0)$. This can be applied to $\hat{\mathcal{I}} \subseteq \{1, \ldots, l_o\}$ and $(\hat{A}, \hat{\rho}, \hat{\mathcal{T}})$.

We define $I_{A,\rho}$ and $\hat{I}_{\hat{A},\hat{\rho}}$ as the set of minimum subsets of $\{1, \ldots, l_s\}$ and $\{1, \ldots, l_o\}$ that satisfy $(A, \rho, \mathcal{T})$ and $(\hat{A}, \hat{\rho}, \hat{\mathcal{T}})$, respectively.

3.1 Authorized Searchable Public Key Encryption

In AS-PKE scheme, keywords $\mathbb{O} = (o_1, o_2, \ldots, o_n)$ of a document are encrypted under an access policy $\mathbb{A}$ and can be searched by an authorized query token. An authorized query token is generated by authority according to a query and user attributes set. An authorized searchable public key encryption (AS-PKE) scheme consists of the following four algorithms:

$\mathsf{Setup}(1^\lambda)$. This setup algorithm takes in the security parameter λ with output of the public parameters PK and a secret key SK.

$\mathsf{Encrypt}(\mathsf{PK}, \mathbb{O} = (o_1, \ldots, o_m), \mathbb{A} = (A, \rho, \mathcal{T}))$. This encryption algorithm takes in the public parameter PK, keywords $\mathbb{O} = (o_1, \ldots, o_m)$, and an access policy $\mathbb{A} = (A, \rho, \mathcal{T})$. It outputs a ciphertext $C_{\mathbb{O},\mathbb{A}}$.

$\mathsf{GenToken}(\mathsf{PK}, \mathsf{SK}, \mathbb{P}, \mathbb{S} = (s_1, \ldots, s_n))$. This algorithm takes in the public key PK, the secret key SK, a user attributes set $\mathbb{S} = (s_1, \ldots, s_n)$ and a query predicate $\mathbb{P}$. It outputs an authorized query token key $\mathsf{TK}_{\mathbb{P},\mathbb{S}}$.

$\mathsf{Test}(\mathsf{PK}, \mathsf{TK}_{\mathbb{P},\mathbb{S}}, C_{\mathbb{O},\mathbb{A}})$. This test algorithm takes in the public key PK, an authorized query token $\mathsf{TK}_{\mathbb{P},\mathbb{S}} = \mathsf{GenToken}(\mathsf{PK}, \mathsf{SK}, \mathbb{P}, \mathbb{S})$ and a ciphtertext $C_{\mathbb{O},\mathbb{A}} = \mathsf{Encrypt}(\mathsf{PK}, \mathbb{O}, \mathbb{A})$. It outputs "Yes" if the keywords in $\mathbb{O}$ satisfy the predicate $\mathbb{P}$ (i.e., $\mathbb{P}(\mathbb{O}) = 1$) and the user attributes in set $\mathbb{S}$ satisfy the access policy $\mathbb{A}$ (i.e. $\mathbb{A}(\mathbb{S}) = 1$); and outputs "No" otherwise.

Correctness. The system must satisfy the following **correctness property**:

- Let $(\mathsf{PK}, \mathsf{SK}) \leftarrow \mathsf{Setup}(1^\lambda)$, $C_{\mathbb{O},\mathbb{A}} \leftarrow \mathsf{Encrypt}(\mathsf{PK}, \mathbb{O}, \mathbb{A})$, $\mathsf{TK}_{\mathbb{P},\mathbb{S}} \leftarrow \mathsf{GenToke}(\mathsf{PK}, \mathsf{SK}, \mathbb{P}, \mathbb{S})$. If $\mathbb{P}(\mathbb{O}) = 1$ and $\mathbb{A}(\mathbb{S}) = 1$, then $\mathsf{Test}(\mathsf{PK}, \mathsf{TK}_{\mathbb{P},\mathbb{S}}, C_{\mathbb{O},\mathbb{A}}) =$ "Yes"; Otherwise, $\Pr[\mathsf{Test}(\mathsf{PK}, \mathsf{TK}_{\mathbb{P},\mathbb{S}}, C_{\mathbb{O},\mathbb{A}}) =$ "No" $] > 1 - \epsilon(\lambda)$ where $\epsilon(\lambda)$ is a negligible function.

