# An Efficient Key-Policy Attribute-Based Searchable Encryption in Prime-Order Groups

Ru Meng[1](✉), Yanwei Zhou[1], Jianting Ning[2], Kaitai Liang[3], Jinguang Han[3], and Willy Susilo[4]

[1] School of Computer Science, Shaanxi Normal University, Xi'an, China
{mengru,zyw}@snnu.edu.cn
[2] Department of Computer Science,
National University of Singapore, Singapore, Singapore
ningjt@comp.nus.edu.sg
[3] Department of Computer Science, University of Surrey, Guildford, UK
ktliang88@gmail.com, jghan22@gmail.com
[4] School of Computing and Information Technology,
Institute of Cybersecurity and Cryptology, University of Wollongong,
Wollongong, Australia
wsusilo@uow.edu.au

**Abstract.** Public key encryption with keyword search (PEKS) is a promising cryptographic mechanism to enable secure search over encrypted data in cloud. The mechanism allows a semi-trusted cloud server to return related encrypted contents without knowing *what the query is* and *what the corresponding contents are*. It has been combined with attribute based encryption (ABE) to support more expressiveness in search. Most of the existing searchable ABE schemes, however, are restricted to heavy complexity. In particular, the size of ciphertext and pairing cost in the test phase are both linear in the size of the keyword set, say $O(n)$, where $n$ is the number of keyword. This limitation hinders the scalability of searchable ABE in practice. To address this long-lasting open problem, this paper proposes a new key-policy attribute-based search encryption (KP-ABSE) scheme. Our construction can be regarded as a *novel* combination of fast decryption, anonymous-like encryption, and KP-ABE technologies. As of independent interest, the scheme is built in asymmetric bilinear groups. The scheme is further proved secure under the asymmetric decisional DBDH, decisional q-BDHE and decisional linear assumptions in the standard model. Compared with existing KP-ABSE schemes, our new scheme achieves the following properties: (1) flexible access structure for search - any monotonic access structure, (2) *constant* ciphertext size, (3) *constant* pairing operations in the test phase.

**Keywords:** Key-policy attribute-based encryption · Searchable encryption · Prime-order groups · Efficiency

T. Okamoto et al. (Eds.): ProvSec 2017, LNCS 10592, pp. 39–56, 2017.
https://doi.org/10.1007/978-3-319-68637-0_3

## 1 Introduction

The proliferation of cloud computing has attracted many attentions from academic and industrial communities since it provides powerful computing capability and considerable storage space. It can reduce Internet users' local data management and maintenance cost significantly. Users can access cloud services whenever and wherever once they are authorized by service providers. Due to its merits, companies and individuals are willing to store their data in a remote cloud. Since users will lose their control on data after outsourcing their data to the cloud, they concern that the data may be illegally accessed by the cloud server administrator and network attackers. Considering the confidentiality of the outsourced data, users often encrypt it first, and then store the ciphertext to cloud servers. However, it is difficult to search an "exact" file among encrypted data stored in cloud.

In 2000, Song *et al.* [45] first proposed the definition of searchable encryption (SE). In [45], a data owner is allowed to encrypt both files and the corresponding keywords, and store the ciphertexts to cloud. When searching for a file with keywords $W$, the data user generates a trapdoor using his/her secret key and further sends the trapdoor to the server. After receiving the trapdoor, the server searches out the encrypted file where the keywords $W$ matches, and returns the search result to the user. Finally, the user can use the secret key to decrypt the ciphertext and obtain the file. In 2004, Boneh *et al.* [8] introduced the concept of public-key encryption with keyword search (PEKS), and constructed a concrete PEKS scheme based on bilinear groups with prime order. In 2006, Khader [25] proposed an identity-based PEKS derived from identity-based encryption (IBE). In 2007, Abdalla *et al.* [1] presented a generic construction of PEKS by using anonymous IBE, and discussed the consistency in PEKS schemes.

Previous PEKS schemes can only support simple query and the size of ciphertexts and trapdoor (search token) is super-polynomial in the number of keywords. In practice, fine-grained access control is required. In 2013, Lai *et al.* [28] proposed an expressive searchable encryption scheme based on KP-ABE scheme. This scheme supports any monotonic formula, for example, ("sender : Bob AND priority : urgent OR subject : recruitment"). However, the trapdoor can leak the information of keywords, namely the test algorithm can detect whether the encrypted data contains some keywords in trapdoor. In 2014, Lv *et al.* [38] proposed an expressive and secure asymmetric searchable encryption (ESASE) scheme, which was based on an asymmetric bilinear group with composite order and supports non-monotonic query. Nevertheless, the scheme only disclosed whether the keywords in the trapdoor are primed or not. In 2016, Cui *et al.* [14] proposed an efficient and expressive keyword searchable encryption scheme constructed in a bilinear group with prime order. The scheme is selectively secure in the standard model. It supports keyword search policies in terms of conjunctive, disjunctive and any monotonic Boolean formula. However, it brings some critical issue to search efficiency. In most existing expressive searchable encryption schemes derived from ABE, both the size of ciphertext and the search cost are linear in the number of keywords. Specifically, in the test (search) algorithm, it usually requires one pairing operation for a single keyword (embedded in a given ciphertext). Hence, the existing expressive searchable encryption schemes built on top of ABE are not efficient and scalable.

Attrapadung *et al.* [2] and Hohenberger *et al.* [24] presented KP-ABE schemes with constant-size ciphertext and fast decryption, respectively. In 2014, Lai *et al.* [27] proposed a new KP-ABE with constant-size ciphertext and fast decryption, which is adaptability secure in the standard model. KP-ABE schemes do not consider the privacy issue of attributes associated with ciphertext. However, searchable encryption requests that ciphertext should not reveal any information about keywords except that a valid trapdoor is provided.In this paper, we propose a new efficient key-policy attribute-based searchable encryption (KP-ABSE) scheme which is derived from an asymmetric bilinear group with prime order. In this scheme, the privacy of keyword in both ciphertext and trapdoor are addressed. Moreover, both the size of ciphertext and the computation cost of the test algorithm are constant. Compared with expressive searchable encryption based on bilinear groups with prime order, our work is more efficient.

### 1.1 Technical Roadmap

*Protecting Privacy of Keywords in Ciphertext.* (1) We use anonymity from the asymmetric technique [15] to encrypt keywords in group $\mathbb{G}$; while trapdoors are generated in group $\hat{\mathbb{G}}$ to prevent cloud servers, and adversaries from raising keyword guess attacks using pairing operations [5]. As claimed in [15], asymmetric bilinear groups provide good properties, including compact representation of group elements, a flexible choice of elliptic curve implementation [18] and strong security [20]. (2) We use the linear splitting technique [9] to split the random exponent used to hide keywords into two parts. As a result, adversaries cannot obtain any information about keywords even if they acquire the ciphertext and public parameters. Secret keys are randomized in the test algorithm.

