Fast Keyword Search over Encrypted Data with Short Ciphertext in Clouds

Yi-Fan Tseng[a], Chun-I Fan[b,c,d,*], Zi-Cheng Liu[b]

[a] *Department of Computer Science, National Chengchi University, Taipei, Taiwan.*
[b] *Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan*
[c] *Information Security Research Center, National Sun Yat-sen University, Kaohsiung, Taiwan*
[d] *Intelligent Electronic Commerce Research Center, National Sun Yat-sen University, Kaohsiung, Taiwan*

Abstract

Nowadays, it is convenient for people to store their data on clouds. To protect the privacy, people tend to encrypt their data before uploading them to clouds. Due to the widespread use of cloud services, public key searchable encryption is necessary for users to search the encrypted files efficiently and correctly. However, the existing public key searchable encryption schemes supporting monotonic queries suffer from either infeasibility in keyword testing or inefficiency such as heavy computing cost of testing, large size of ciphertext or trapdoor, and so on. In this work, we first propose a novel and efficient anonymous key-policy attribute-based encryption (KP-ABE). Then by applying Shen *et al.*'s generic construction proposed to the proposed anonymous KP-ABE, we obtain an efficient and expressive public key searchable encryption, which to the best of our knowledge achieves the best performance in testing among the existing such schemes. Only 2 pairings is needed in testing. Besides, we also implement our scheme and others with Python for comparing the performance. From the implementation results, our scheme owns the best performance on testing, and the size of ciphertexts and trapdoors are smaller than most of the

*Corresponding author
Email addresses: yftseng@cs.nccu.edu.tw (Yi-Fan Tseng), cifan@mail.cse.nsysu.edu.tw (Chun-I Fan), tcdiadem@gmail.com (Zi-Cheng Liu)

existing schemes.

Keywords: Public Key Searchable Encryption, Key-Policy Attribute-Based Encryption, Anonymous KP-ABE, The Standard Model, Monotonic Access Structure

1. Introduction

Cloud computing has been thriving around the world recently. People tend to store their data on clouds so that they can back up the data and retrieve them anytime and anywhere. Consider the following scenario. In a company, employees are asked to store the commercial documents in the company's private cloud. In order to prevent unauthorized access, it is necessary to store the documents in encrypted form. Besides, the documents may come from customers, and thus they should be transmitted in encrypted form. This is a common business model of nowadays. In such scenario, it is significant for the employees to efficiently and securely search the required encrypted files. A practical solution to this problem is to apply searchable encryption.

1.1. Related Works

In 2000, Song et al. [1] first gave the definition of searchable encryption (SE). In an SE scheme, a data owner can encrypt keywords and upload it with the encrypted data so that users can find the desired data by searching the encrypted keywords. In 2004, Boneh et al. [2] first proposed a public key encryption with keyword search (PEKS) (a.k.a. public key searchable encryption). They combined the public key setting and the keyword search encryption, and discussed the relationship between PEKS and identity-based encryption (IBE). Note that public key searchable encryption is different from private key searchable encryption (a.k.a. searchable symmetric encryption) [3]. The former belongs to the family of public key primitives, where an encryptor is allowed to be different to the owner of private key, while in a private key searchable encryption, the

one who encrypts the data must be the same as the one who is able to decrypt the data. In this manuscript, we focus on solving the emerging problems in the realm of PEKS. Following Boneh's pioneering work, Abdalla et al. [4] proposed a generic construction of PEKS from anonymous IBE, where an encryption reveals nothing about its receiver. However, [4, 2] only support equality queries. It is necessary to construct an PEKS scheme with more expressive queries such as conjunction, disjunction and monotonic formulas to make the search more accurate and flexible. In 2007, Boneh et al. [5] proposed a searchable encryption scheme supporting conjunctive, subset, and range queries in public key setting. In the next year, Katz et al. [6] first introduced inner-product predicate encryption (IPE) [7, 8] which can be extended to PEKS supporting disjunctive queries. Nevertheless, it is inefficient in this way because the size of the ciphertext and the search token would superpolynomially blow up [9]. Another shortcoming of Katz et al.'s work is that, their scheme is constructed under composite-order bilinear groups whose performance is notoriously worse than prime-order bilinear groups. According to [10], the length of a group element in a composite-order group is 12 times larger than that in a prime-order group. Besides, the bilinear pairings in composite-order groups are 254 times slower than those in prime-order groups for the same 128-bit security. In 2018, there are several related works [11], [12],[13],[14] that have been proposed. In these schemes, the authors explored keyword search in attribute-based encryption so that the scheme can support access control and keyword search simultaneously. However, these schemes cannot support expressive search queries. They only support searching on single keyword. To achieve expressive queries (i.e. conjunction/disjunction of keywords) is significant for cloud applications due to its convenience. Fig 1 shows an example for PEKS supporting expressive queries. Any user can upload an encrypted file tagged with some keywords to a cloud service provider. Users can also make some queries of the conjunction/disjunction for some keywords to search the files they want.