3.2 Security Model for AS-PKE

We define a security model for AS-PKE in the sense of semantic-security using the following game between a challenger and an attacker.

Setup. The challenger runs $\mathsf{Setup}(1^\lambda)$ to obtain a public PK and a secret key SK. It gives the public key PK to the adversary and keeps SK by itself.

Query phase 1. The adversary $\mathcal{A}$ adaptively queries the challenger for token keys for pairs of user attributes set and predicate $(\mathbb{S}, \mathbb{P})$. In response, the challenger runs $\mathsf{TK}_{\mathbb{P}_i,\mathbb{S}_i} \leftarrow \mathsf{GenToken}(\mathsf{PK}, \mathsf{SK}, \mathbb{P}_i, \mathbb{S}_i)$ and gives the authorized query token $\mathsf{TK}_{\mathbb{P}_i,\mathbb{S}_i}$ to $\mathcal{A}$, for $1 \leq i \leq q$.

Challenge. The adversary $\mathcal{A}$ submits two pairs of keywords and access policy $(\mathbb{O}_0, \mathbb{A}_0 = (A, \rho, \mathcal{T}_0))$, $(\mathbb{O}_1, \mathbb{A}_1 = (A, \rho, \mathcal{T}_1))$ subject to the restriction that, for any previous query $(\mathbb{P}_i, \mathbb{S}_i)$ in phase 1, either $\mathbb{O}_j$ does not satisfy $\mathbb{P}_i$ or $\mathbb{S}_i$ does not satisfy $\mathbb{A}_j$ for all $j \in [0, 1]$. The challenger selects a random bit $\beta \in \{0, 1\}$, sets $C_{\mathbb{O}_\beta, \mathbb{A}_\beta} = \mathsf{Encrypt}(\mathsf{PK}, \mathbb{O}_\beta, \mathbb{A}_\beta)$, and sends $C_{\mathbb{O}_\beta, \mathbb{A}_\beta}$ to the adversary as its challenge ciphertext.
Note that, the LSSS matrix A and ρ are the same in the two access structures provided by the adversary. In an AS-PKE scheme, one can distinguish the ciphertexts if the associated access structures have different (A, ρ), since (A, ρ) is sent along with the encrypted document explicitly.

Query phase 2. The adversary continues to adaptively query the challenger for token keys corresponding to predicates and user attribute sets with the same restriction in **Challenge** phrase.

Guess. The adversary $\mathcal{A}$ outputs its guess $\beta' \in \{0, 1\}$ for β and wins the game if $\beta' = \beta$.

The advantage of the adversary in this game is defined as $|\mathsf{Pr}[\beta = \beta'] - \frac{1}{2}|$ where the probability is taken over the random bits used by the challenger and the adversary.

Definition 7. *An AS-PKE scheme is secure if all polynomial time adversaries have at most a negligible advantage in this security game.*

3.3 Constructions

Our construction of a secure AS-PKE scheme is shown as follows.

$\mathsf{Setup}(1^\lambda)$. The setup algorithm first runs $\mathcal{G}(1^\lambda)$ to obtain $(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e)$ with $\mathbb{G} = \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3} \times \mathbb{G}_{p_4}$, where $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $N = p_1 p_2 p_3 p_4$. Then, it chooses random elements $g, u, h_1, \ldots, h_n, \hat{h}_1, \ldots, \hat{h}_m \in \mathbb{G}_{p_1}$, $X_3 \in \mathbb{G}_{p_3}$, $X_4, Z, Z', Z_0, Z_1, \ldots, Z_n, \hat{Z}_1, \ldots, \hat{Z}_m \in \mathbb{G}_{p_4}$ and random number $a \in \mathbb{Z}_N$. The public key is published as $\mathsf{PK} = (N, gZ, g^a Z', U = uZ_0, \{H_i = h_i \cdot Z_i\}_{1 \le i \le n}, \{\hat{H}_i = \hat{h}_i \cdot \hat{Z}_i\}_{1 \le i \le m}, X_4)$. The secret key is $\mathsf{SK} = (g, u, h_1, \ldots, h_n, \hat{h}_1, \ldots, \hat{h}_m, X_3, a)$.