*Protecting Privacy of Keywords in Access Structure.* We divide each keyword into two parts: the keyword name and the keyword value [26]. In practice, keyword values are more sensitive than keyword names. If the set of attributes associated with a users private key does not satisfy the access structure associated with a ciphertext, attribute values in the access structure are hidden, while other information, such as attribute names, about the access structure is public. Suppose that the access structure in personal health database is (illness = diabetes) OR (gender = male) OR (department = medical) OR (affiliation= city hospital) where illness, gender, department and affiliation are keyword names and diabetes, male, medical and city hospital are keyword values. The keyword names contains less sensitive information and can be released, while keyword values are very sensitive and should be kept secret. Hence, in our scheme, we mainly consider to protect the privacy of keyword values. PKES is subject to the offline keywords dictionary guessing attacks since anyone who knows the trapdoor and public parameters can conclude the value embedded in the trapdoor by executing exhaustive search. To prevent the above attacks, the designated technique [43] is used. The idea is that trapdoors are encrypted under the public key of the cloud server such that adversaries cannot acquire any information about keywords without knowing the secret key. Therefore, trapdoors can be transferred in public channels.

### 1.2 Contributions

We propose a new key-policy attribute-based search encryption scheme (KP-ABSE) which is derived from KP-ABE in asymmetric bilinear group with prime order. The proposed scheme has the following good properties: (1) It is expressive and supports any monotonic access structure; (2) It has constant-size ciphertext and supports fast decryption; (3) The number of pairing operations needed in the test algorithm is constant. Therefore, it reduces the computation cost on cloud server side as well as communication cost between the data users and cloud. One disadvantage of our scheme is that the size of trapdoors is $\mathcal{O}(n \cdot \ell)$, where $n$ is the number of attributes in the system and $\ell$ is the number of leaf nodes in the access structure. Note that we will regard this as an open problem of our research work. However, depending on applications, one should take into consideration if the increase of trapdoor size is worthy.

### 1.3 Related Work

*Attribute-Based Encryption.* To implement fine-grained access control on sensitive data, Sahai and Water [44] introduced the definition of attribute-based encryption (ABE). ABE schemes can be classified into two types: key-policy ABE (KP-ABE) [21] and ciphertext-policy ABE (CP-ABE) [4]. In a KP-ABE scheme [21], secret keys are associated with access structures; while ciphertexts are labeled with sets of attributes. A user can decrypt a ciphertext if and only if the access structure associated with his secret key can be satisfied by the attributes labeled in ciphertexts. On the contrary, in a CP-ABE scheme [4], secret keys are labeled with sets of attributes; while ciphertexts are associated with access structures.

Goyal *et al.* [21] proposed a KP-ABE scheme which supports any monotonic access structure. Later, Ostrovsky *et al.* [41] presented a KP-ABE system which supports non-monotonic access structures. Lewko *et al.* [29] proposed the first fully secure KP-ABE scheme supporting any monotonic access structure. Chase *et al.* [10,11] considered multi-authority KP-ABE schemes. The first CP-ABE was proposed by Bethencourt *et al.* [4] and was proven to be secure in the generic group model. Later, Cheung and Newport [12] presented a CP-ABE scheme which is secure in the standard model; while, it can only support restricted access structures, for example AND gate. Lewko *et al.* [30] considered multi-authority CP-ABE schemes to reduce the trust on central authority. Some ABE variants and applications can be seen in [32,33,39,40,46].

*Attribute-Based Encryption with Fast Decryption.* In KP-ABE schemes, both the size of the ciphertext and the decryption cost are linear with the number of required attributes. To reduce the size of ciphertext and decryption cost, some new KP-ABE were presented [2,27,44]. Meanwhile, in CP-ABE scenario, the size of ciphertext and decryption cost were also considered. Emura *et al.* [16] proposed a CP-ABE scheme with constant-size ciphertext which can only supports restricted access structures, such as AND gate. Herranz *et al.* [23] described a CP-ABE scheme with constant-size ciphertext which supports threshold access structures. Hohenberger [24] proposed a KP-ABE with fast decryption. In [24], the decryption cost is constant, instead of linear with the number of required attributes. In 2014, Lai *et al.* [27] proposed a KP-ABE with constant-size ciphertext and fast decryption.

*Keyword search over Encrypted Data.* Boneh *et al.* [8] initiated the research on PEKS and gave a specific construction which only supports equality queries. Abdalla *et al.* [1] addressed the consistency in PEKS schemes, and analyzed the relationship between PEKS and anonymous IBE. To guarantee the correctness of the searching results, verifiable keyword search schemes have been proposed [3,17,42]. In these schemes, each keyword is represented as the root of one polynomial. It is easy to check whether a keyword is included by evaluating the polynomial on the keyword and verify whether the output is zero or not. Zheng *et al.* [48] proposed a novel PEKS called verifiable attribute-based keyword search (VABKS). This allows legitimate data users to outsource the (often costly) search operations to cloud servers and verify whether cloud servers have faithfully executed the search operations. Some variants of ABE searchable encryption have been proposed in [34–37].

### 1.4 Organization

The rest of this paper is organized as follows. In Sect. 2, we briefly review definitions and models used in this paper. Section 3 describes the preliminaries used throughout this paper and notions of KP-ABSE. In Sect. 4, a concrete KP-ABSE scheme is presented. We compare our work with other related works in Sect. 5. Section 6 concludes the paper.

## 2 System Definitions

### 2.1 System Algorithms

A key-policy attribute-based search encryption (KP-ABSE) system includes four parties, namely, data owner, cloud server, Trusted Key Generator (TKG), and data user.

**Definition 1.** *A KP-ABSE system consists of the following algorithms* [14]*:*

1. *Setup*$(1^\lambda) \rightarrow (pars, msk)$*: intaking a security parameter* $\lambda$*, the TKG runs the setup algorithm to construct the public parameters* $pars$*, and the master secret key* $msk$*. The* $pars$ *is published, while the* $msk$ *is kept secret.*

2. *$sKeyGen(pars) \rightarrow (pk_s, sk_s)$: intaking pars, the TKG runs the server key generation algorithm to construct the public key $pk_s$ and the private key $sk_s$ for the cloud server.*
3. *$Encrypt(pars, \boldsymbol{W}) \rightarrow CT$: intaking pars, and a set of keywords $\boldsymbol{W}$, a data owner runs the encryption algorithm to output a ciphertext $CT$.*
4. *$Trapdoor(pars, msk, pk_s, \mathbb{A}) \rightarrow T_{\mathbb{M}}$: intaking pars, msk, $pk_s$ and an access structure $\mathbb{A}$ (corresponding to some keyword set), the TKG runs the trapdoor generation algorithm to construct a trapdoor $T_{\mathbb{M}}$, and further sends $T_{\mathbb{M}}$ to the cloud server.*
5. *$Test(pars, sk_s, CT, T_{\mathbb{M}}) \rightarrow 0/1$: Intaking pars, $sk_s$, $CT$ and $T_{\mathbb{M}}$, the cloud server runs the test algorithm. It outputs 1 if the keyword set embedded in $CT$ matches the access structure in $T_{\mathbb{M}}$, and 0 otherwise.*

**Correctness:** A key-policy attribute-based search encryption is correct if

$$\Pr\left[ Test(pars, sk_s, CT, T_{\mathbb{M}}) \rightarrow 1 \middle| \begin{array}{l} Setup(1^{\lambda}) \rightarrow (pars, msk); \\ Encrypt(pars, \mathbf{W}) \rightarrow CT; \\ sKeyGen(pars) \rightarrow (pk_s, sk_s); \\ Trapdoor(pars, msk, pk_s, \mathbb{A}) \rightarrow T_{\mathbb{M}} \end{array} \right] = 1.$$

## 2.2 System Workflow

The architecture of our system workflow is shown in Fig. 1, which is composed of four entities: ***a trusted key generator (TKG)*** who publishes the system parameter and holds a master private key and is responsible for trapdoor generation for the system. We may regard the TKG as trusted device(s), like TPM. A user may make use of this device in some untrusted computers (like those in library or public area) to generate a token for further search. But the device may not have sufficient knowledge about positive or negative cases (on access control rules). Because it may not be allowed to access, say the access control list.