To achieve more expressive queries, Lai et al. [9] proposed a PEKS scheme

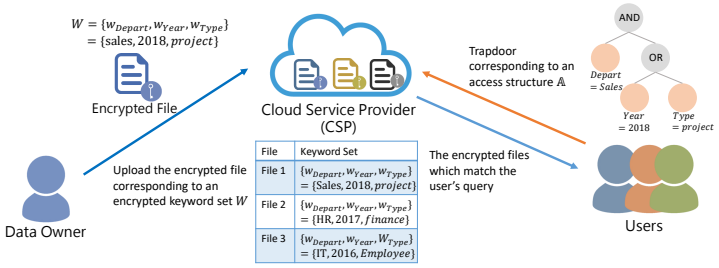

Figure 1: Example for the Application of PEKS Supporting Expressive Queries

motivated from Lewko et al.'s [15] key-policy attribute-based encryption (KP-ABE) in 2013. In 2012 Han et al. [16, 17] proposed a generic construction for attribute-based encryption with keyword search (ABEKS). In 2020, Shen *et al.* [18] further gave a generic construction for building PEKS from anonymous KP-ABE. Note that the notion of ABEKS is different from PEKS. The former needs a trusted third party for issuing attribute keys to users, while in a PEKS scheme, users generate their public/secret key by themselves. However, there exists a common problem of these schemes [9, 17, 18, 19]. The test algorithm in these schemes will not be conducted successfully. For instance, the test algorithm in [9] needs to compute the following formula.

$$\hat{C} = \prod_{i \in \mathcal{I}} \left(\frac{e(C_0, K_{1,i})}{e(C_{\rho(i)}, K_{2,i})} \right)^{\omega_i} \tag{1}$$

In formula 1, $K_{1,i}, K_{2,i}$ represent the search token associated with keyword $\rho(i)$ and $C_{\rho(i)}$ represents the ciphertext component associated with keyword $\rho(i)$, where ρ is a map from indices to keywords. We can observe that it is necessary to know the correlation between $(K_{1,i}, K_{2,i})$ and $C_{\rho(i)}$, which implies that the test algorithm needs to know the corresponding keyword of $C_{\rho(i)}$. If the underlying KP-ABE is anonymous, it will be infeasible in conducting test algorithm. The test algorithms in [16, 17, 18, 19, 20] are similar to that in [9], and thus they suffer from the same problem.

In order to solve the correlation problem, in 2018, Cui et al. [21] proposed

a PEKS scheme with weaker anonymity notion. They separate a keyword into a keyword name and a keyword value. For instance, in the case of ("gender" = "female"), "gender" is the keyword name and "female" is the keyword value. The weaker anonymity in [21] only guarantees that a ciphertext reveals nothing on its keyword values, while the keyword names are attached to the ciphertext. In the same year, Meng et al. [22] improved the efficiency of [21] by aggregating the ciphertext components for each attribute into a group element. However, their Test algorithm requires that all attributes in a ciphertext should appear in the access structure, or it would fail. Besides, there exists a common problem in [21, 22]. That is their schemes need an online and trusted third party to generate search tokens which is an unreasonable assumption in cryptography.

1.2. Contribution

In this work, we aim at proposing an efficient PEKS supporting expressive search queries. Due to [18], we have a new approach to build a PEKS scheme. Therefore, we first propose a novel anonymous KP-ABE with provably security. Then, by adopting the generic construction shown in [18], we obtain a novel PEKS from KP-ABE supporting monotonic access structure with the following advantages.

1. **Expressive queries:** The proposed scheme supports monotonic formula in search queries.
2. **High efficiency:** The proposed scheme is constructed under prime-order bilinear groups. Moreover, the pairings performed in the Test algorithm is independent of the number of attributes in ciphertexts and search tokens. Besides the length of ciphertexts in the proposed scheme is shorter than most of the existing schemes.
3. **Formal security proof:** The proposed scheme is proven to be fully secure in the standard model.

2. Preliminaries

In this section, we introduce the related formal definitions of public key searchable encryption and other preliminaries.

2.1. Bilinear Mapping

In this subsection, we will introduce the definition of bilinear mappings.

Definition 2.1. Let $\mathbb{G}, \hat{\mathbb{G}}$ and $\mathbb{G}_T$ be all multiplicative cyclic groups of prime order p.
A bilinear mapping $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ satisfies the following properties in which $g, \hat{g}$ are generators of $\mathbb{G}, \hat{\mathbb{G}}$, respectively.