$\mathsf{Encrypt}(\mathsf{PK}, \mathbb{O} = (o_1, \ldots, o_m) \in \mathbb{Z}_N^m, \mathbb{A} = (A, \rho, \mathcal{T}))$. A is an $l_s \times n$ matrix, ρ is a map from each row A_i of A to a user attribute $\rho(i)$, and $\mathcal{T} = (t_{\rho(1)}, \ldots, t_{\rho(l_s)})$. The encryption algorithm chooses a random vector $v = (s, v_2, \ldots, v_n) \in \mathbb{Z}_N^n$. For each row A_i of A, it chooses a random $r_i \in \mathbb{Z}_N$. It also chooses random elements $\tilde{Z}_{1,0}, \{\tilde{Z}_{1,i}\}_{1 \le i \le n}, \{\tilde{Z}'_{1,i}\}_{1 \le i \le n} \in \mathbb{G}_{p_4}$, $\{Z_{2,x}\}_{1 \le x \le m} \in \mathbb{G}_{p_4}$. The ciphertext $\mathsf{CT} = ((A, \rho), C, C_i, D_i, \hat{C}_x)$ is computed as:

$$C = (gZ)^s \cdot \tilde{Z}_{1,0} = g^s \cdot Z_{1,0},$$

$$C_i = (g^a Z')^{A_i \cdot v} (U^{t_{\rho(i)}} H_{\rho(i)})^{r_i} \cdot \tilde{Z}_{1,i} = g^{a A_i \cdot v} (U^{t_{\rho(i)}} H_{\rho(i)})^{r_i} \cdot Z_{1,i},$$

$$D_i = (gZ)^{-r_i} \cdot \tilde{Z}'_{1,i} = g^{-r_i} \cdot Z'_{1,i}, \quad \hat{C}_x = (U^{o_x} \cdot \hat{H}_x)^s \cdot Z_{2,x} \ \forall x,$$

where $Z_{1,0} = Z^s \cdot \tilde{Z}_{1,0}, Z_{1,i} = Z'^{A_i \cdot v} \cdot \tilde{Z}_{1,i}, Z'_{1,i} = Z^{-r_i} \cdot \tilde{Z}'_{1,i}$.

$\mathsf{GenToken}(\mathsf{PK}, \mathsf{SK}, \hat{\mathcal{P}} = (\hat{A}, \hat{\rho}, \hat{\mathcal{T}}), \mathbb{S} = (s_1, \ldots, s_n))$. $\hat{A}$ is an $l_o \times m$ matrix, $\hat{\rho}$ is a map from each row $\hat{A}_x$ of $\hat{A}$ to a keyword field $\hat{\rho}(x)$, and $\hat{\mathcal{T}} = (\hat{t}_{\hat{\rho}(1)}, \ldots, \hat{t}_{\hat{\rho}(l_o)})$. The algorithm first chooses two random numbers t_1, $t_2 \in \mathbb{Z}_N$ and a random vector $\hat{v} = (t_2, \hat{v}_2, ..., \hat{v}_m) \in \mathbb{Z}_N^m$. It also chooses random elements $R_0, R_0', R_x', R_i, \hat{R}_x \in G_{p_3}$. The authorized query token key $\mathsf{TK}_{\hat{\mathcal{P}},\mathcal{S}} = ((\hat{A}, \hat{\rho}), K, L, K_i, \hat{K}_x, K_x')$ is computed as:

$$K = g^{a(t_1+t_2)} R_0, \quad L = g^{t_1} R_0', \quad K_i = (u^{s_i} h_i)^{t_1} R_i$$

$$\hat{K}_x = g^{a\hat{A}_x \cdot \hat{v}} (u^{\hat{t}_{\hat{\rho}(x)}} \hat{h}_{\hat{\rho}(x)})^{t_x} \hat{R}_x \ \forall x, \quad K_x' = g^{-t_x} R_x' \ \forall x$$

