Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 314621, 7 pages
http://dx.doi.org/10.1155/2015/314621

Research Article

A Novel Model for Lattice-Based Authorized Searchable Encryption with Special Keyword

Fugeng Zeng[1,2] and Chunxiang Xu[1]

[1]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*
[2]*Department of Mathematics, Qiongzhou University, Hainan 572000, China*

Correspondence should be addressed to Fugeng Zeng; zengfugeng@foxmail.com

Received 25 December 2014; Accepted 26 January 2015

Academic Editor: Chin-Chia Wu

Data stored in the cloud servers, keyword search, and access controls are two important capabilities which should be supported. Public-keyword encryption with keyword search (PEKS) and attribute based encryption (ABE) are corresponding solutions. Meanwhile, as we step into postquantum era, pairing related assumption is fragile. Lattice is an ideal choice for building secure encryption scheme against quantum attack. Based on this, we propose the first mathematical model for lattice-based authorized searchable encryption. Data owners can sort the ciphertext by specific keywords such as time; data users satisfying the access control hand the trapdoor generated with the keyword to the cloud sever; the cloud sever sends back the corresponding ciphertext. The security of our schemes is based on the worst-case hardness on lattices, called learning with errors (LWE) assumption. In addition, our scheme achieves attribute-hiding, which could protect the sensitive information of data user.

1. Introduction

Nowadays, more and more people use service from cloud server [1], which provides scalable and elastic storage and computation resources by the Internet. Outsourcing data services to the cloud enables companies to not only save equipment investment, but also simplify the local IT management. Cloud infrastructures are physically hosted and maintained by the cloud providers. To minimize the risk of data leakage to cloud service providers and protect data security and privacy, data owners choose to encrypt sensitive data, such as health records, and property information, before outsourcing it to the cloud, while retaining the decryption key by itself and other authorized users. However, simple encryption scheme is not enough, because the data owners tend to strengthen the sharing of sensitive data under fine-grained access control. Cloud server cannot be fully trusted by the data owner, so traditional server-based access control methods are no longer suitable solution for cloud computing.

In order to address the problem of secure and decentralized access control, Sahai and Waters [2] proposed the concept of ABE by extending identity-based encryption, which achieved flexibility and one-to-many encryption and provided a fine-grained data sharing scheme. Later, there are two kinds of ABE that were put forward: key policy (KP) ABE, which is the ciphertext associated with the attributes and the secret key associated with the decryption policy, and ciphertext policy (CP) ABE, where the secret key associated a list of attributes and the ciphertext associated with access policy. Goyal et al. [3] proposed the first construction of KP-ABE which supported any monotone access policy. After then, the first CP-ABE scheme was provided by Bethencourt et al. [4]; unfortunately the security proof of their scheme was only proved in the generic group model. Subsequently, Ostrovsky et al. broaden the two programs, to support any nonmonotonic structure [5]. The first CP-ABE scheme which could be proved in the standard model was proposed by Cheung and Newport [6] including only AND-gate. Later on, Waters [7] gave the first CP-ABE proved in the standard model supporting fully expressive access structure.

All the schemes mentioned above are constructed from pairings. But unluckily, if we move into the era of postquantum, pairing related assumption is fragile. Lattice is an ideal choice for building secure encryption scheme according to

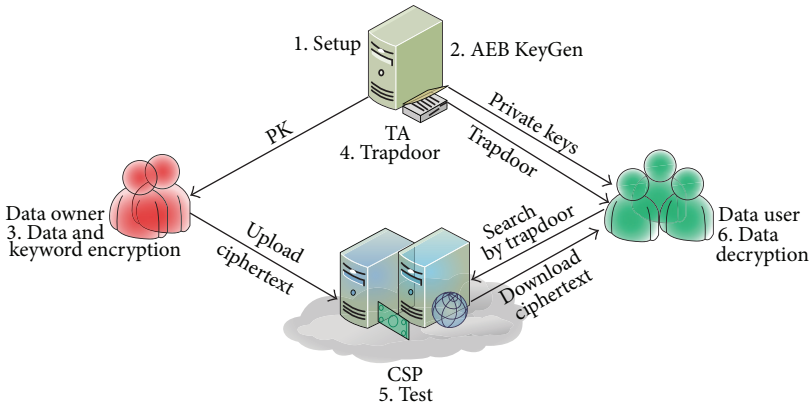

Figure 1: System architecture of ASE in cloud computing.

two facts: firstly, there is no known algorithm even with the help of quantum computer that can efficiently solve lattice hard problems; secondly, lattice-based cryptographic constructions enjoy several potential advantages: asymptotic efficiency, conceptual simplicity, and security proof based on worst-case hard problem. Recently, ABE from lattice assumptions are ascendant. J. Zhang and Z. Zhang [8] proposed a CP-ABE without pairings scheme, which supports AND-gates access structure. Boyen [9] built a KP-ABE from lattice assumptions and pointed to the future work of the study of CP-ABE as an open problem.