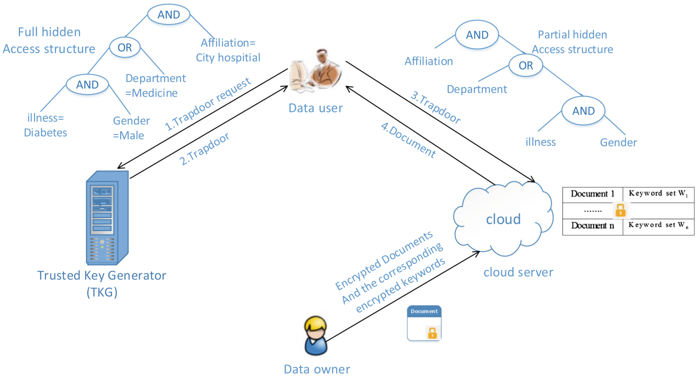

**Fig. 1.** System workflow

***data owners*** who outsource encrypted data to a public cloud, ***data users*** who are privileged to search and access encrypted data, and ***a cloud server*** who executes the keyword search operations for data users. To enable the cloud server to search over ciphertexts, the data owners append every encrypted document with encrypted keywords. A data user issues a trapdoor request by sending a keyword access structure to the TKG which generates and returns a trapdoor corresponding to the access structure. After obtaining a trapdoor, the data user sends the trapdoor and the corresponding partial hidden access structure (i.e., the access structure without keyword values) to the designated cloud server. The latter performs the testing operations between each ciphertext and the trapdoor using its private key, and forwards the matching ciphertexts to the data user.

### 2.3 Adversary Models

In this paper, we assume that data owner, data user and the cloud server are semi-trusted, while the TKG is fully trusted. However, for a data user, he/she may choose to guess the keyword set embedded into a given ciphertext without the help of the server. For a "curious" server, it may curiously guess the keyword set in the ciphertext of which the corresponding search trapdoor is not given; it may also guess the keyword information from a given search trapdoor. Therefore, we define the following three security models.

**Indistinguishability Against Chosen Keyword-Set Attacks (IND-CKA).** This security model focuses on the privacy of the keyword set associated with a given ciphertext. There are two kinds of adversaries in this model, one is outside-attacker, and the other is the cloud server itself. Below, we define two security games by constructing interactions between a challenger $\mathcal{B}$ and an adversary $\mathcal{A}$.

**IND-CKA Security for Outsider.** This security game between $\mathcal{A}_1$ and $\mathcal{B}$ is used to show that a system outsider, without the help of the cloud server, cannot tell if a given ciphertext contains some specified keyword set (here the outsider is allowed to commit to two known keyword sets at the outset of the game).

**Definition 2.** *A KP-ABSE scheme is $IND\text{-}CKA^{\mathcal{A}_1}$ secure if no PPT adversary $\mathcal{A}_1$ can win the game below with non-negligible advantage* [14].

1. ***Init.*** *$\mathcal{A}_1$ commits to two equal length challenge keyword sets $\boldsymbol{W}_0^*$, $\boldsymbol{W}_1^*$.*
2. ***Setup.*** *$\mathcal{B}$ runs Setup$(1^\lambda)$, and further sends pars to $\mathcal{A}_1$. It runs sKeyGen(pars) and next returns $pk_s$ to $\mathcal{A}_1$.*
3. ***Phase 1.*** *$\mathcal{A}_1$ issues search trapdoor queries to $\mathcal{B}$ by submitting ($\mathbb{M}_1$, $\rho_1$, $\{W_{\rho_1(i)}\}$), ..., ($\mathbb{M}_{q_1}$, $\rho_{q_1}$, $\{W_{\rho_{q_1}(i)}\}$). $\mathcal{B}$ returns the corresponding trapdoors to $\mathcal{A}_1$ by running the algorithm Trapdoor.*
4. ***Challenge.*** *$\mathcal{B}$ returns the challenge ciphertext $CT^* = Encrypt(pars,\ \boldsymbol{W}_\beta^*)$ to $\mathcal{A}_1$, where $\beta \in_R \{0,1\}$. Note that the challenge ciphertext cannot match any trapdoor constructed in Phase 1 (namely, both of the challenge keyword sets cannot match the given trapdoors).*

5. ***Phase 2.*** *$\mathcal{A}_1$ continues making queries as in Phase 1, by issuing ($\mathbb{M}_{q_1+1}$, $\rho_{q_1+1}$, $\{W_{\rho_{q_1+1}(i)}\}$), ..., ($\mathbb{M}_q$, $\rho_q$, $\{W_{\rho_q(i)}\}$), with a restriction that the queries cannot match the given challenge keyword sets.*
6. ***Guess.*** *$\mathcal{A}_1$ outputs a guess bit $\beta' \in \{0,1\}$. If $\beta = \beta'$, $\mathcal{A}_1$ wins.*

*The advantage of $\mathcal{A}_1$ is defined as $Adv_{\mathcal{A}_1}(1^\lambda) = |Pr[\beta' = \beta] - \frac{1}{2}|$.*

**IND-CKA Security for the Cloud Server.** This security game between $\mathcal{A}_2$ and $\mathcal{B}$ is used to show that the cloud server, without a valid search trapdoor, cannot tell if a given ciphertext contains some specified keyword set (here the cloud server is allowed to commit to two "known" keyword sets in advance).

**Definition 3.** *A KP-ABSE scheme is $IND\text{-}CKA^{\mathcal{A}_2}$ secure if no PPT adversary $\mathcal{A}_2$ can win the game below with non-negligible advantage [14].*

1. ***Init.*** *$\mathcal{A}_2$ commits to two equal length challenge keyword sets $\boldsymbol{W}_0^*$, $\boldsymbol{W}_1^*$.*
2. ***Setup.*** *$\mathcal{B}$ runs $Setup(1^\lambda)$ to send pars to $\mathcal{A}_2$. It further runs $sKeyGen(pars)$ to return $pk_s, sk_s$ to $\mathcal{A}_2$.*
3. ***Phase 1.*** *$\mathcal{A}_2$ issues search trapdoor queries to $\mathcal{B}$ by submitting ($\mathbb{M}_1$, $\rho_1$, $\{W_{\rho_1(i)}\}$), ..., ($\mathbb{M}_{q_1}$, $\rho_{q_1}$, $\{W_{\rho_{q_1}(i)}\}$). For each query ($\mathbb{M}_j, \rho_j, \{W_{\rho_j(i)}\}$), $j \in [1, q_1]$, $\mathcal{B}$ returns the corresponding trapdoor $T_{\mathbb{M}_j}$ to $\mathcal{A}_2$ by running the algorithm $Trapdoor$.*
4. ***Challenge.*** *$\mathcal{B}$ randomly chooses $\beta \in \{0,1\}$ and returns the challenge ciphertext $CT^* = Encrypt(pars,\ \boldsymbol{W}_\beta^*)$ to $\mathcal{A}_2$ with a restriction that the challenge ciphertext cannot match any trapdoor given in Phase 1.*
5. ***Phase 2.*** *$\mathcal{A}_2$ continues making queries by issuing ($\mathbb{M}_{q_1+1}$, $\rho_{q_1+1}$, $\{W_{\rho_{q_1+1}(i)}\}$), ..., ($\mathbb{M}_q$, $\rho_q$, $\{W_{\rho_q(i)}\}$), with a restriction that the queries cannot match the given challenge keyword sets.*
6. ***Guess.*** *$\mathcal{A}_2$ outputs a guess bit $\beta' \in \{0,1\}$. If $\beta = \beta'$, $\mathcal{A}_2$ wins.*