$\mathsf{Test}(\mathsf{PK}, \mathsf{TK}_{\hat{\mathcal{P}},\mathbb{S}}, \mathsf{CT})$. Let $\mathsf{CT} = ((A, \rho), C, C_i, D_i, \hat{C}_x)$ and $\mathsf{TK}_{\hat{\mathcal{P}},\mathcal{S}} = ((\hat{A}, \hat{\rho}), K, L, K_i, \hat{K}_x, K_x')$. The test algorithm first calculates $I_{A,\rho}$ from (A, ρ), where $I_{A,\rho}$ denotes the set of minimum subsets of $(1, \ldots, l_s)$ that satisfies $I_{A,\rho}$. It similarly calculates $\hat{I}_{\hat{A},\hat{\rho}}$ from $(\hat{A}, \hat{\rho})$. Then, it checks if there exist an $I \in I_{A,\rho}$ and an $\hat{I} \in \hat{I}_{\hat{A},\hat{\rho}}$ that satisfies

$$e(C, K) = \prod_{i \in I} (e(C_i, L) e(K_i, D_i))^{\omega_i} \cdot \prod_{x \in \hat{I}} (e(\hat{K}_x, C) e(\hat{C}_x, K_x'))^{\hat{\omega}_x} \tag{1}$$

where $\sum_{i \in I} \omega_i A_i = (1, 0, \ldots, 0)$ and $\sum_{x \in \hat{I}} \hat{\omega}_x \hat{A}_x = (1, 0, \ldots, 0)$. If no elements in $I_{A,\rho}$ and $\hat{I}_{\hat{A},\hat{\rho}}$ satisfy the above equation, it outputs "No"; otherwise, it outputs "Yes".

The **correctness** is shown as follows. Suppose $\mathbb{P}(\mathbb{O}) = 1$ and $\mathbb{A}(\mathbb{S}) = 1$, i.e. there exist $\mathcal{I} \subseteq \{1, \ldots, l_s\}$, $\hat{\mathcal{I}} \subseteq \{1, \ldots, l_o\}$ and constants $\{w_i\}_{i \in \mathcal{I}}$, $\{\hat{w}_x\}_{x \in \hat{\mathcal{I}}}$ such that $\sum_{i \in \mathcal{I}} w_i A_i = (1, 0, \ldots, 0)$ and $s_{\rho(i)} = t_{\rho(i)}$ for $\forall i \in \mathcal{I}$, $\sum_{x \in \hat{\mathcal{I}}} \hat{w}_x \hat{A}_x = (1, 0, \ldots, 0)$ and $o_{\hat{\rho}(x)} = \hat{t}_{\hat{\rho}(x)}$ for $\forall x \in \hat{\mathcal{I}}$. Then, the left side of Equation (1) is equal to

$$e(C, K) = e(g^s \cdot Z_{1,0}, g^{a(t_1+t_2)} R_0) = e(g, g)^{as(t_1+t_2)}$$

and the right side of Equation (1) is equal to

$$\prod_{i \in I} (e(C_i, L) e(K_i, D_i))^{\omega_i} \cdot \prod_{x \in \hat{I}} (e(\hat{K}_x, C) e(\hat{C}_x, K_x'))^{\hat{\omega}_x}$$