ABE resolves the problem of fine-grained access control and provides a one-to-many encryption which can improve the efficiency of the data owner; however, data utilization is still a challenging problem. For example, in order to search some relevant documents amongst an encrypted data set stored in the cloud, one may have to download and decrypt the entire data set. This is apparently impractical when the data volume is large. Thus, mechanisms that allow users to search directly on the encrypted data are of great interest in the era of cloud computing. Based on the traditional plaintext keyword search data services will result in bad quality of service because the data are encrypted. Boneh et al. [10] proposed a public key encryption with keyword search (PEKS) scheme to address the problem of searching encrypted data.

There are also many existing searchable encryption schemes from parings. Lai et al. [11] present a more efficient construction based on Lewko et al.'s KP-ABE scheme [12]. However, scheme [11] discloses the searching keywords in the trapdoor, which will let the server learn whether the encrypted data contains the keywords in the trapdoor. Compared with [13], the size of a ciphertext (or a trapdoor) in [11] is linear with the number of keywords. Recently, Lv et al. [14] present an expressive and secure asymmetric searchable encryption scheme, which is the first to simultaneously support conjunctive, disjunctive, and negation search operations. However, there has been no ASE scheme from lattice assumptions so far. In this paper, we integrate CP-ABE with PEKS and propose authorized searchable encryption with attribute-hiding from lattices, which enables only authorized users to perform keyword search and then decrypt ciphertext.

Meanwhile, by setting the keyword such as year, month, and day, data owners can sort ciphertext. If data users want to extract the ciphertext from some time point, they only need to submit trapdoor corresponding to keyword the cloud server.

Therefore, there are two main contributions of our scheme in detail as follows.

(1) To the best of our knowledge, this is the first work that addresses ASE from lattice assumptions.

(2) In contrast to previous solutions [11, 14], our scheme achieves attribute-hiding, which could protect sensitive user information from being leaked.

The rest of the paper is organized as follows. Section 2 states the preliminaries about definitions for ASE, security model for PEKS and CP-ABE, and lattice knowledge. Section 3 describes our ASE with attribute-hiding from lattice assumptions in detail. Section 4 gives the security proof of our scheme. Section 5 presents our conclusion for this paper.

2. Preliminaries

2.1. Definitions for ASE. We consider ASE in cloud computing. The system architecture is similar to that in [15] which is illustrated as Figure 1. There exist four participants in our system.

Trusted Authority (TA). The entity is fully trusted by the other participants of the system. The responsibility of TA is to initialize system parameters, to generate attributed-based private keys, and to generate trapdoor keys for data users.

Cloud Services Provider (CSP). The entity provides data storage and retrieval services. It stores the outsourcing data content of the data owner. Only the specified receiver who meets the access policy can search and download the content. We adopt the honest-but-curious model for the cloud server as in [16]. It assumes that the cloud server would honestly follow the designated protocols and procedures to fulfill its service providers role, while it may analyze the information stored and processed on the server in order to learn additional information about its customers.

Data Owner (DO). The entity is a cloud storage subscriber who wants to encrypt its data content first and then upload to the cloud storage service. Intended receivers who satisfy the access policy can read the encrypted content. The responsibility of data owner is to create encrypted data and to choose keywords to encrypt.

Data User (DU). The entity is another cloud storage subscriber who queries encrypted data from CSP. Only retrievers who satisfy the access policy can have the legal rights to access the encrypted content and read the original message. The responsibility of data users is to choose keywords to create trapdoor for search, to initiate search requests, and to decrypt data.

In our setting, a user will be identified by a set of attributes; let S be the users attributes. An ASE scheme consists of six polynomial-time algorithms described as follows.

Setup. The setup algorithm is run by TA, which inputs a security parameter k. It outputs the master secret key msk and public system parameters $params$ which include the description of attribute universe and keyword universe. TA publishes $params$ and keeps msk secret. We describe it as $Setup(1^k) \rightarrow (params, msk)$.

ABE-KeyGen. The attribute private key generation algorithm is an interactive protocol implemented between DU and TA. The public input to TA and DU consists of the system public parameters $params$, the users attributes set S owned by DU. The private input to TA is the master secret key msk. Finally, DU can extract an attribute private key SK_S. We describe it as $ABE\text{-}KeyGen(params, msk, S) \rightarrow SK_S$.

KS-CPABE. DO runs the encryption algorithm, which inputs the system public parameters $params$, an access structure W, and a message msg. The algorithm encrypts msg and produces a ciphertext ct. Note that, in our ASE, the ciphertext does not contain W, which achieves attribute-hiding. We describe it as $Encrypt(params, W, msg, kw) \rightarrow ct$.

Trapdoor. The query private key generation algorithm is an interactive protocol implemented between DU and TA. The public input to TA and DU consists of the system public parameters $params$, the users attributes set S owned by DU, and a keyword kw. TA inputs the master secret key msk. In addition, a sequence of random coin tests may be used by TA and DU as private inputs. Finally, DU can extract an attribute trapdoor T_{kw}. We describe it as $Trapdoor(params, msk, S) \rightarrow T_{kw}$. After then, DU sends T_{kw} to CSP.