*The advantage of $\mathcal{A}_2$ is defined as $Adv_{\mathcal{A}_2}(1^\lambda) = |Pr[\beta' = \beta] - \frac{1}{2}|$.*

*For $\mathcal{A} \in \{\mathcal{A}_1, \mathcal{A}_2\}$, an KP-ABSE system is selectively IND-CKA secure if the advantage function referring to the security $Game_{\Pi,\mathcal{A}}^{(IND)}$, $Adv_{\Pi,\mathcal{A}}^{(IND)}(\lambda) = Pr[\beta \neq \beta'] - \frac{1}{2}$ is negligible in the security parameter $\lambda$ for any probabilistic polynomial time adversary algorithm $\mathcal{A}$.*

# 3 Preliminaries

## 3.1 Bilinear Maps

Let $\mathbb{G}$, $\hat{\mathbb{G}}$ and $\mathbb{G}_T$ be all multiplicative groups of prime order $p \in \Theta(2^\lambda)$, respectively generated by $g$,$\hat{g}$ and $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ is an efficient bilinear map with the following properties: (1) *Bilinearity*: for all $a, b \in_R \mathbb{Z}_p$, $e(g^a, \hat{g}^b) = e(g, \hat{g})^{ab}$; (2) *Non-degeneracy*: $e(g, \hat{g}) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ is the unit of $\mathbb{G}_T$; (3) *Computability*: for all $g \in \mathbb{G}$ and $\hat{g} \in \hat{\mathbb{G}}$, $e(g, \hat{g})$ can be computed efficiently.

### 3.2 Complexity Assumptions

**Definition 4. Asymmetric Decisional Bilinear Diffie-Hellman (DBDH) Assumption** *[47] is that all Probabilistic Polynomial Time (PPT) algorithm $\mathcal{A}$ have an advantage negligible in $\lambda$ of distinguishing $e(g, \hat{g})^{abc} \in \mathbb{G}_T$ from a random element in $\mathbb{G}_T$ by given the vector $y = (g, g^a, g^c, \hat{g}, \hat{g}^a, \hat{g}^b)$. The advantage of $\mathcal{A}$ is defined as $|Pr[\mathcal{A}(y, e(g, \hat{g})^{abc}) = 1] - Pr[\mathcal{A}(y, Z) = 1]|$, where the probability is over the randomly chosen $g \leftarrow \mathbb{G}$,$\hat{g} \leftarrow \hat{\mathbb{G}}$, $a, b, c$, and the random bits consumed by $\mathcal{A}$.*

**Definition 5. Asymmetric Decisional q-Bilinear Diffie-Hellman Exponent (q-BDHE) Assumption** *[6] is that all PPT algorithms $\mathcal{A}$ have an advantage negligible in $\lambda$ of distinguishing $e(g, \hat{g})^{a^{q+1}b} \in \mathbb{G}_T$ from a random element in $\mathbb{G}_T$ by given the vector*

$$y = g, g^b, g^a, g^{a^2}, ..., g^{a^q}, g^{a^{q+2}}, \hat{g}, \hat{g}^a, \hat{g}^{a^2}, ..., \hat{g}^{a^q}, \hat{g}^{a^{q+2}} ..., \hat{g}^{a^{2q}}, T$$

*The advantage of $\mathcal{A}$ is defined as $|Pr[\mathcal{A}(y, e(g, \hat{g})^{a^{q+1}b}) = 1] - Pr[\mathcal{A}(y, T) = 1]|$, where the probability is over the randomly chosen $a, b$, and the generator $g$,$\hat{g}$, and the random bits consumed by $\mathcal{A}$.*

**Definition 6. Asymmetric Decisional Linear Assumption** *[7] is that all PPT algorithms $\mathcal{A}$ have an advantage negligible in $\lambda$ of distinguishing $Z = g^{x_3+x_4} \in \mathbb{G}$ from a random element in $\mathbb{G}$ by given the vector $y = \{g,\ g^{x_1},\ g^{x_2}, g^{x_1x_3},\ g^{x_2x_4},\ \hat{g},\ g^{\hat{x_1}}, g^{\hat{x_2}}\}$. The advantage of $\mathcal{A}$ is defined as $|Pr[\mathcal{A}(y, g^{x_3+x_4} = 1] - Pr[\mathcal{A}(y, Z) = 1]|$, where the probability is over the randomly chosen $x_1, x_2, x_3, x_4 \in \mathbb{Z}_p$, and the random bits consumed by $\mathcal{A}$. We remark that the elements $\hat{g}, g^{\hat{x_1}}, g^{\hat{x_2}}$ were not explicitly included in Boenh's et al. original formulation.*

### 3.3 Building Blocks

**Definition 7. Access Structure** *[31]. Let $\{P_1, ..., P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,...,P_n\}}$ is monotone if $\forall B, C$: $B \in \mathbb{A}$ 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 $\{P_1, ..., P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1,...,P_n\}} \setminus \{\}$. The set in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.*

Note in our setting keywords will play the role of parties and we only consider the monotone access structures, and the negation of a keyword is regarded as a separate keyword.

**Definition 8. Linear Secret-Sharing Schemes (LSSS)** *[31]. A secret sharing scheme $\Pi$ over a set of parties $P$ 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 $\mathbb{M}$ called the share-generating matrix for $\Pi$. The matrix $\mathbb{M}$ has $l$ rows and $n$ columns. For all $i = 1, ...., l$, the ith row of $\mathbb{M}$ is labeled by a party $\rho(i)$ ($\rho$ is a function from $\{1, ..., l\}$ to $P$). When we consider the column vector $v = (\alpha, r_2, ..., r_n)$, where $\alpha \in \mathbb{Z}_p$ is the secret to be shared and $r_2, ..., r_n \in \mathbb{Z}_p$ are randomly chosen, then $\mathbb{M}\boldsymbol{v}$ is the vector of $l$ shares of the secret $\alpha$ according to $\Pi$. The share $(\mathbb{M}\boldsymbol{v})_i$ belongs to party $\rho(i)$.*

The linear reconstruction property: let $\Pi$ be an LSSS for access structure $\mathbb{A}$, $\boldsymbol{W}$ denote an authorized set, and define $I \subseteq \{1, ..., l\}$ as $I = \{i|\rho(i) \in \boldsymbol{W}\}$. The vector $(1, 0, ..., 0)$ is in the span of rows of $\mathbb{M}$ indexed by $I$, and there exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i\in I}$ such that, for any valid shares $\{\lambda_i\}$ of a secret $\alpha$ according to $\Pi$, we have $\sum_{i\in I} \omega_i\lambda_i = \alpha$. These constants $\{\omega_i\}$ can be found in time polynomial in the size of share-generating matrix $\mathbb{M}$. But for unauthorized sets of rows $I$, the target vector is not in the span of the rows of the set $I$. Moreover, there will exists a vector $\omega$, such that $\omega\cdot(1, 0, ..., 0) = -1$ and $\omega\cdot\mathbb{M}_i = 0$ for all $i \in I$.