$$= \prod_{i \in I} (e(g^{aA_i \cdot v (U^{t_{\rho(i)}} H_{\rho(i)})^{r_i}} \cdot Z_{1,i}, g^{t_1} R_0') \cdot e((u_{s_i} h_i)^{t_i} R_i, g^{-r_i} Z_{1,i}'))^{w_i}$$

$$\cdot \prod_{x \in \hat{I}} (e(g^{a\hat{A}_x \cdot \hat{v}} (u^{\hat{(t)}_{\hat{\rho}(x)}} \hat{h}_{\hat{\rho}(x)})^{t_x} \hat{R}_x, g^s Z_{1,0}) \cdot e((U^{o_x} \hat{H}_x)^s \cdot Z_{2,x}, g^{-t_x} R_x'))^{\hat{w}_x}$$

$$= \prod_{i \in I} (e(g^{aA_i \cdot v (u^{t_{\rho i}} \cdot h_i)^{r_i}, g^{t_1}}) \cdot e((u^{s_i} h_i)^{t_1}, g^{-r_i}))^{w_i}$$

$$\cdot \prod_{x \in \hat{I}} (e(g^{a\hat{A}_x \cdot \hat{v}} (u^{\hat{t}_{\hat{\rho}(x)}} \hat{h}_{\hat{\rho}(x)})^{t_x}, g^s) \cdot e((u^{o_x}, \hat{h}_x)^s, g^{-t_x}))^{\hat{w}_x}$$

$$= e(g, g)^{at_1 s} \cdot e(g, g)^{at_2 s} = e(g, g)^{as(t_1+t_2)}$$

which is equal to the left side of Equation (1).

3.4 Security

Theorem 1. *If assumptions 1, 2, 3 and 4 hold, then the proposed AS-PKE scheme is secure.*

Proof. Following the approach by Lewko and Waters [7], we define two additional structures: *semi-functional* ciphertexts and *semi-functional* keys. They are not used in the real system, only in our proof.

Semi-functional Ciphertext. Let g_2 denote a generator of the subgroup $\mathbb{G}_{p_2}$. A semi-functional ciphertext is created as follows. We first use the encryption algorithm to form a normal ciphertext $\mathsf{CT}' = ((A, \rho), C', C'_i, D'_i, \hat{C}'_x)$. Then, we choose random exponent $c, b' \in \mathbb{Z}_N$ and random values $z_i \in \mathbb{Z}_N$ associated to user attributes, random values $\gamma_i \in \mathbb{Z}_N$ associated to rows i of matrix A, random values $z'_x \in \mathbb{Z}_N$ associated to keywords and a random vector $w \in \mathbb{Z}_N^n$. Then, the semi-functional ciphertext is set to be:

$$(A, \rho), \quad C = C' \cdot g_2^c, \quad C_i = C'_i \cdot g_2^{A_i \cdot w + \gamma_i z_{\rho(i)}},$$

$$D_i = D'_i \cdot g_2^{-\gamma_i} \ \ \forall i \in [1, n], \quad \hat{C}_x = \hat{C}'_x \cdot g_2^{b' z'_x} \ \ \forall x \in [1, m]$$

It should be noted that the values z_i and z'_x are chosen randomly once and then fixed — the same values are also involved in semi-functional keys as defined below.

Semi-functional Key. A semi-functional key will take on one of the following two forms. In order to create a semi-functional key, we first use the key generation algorithm to form a normal key $\mathsf{TK}'_{\hat{\mathcal{P}},\mathcal{S}} = ((\hat{A}, \hat{\rho}), K', L', K'_i, \hat{K}'_x, \tilde{K}'_x)$. Then, we choose random exponents $d, b \in \mathbb{Z}_N$, random values $\gamma'_x \in \mathbb{Z}_N$ associated to row x of matrix $\hat{A}$ and a random vector $\hat{w} \in \mathbb{Z}_N^n$. The semi-functional key of type 1 is set as:

$$(\hat{A}, \hat{\rho}), \quad K = K' \cdot g_2^d \quad L = L' \cdot g_2^b, \quad K_i = K'_i \cdot g_2^{b z_i} \ \ \forall i \in [1, n]$$

$$\hat{K}_x = \hat{K}'_x \cdot g_2^{\hat{A}_x \cdot \hat{w} + \gamma'_x z'_{\hat{\rho}(x)}} \ \ \forall x \in [1, m], \quad K'_x = \tilde{K}'_x \cdot g_2^{\gamma'_x} \ \ \forall x \in [1, m]$$