Test. The keyword test algorithm is run by CSP, which takes as input system parameters $params$ and a trapdoor T_{kw} corresponding to the keyword kw from a DU and tests the ct for keyword set kw'. Output 1 if $kw = kw'$ and 0 otherwise.

Decrypt. DU runs decryption algorithm, which takes the ciphertext ct and SK_S as input. Only if S satisfies the access control W, it will return the message msg.

2.2. Security Model for PEKS and CPABE. In this subsection, we introduce the functionality of PEKS and CP-ABE independently.

2.2.1. PEKS. A *PEKS* scheme includes four polynomial-time algorithms: *KeyGen*, *PEKS*, *Trapdoor*, and *Test*. The algorithm generates a public/private key pair (pk, msk). The *PEKS* algorithm generates a searchable encryption form of keyword kw corresponding to intended receivers public key. The *Trapdoor* algorithm produces a trapdoor T_{kw} for keyword kw corresponding to receiver's private key. And the *Test* algorithm verifies whether a ciphertext matches a trapdoor.

The general security property of *PEKS* scheme is the indistinguish ability against chosen keyword attack. The *PEKS* scheme is semantic security if a polynomial adversary has no nonnegligible advantage against the challenger in the following security game [10].

Security Game

KeyGen. The challenger $\mathscr{C}$ runs *KeyGen* algorithm to generate a key pair (pk, sk) and give pk to the adversary $\mathscr{A}$.

Phase 1. $\mathscr{A}$ queries the challenger for the trapdoor for any keyword $kw \in \{0,1\}^*$ of his choice.

Challenge. At some time, $\mathscr{A}$ sends the challenger two keywords kw_0 and kw_1 which it wishes to challenge. The only restriction is that $\mathscr{A}$ has never previously queried the trapdoors T_{kw_0} and T_{kw_1} for kw_0 and kw_1, respectively. The challenger selects $b \in \{0,1\}$ randomly and sends the adversary $C = PEKS(pk, kw_b)$ as the challenge *PEKS* ciphertext.

Phase 2. $\mathscr{A}$ can continue to adaptively ask the challenger for the trapdoor for the keyword kw of his choice which satisfies $kw \neq kw_0, kw_1$.

Guess. In the end, the adversary $\mathscr{A}$ outputs $b' \in \{0,1\}$. If $b' = b$, $\mathscr{A}$ wins the game. Define the advantage of $\mathscr{A}$ in this game as $|\Pr[b' = b] - 1/2|$.

Definition 1. A PEKS scheme is (t, Q_t, ε) IND-PEKS CPA secure if all t polynomial time adversaries making at most Q_t token queries have at most a negligible advantage ε in the above security game.

2.2.2. A CP-ABE Scheme with Attribute-Hiding. The scheme consists of four algorithms [17].

Setup. This algorithm inputs a security parameter k and generates the public key PK and a master secret key msk. PK is used for encryption; msk is used to generate user secret keys. It is held by the central authority.

Encrypt. This algorithm inputs the public key PK, a message msg, and an access policy W. It outputs the ciphertext ct. Note that, in CP-ABE supporting attribute-hiding, the ciphertext does not contain W.

KeyGen. This algorithm inputs a set of attributes S associated with the user and outputs a secret key SK_S.

Decrypt. This algorithm takes as input the ciphertext ct and a secret key SK_S. Only if S satisfies the access policy W, it returns the message msg.

Selective Game for CP-ABE with Hiding Attributes

Init. The adversary $\mathcal{A}$ gives the challenge ciphertext policies W_0, W_1 before setup.

Setup. The challenger $\mathcal{C}$ runs the setup algorithm and gives PK to the adversary $\mathcal{A}$.

Phase 1. The adversary $\mathcal{A}$ submits the attribute list L for a *KeyGen* query. If $L \subseteq W_0 \wedge L \subseteq W_1$ or $L \notin W_0 \wedge L \notin W_1$, the challenger gives the adversary the secret key SK_L. The adversary $\mathcal{A}$ can repeat this query polynomial times.

Challenge. The adversary $\mathcal{A}$ submits messages M_0, M_1 to the challenger. If the adversary obtained the SK_L whose associated attribute list L satisfies both W_0 and W_1 in Phase 1, then it is required that $M_0 = M_1$. The challenger flips a random coin b and passes the ciphertext $Encrypt(PK, M_b, W_b)$ to the adversary.

Phase 2. Phase 1 is repeated. If $M_0 \neq M_1$, the adversary cannot submit L such that $L \subseteq W_0 \wedge L \subseteq W_1$.

Guess. The adversary outputs a guess b' of b. The advantage of an adversary in this game is defined as $|\Pr[b' = b] - 1/2|$.

Definition 2. A CP-ABE scheme with hiding attributes is selective CPA secure if all polynomial-time adversaries have at most a negligible advantage ε in the above security game.