- **Bilinearity:** $e(h^a, \hat{h}^b) = e(h, \hat{h})^{ab}$, $\forall h \in \mathbb{G}, \hat{h} \in \hat{\mathbb{G}}$ and $a, b \in \mathbb{Z}_p$.
- **Non-Degeneracy:** There exist $h \in \mathbb{G}$ and $\hat{h} \in \hat{\mathbb{G}}$ such that $e(h, \hat{h}) \neq 1$.
- **Computability:** There exists an efficient algorithm to compute $e(h, \hat{h}), \forall h \in \mathbb{G}, \hat{h} \in \hat{\mathbb{G}}$.

2.2. The DBDH-3 Problem

In this subsection, we will introduce the definition of the DBDH-3 problem shown in [23], which is a variant of the decisional bilinear Diffie-Hellman in asymmetric paring groups.

Definition 2.2. Given $(g, \hat{g}, g^a, g^b, g^c, \hat{g}^a, \hat{g}^b, Y)$, where $a, b, c \xleftarrow{\$} \mathbb{Z}_p$, decide whether $Y = e(g, \hat{g})^{abc}$ or a random element in $\mathbb{G}_T$.

We say that an algorithm $\mathcal{B}$ that outputs a bit has the advantage ϵ in solving the DBDH problem if

$$\left| \Pr[\mathcal{B}(g, \hat{g}, g^a, g^b, g^c, \hat{g}^a, \hat{g}^b, e(g, \hat{g})^{abc}) = 1] - \Pr[\mathcal{B}(g, \hat{g}, g^a, g^b, g^c, \hat{g}^a, \hat{g}^b, Y \xleftarrow{\$} \mathbb{G}_T) = 1] \right| \geq \epsilon.$$

2.3. Linear Secret-Sharing Scheme (LSSS)

We adapt the definition from those given in [24].

Definition 2.3. [24] A secret-sharing scheme $\prod$ 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 $\mathbb{M}$ with ℓ rows and n columns called the share-generation matrix for $\prod$, which can be computed from the access structure $\mathbb{A}$ of attribute names. For the i-th row of $\mathbb{M}$, $i = 1, \ldots, \ell$, we let the function ρ define the party labelling row i as $\rho(i)$ which maps the row i to an attribute name. We consider a column vector $\vec{v} = (s, r_2, \ldots, r_n)$, where s is the value to be shared, and $r_2, ..., r_n \in \mathbb{Z}_p$ are randomly chosen, and thus $\mathbb{M}\vec{v}$ is the vector of ℓ shares of the value s according to $\prod$. The share $\lambda_i = \mathbb{M}_i \vec{v}^\top$ belongs to party $\rho(i)$, where $\mathbb{M}_i$ is the i-th row of $\mathbb{M}$.

According to [24], every linear secret sharing-scheme satisfying the above definitions also enjoys the linear reconstruction property. Let S be an authorized set and $I = \{i : \rho(i) \in S\} \subseteq \{1, 2, \ldots, \ell\}$. Then, there exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$, such that $\sum_{i \in I} \omega_i \lambda_i = s$.

2.4. Access Structure

In this subsection, we will introduce the definition of access structures used in the proposed scheme.

Definition 2.4. An access structure $\mathbb{A}$ in our scheme conatins an $(\mathbb{M}, \rho)$ corresponding to attribute names and a set $L = (z_{\rho(1)}, \ldots, z_{\rho(\ell)})$ corresponding to attribute values of $\rho(1), \ldots, \rho(\ell)$, respectively. Given an access structure $\mathbb{A} = (\mathbb{M}, \rho, L)$ a set $S = (v_1, \ldots, v_t)$ corresponding to the values of attribute names $1, \ldots, t$, respectively, we say that S satisfies $\mathbb{A}$ (denoted by $S \preceq \mathbb{A}$), if:

- there exists an index set $I = \{i : \rho(i) \in [1, t]\}$, such that, there exsit constants $\{\omega_i\}_{i \in I}$ satisfing $\sum_{i \in I} \omega_i \mathbb{M}_i = (1, 0, \ldots, 0)$;
- for $i \in I$, $v_{\rho(i)} = z_{\rho(i)}$.

2.5. Public Key Searchable Encryption

A public key searchable encryption [5, 9] consists of four algorithms.

- **Setup**(1^λ) : Take as input a security parameter λ. It outputs a public/secret key pair (PK, SK).
- **Encrypt**(PK, W) : Take as inputs the public key PK and a keyword set W. It outputs a ciphertext CT_W.
- **Trapdoor**$(PK, SK, \mathcal{P})$: Take as inputs the public key PK, the secret key SK and a predicate $\mathcal{P}$. It outputs a search token $TK_{\mathcal{P}}$. Note that the search tokens are also known as trapdoors in the literatures. In this work, we sometimes use "trapdoor" to denote a search token.
- **Test**$(PK, TK_{\mathcal{P}}, CT_W)$: Take as inputs the public key, a search token $TK_{\mathcal{P}}$ and a ciphertext CT_W. If the keyword set W satisfies the predicate $\mathcal{P}$, the algorithm outputs 1; otherwise, outputs 0.