**Definition 9.** ***Target Collision Resistant Hash Function*** *[13]. A TCR hash function $H$ guarantees that given a random element $x$ which is from the valid domain of $H$, a PPT adversary $A$ cannot find $y \neq x$ such that $H(x) = H(y)$. We let $Adv_{H,A}^{TCR} = Pr[(x, y) \leftarrow \mathcal{A}(1^k) : H(x) = H(y), x \neq y, x, y \in DH]$ be the advantage of $A$ in successfully finding collisions from a TCR hash function $H$, where $DH$ is the valid input domain of $H$, $k$ is the security parameter. If a hash function is chosen from a TCR hash function family, $Adv_{H,A}^{TCR}$ is negligible.*

# 4 A New KP-ABSE

## 4.1 Construction

- **Setup**$(1^\lambda) \to (pars, msk)$. The setup algorithm takes as input a security parameter $1^\lambda$. It chooses bilinear groups $\mathbb{G}, \hat{\mathbb{G}}$ of prime order $p$ with generators $g, \hat{g}$, respectively. It symmetrically random chooses $u, h, \delta \in \mathbb{G}, \hat{u}, \hat{h}, \hat{\delta} \in \hat{\mathbb{G}}$ and $\alpha, d_1, d_2, d_3, d_4 \in \mathbb{Z}_p^*$. It then sets $g_1 = g^{d_1}, g_2 = g^{d_2}, g_3 = g^{d_3}, g_4 = g^{d_4}$. It also chooses a collision-resistant hash function $H$ that maps group elements in $\mathbb{G}_T$ to group elements in $\mathbb{G}$. The public parameters $pars$ and the master secret key $msk$ are given by

$$pars = (H, g, u, h, \delta, \hat{u}, g_1, g_2, g_3, g_4, e(g, \hat{g})^\alpha),$$

$$msk = (\alpha, \hat{g}, \hat{h}, \hat{\delta}, d_1, d_2, d_3, d_4).$$

- **sKeyGen**$(pars) \to (pk_s, sk_s)$. The algorithm takes as input the public parameter $pars$. It randomly chooses $\kappa \in \mathbb{Z}_p^*$ and outputs the public and private key pair $(pk_s, sk_s) = (g^\kappa, \kappa)$ for the cloud server.
- **Trapdoor**$(pars,\ pk_s,\ msk,\ \mathbb{A} = (\mathbb{M},\ \rho,\mathcal{T})) \to T_{\mathbb{M},\rho}$. The algorithm takes as input the public parameter $pars$, the server public key $pk_s$, the master private key $msk$ and an LSSS access structure $(\mathbb{M}, \rho, \mathcal{T})$, where $\mathbb{M}$ is $l \times n$

share-generating matrix, $\rho$ is a map from each row of $\mathbb{M}$ to an attribute name, $\mathcal{T} = (z_{\rho(1)}, ..., z_{\rho(l)})$ and $z_{\rho(i)}$ is the value of keyword name $\rho(i)$ specified by the access formula. It randomly chooses a vector $\boldsymbol{v} = (\alpha, y_2, ..., y_n) \in \mathbb{Z}_p^n$, and computes $\lambda_i = \boldsymbol{v} \cdot \mathbb{M}_i$ for each $i = [l]$. Let $Q_i$ denote the set $[n] \setminus \{\rho(i)\}$ for each $i \in [l]$. For each row $\mathbb{M}_i$ of $\mathbb{M}$, it chooses random $r, r', t_{1,1}, t_{1,2}, ..., t_{l,1}, t_{l,2} \in \mathbb{Z}_p$, computes $D = g^r, \hat{D} = \hat{g}^{r'}$, and outputs the trapdoor as $T_{\mathbb{M},\rho} = ((\mathbb{M},\rho)$, $D$, $\hat{D}$,$\{D_i, R_i, T_{i,1}, T_{i,2}, T_{i,3}, T_{i,4}, \{Q_{i,j}, Q'_{i,j}, Q''_{i,j}, Q'''_{i,j}\}_{j \in Q_i}\}_{i \in [l]})$

$$D_i = \hat{g}^{\lambda_i} \hat{\delta}^{d_1 d_2 t_{i,1} + d_3 d_4 t_{i,2}}, R_i = H(e(pk_s, \hat{D})^r) \cdot \hat{g}^{d_1 d_2 t_{i,1} + d_3 d_4 t_{i,2}},$$

$$T_{i,1} = (\hat{u}^{z_{\rho(i)}} \hat{h})^{-d_2 t_{i,1}}, Q_{i,j} = (\hat{u}^{z_j})^{-d_2 t_{i,1}};$$

$$T_{i,2} = (\hat{u}^{z_{\rho(i)}} \hat{h})^{-d_1 t_{i,1}}, Q'_{i,j} = (\hat{u}^{z_j})^{-d_1 t_{i,1}};$$

$$T_{i,3} = (\hat{u}^{z_{\rho(i)}} \hat{h})^{-d_4 t_{i,2}}, Q''_{i,j} = (\hat{u}^{z_j})^{-d_4 t_{i,2}};$$

$$T_{i,4} = (\hat{u}^{z_{\rho(i)}} \hat{h})^{-d_3 t_{i,2}}, Q'''_{i,j} = (\hat{u}^{z_j})^{-d_3 t_{i,2}}.$$

- **Encrypt**$(pars, \boldsymbol{W} = (\mathrm{w_1}, ..., \mathrm{w_n})) \to CT$. The algorithm takes as input the public parameter $pars$ and a keyword set $\boldsymbol{W}$ (each keyword is denoted as keyword name and keyword value, $i$ is the generic keyword name and $\mathrm{w_i}$ is the corresponding keyword value), where $\mathrm{w_1}, ..., \mathrm{w_n} \in \mathbb{Z}_\mathrm{p}$ are the values of $\boldsymbol{W}$. It chooses random $\mu, s, s_1, s_2 \in \mathbb{Z}_p$, and outputs a ciphertext $CT = (C, C', C'', E_1, E_2, E_3, E_4)$ as

$$C = e(g, \hat{g})^{\alpha\mu}, C' = g^{\mu}, C'' = \delta^{-\mu}(h \prod_{i=1}^{n} u^{\mathrm{w_i}})^s$$