2.3. Lattice and Hardness Assumption

2.3.1. Integer Lattices

Definition 3. Let $B = (b_1|\cdots|b_n) \in Z^{n\times n}$ be an $n \times n$ matrix which consists of n linearly independent vectors $b_1, \ldots, b_n \in Z^n$. The n dimensional full-rank lattice Λ generated by B is $\Lambda = \{y \in Z^n \mid y = Bc = \sum_{i=1}^{n} c_i b_i,\ c \in Z^n\}$; B is called a basis of the lattice Λ.

For a basis B, let $\widetilde{B}$ denote its Gram-Schmidt orthogonalization, defined iteratively as $\widetilde{b}_1 = b_1$, and $\widetilde{b}_i$ is the component of b_i orthogonal to span $(b_1, \ldots, b_{i-1})$. $\|B\|$ denotes the longest Euclid norm of the column vectors in B.

Given a matrix $A \in Z_q^{n\times m}$ for a prime q, integers m and n, we consider two kinds of full-rank m-dimensional integer defined by $\Lambda_q^{\perp}(A) = \{e \in Z^m \mid Ae = 0(\mathrm{mod} q)\}$, $\Lambda_q(A) = \{y \in Z^m \mid \exists s \in Z^n,\ A^T s = y(\mathrm{mod} q)\}$.

Proposition 4 (see [18]). *For any prime $q \geq 2$ and $m \geq 5n\log q$, there is a probabilistic polynomial-time algorithm $TrapGen(q, n)$ that outputs a matrix $A \in Z_q^{n\times m}$ and a full-rank set $T_A \in Z^{m\times m}$ such that A is statistically close to uniform over $Z_q^{n\times m}$ and T_A is a basis for $\Lambda_q^{\perp}(A)$.*

2.3.2. Discrete Gaussian. For any $s > 0$, the Gaussian function on $\Lambda \subset Z^n$ centered at c with parameter s is defined as $\forall x \in \Lambda$, $\rho_{s,c}(x) = \exp(-\pi(\|x - c\|/s^2))$. Let $\rho_{s,c}(\Lambda) = \sum_{x\in\Lambda} \rho_{s,c}(x)$. The discrete Gaussian distribution over Λ with center c and parameter s is defined as $\forall y \in \Lambda$, $D_{\Lambda,s,c}(y) = \rho_{s,c}(y)/\rho_{s,c}(\Lambda)$. The subscripts s are taken to be 0 when omitted.

Gentry et al. [19] defined and constructed the preimage sampleable functions. Let T_A be a basis for an m-dimensional lattice Λ satisfying $s \geq \|\widetilde{T}_A\|\omega(\sqrt{\log m})$, the algorithm samples from the discrete Gaussian distribution $D_{\Lambda,s,c}$.

The preimage sampleable function is defined as follows.

$Samplepre(A, T_A, u, s)$. The algorithm takes as input $A \in Z_q^{n\times m}$, the short basis T_A for $\Lambda_q^{\perp}(A)$, the target image $u \in Z_q^n$, and Gaussian parameters $s \geq \|\widetilde{T}_A\|\omega(\sqrt{\log m})$ and outputs $e \in Z^n$ which is statistically close to $D_{\Lambda_q^u(A),s}$.

2.3.3. Learning with Error Problem. Our construction can be reduced to learning with errors *LWE* problem, which is a classical problem defined by Regev [20].

For an integer $q = q(n)$ and a distribution χ on Z_q, the goal of the (average case) learning with errors problem $LWE_{q,\chi}$ is to distinguish the distribution $A_{s,\chi}$ for some uniform secret $s \in Z_q^n$ and the uniform distribution on $Z_q^n \times Z_q$. The hardness of *LWE* problem means the distribution $A_{s,\chi}$ is pseudorandom. Regev demonstrated that, for certain modulo q and Gaussian error distributions χ, $LWE_{q,\chi}$ is as hard as solving several standard worst-case lattice problems using quantum algorithm.

Proposition 5 (see [20]). *For an $\alpha \in (0, 1)$ and a prime $q > 2\sqrt{n}/\alpha$, let $\overline{\psi_\alpha}$ denote the distribution over Z_q of the random variable $\lfloor qX + 1/2 \rfloor \bmod q$, where X is a normal random variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$. If there exists an efficient, possibly quantum, algorithm for deciding the (Z_q, n, χ)-LWE problem, then there exists an efficient quantum algorithm for approximating the SIVP and GapSVP problems, to within $\widetilde{O}(n/\alpha)$ factors in the l_2 norm, in the worst case.*

3. Authorized Searchable Encryption Scheme

In this section, we put forward our ASE scheme where the access structures include positive and negative attributes based on AND-gates. Define some symbols simply as follows: let the set of attributes be $N = \{1, 2, \ldots, l\}$ for a fixed natural number l. Mark attributes i and their negations $\neg i$ as literals. Consider access structures that consist of an AND-gate policy whose inputs are literals, which is denoted by $W = \wedge_{i\in I} \bar{i}$, where $W \subseteq N$ and every $\bar{i}$ is literal (i.e., i or $\neg i$). Our construction is defined as follows, which is parameterized by dimension m, Gaussian parameter s, modulus q, and α that determines the error distribution χ.