The predicate supported by our scheme is monotonic formula represented by linear secret sharing schemes. Next we show the definition for public key searchable encryption, called IND-CKA security (i.e. indistinguishability against chosen keyword attacks). The notion states that a ciphertext in an SE scheme reveals no information about its keywords.

Definition 2.5. (IND-CKA Security)

- Setup: A challenger runs the **Setup** algorithm and gives the public parameters to the adversary.
- Phase 1: The adversary is allowed to issue polynomially many queries for trapdoors T with access structures $\mathbb{A}_j$'s.
- Challenge: The adversary submits two equal-size keywords sets W_0^* and W_1^*. The sets W_0^* and W_1^* should not satisfy any trapdoor that has been queried in Phase 1. The challenger flips a random coin b, and gnerate a ciphertext C^* with W_b^*. The ciphertext C^* is passed to the adversary.

- Phase 2: The adversary repeats the steps in Phase 1.

- Guess: The adversary outputs the guess $b' \in \{0, 1\}$ of b and wins the game if $b' = b$.

The advantage of the adversary in this game is defined as $Adv_{\mathcal{A}}^{IND\text{-}CKA} = |Pr[b' = b] - \frac{1}{2}|$. A PEKS scheme is said to be semantically secure if for every polynomial-time adversary $\mathcal{A}$, $Adv_{\mathcal{A}}^{IND\text{-}CKA}$ is at most negligible.

Remark 1. There is another security notion called "keyword privacy", which is analogous to the notion "function-private" [25] in functional encryption. Keyword privacy states that a trapdoor reveals no information about its keywords (or policy). However, as that stated in [20], it is impossible to achieve for PEKS. The reason is obvious, i.e. given a trapdoor, anyone can generate a ciphertext for any keywords under her/his choice, and thus reveal the information of the given trapdoor by performing the Test algorithm with the trapdoor and the ciphertext. Since our goal is to solve the problems for PEKS, we only focus on the IND-CKA security of the proposed scheme.

2.6. Definition and Security Model for Key-Policy Attribute-Based Encryption

KP-ABE was first proposed by Goyal et al. in [26], which is the dual construction of ciphertext-policy attribute-based encryption (CP-ABE) [27, 28]. A KP-ABE consists of the following four algorithms.

- **Setup**(1^λ) : The input is the security parameter 1^λ. The algorithm outputs the public parameter PK and the master secret key MSK.

- **KeyGen**$(PK, MSK, \mathbb{A})$: The inputs are the public parameter PK, master secret key MSK, and the access structure $\mathbb{A}$ which is assigned by Key Generation Center (KGC) to the user. The algorithm outputs a decryption key $SK_{\mathbb{A}}$ which contains the information of access structure.

- **Encrypt**(PK, S, M) : The inputs are the public parameter $param$, a set of descriptive attributes S, and a message M. The algorithm outputs a ciphertext CT.

- **Decrypt**$(CT, SK_{\mathbb{A}})$: This algorithm is run by the receiver. The inputs are a ciphertext CT which was encrypted under the set of attributes S, and the decryption key $SK_{\mathbb{A}}$ for access structure $\mathbb{A}$. The algorithm outputs the message M if $S \preceq \mathbb{A}$.

Next we give the security notion of KP-ABE. There are two security notions for KP-ABE, IND-CPA security and ANON-CPA security. The IND-CPA security defines that a ciphertext does not reveal any information about the encrypted message, and the ANON-CPA defines that a ciphertext reveals nothing to the attribute set.

Definition 2.6. (IND-CPA Security)

We provide the IND-CPA security model for a KP-ABE scheme.

- Setup: A challenger runs the **Setup** algorithm to generate public parameters and master secret key, and gives the public parameters to the adversary.

- Phase 1: The adversary is allowed to issue polynomially many queries for private keys with access structures $\mathbb{A}_j$.

- Challenge: The adversary submits two equal-length messages M_0 and M_1 along with an attribute set S^*, where S^* should not satisfy any access structure $\mathbb{A}_j$ queried in Phase 1. The challenger flips a random coin b, and encrypts M_b with $\mathbb{A}^*$. The ciphertext is passed to the adversary.

- Phase 2: The adversary repeats the steps in Phase 1.

- Guess: The adversary outputs the guess $b' \in \{0, 1\}$ of b and wins the game if $b' = b$.