$$E_1 = g_1^{s-s_1}, E_2 = g_2^{s_1}, E_3 = g_3^{s-s_2}, E_4 = g_4^{s_2}.$$

- **Test**$(pars, sk_s, CT, T_{\mathbb{M},\rho})$. The algorithm takes as input the public parameter $pars$, the server private key $sk_s$, a ciphertext $CT = (C, C', C'', E_1, E_2, E_3, E_4)$ on a keyword set $\boldsymbol{W}$ and a trapdoor $T_{\mathbb{M},\rho}$ associated with an access structure $\mathbb{A} = (\mathbb{M}, \rho, \mathcal{T})$. If the keyword set $\boldsymbol{W}$ does not satisfy $\mathbb{A}$, output $\perp$. Otherwise, if the keyword set $\boldsymbol{W}$ satisfies $\mathbb{A}$, the test algorithm first finds $\mathcal{I} \subseteq [1, l]$ and constants $\{\omega_i\}_{i \in \mathcal{I}} \in \mathbb{Z}_p$ such that $\sum_{i \in \mathcal{I}} \omega_i \mathbb{M}_i = (1, 0, ..., 0)$ and $\mathrm{w}_{\rho(\mathrm{i})} = \mathrm{z}_{\rho(\mathrm{i})}$ for $\forall i \in \mathcal{I}$. The algorithm then does as follows:
  (1) Pre-processing step on the private key
  Let $Q_i$ denote the set $[n] \setminus \{\rho(i)\}$ for each $i \in \mathcal{I}$. Note that if $j \in Q_i$, then $j \neq \rho(i)$. Since for each $i \in \mathcal{I}$, $\mathrm{w}_{\rho(\mathrm{i})} = \mathrm{z}_{\rho(\mathrm{i})}$, then we have

$$\hat{T}_{i,1} = T_{i,1} \prod_{j \in Q_i} Q_{i,j}^{\mathrm{w_j}} = (\hat{h} \prod_{j=1}^{n} \hat{u}^{\mathrm{w_j}})^{-d_2 t_{i,1}},$$

$$\hat{T}_{i,2} = T_{i,2} \prod_{j \in Q_i} (Q'_{i,j})^{\mathrm{w_j}} = (\hat{h} \prod_{j=1}^{n} \hat{u}^{\mathrm{w_j}})^{-d_1 t_{i,1}},$$

$$\hat{T}_{i,3} = T_{i,3} \prod_{j \in Q_i} (Q''_{i,j})^{\mathrm{w_j}} = (\hat{h} \prod_{j=1}^{n} \hat{u}^{\mathrm{w_j}})^{-d_4 t_{i,2}},$$

$$\hat{T}_{i,4} = T_{i,4} \prod_{j \in Q_i} (Q'''_{i,j})^{\mathrm{w_j}} = (\hat{h} \prod_{j=1}^{n} \hat{u}^{\mathrm{w_j}})^{-d_3 t_{i,2}},$$

(2) $I_{\mathbb{M},\rho}$ is a set of minimum subsets satisfied $(\mathbb{M}, \rho)$, it then checks whether there is an $\mathcal{I} \in I_{\mathbb{M},\rho}$ statisfying

$$\begin{aligned}
& e(C', \prod_{i \in \mathcal{I}} D_i^{\omega_i}) e(C'', \prod_{i \in \mathcal{I}} (\frac{R_i}{H_2(e(D, \hat{D})^{\kappa})})^{\omega_i}) e(E_1, \prod_{i \in \mathcal{I}} (\hat{T}_{i,1})^{\omega_i}) e(E_2, \prod_{i \in \mathcal{I}} (\hat{T}_{i,2})^{\omega_i}) \\
& \qquad \cdot e(E_3, \prod_{i \in \mathcal{I}} (\hat{T}_{i,3})^{\omega_i}) e(E_4, \prod_{i \in \mathcal{I}} (\hat{T}_{i,4})^{\omega_i}) \\
& = e(g^{\mu}, \prod_{i \in \mathcal{I}} (\hat{g}^{\lambda_i} \hat{\delta}^{d_1 d_2 t_{i,1} + d_3 d_4 t_{i,2}})^{\omega_i}) \cdot e(\delta^{-\mu} (h \prod_{i=1}^{n} u^{\mathrm{w_i}})^{s}, \prod_{i \in \mathcal{I}} (\hat{g}^{d_1 d_2 t_{i,1} + d_3 d_4 t_{i,2}})^{\omega_i}) \\
& \qquad e(g_1^{s-s_1}, (\hat{h} \prod_{j=1}^{n} \hat{u}^{\mathrm{w_j}})^{-d_2 t_{i,1} w_i}) e(g_2^{s_1}, (\hat{h} \prod_{j=1}^{n} \hat{u}^{\mathrm{w_j}})^{-d_1 t_{i,1} w_i}) \\
& \qquad \cdot e(g_3^{s-s_2}, (\hat{h} \prod_{j=1}^{n} \hat{u}^{\mathrm{w_j}})^{-d_4 t_{i,2} w_i}) e(g_4^{s_2}, (\hat{h} \prod_{j=1}^{n} \hat{u}^{\mathrm{w_j}})^{-d_3 t_{i,2} w_i}) \\
& = e(g^{\mu}, \prod_{i \in \mathcal{I}} \hat{g}^{\lambda_i \omega_i}) e(g^{\mu}, \prod_{i \in \mathcal{I}} (\hat{\delta}^{d_1 d_2 t_{i,1} + d_3 d_4 t_{i,2}})^{\omega_i}) \\
& \qquad \cdot e(\delta^{-\mu}, \prod_{i \in \mathcal{I}} (\hat{g}^{d_1 d_2 t_{i,1} + d_3 d_4 t_{i,2}})^{\omega_i}) e((h \prod_{i=1}^{n} u^{\mathrm{w_i}})^{s}, \prod_{i \in \mathcal{I}} \hat{g}^{(d_1 d_2 t_{i,1} + d_3 d_4 t_{i,2}) \omega_i}) \\
& \quad e(g^{s d_1}, (\hat{h} \prod_{j=1}^{n} \hat{u}^{\mathrm{w_j}})^{-d_2 t_{i,1} w_i}) e(g^{-d_1 s_1}, (\hat{h} \prod_{j=1}^{n} \hat{u}^{\mathrm{w_j}})^{-d_2 t_{i,1} w_i}) e(g^{d_2 s_1}, (\hat{h} \prod_{j=1}^{n} \hat{u}^{\mathrm{w_j}})^{-d_1 t_{i,1} w_i}) \\
& \quad e(g^{s d_3}, (\hat{h} \prod_{j=1}^{n} \hat{u}^{\mathrm{w_j}})^{-d_4 t_{i,2} w_i}) e(g^{-d_3 s_2}, (\hat{h} \prod_{j=1}^{n} \hat{u}^{\mathrm{w_j}})^{-d_4 t_{i,2} w_i}) e(g^{d_4 s_2}, (\hat{h} \prod_{j=1}^{n} \hat{u}^{\mathrm{w_j}})^{-d_3 t_{i,2} w_i}) \\
& = e(g, \hat{g})^{\alpha \mu} = C
\end{aligned}$$

### 4.2 Security Proof

**Theorem 1.** Under the asymmetric decisional DBDH assumption, the asymmetric decisional q-BDHE assumption and the asymmetric decisional linear assumption, our scheme is selectively indistinguishable against chosen keyword-set attacks (selectively IND-CKA).

*Proof.* The proof is divided into two parts, depending on the role of the adversary. In the first part, the adversary is assumed to be an outside attacker, and in the second part, the adversary is assumed to be the cloud sever who performs search operations. The proof details will be given in the full version of the paper due to space limit.