Setup(n, m, q, N). TA chooses a cryptographic secure hash function H, which maps each keyword kw to a vector in Z_q. Compute $(A_0, T_{A_0}) \leftarrow TrapGen(n, m, q)$; then, for each $i \in N$, randomly choose $A_{i^+} \leftarrow Z_q^{n\times m}$, $A_{i^-} \leftarrow Z_q^{n\times m}$. Intuitively, the public key elements A_{i^+}, A_{i^-} associate with two cases of i: positive and negative. Next, randomly choose a vector $u \leftarrow Z_q^n$ and set public key $PK = (A_0, \{A_{i^+}, A_{i^-}\}_{i\in R}, u, H)$, while keeping the master secret key $msk = (PK, T_{A_0})$.

ABE-KeyGen. Denote S as the input attribute set of DU. Every $i \notin S$ is implicitly as a negative attribute. For each $i \in N$, if $i \in S$, define $\overline{A_i} = A_{i^+}$; else define $\overline{A_i} = A_{i^-}$; then, for each $i \in R$, randomly choose $e_i \leftarrow D_{Z^m,s}$, and compute $y = u - \sum_{i\in R} \overline{A_i} e_i$; finally, compute $e_0 \leftarrow Samplepre(A_0, T_{A_0}, s, y)$, and return secret key $SK_S = [e_0, e_1, \ldots, e_{|l|}]$. Observe that if letting $E = [A_0 \parallel \overline{A_1} \parallel \cdots \parallel \overline{A_{|l|}}]$, we have $E \cdot SK_S = u$.

KS-CPABE. Given a message bit $M \in \{0, 1\}$ and an AND-gate access structure $W = \wedge_{i\in I} \tilde{i}$, let $S^+(S^-)$ be the set of positive (negative) attributes in W, respectively, and denote $S' = S^+ \cup S^-$; then, for each $i \in S'$, if $i \in S^+$, define c_{i1} as a well-formed ciphertext and c_{i2} as a malfunction ciphertext. If $i \in S^-$, the situation is converse; define c_{i2} as a well-formed ciphertext and c_{i1} as a malfunction ciphertext. If $i \in N \setminus S'$, both c_{i1} and c_{i2} are well-formed ciphertext, and, for each keyword kw, $H(kw) \in Z_q^n$. Randomly choose $x \in Z_q^n$, $x_c, x_p \in \chi$, and $x_0, x_{i^+}, x_{i^-} \in \chi^m$ as noise distributions; compute $c = u^T x + x_c + M\lfloor q/2 \rfloor$, $p = H(kw)^T x + x_p$, and $c_0 = A_0^T x + x_0$. If $i \in S^+$, $c_{i1} = A_{i^+}^T x + x_{i^+}$, and c_{i2} is a random m dimension vector and could be achieved by randomly choosing $x_i' \in Z_q^n$, $c_{i2} = A_{i^-}^T x_i' + x_{i^-}$. If $i \in S^-$, $c_{i1} = A_{i^+}^T x_i' + x_{i^+}$, $c_{i2} = A_{i^-}^T x + x_{i^-}$. If $i \in N \setminus S'$, $c_{i1} = A_{i^+}^T x + x_{i^+}$, $c_{i2} = A_{i^-}^T x + x_{i^-}$.

Finally, return ciphertext $C = (c, c_0, \{c_{i1}, c_{i2}\}_{i\in N})$ and secure keyword attachment p.

Trapdoor. To generate a trapdoor for a keyword, DU must contact with TA. TA enforces the trapdoor generation similar to the process of ABE-KeyGen phase. For each $i \in N$, if $i \in S$, define $\overline{A_i} = A_{i^+}$; else define $\overline{A_i} = A_{i^-}$; then, for each $i \in N$, randomly choose $f_i \leftarrow D_{Z^m,s}$ and compute $z = H(kw) - \sum_{i\in R} \overline{A_i} f_i$; finally, compute $f_0 \leftarrow Samplepre(A_0, T_{A_0}, s, z)$ and return secret key $T_{kw} = [f_0, f_1, \ldots, f_{|R|}]$.

Observe that if we let $E = [A_0 \parallel \overline{A_1} \parallel \cdots \parallel \overline{A_{|l|}}]$, we have $E \cdot T_{kw} = H(kw)$. TA securely transform the query trapdoor to DU. When users want to download ciphertext related to keywords kw, DU sends $T_{kw} = [f_0, f_1, \ldots, f_{|R|}]$ and a list L corresponds to attribute positive or negative to CSP; ask the CSP to enforce the search ciphertext. Note that DU does not reveal the attribute name to CSP except the positive or negative information of the attributes.