The advantage of the adversary in this game is defined as $Adv_{\mathcal{A}}^{IND-CPA} = |Pr[b' = b] - \frac{1}{2}|$. A KP-ABE scheme is said to be IND-CPA secure if for every polynomial-time adversary $\mathcal{A}$, $Adv_{\mathcal{A}}^{IND-CPA}$ is at most negligible.

Definition 2.7. (ANON-CPA Security)

We provide the ANON-CPA security models for a KP-ABE scheme.

- Setup: A challenger runs the **Setup** algorithm and gives the public parameters to the adversary.

- Phase 1: The adversary is allowed to issue polynomially many queries for private keys with access structures $\mathbb{A}_j$'s.

- Challenge: The adversary submits two equal-size sets S_0^* and S_1^* and a message M. The sets S_0^* and S_1^* should not satisfy any key that has been queried in Phase 1. The challenger flips a random coin b, and encrypts M with S_b^*. The ciphertext is passed to the adversary.

- Phase 2: The adversary repeats the steps in Phase 1.

- Guess: The adversary outputs the guess $b' \in \{0, 1\}$ of b and wins the game if $b' = b$.

The advantage of the adversary in this game is defined as $Adv_{\mathcal{A}}^{ANON-CPA} = |Pr[b' = b] - \frac{1}{2}|$. A KP-ABE scheme is said to be ANON-CPA secure if for every polynomial-time adversary $\mathcal{A}$, $Adv_{\mathcal{A}}^{ANON-CPA}$ is at most negligible.

The models can be easily extended to handle chosen-ciphertext attacks by allowing for decryption queries in Phase 1 and Phase 2.

2.7. Public Key Searchable Encryption from KP-ABE

In [18], Shen *et al.* give a generic construction of a public key searchable encryption scheme from a KP-ABE scheme supporting monotonic queries. The construction is an extension of that proposed in [2, 4], which builds a PEKS from a given identity-based encryption. Intuitively, the keywords can be regarded as attributes in KP-ABE. Additionally, we can regard the access structure as a search query associated with a trapdoor. Then, the KeyGen algorithm of KP-ABE can be performed as the Trapdoor algorithm of PEKS to generate a trapdoor for an access policy. The Encrypt algorithm of KP-ABE can be performed as the Encrypt algorithm of PEKS, the Decrypt algorithm of KP-ABE can be used for the Test algorithm of PEKS. More precisely, given a KP-ABE $\Pi = (Setup, Encrypt, KeyGen, Decrypt)$, a PEKS is given as follows.

- Setup(1^λ): Run $\Pi.Setup(1^\lambda) \rightarrow (PK, MSK)$, and output $(PK, SK) = (PK, MSK)$.

- Encrypt(PK, W): Choose a random message m, and compute $\Pi.Encrypt(PK, W, m) \rightarrow CT$. Output $C = (CT, m)$.

- Trapdoor($PK, SK, \mathbb{A}$): Generate a trapdoor $TK_{\mathbb{A}}$ for an access policy $\mathbb{A}$ as $TK_{\mathbb{A}} \leftarrow \Pi.KeyGen(SK, \mathbb{A})$.

- Test($PK, TK_{\mathbb{A}}, C = (CT, m)$): Output 1 if $m = \Pi.Decrypt(CT, TK_{\mathbb{A}})$; output 0 otherwise.

The correctness of the generic construction is easily derived from the correctness of the underlying KP-ABE. On the other hand, like the generic construction given in [2, 4], the security of the PEKS relies upon the anonymity of the underlying KP-ABE, since the transformation regards keywords as the attributes in the underlying KPABE. Next we show that the construction is secure based on the security of the underlying KP-ABE.

Theorem 2.1. If the KP-ABE is ANON-CPA secure, then the PEKS form the KP-ABE is IND-CKA secure.

The detailed proof of this theorem can be refered to [18].

3. The Proposed Anonymous KP-ABE and PEKS

In this section, we first give a new anonymous KP-ABE scheme, and then give an efficient PEKS scheme via the transformation shown in Section 2.7.

3.1. The Proposed KP-ABE

Let $\mathbb{G}$ and $\hat{\mathbb{G}}$ be two multiplicative groups of prime order p, and $e : \mathbb{G} \times \hat{\mathbb{G}} \rightarrow \mathbb{G}_T$ be the bilinear map. The details of our scheme are described as follows.

3.1.1. Setup

Setup$(1^\lambda) \to (PK, MSK)$. Given a security parameter 1^λ. To generate the system parameters, the KGC performs the following steps:

1. Generate generators g and $\hat{g}$ of $\mathbb{G}$ and $\hat{\mathbb{G}}$, respectively.
2. Choose a hash functions $H : \{0,1\}^* \to \mathbb{Z}_p$.
3. Randomly select $\phi \in \mathbb{Z}_p$, and compute $h = g^\phi$ and $\hat{h} = \hat{g}^\phi$.
4. Randomly select $\alpha, \beta \in \mathbb{Z}_p$, and compute $U = e(g,\hat{g})^{\alpha(\beta-1)}$ and $V = e(g,\hat{g})^{\alpha\beta}$.
5. The public parameters is $PK = \{\mathbb{G}, \hat{\mathbb{G}}, e, g, U, V, h, H\}$, and the master secret key is $MSK = \{\hat{g}, \hat{g}^\alpha, \hat{h}\}$. Keep the master secret key MSK secret.