## 5 Comparison

To specifically highlight the contributions of our research work, we compare our scheme with three related works, namely [14,28,48]. Lai *et al.* [28] is an expressive searchable encryption protocol built in composite order group, while [14,48] are expressive searchable encryption schemes with prime order group. Below, we compare the above schemes in terms of communication cost, computation cost, features and security. In [48], $S$ is the number of the data user's attributes, and $N$ is the number of attributes that are involved in the data owner's access control policy. Let $|par|$, $|msk|$, $|T_{M,\rho}|$, $|M|$ be the size of the public parameter, the master private key, the trapdoor and the access structure, respectively. We let $|\mathbb{G}|$, $|\hat{\mathbb{G}}|$, and $|\mathbb{G}_T|$ denote the size of the element in $\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T$, respectively. Let $l$ be the number of keywords in an access structure, $n$ be the maximum number of keywords in the system, and $m$ be the size of a keyword set associated with a ciphertext. Denote $E$ as an exponentiation operation, $P$ as a pairing operation, $x_1$ as the number of elements in $I_{\mathbb{M},\rho} = \{\mathcal{I}_1, ..., \mathcal{I}_{x_1}\}$, $x_2$ as $|\mathcal{I}_1| + \cdots + |\mathcal{I}_{x_1}|$.

**Table 1.** Storage and communication overhead comparison

| | Public parameter | Master private key | Trapdoor | Ciphertext | Bilinear group |
|---|---|---|---|---|---|
| [28] | $n+5$ | $n+4$ | $2l+\|M\|$ | $m+2$ | Composite |
| [48] | 5 | 3 | $N+3$ | $S+3$ | Prime |
| [14] | 9 | 5 | $6l+\|M\|$ | $5m+2$ | Prime |
| Ours | 10 | 8 | $(4n+2)l+\|M\|$ | 7 | Asymmetric prime |

From Table 1, it can be seen that only [28] is built on composite order group, suffering from the heaviest communication cost (with linear cost in all metrics), while others are in prime order group. According to [22], prime order group have clear advantage in the parameter size over composite order group pairing-friendly elliptic curves. Although being constructed in prime order group, [14,48] come at $O(S)$ and $O(m)$ price in ciphertext storage/communication. However, ours only requires constant value in the same metric. The reason behind the constant cost (in our construction) relies on the "aggregation" of ciphertext components, aggregating keyword set as a whole (much like some technique used in hierarchical IBE). We note that the size of trapdoor in our scheme is bound at $O(nl)$. This seems as a trade-off between reducing the cost in ciphertext and (meanwhile) enlarging the size of trapdoor. However, we here state that increasing the size of trapdoor does bring efficiency in test phase. We will discuss this in the next paragraph.

Table 2 shows that our scheme only requires constant pairing cost ($7P$) in test phase, while others are restricted to linear pairing cost. In the same metric, the exponentiation cost of our scheme maintains the same magnitude as that of others. Except [48], pairing computation exists in the encryption phase of all other schemes. Compared to [28] (with composite order group), our scheme may enjoy around 50 times faster in pairings (if [28] equips a 1024-bit composite order elliptic curve) [19]. The decryption techniques used in [14,48] drag down the efficiency of decryption. This is so because the pairings mainly depend on the size of attribute set, in particular, an attribute needs one pairing computation.

**Table 2.** Computation cost comparison

| | Trapdoor | Enc | Test |
|---|---|---|---|
| [28] | $4lE$ | $2(m+1)E+P$ | $\leq 2x_2P+x_2E$ |
| [48] | $(2N+2)E$ | $(S+4)E$ | $(2S+2)P+SE_T$ |
| [14] | $(16l+1)E$ | $(7m+2)E+P$ | $\leq (6x_2+1)P+(x_2+1)E$ |
| Ours | $(15l+1)E$ | $(n+6)E+P$ | $\leq 7P+(x_2+1)E$ |

However, we employ "fast decryption" technology and auxiliary components $Q_{i,j}$ into our construction, so that the test algorithm are free of linear cost, namely, the efficiency of the test algorithm is not restricted to the size of attribute set.

We show the feature and security comparison in Table 3. We use KGA to denote keyword guessing attacks. It is clear to see that our scheme supports any monotonic assess structure while others only provide AND and OR level of expressiveness. Enjoying more expressiveness, our scheme maintains the same security level with Cui *et al.*'s scheme [14]. Zheng *et al.* [48] opted to use an authenticated private channel to eliminate the keyword guessing attacks. However, it may not be scalable in practice. To enable publicly trapdoor delivery, [14] and our schemes slightly degrade the keyword privacy level to only allow a designated server to launch KGA. We state that our scheme is the first of its type, in the literature, to provide security and expressiveness simultaneously without significantly jeopardizing the efficiency. It is worthy of mentioning that the generation/computation cost of trapdoor (in our scheme) can be further off-loaded to the a trusted party holding the master private key (because of our sophisticated construction technique), so that system user can enjoy lighter computation complexity.

**Table 3.** Property and security comparison

| | Expressiveness | Security | Trapdoor delivery |
|---|---|---|---|
| [28] | AND, OR | Adaptive chosen keyword attacks in standard model | Public channel |
| [48] | AND, OR | Selective security against chosen-keyword attack in ROM | Authenticated private channel |
| [14] | Any monotonic access structure | Selective indistinguishability against chosen keyword set attack in standard model | Public channel |
| Ours | Any monotonic access structure | Selective indistinguishability against chosen keyword set attack in standard model | Public channel |

## 6 Conclusions

Attribute-based keyword search has attracted many attentions since it can support secure search over encrypted data with expressive access structure. Nevertheless, the size of ciphertexts but also the pairing cost (incurred in the test phase) are linear in the number of keyword. That is the main drawback of the most of the existing searchable encryption systems with ABE. To tackle the above opened problem, we propose a new KP-ABES scheme with outstanding features, namely expressive access structures, constant size ciphertext, and constant pairing cost (in search). There are some interesting open problems brought by this research work as well, for example, how to reduce the size of search trapdoor, and how to renew/provoke attribute.

**Acknowledgements.** The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. This work is supported by the National Natural Science Foundation of China (61572303), the Summit of the Six Top Talents Program of Jiangsu Province (Grant No. 2015-DZXX-020). The National Research Foundation, Prime Minister Office, Singapore under its Corporate Laboratory@University Scheme, National University of Singapore, and Singapore Telecommunications Ltd.