Test. CSP receives the trapdoor T_S and list L about the positive or negative information of attributes; let $y_0 = c_0$ if i is a positive attribute, and let $y_i = c_{i1}$; else let $y_i = c_{i2}$. Define $y = [y_0, y_1, \ldots, y_{|R|}]$; compute $a = T_{kw}^T \cdot y = H(kw)^T x + x'$; let $b = p - a$; if $|b| < \lfloor q/4 \rfloor$, CSP accepts it as a valid ciphertext and outputs 1, otherwise, CSP refuses it as an invalid ciphertext and outputs 0.

Decrypt. After receiving the cipthertext from CSP, DU does the decryption procedure as the test phase. Define $y = [y_0, y_1, \ldots, y_{|R|}]$ as above; compute $f = SK_S^T \cdot y = u^T x + x'$. Define $g = c - f = x_c - x' + M\lfloor q/2 \rfloor$. Finally if $|g - q/2| < \lfloor q/4 \rfloor$ in Z, return 1; otherwise, return 0.

4. Security Proof

In this section, we discuss the security proof of our ASE scheme. Comparing ASE scheme with CP-ABE with attribute-hiding and PEKS scheme, we divide our ASE scheme into two parts. If we only choose setup, ABE-KeyGen, encrypt (do not take over the keyword ciphertext p), and decrypt from ASE scheme, our scheme is a CP-ABE scheme with attribute-hiding. If we only choose setup, encrypt (do not take over the first ciphertext c), trapdoor, and test from ASE scheme, our scheme is a PEKS scheme. So we give our security proof of our ASE schemes by the following two theorems.

Theorem 6. *If $LWE_{q,\chi}$ is hardness problem, then this CP-ABE scheme with attribute-hiding is secure against selective chosen plaintext attack. It means that if there exists an adversary A that breaks the selective chosen plaintext attack game with advantage ε, then there exists an algorithm $\mathcal{B}$ cloud solve $LWE_{q,\chi}$ with probability ε.*

Proof. Algorithm $\mathcal{B}$ has an oracle $O(\cdot)$, the goal of $\mathcal{B}$ is to decide whether the samples output by $O(\cdot)$ is from $A_{s,\chi}$ or uniform. $\mathcal{B}$ runs adversary $\mathcal{A}$ and simulates $\mathcal{A}$'s view selective chosen plaintext attack game as follows.

Init. Adversary $\mathcal{A}$ chooses two challenge ciphertext policies $W_0 = [S_{0,1}, S_{0,2}, \ldots, S_{0,l}]$ and $W_1 = [S_{1,1}, S_{1,2}, \ldots, S_{1,l}]$ and gives them to $\mathcal{B}$. Let $S^+(S^-)$ be the set of positive (negative) attributes in $W_0 \cup W_1$ and let $S' = S^+ \cup S^-$.

Setup. After receiving W_0, W_1, $\mathcal{B}$ obtains $(A_0, v_0) \in Z_q^{n\times m} \times Z_q^m$ and $(u, v_u) \in Z_q^n \times Z_q$ from $O(\cdot)$.

For each $i \in R \setminus S'$, $\mathcal{B}$ obtains (A_{i^+}, v_{i^+}), $(A_{i^-}, v_{i^-}) \in Z_q^{n\times m} \times Z_q^m$ from $O(\cdot)$. For each $i \in S^+$, $\mathcal{B}$ obtains $(A_{i^+}, v_{i^+}) \in Z_q^{n\times m} \times Z_q^m$ from $O(\cdot)$ and then computes $(A_{i^-}, T_{A_{i^-}}) \leftarrow TrapGen(n, m, q)$. For each $i \in S^-$, $\mathcal{B}$ obtains $(A_{i^-}, v_{i^-}) \in Z_q^{n\times m} \times Z_q^m$ from $O(\cdot)$ and then computes $(A_{i^+}, T_{A_{i^+}}) \leftarrow TrapGen(n, m, q)$.

Finally $\mathcal{B}$ sets public key $PK = (A_0, \{A_{i^+}, A_{i^-}\}_{i\in R}, u)$, while keeping the master secret key $(\{T_{A_{i^-}}, v_{i^+}\}_{i\in S^+}, \{T_{A_{i^+}}, v_{i^-}\}_{i\in S^-}, \{v_{i^+}, v_{i^-}\}_{i\in R\setminus S'})$.

KeyGen Queries. $\mathcal{B}$ receives a query from $\mathcal{A}$ with attribute sets $S \subseteq N$. If S satisfies W_0 and W_1, $\mathcal{B}$ simply outputs $\perp$. Otherwise, for each $i \in R$, if $i \in S$, $\mathcal{B}$ lets $\overline{A_i} = A_{i^+}$; else it lets $\overline{A_i} = A_{i^-}$.

Since S does not satisfy W_0 and W_1, namely, $S^+ \cap S \neq S^+$ or $S^- \cap S \neq S^-$, there must exist a $j \in R$, such that $\overline{A_j}$ is generated by TrapGen. Hence, $\mathcal{B}$ knows its trapdoor $T_{\overline{A_j}}$.