3.1.2. KeyGen

KeyGen$(PK, MSK, \mathbb{A} = (M, \rho, L)) \to SK_{\mathbb{A}}$. The KGC takes as input the master secret key and an LSSS access structure $\mathbb{A} = (M, \rho, L)$. Let M be an $\ell \times n$ matrix. The function ρ associates rows of M to attribute names. Let $L = (z_{\rho(1)}, ..., z_{\rho(\ell)})$ be the attribute values corresponding to $\rho(1), \ldots, \rho(\ell)$, respectively. Let τ be the set of distinct attribute names existing in the access structure matrix M. To generate a secret key associated with the access structure $\mathbb{A}$, the KGC performs the following steps:

1. Select a random vector $\vec{v} = (s, y_2, ..., y_n)$ where $s = 1$.
2. Compute $\lambda_i = M_i \vec{v}^\top$ for $i = 1$ to ℓ, where M_i is the i-th row of M.
3. Select random numbers $r_i \in \mathbb{Z}_p$ for $i = 1$ to ℓ.
4. Compute $\hat{\sigma}_i = \hat{g}^{H(\rho(i)\|z_{\rho(i)})} \cdot \hat{h}$ for $i = 1$ to ℓ.
5. For $i = 1$ to ℓ, compute $d_{i,0} = \hat{g}^{\alpha \cdot \lambda_i} \cdot \hat{\sigma_i}^{r_i}, \forall j \in \tau/\rho(i), Q_{i,j} = \hat{\sigma_j}^{r_i}$, and $d_{i,1} = \hat{g}^{r_i}$.
6. The secret key is $SK = (\{d_{i,0}, d_{i,1}, \{Q_{i,j}\}_{j \in \tau/\rho(i)}\}_{i=1}^{\ell})$.

3.1.3. Encrypt

Encrypt$(PK, S, m) \to CT$. The algorithm takes as input the public parameters PK, a message $m \in \mathbb{G}_T$, and an attribute value set $S = (v_1, ..., v_t)$,

where v_i is the value of attribute name i. Let $\tilde{S}$ be the the set of attribute names from S. $\tilde{S}$ should be published with the ciphertext in order to decrypt. Note that the attribute values are not revealed to others. To generate a ciphertext of m associated with S, the algorithm performs the following steps:

1. Select a random number $k \in \mathbb{Z}_p$.
2. Compute $C_1 = V^k \cdot m = e(g, \hat{g})^{\alpha\beta k} \cdot m$.
3. Compute $C_2 = U^k = e(g, \hat{g})^{\alpha(\beta-1)k}$.
4. Compute $C_3 = g^k$.
5. Compute $\sigma_i = g^{H(i||v_i)} \cdot h$ for $i = 1$ to t.
6. Compute $C_{4,i} = {\sigma_i}^k$ for $i = 1$ to t.
7. The ciphertext is $CT = (C_1, C_2, C_3, \{C_{4,i}\}_{i=1}^t, \tilde{S})$.

3.1.4. Decrypt

Decrypt$(CT, SK_{\mathbb{A}})$. If the set of attribute names $\tilde{S}$ does not satisfy the access structure $\mathbb{A}$, it outputs $\perp$. Otherwise, let $I \subseteq \{1, 2, ..., \ell\}$ be a set of indices and $\{\omega_i\}_{i\in I} \in \mathbb{Z}_p$ be a set of constants such that $\forall i \in I, \rho(i) \in \tilde{S}$ and $\prod_{i\in I} \omega_i M_i = (1, 0, ..., 0)$. Then we define $\Delta = \{x : \exists i \in I, \rho(i) = x\}$ and $\hat{f}(\Delta) = \prod_{x\in\Delta} \hat{\sigma_x}, f(\Delta) = \prod_{x\in\Delta} \sigma_x$. For each $i \in I$, compute $\hat{d_{i,0}} = d_{i,0} \cdot \prod_{x\in\Delta/\rho(i)} Q_{i,x} = \hat{g}^{\alpha\cdot\lambda_i} \hat{f}(\Delta)^{r_i}$. Compute $L = \prod_{x\in\Delta} C_{4,x} = \prod_{x\in\Delta} \sigma_x^k = f(\Delta)^k$. Then compute the following algorithm to decrypt the message m as follows:

$$
\begin{aligned}
Z &= \frac{e(C_3, \prod_{i\in I} \hat{d}_{i,0}^{\omega_i})}{e(L, \prod_{i\in I} d_{i,1}^{\omega_i})} \\
&= \frac{e(g^k, \hat{g}^{\alpha \sum_{i\in I} \lambda_i\omega_i} \hat{f}(\Delta)^{\sum_{i\in I} \omega_i r_i})}{e(f(\Delta)^k, \hat{g}^{\sum_{i\in I} r_i\omega_i})} \\
&= e(g, \hat{g})^{\alpha k} \\
m &= \frac{C_1}{C_2 \cdot Z} = \frac{e(g, \hat{g})^{\alpha\beta k} \cdot m}{e(g, \hat{g})^{\alpha(\beta-1)k} \cdot e(g, \hat{g})^{\alpha k}}
\end{aligned}
$$

The proposed KP-ABE scheme adopts the technique used in [29] to achieve fast decryption. The decryption in our KP-ABE needs only 2 parings, which is independent of the numbers of attributes in the ciphertext or the secret key.

Furthermore, by adopting the transformation shown in Subsection 2.7, we obtain an efficient searchable encryption with constant pairings in the Test algorithm.

3.2. The PEKS from The Proposed KP-ABE

In Subsection 3.1, we proposed an anonymous KP-ABE with high efficiency. As mentioned in Subsection 2.7, an anonymous KP-ABE can be transformed into a searchable encryption. In order to avoid the unnecessary repetition, we only give an intuition in this section. The main idea is to view keywords in PEKS as attributes in KP-ABE. First, we regard the access structure as a search query associated with a trapdoor. Then, the KeyGen algorithm of KP-ABE can be performed as the Trapdoor algorithm of PEKS to generate a trapdoor for an access policy. Since the underlying KP-ABE supports expressive access structures, and thus the proposed PEKS supports expressive queries. The Encrypt algorithm of KP-ABE can be slightly modified into the Encrypt algorithm of PEKS by encrypting a randomly chosen message M, and outputting M along with the ciphertext C outputted from the Encryption algorithm of KP-ABE. As for the Test algorithm, one uses the Decrypt algorithm of KP-ABE to decrypt C and obtain a message M', and then check whether $M = M'$. The reader is referred to Section 2.7 for details.

4. Conclusion

Public key searchable encryption with expressive queries is necessary for people to search encrypted files, due to the widely usage of cloud services nowadays. However, the existing schemes which support monotonic queries suffer from problems such as heavy computation cost, infeasibility or incorrect in testing, weaker security notion, polynomial keyword space, and the requirement of online trusted third party. In this work, we focus on constructing a new public key searchable encryption to overcome the aforementioned drawbacks. The security proof and the comparison will be presented in the full version of this paper.

Acknowledgments

This work was partially supported by the Ministry of Science and Technology of Taiwan under grants MOST 110-2218-E-110-007-MBK, MOST 109-2221-E-110-044-MY2, MOST 110-2923-E-110-001-MY3, MOST 108-2218-E-004-002-MY2, and MOST 110-2221-E-004-003-.

References

[1] D. X. Song, D. Wagner, A. Perrig, Practical techniques for searches on encrypted data, in: Proceeding 2000 IEEE Symposium on Security and Privacy. S P 2000, 2000, pp. 44–55.

[2] D. Boneh, G. D. Crescenzo, R. Ostrovsky, G. Persiano, Public key encryption with keyword search, in: Advances in Cryptology - EUROCRYPT 2004, 2004, pp. 506–522.

[3] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky, Searchable symmetric encryption: Improved definitions and efficient constructions, in: Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS '06, ACM, New York, NY, USA, 2006, pp. 79–88. doi:10.1145/1180405.1180417.
URL http://doi.acm.org/10.1145/1180405.1180417

[4] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, H. Shi, Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions, in: Advances in Cryptology – CRYPTO 2005, 2005, pp. 205–222.

[5] D. Boneh, B. Waters, Conjunctive, subset, and range queries on encrypted data, in: Theory of Cryptography, Springer Berlin Heidelberg, 2007, pp. 535–554.

[6] J. Katz, A. Sahai, B. Waters, Predicate encryption supporting disjunctions, polynomial equations, and inner products, in: Advances in Cryptology – EUROCRYPT 2008, 2008, pp. 146–162.

[7] C.-I. Fan, V. S.-M. Huang, H.-M. Ruan, Arbitrary-state attribute-based encryption with dynamic membership, IEEE Transactions on Computers 63 (8) (2014) 1951–1961. doi:10.1109/TC.2013.83.