The semi-functional key of type 2 is set as:

$$(\hat{A}, \hat{\rho}), \quad K = K' \cdot g_2^d \quad L = L', \quad K_i = K'_i \ \ \forall i \in [1, n]$$

$$\hat{K}_x = \hat{K}'_x \cdot g_2^{\hat{A}_x \cdot \hat{w}} \ \ \forall x \in [1, m], \quad K'_x = \tilde{K}'_x \ \ \forall x \in [1, m]$$

We will prove the security of the proposed scheme based on the Assumptions 1, 2, 3 and 4 using a hybrid argument over a sequence of games. The first game, Game_{real}, is the real security game where the ciphertext and all token keys are normal. In the next game, Game_0, all of token keys are normal, but the challenge ciphertext is semi-functional. Let q denote the number of token key queries made by the attacker. For k from 1 to q and l from 1 to m, we define:

$\mathsf{Game}_{k,1}$. In this game, the challenge ciphertext is semi-functional, the first $k-1$ token keys are semi-functional of type 2, the k^{th} token key is semi-functional of type 1, and the remaining token keys are normal.

$\mathsf{Game}_{k,2}$. In this game, the challenge ciphertext is semi-functional, the first k token keys are semi-functional of type 2, the remaining keys are normal.

$\mathsf{Game}_{keyword_l}$. In this game, all token keys are semi-functional of type 2, and the challenge ciphertext $CT = (C, C_i, D_i, \hat{C}_x)$ is a semi-functional ciphertext with $\hat{C}_1, \ldots, \hat{C}_l$ randomly chosen from $\mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}$.

Game_{Final_0}. This game is the same as $\mathsf{Game}_{keyword_m}$.

Game_{Final_1}. This game is the same as Game_{Final_0}, except that in the challenge ciphertext C_i are chosen from $\mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{G_4}$ at random.

We prove that these games are indistinguishable in five lemmas, which are given in the Appendix. Therefore, we conclude that the advantage of the adversary in Game_{real}, i.e. the real security game, is negligible. This completes the proof of Theorem 1.

3.5 Efficiency

Let $|\mathbb{G}|$ be the length of the bit-representation of a group in $\mathbb{G}$. The size of the public key, a token key, and a ciphertext are $(n + m + 4)|\mathbb{G}|$, $(n + 2m + 2)|\mathbb{G}|$, and $(2n + m + 1)|\mathbb{G}|$, respectively. For a predicate $(A, \rho, \mathcal{T})$, let $l_1 = |I_{A,\rho}|, I_{A,\rho} = \{I_1, \ldots, I_{l_1}\}$ and $l_2 = |I_1| + \ldots + |I_{l_1}|$; for a predicate $(\hat{A}, \hat{\rho}, \hat{\mathcal{T}})$, let $\hat{l}_1 = |\hat{I}_{\hat{A},\hat{\rho}}|, \hat{I}_{\hat{A},\hat{\rho}} = \{\hat{I}_1, \ldots, \hat{I}_{\hat{l}_1}\}$ and $\hat{l}_2 = |\hat{I}_1| + \ldots + |\hat{I}_{\hat{l}_1}|$. Then, the computational costs of an encryption and a test are $(4n + 2m + 1)t_e + (4n + 2m + 1)t_m$ and $(2l_1\hat{l}_2 + 2l_2\hat{l}_1 + 1)t_b + (l_1\hat{l}_2 + 2l_2\hat{l}_1)t_{T\_m} + (l_1\hat{l}_2 + l_2\hat{l}_1)t_{T\_e}$, respectively, where t_b, t_e, t_m, t_{T_e}, and t_{T_m} denote the computational costs of bilinear map, exponentiation in $\mathbb{G}$, multiplication in $\mathbb{G}$, exponentiation in $\mathbb{G}_T$, and multiplication in $\mathbb{G}_T$, respectively. We note that the proposed AS-PKE scheme may not be highly practical due to the use of composite order bilinear groups. The major contribution of this paper is more on the theoretical aspects, including the concept and the security model of AS-PKE, and the first AS-PKE scheme and its security proof. In the future, we will investigate how to construct more efficient AS-PKE schemes.