Let $E = [A_0 \parallel \overline{A_1} \parallel \cdots \parallel \overline{A_{|l|}}]$; then, for each $i \in R \setminus j$, randomly choose $e_i \leftarrow D_{Z^m,s}$ and compute $y = u - \sum_{i \in R} \overline{A_i} e_i$, and $\mathcal{B}$ computes $e_j \leftarrow Samplepre(\overline{A_j}, T_{\overline{A_j}}, s, y)$ and returns secret key $SK_S = [e_0, e_1, \ldots, e_{|l|}]$ and returns SK_S to $\mathcal{A}$.

Challenge. The adversary $\mathcal{A}$ submits messages M_0, M_1 to the challenger. If the adversary obtained the SK_L whose associated attribute list L satisfies both W_0 and W_1 in Phase 1, then it is asked that $M_0 = M_1$. $\mathcal{B}$ randomly chooses $b \in \{0, 1\}$ and computes $c = v_u + M_b \lfloor q/2 \rfloor$ and $c_0 = v_0$. For each $i \in S^+$, let $c_{i1} = v_{i^+}$; c_{i2} is a random vector. For each $i \in S^-$, let $c_{i2} = v_{i^-}$; c_{i1} is a random vector. For each $i \in N \setminus S'$, let $c_{i1} = v_{i^+}$, $c_{i2} = v_{i^-}$.

Finally, $\mathcal{B}$ returns $C^* = (c, c_0, \{c_{i1}, c_{i2}\}_{i \in l})$.

Phase 2. Phase 2 has similar operations to phase 1. If $M_0 \neq M_1$, the adversary cannot submit L such that $L \subseteq W_0 \wedge L \subseteq W_1$.

$\mathcal{A}$ can make more key generation queries with the limitation that the attribute set S does not satisfy W_0 and W_1. Finally, $\mathcal{A}$ outputs a bit b' as a guess for b. If $b' = b$, $\mathcal{B}$ outputs 1; else it outputs 0.

On one hand, if $O(\cdot)$ is a *LWE* oracle for some x^*, C^* is a valid ciphertext; thus the distribution of $\mathcal{A}$'s view is statistically close to that in the real game. On the other hand, if $O(\cdot)$ is chosen from uniform, then the ciphertext c is uniform from Z_q; then the probability that $\mathcal{A}$ guesses the right b is exactly $1/2$. Therefore, if $\mathcal{A}$ can break our system, $\mathcal{B}$ can solve the *LWE* problem. □

Theorem 7. *Assuming the LWE assumption is hardness, this PEKS scheme is IND-PEKS CPA secure in the random oracle model.*

Proof. In the random oracle mode, suppose there is a polynomial-time adversary $\mathcal{A}$ that has nonnegligible advantage ε attacking the scheme; let the maximum number of H queries be Q_H, and construct an algorithm $\mathcal{B}$ to solve the *LWE* problem. $\mathcal{B}$ runs $\mathcal{A}$ as a subroutine. $\mathcal{B}$ uniformly chooses a random index $j \leftarrow [Q_H]$ and interacts with $\mathcal{A}$ as follows.

Setup. $\mathcal{B}$ sends $PK = (A_0, \{A_{i^+}, A_{i^-}\}_{i \in R})$ to $\mathcal{A}$.

Phase 1. $\mathcal{B}$ answers queries of $\mathcal{A}$ as follows.

Phase 1: Hash Queries. $\mathcal{B}$ keeps a list L which is originally empty. The form of L is (kw, h, T_{kw}). Receiving $\mathcal{A}$'s ith distinct query kw to H, if $i = j$, then $\mathcal{B}$ sets $h = h^*$, $SK_T = \perp$ and gives h^* to $\mathcal{A}$; else, for $i \neq j$, $\mathcal{B}$ randomly chooses $f_i \leftarrow D_{Z^m,s}$ and returns secret key $T_{kw} = [f_0, f_1, \ldots, f_{|l|}]$. Let $E = [A_0 \parallel \overline{A_1} \parallel \cdots \parallel \overline{A_{|l|}}]$; we have $E \cdot T_{kw} = h$, and return h to $\mathcal{A}$.

Phase 1: Trapdoor Queries. When $\mathcal{A}$ asks for the trapdoor for a keyword kw, if $\mathcal{A}$ has already queried H about kw, let (kw, h, T_{kw}) be the corresponding tuple in the list L. If $T_{kw} = \perp$, then $\mathcal{B}$ aborts; otherwise it gives T_{kw} to $\mathcal{A}$.

Challenge. $\mathcal{A}$ submits two target keywords kw_0, kw_1 to $\mathcal{B}$, if $\mathcal{A}$ has already queried H about kw_0, kw_1 and meets $H(kw_0) \neq h^*$ and $H(kw_1) \neq h^*$; then $\mathcal{B}$ aborts. Otherwise, compute $c_0, \{c_{i1}, c_{i2}\}_{i \in N}$ as normal; p is chosen from challenge oracle O; $p = (h^*)^T x + x_p$ or a random number from Z_q. Finally, return ciphertext $ct = (p, c_0, \{c_{i1}, c_{i2}\}_{i \in N})$.