## References

1. Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable encryption revisited: consistency properties, relation to anonymous IBE, and extensions. J. Cryptol. **21**(3), 350–391 (2008)
2. Attrapadung, N., Libert, B., de Panafieu, E.: Expressive key-policy attribute-based encryption with constant-size ciphertexts. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 90–108. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19379-8_6
3. Benabbas, S., Gennaro, R., Vahlis, Y.: Verifiable delegation of computation over large datasets. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 111–131. Springer, Heidelberg (2011). doi:10.1007/978-3-642-22792-9_7
4. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: S&P 2007, pp. 321–334. IEEE Computer Society (2007)
5. Boneh, D., Boyen, X.: Efficient selective identity-based encryption without random oracles. J. Cryptol. **24**(4), 659–693 (2011)
6. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005). doi:10.1007/11426639_26
7. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004). doi:10.1007/978-3-540-28628-8_3
8. 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). doi:10.1007/978-3-540-24676-3_30

9. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006). doi:10.1007/11818175_17
10. Chase, M.: Multi-authority attribute based encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 515–534. Springer, Heidelberg (2007). doi:10.1007/978-3-540-70936-7_28
11. Chase, M., Chow, S.S.M.: Improving privacy and security in multi-authority attribute-based encryption. In: CCS 2009, pp. 121–130. ACM (2009)
12. Cheung, L., Newport, C.C.: Provably secure ciphertext policy ABE. In: CCS 2007, pp. 456–465. ACM (2007)
13. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM J. Comput. **33**(1), 167–226 (2004)
14. Cui, H., Wan, Z., Deng, R., Wang, G., Li, Y.: Efficient and expressive keyword search over encrypted data in the cloud. IEEE Trans. Dependable Secure Comput. **PP**(99), 1 (2016)
15. Ducas, L.: Anonymity from asymmetry: new constructions for anonymous HIBE. In: Pieprzyk, J. (ed.) CT-RSA 2010. LNCS, vol. 5985, pp. 148–164. Springer, Heidelberg (2010). doi:10.1007/978-3-642-11925-5_11
16. Emura, K., Miyaji, A., Nomura, A., Omote, K., Soshi, M.: A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In: Bao, F., Li, H., Wang, G. (eds.) ISPEC 2009. LNCS, vol. 5451, pp. 13–23. Springer, Heidelberg (2009). doi:10.1007/978-3-642-00843-6_2
17. Fiore, D., Gennaro, R.: Publicly verifiable delegation of large polynomials and matrix computations, with applications. In: CCS 2012, pp. 501–512. ACM (2012)
18. Freeman, D., Scott, M., Teske, E.: A taxonomy of pairing-friendly elliptic curves. J. Cryptol. **23**(2), 224–280 (2010)
19. Freeman, D.M.: Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 44–61. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13190-5_3
20. Galbraith, S.D., Paterson, K.G., Smart, N.P.: Pairings for cryptographers. Discrete Appl. Math. **156**(16), 3113–3121 (2008)
21. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: CCS 2006, pp. 89–98. ACM (2006)
22. Guillevic, A.: Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In: Jacobson, M., Locasto, M., Mohassel, P., Safavi-Naini, R. (eds.) ACNS 2013. LNCS, vol. 7954, pp. 357–372. Springer, Heidelberg (2013). doi:10.1007/978-3-642-38980-1_22
23. Herranz, J., Laguillaumie, F., Ràfols, C.: Constant size ciphertexts in threshold attribute-based encryption. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 19–34. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13013-7_2
24. Hohenberger, S., Waters, B.: Attribute-based encryption with fast decryption. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 162–179. Springer, Heidelberg (2013). doi:10.1007/978-3-642-36362-7_11
25. Khader, D.: Public key encryption with keyword search based on K-resilient IBE. In: Gavrilova, M., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganá, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3982, pp. 298–308. Springer, Heidelberg (2006). doi:10.1007/11751595_33
26. Lai, J., Deng, R.H., Li, Y.: Expressive CP-ABE with partially hidden access structures. In: ASIACCS 2012, pp. 18–19. ACM (2012)

27. Lai, J., Deng, R.H., Li, Y., Weng, J.: Fully secure key-policy attribute-based encryption with constant-size ciphertexts and fast decryption. In: ASIACCS 2014, pp. 239–248. ACM (2014)
28. Lai, J., Zhou, X., Deng, R.H., Li, Y., Chen, K.: Expressive search on encrypted data. In: ASIACCS 2013, pp. 243–252. ACM (2013)
29. Lewko, A., 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). doi:10.1007/978-3-642-13190-5_4
30. Lewko, A., Waters, B.: Decentralizing attribute-based encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 568–588. Springer, Heidelberg (2011). doi:10.1007/978-3-642-20465-4_31
31. Lewko, A., Waters, B.: New proof methods for attribute-based encryption: achieving full security through selective techniques. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 180–198. Springer, Heidelberg (2012). doi:10.1007/978-3-642-32009-5_12
32. Li, X., Liang, K., Liu, Z., Wong, D.S.: Attribute-based encryption: traitor tracing, revocation and fully security on prime order groups. In: CLOSER 2017, pp. 281–292. SciTePress (2017)
33. Li, Y., Liang, K., Su, C., Wu, W.: DABEHR: decentralized attribute-based electronic health record system with constant-size storage complexity. In: Au, M.H.A., Castiglione, A., Choo, K.-K.R., Palmieri, F., Li, K.-C. (eds.) GPC 2017. LNCS, vol. 10232, pp. 611–626. Springer, Cham (2017). doi:10.1007/978-3-319-57186-7_44
34. Liang, K., Huang, X., Guo, F., Liu, J.K.: Privacy-preserving and regular language search over encrypted cloud data. IEEE Trans. Inf. Forensics Secur. **11**(10), 2365–2376 (2016)
35. Liang, K., Su, C., Chen, J., Liu, J.K.: Efficient multi-function data sharing and searching mechanism for cloud-based encrypted data. In: ASIACCS 2016, pp. 83–94. ACM (2016)
36. Liang, K., Susilo, W.: Searchable attribute-based mechanism with efficient data sharing for secure cloud storage. IEEE Trans. Inf. Forensics Secur. **10**(9), 1981–1992 (2015)
37. Liu, J.K., Au, M.H., Susilo, W., Liang, K., Lu, R., Srinivasan, B.: Secure sharing and searching for real-time video data in mobile cloud. IEEE Netw. **29**(2), 46–50 (2015)
38. Lv, Z., Hong, C., Zhang, M., Feng, D.: Expressive and secure searchable encryption in the public key setting. In: Chow, S.S.M., Camenisch, J., Hui, L.C.K., Yiu, S.M. (eds.) ISC 2014. LNCS, vol. 8783, pp. 364–376. Springer, Cham (2014). doi:10.1007/978-3-319-13257-0_21
39. Ning, J., Cao, Z., Dong, X., Wei, L.: Traceable and revocable CP-ABE with shorter ciphertexts. Sci. China Inf. Sci. **59**(11), 119102:1–119102:3 (2016)
40. Ning, J., Dong, X., Cao, Z., Wei, L., Lin, X.: White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes. IEEE Trans. Inf. Forensics Secur. **10**(6), 1274–1288 (2015)
41. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: CCS 2007, pp. 195–203. ACM (2007)
42. Papamanthou, C., Shi, E., Tamassia, R.: Signatures of correct computation. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 222–242. Springer, Heidelberg (2013). doi:10.1007/978-3-642-36594-2_13
43. Rhee, H.S., Park, J.H., Susilo, W., Lee, D.H.: Improved searchable public key encryption with designated tester. In: ASIACCS 2009, pp. 376–379. ACM (2009)

44. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005). doi:10.1007/11426639_27
45. Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: S&P 2000, pp. 44–55. IEEE Computer Society (2000)
46. Wang, S., Liang, K., Liu, J.K., Chen, J., Jianping, Y., Xie, W.: Attribute-based data sharing scheme revisited in cloud computing. IEEE Trans. Inf. Forensics Secur. **11**(8), 1661–1673 (2016)
47. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005). doi:10.1007/11426639_7
48. Zheng, Q., Shouhuai, X., Giuseppe Ateniese, V.: VABKS: verifiable attribute-based keyword search over outsourced encrypted data. In: INFOCOM 2014, pp. 522–530. IEEE (2014)