3.6 Discussion

The proposed AS-PKE scheme is based on the KP-ABE scheme proposed by Lewko et al. and the CP-ABE with hidden access structures proposed by Lai et al. [7,11]. Different from the KP-ABE scheme [7] which works in a small universe of attributes, the keywords in the proposed AS-PKE scheme have a large universe (i.e. $\mathbb{Z}_N$). The proposed AS-PKE scheme can be easily extended to obtain an anonymous dual-policy ABE scheme which implies a fully secure dual-policy ABE scheme [14].

Similar to the KP-ABE scheme in [7], the proposed AS-PKE scheme has a restriction that each keyword field can only be used once in a predicate, which is

called one-use AS-PKE. We can construct a secure AS-PKE scheme where the keyword fields can be used multiple times (up to a constant number of uses fixed at setup) from a one-use AS-PKE scheme by applying the generic transformation given in Lewko et al. [7].

4 Conclusion

This paper presented AS-PKE, a public-key encryption scheme supporting both keyword search and access control capabilities. The AS-PKE scheme is constructed based on the KP-ABE scheme proposed by Lewko et al. [7] and the CP-ABE with hidden access structure proposed by Lai et al. [11]. The scheme supports monotone boolean predicates and is proven to be fully secure in the standard model.

Acknowledgments. The work of Jie Shi was supported by the National Natural Science Foundation of China (No. 61300227), and the Guangdong Provincial Natural Science Foundation (No. S2013040015711). The work of Junzuo Lai was supported by the National Natural Science Foundation of China (Nos. 61300226, 61272534), the Research Fund for the Doctoral Program of Higher Education of China (No. 20134401120017), the Guangdong Provincial Natural Science Foundation (No. S2013040014826), and the Fundamental Research Funds for the Central Universities. The work of Jian Weng was supported by the National Science Foundation of China (Nos. 61272413, 61133014), the Fok Ying Tung Education Foundation (No. 131066), the Program for New Century Excellent Talents in University (No. NCET-12-0680), the Research Fund for the Doctoral Program of Higher Education of China (No. 20134401110011), and the Foundation for Distinguished Young Talents in Higher Education of Guangdong (No. 2012LYM 0027).

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing. Commun. ACM 53(4), 50–58 (2010)
2. Li, M., Yu, S., Cao, N., Lou, W.: Authorized private keyword search over encrypted data in cloud computing. In: ICDCS, pp. 383–392 (2011)
3. Benaloh, J., Chase, M., Horvitz, E., Lauter, K.: Patient controlled encryption: ensuring privacy of electronic medical records. In: CCSW, pp. 103–114 (2009)
4. Li, M., Yu, S., Ren, K., Lou, W.: Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings. In: Jajodia, S., Zhou, J. (eds.) SecureComm 2010. LNICST, vol. 50, pp. 89–106. Springer, Heidelberg (2010)
5. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)

6. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communications Security, pp. 89–98 (2006)
7. Lewko, A.B., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010)
8. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334 (2007)
9. Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011)
10. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)
11. Lai, J., Deng, R.H., Li, Y.: Expressive CP-ABE with partially hidden access structures. In: ASIACCS, pp. 18–19 (2012)
12. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004)
13. Park, D.J., Kim, K., Lee, P.J.: Public key encryption with conjunctive field keyword search. In: Lim, C.H., Yung, M. (eds.) WISA 2004. LNCS, vol. 3325, pp. 73–86. Springer, Heidelberg (2005)
14. Attrapadung, N., Imai, H.: Dual-policy attribute based encryption. In: Abdalla, M., Pointcheval, D., Fouque, P.-A., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 168–185. Springer, Heidelberg (2009)
15. Goyal, V., Jain, A., Pandey, O., Sahai, A.: Bounded ciphertext policy attribute based encryption. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 579–591. Springer, Heidelberg (2008)
16. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 195–203. ACM (2007)
17. Cheung, L., Newport, C.C.: Provably secure ciphertext policy ABE. In: ACM Conference on Computer and Communications Security, pp. 456–465 (2007)
18. Nishide, T., Yoneyama, K., Ohta, K.: Attribute-based encryption with partially hidden encryptor-specified access structures. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 111–129. Springer, Heidelberg (2008)
19. Lai, J., Deng, R.H., Li, Y.: Fully secure cipertext-policy hiding CP-ABE. In: Bao, F., Weng, J. (eds.) ISPEC 2011. LNCS, vol. 6672, pp. 24–39. Springer, Heidelberg (2011)
20. Shi, E., Waters, B.: Delegating capabilities in predicate encryption systems. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 560–578. Springer, Heidelberg (2008)
21. Okamoto, T., Takashima, K.: Hierarchical predicate encryption for inner-products. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 214–231. Springer, Heidelberg (2009)
22. Shen, E., Shi, E., Waters, B.: Predicate privacy in encryption systems. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 457–473. Springer, Heidelberg (2009)

23. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (2010)
24. Hwang, Y.H., Lee, P.J.: Public key encryption with conjunctive keyword search and its extension to a multi-user system. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 2–22. Springer, Heidelberg (2007)
25. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (2007)
26. Sun, W., Yu, S., Lou, W., Hou, Y.T., Li, H.: Protecting your right: Attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud. In: INFOCOM (2014)
27. Narayan, S., Gagné, M., Safavi-Naini, R.: Privacy preserving EHR system using attribute-based infrastructure. In: CCSW, pp. 47–52 (2010)
28. Beimel, A.: Secure schemes for secret sharing and key distribution. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel (1996)
29. Lewko, A.B., Waters, B.: Decentralizing attribute-based encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 568–588. Springer, Heidelberg (2011)
30. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
31. Lai, J., Zhou, X., Deng, R.H., Li, Y., Chen, K.: Expressive search on encrypted data. In: Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, ASIA CCS 2013, pp. 243–252. ACM, New York (2013)

A Lemmas

The following five lemmas are used in the proof of Theorem 1. The proof of the lemmas is detailed in (the appendix of) the full version of this paper, which is accessible at http://www.mysmu.edu/faculty/yjli/ASPKE-full.pdf.

Lemma 1. *Suppose that $\mathcal{G}$ satisfies Assumption* 1*. Then* Game_{real} *and* Game_0 *are computationally indistinguishable.*

Lemma 2. *Suppose that $\mathcal{G}$ satisfies Assumption* 2*. Then* $\mathsf{Game}_{k-1,2}$ *and* $\mathsf{Game}_{k,1}$ *are computationally indistinguishable.*

Lemma 3. *Suppose that $\mathcal{G}$ satisfies Assumption 2. Then* $\mathsf{Game}_{k,1}$ *and* $\mathsf{Game}_{k,2}$ *are computationally indistinguishable.*

Lemma 4. *Suppose that $\mathcal{G}$ satisfies Assumption 3. Then* $\mathsf{Game}_{keyword_{l-1}}$ *and* $\mathsf{Game}_{keyword_l}$ *are computationally indistinguishable.*

Lemma 5. *Suppose that $\mathcal{G}$ satisfies Assumption 4. Then* Game_{final_0} *and* Game_{final_1} *are computationally indistinguishable.*