Notice that if O is pseudorandom *LWE*, p is a part of an effective encryption; if O is random *LWE*, p is uniform distribution from Z_q.

Phase 2. $\mathcal{B}$ answers $\mathcal{A}$'s query about $kw \in \{0, 1\}^*$ as the phase 1; the only limitation is $kw \neq kw_0, kw_1$.

Guess. $\mathcal{A}$ outputs $b' \in \{0, 1\}$; finally, if $b = b'$, then $\mathcal{B}$ outputs 1; otherwise $\mathcal{B}$ outputs 0.

We now analyze the reduction. The probability of $\mathcal{B}$ does not abort in the trapdoor query $1 - 1/Q_H$. In the phase of challenge, the probability of $H(kw_0) = h^*$ or $H(kw_1) = h^*$ is $2/Q_H$, so we can get that the advantage of $\mathcal{B}$ solving *LWE* is $2\varepsilon(Q_H - 1)/Q_H^2$. □

5. Conclusion

We propose an authorized searchable encryption with attribute-hiding from lattices, which only enables authorized users to perform keyword search and then decrypt ciphertext. We are the first to integrate PEKS with CP-ABE based lattices assumption. In contrast to previous solutions [11, 14], our scheme achieves attribute-hiding, which could prevent the revelation of sensitive user information. The security of our schemes is based on LWE assumption; meanwhile data owners can sort ciphertext. If data users want to extract the ciphertext from some time point, they only need to submit trapdoor corresponding to keyword the cloud server.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant nos. 61272525 and 61370203) and Science and Technology on Communication Security Laboratory Foundation (no. 9140C110301110C1103).

References

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," *National Institute of Standards and Technology*, vol. 53, no. 6, 2009.

[2] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology—EUROCRYPT 2005*, vol. 3494 of *Lecture Notes in Computer Science*, pp. 457–473, Springer, Berlin, Germany, 2005.

[3] V. Goyal, O. Pandey, A. Sahai, and 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 (CCS '06), pp. 89–98, November 2006.

[4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the IEEE Symposium on Security and Privacy (SP '07)*, pp. 321–334, May 2007.

[5] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pp. 195–203, November 2007.

[6] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pp. 456–465, November 2007.

[7] B. Waters, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," in *Public Key Cryptography—PKC 2011*, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds., vol. 6571 of *Lecture Notes in Computer Science*, pp. 53–70, Springer, Berlin, Germany, 2011.

[8] J. Zhang and Z. Zhang, "A ciphertext policy attribute-based encryption scheme without pairings," in *Information Security and Cryptology*, vol. 7537 of *Lecture Notes in Computer Science*, pp. 324–340, Springer, Berlin, Germany, 2012.

[9] X. Boyen, "Attribute-based functional encryption on lattices," in *Theory of Cryptography: 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3–6, 2013. Proceedings*, vol. 7785 of *Lecture Notes in Computer Science*, pp. 122–142, Springer, Berlin, Germany, 2013.

[10] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology—EUROCRYPT 2004*, vol. 3027 of *Lecture Notes in Computer Science*, pp. 506–522, Springer, Berlin, Germany, 2004.

[11] J. Lai, X. Zhou, R. H. Deng, Y. Li, and K. Chen, "Expressive search on encrypted data," in *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security (ASIA CCS '13)*, pp. 243–252, May 2013.

[12] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption," in *Advances in Cryptology—EUROCRYPT 2010*, H. Gilbert, Ed., vol. 6110 of *Lecture Notes in Computer Science*, pp. 62–91, Springer, Heidelberg, Germany, 2010.

[13] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *Advances in Cryptology—EUROCRYPT 2008*, vol. 4965 of *Lecture Notes in Computer Science*, pp. 146–162, Springer, Berlin, Germany, 2008.

[14] Z. Lv, H. Cheng, M. Zhang, and D. Feng, "Expressive and secure searchable encryption in the public key setting," in *Information Security*, Lecture Notes in Computer Science, pp. 364–376, Springer, Berlin, Germany, 2014.

[15] C. Wang, W. Li, Y. Li et al., "A ciphertext-policy attribute-based encryption scheme supporting keyword search function," in *Cyberspace Safety and Security*, pp. 377–386, Springer, Berlin, Germany, 2013.

[16] M. Chuah and W. Hu, "Privacy-aware bedtree based solution for fuzzy multi-keyword search over encrypted data," in *Proceedings of the 31st International Conference on Distributed Computing Systems Workshops (ICDCSW '11)*, pp. 273–281, June 2011.

[17] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Applied Cryptography and Network Security*, vol. 5037 of *Lecture Notes in Computer Science*, pp. 111–129, Springer, Berlin, Germany, 2008.

[18] J. Alwen and C. Peikert, "Generating shorter bases for hard random lattices," in *Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science (STACS '09)*, pp. 75–86, Freiburg, Germany, 2009.

[19] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC '08)*, R. E. Ladner and C. Dwork, Eds., pp. 197–206, ACM, 2008.

[20] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC '05)*, pp. 84–93, ACM, Baltimore, Md, USA, May 2005.