[8] S.-Y. Huang, C.-I. Fan, Y.-F. Tseng, Enabled/disabled predicate encryption in clouds, Future Generation Computer Systems 62 (2016) 148–160. doi:https://doi.org/10.1016/j.future.2015.12.008.
URL https://www.sciencedirect.com/science/article/pii/S0167739X15003921

[9] J. Lai, X. Zhou, R. Deng, X. Li, K. Chen, Expressive search on encrypted data, in: Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, 2013, pp. 243–252.

[10] A. Guillevic, Comparing the pairing efficiency over composite-order and prime-order elliptic curves, in: Applied Cryptography and Network Security, 2013, pp. 357–372.

[11] M. H. Ameri, M. Delavar, J. Mohajeri, M. Salmasizadeh, A key-policy attribute-based temporary keyword search scheme for secure cloud storage, IEEE Transactions on Cloud Computing (2018) 1–1.

[12] Y. Miao, J. Ma, X. Liu, X. Li, Q. Jiang, J. Zhang, Attribute-based keyword search over hierarchical data in cloud computing, IEEE Transactions on Services Computing (2018) 1–1.

[13] Y. Miao, J. Ma, X. Liu, J. Weng, H. Li, Lightweight fine-grained search over encrypted data in fog computing, IEEE Transactions on Services Computing (2018) 1–1.

[14] H. Wang, X. Dong, Z. Cao, Multi-value-independent ciphertext-policy attribute based encryption with fast keyword search, IEEE Transactions on Services Computing (2018) 1–1.

[15] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters, Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption, in: Advances in Cryptology – EUROCRYPT 2010, 2010, pp. 62–91.

[16] H. Fei, Q. Jing, Z. Huawei, H. Jiankun, A general transformation from KP-ABE to searchable encryption, in: Cyberspace Safety and Security, Springer Berlin Heidelberg, 2012, pp. 165–178.

[17] F. Han, J. Qin, H. Zhao, J. Hu, A general transformation from KP-ABE to searchable encryption, Future Generation Computer Systems 30 (2014) 107 – 115, special Issue on Extreme Scale Parallel Architectures and Systems, Cryptography in Cloud Computing and Recent Advances in Parallel and Distributed Systems, ICPADS 2012 Selected Papers. doi:https://doi.org/10.1016/j.future.2013.09.013.

[18] C. Shen, Y. Lu, J. Li, Expressive public-key encryption with keyword search: Generic construction from kp-abe and an efficient scheme over prime-order groups, IEEE Access 8 (2020) 93–103. doi:10.1109/ACCESS.2019.2961633.

[19] Z. Lv, C. Hong, M. Zhang, D. Feng, Expressive and secure searchable encryption in the public key setting, in: Information Security, 2014, pp. 364–376.

[20] Q. Zheng, S. Xu, G. Ateniese, VABKS: Verifiable attribute-based keyword search over outsourced encrypted data, in: IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, 2014, pp. 522–530.

[21] H. Cui, Z. Wan, R. H. Deng, G. Wang, Y. Li, Efficient and expressive

keyword search over encrypted data in cloud, IEEE Transactions on Dependable and Secure Computing 15 (3) (2018) 409–422.

[22] R. Meng, Y. Zhou, J. Ning, K. Liang, J. Han, W. Susilo, An efficient key-policy attribute-based searchable encryption in prime-order groups, in: Provable Security, Cham, 2017, pp. 39–56.

[23] S. Chatterjee, A. Menezes, On cryptographic protocols employing asymmetric pairings - the role of ψ revisited, Discrete Applied Mathematics 159 (13) (2011) 1311 – 1322. doi:https://doi.org/10.1016/j.dam.2011.04.021.

[24] A. Beimel, Secure schemes for secret sharing and key distribution, Technion-Israel Institute of technology, Faculty of computer science, 1996.

[25] D. Boneh, A. Raghunathan, G. Segev, Function-private identity-based encryption: Hiding the function in functional encryption, in: R. Canetti, J. A. Garay (Eds.), Advances in Cryptology – CRYPTO 2013, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 461–478.

[26] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in: Proceedings of the 13th ACM conference on Computer and communications security, 2006, pp. 89–98.

[27] Y.-F. Tseng, C.-I. Fan, C.-W. Lin, Provably secure ciphertext-policy attribute-based encryption from identity-based encryption, Journal of Universal Computer Science 25 (3) (2019) 182–202.

[28] C.-I. Fan, Y.-F. Tseng, J.-J. Huang, S.-F. Chen, H. Kikuchi, Multireceiver predicate encryption for online social networks, IEEE Transactions on Signal and Information Processing over Networks 3 (2) (2017) 388–403. doi:10.1109/TSIPN.2017.2697580.

[29] S. Hohenberger, B. Waters, Attribute-based encryption with fast decryption, in: K. Kurosawa, G. Hanaoka (Eds.), Public-Key Cryptography –

PKC 2013, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 162–179.