JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 32, 425-438 (2016)

DOI:10.1688/JISE.2016.32.2.10

Semantic Searchable Encryption Scheme Based on Lattice in Quantum-Era*

YANG YANG[1] AND MAODE MA[2]
[1]*College of Mathematics and Computer Science*
Fuzhou University
Fuzhou, 350108 P.R. China
E-mail: yang.yang.research@gmail.com
[2]*School of Electrical and Electronic Engineering*
Nanyang Technological University
639798 Singapore
E-mail: emdma@ntu.edu.sg

When data is outsourced to a remote storage server, searchable encryption plays an important role to protect data privacy while allowing users to retrieve data in the massive ciphertext. As far as we know, most of the existing searchable encryption schemes work based on the bilinear map. These schemes may not be secure in the quantum age. Both discrete logarithms and factorization can be solved by quantum computer in a polynomial time. There are very few searchable encryption schemes that can be proved secure in post quantum age. In this paper, in order to construct a post-quantum secure scheme for future cloud storage, we suggest a public key encryption with semantic keyword search using the lattice based mechanism. The suggested scheme is proved secure against indistinguishable chosen-keyword attacks (IND-CKA) based on learning with errors (LWE) problem. This scheme is believed to be secure in quantum-era due to the quantum intractability of the LWE problem.

Keywords: lattice-based cryptography, semantic searchable encryption, chosen keyword attack, learning with errors, cloud computing

1. INTRODUCTION

The rapid development of cloud computing has attracted a lot of attention from both business community and the academy. In recent years, a lot of commercial cloud services emerge which make storage and computing outsourcing possible. In order to reduce the purchase and maintaining cost of computing and storage equipment, many enterprises turn their heads to the convenience of using a public cloud infrastructure. A lot of individual users also begin to try uploading their personal information to the cloud. With the expansion of application scope, a large quantum of information is gathered at the cloud server which may contain a lot of personal sensitive information. In the public cloud environment, it is very hard to assure the customers that their private data will not be watched stealthily or sold to a third party for benefits. Significant security and privacy concerns begin to rise which will hinder the further development of the cloud service. Data encryption is a fundamental and classical way to protect data from eavesdropping.

Received September 24, 2014; revised December 2, 2014; accepted January 16, 2015.
Communicated by Hung-Min Sun.
* This work is supported by National Natural Science Foundation of China under Grant No. 61402112, 61472307, 61472309, 61303198, Science and Technology Project of Fujian Education Department under Grant No. JA12028, Science and Technology Development Foundation of Fuzhou University under Grant No. 2012-XY-17.

It is vital to provide search functions over the encrypted information when an individual wants to seek documents relevant to certain topics from a huge quantity of encrypted data. Traditional encryption scheme could not provide such functions.

In 2004, Boneh *et al.* [1] proposed a public key encryption scheme with keyword search (PEKS) to facilitate the encrypted information retrieval without compromising the security of the plaintext. In the suggested scenario in [1], user Alice plans to read her email on different devices, such as desktop, laptop and smart phone. For example, she wants to check her smart phone on those emails that is urgent for her quick reply. Alice's email gateway is supposed to route email to the designated device based on the keyword in the email. When Jack sends an email with the keyword "dinner", the email will be routed to Alice's desktop for later reading. After that, Gu *et al.* [2] gave out a new PEKS construction from pairings based on n-Bilinear Diffie-Hellman Inverse (*n*-BDHI) problem. Long *et al.* [3] designed a new index structure to accelerate the search procedure. In order to retrieve several keywords at one time, Chen *et al.* [4] presented a conjunctive keyword searchable encryption scheme with constant bilinear pairing operations, short ciphertext and trapdoor. Later, many searchable encryption schemes have been designed to be used in various application scenarios, such as cloud computing [5, 6], smart grid [7, 8] and e-health record system [9, 10].

Up to date, most of the proposed searchable encryption schemes rely on the number theory assumptions such as discrete logarithm problem (DLP), big integer factorization problem and bilinear pairings with the Diffie-Hellman problem (DHP). Unfortunately, with the development of quantum computing, those number theory based hard problems will be solvable. Then, the security schemes with security functionality built on those hard problems will be absolutely undermined. Thus, it is necessary to construct new searchable encryption system that will be secure in the quantum age. Lattice based cryptography is a typical post-quantum cryptography and recently becomes a hot research topic in the public key cryptography field. In 2005, Regev *et al.* [11] defined a hardness assumption called learning from errors (LWE) problem and make the quantum reduction between LWE problem and a standard lattice hard problem, shortest vector problem (SVP). A LWE-based cryptography scheme was also given in [11] with a security proof. Since then, the LWE problem becomes a foundation of many lattice based cryptography systems, such as the public key encryption (PKE) scheme [12, 13], hierarchical identity-based encryption (HIBE) schemes [14-17] and other cryptography schemes [18-20].

In 2012, Zhang *et al.* [21] claimed that they have proposed the first lattice based searchable encryption scheme to resist quantum attack, which works based on LWE problem. However, their scheme suffers from some serious problems. The critical problem is that no public and private keys are generated for users. Then, Gu [22] and Hou [23] proposed two PEKS schemes based on lattice. However, they look also quite similar because their schemes are all constructed based on the scheme shown in [17]. Moreover, both of them can only support exact keyword search.

In this paper, we utilize the lattice delegation technique to construct a lattice based public key encryption scheme with semantic keyword search in order to provide a new primitive for the post-quantum age cloud computing. This scheme enables privacy-aware semantic keyword search over encrypted data. In real world application, it is quite common for a user to remember a synonym of the pre-extracted keyword, for example, "computer/PC" and "search/query". The available searchable encryption constructed on

lattice can only realize exact keyword search. If a user queries a synonym of the predefined keyword, the system cannot return the encrypted documents that they really want. In the proposed system, WordNet [25] database is utilized to construct the synonym set of keyword in order to support semantic keyword search. The size of public key, private key and ciphertext are $O(1)$. Based on the learning with errors (LWE) assumption, this scheme is proved secure against chosen keyword attack (CKA).

The remainder of the paper is structured as follows. In Section 2, we introduce the concept of lattice, the hardness assumption, the lattice delegation. In Section 3, we present the system model, the security model and the proposed lattice based searchable encryption scheme. Section 4 provides the security proof of the construction. Section 5 concludes this paper.

2. PRELIMINARIES

2.1 Notations

Let $A = [a_1, \cdots, a_m]$ be a matrix and $\|\cdot\|$ be the Euclidean norm. A norm of a matrix is defined as the norm of its longest column (*e.g.*, $\| A \| = \max_{i \in \{1,\cdots,m\}} \| a_i \|$). For any matrix A, $\tilde{A}$ represents the Gram-Schmidt Orthogonal matrix. D_α denotes the Gaussian distribution over R with parameter α. If O is a representation to classify the growth of functions, then $poly(n)$ denotes an unspecified function $f(n) = O(n^c)$ for some constant c. A function $g(n)$ is deemed as $\omega(f(n))$ if it grows faster than $c(f(n))$ for any constant $c > 0$.

2.2 Lattice

Let $A = [a_1, \cdots, a_m]$ denote a set of linearly independent vectors. An n-dimensional lattice generated by A is defined as $\Lambda = \{Ac = \sum_{i \in \{1,\cdots,n\}} c_i a_i, c_i \in Z\}$, where A plays the role of basis for this lattice.

A trapdoor basis of a lattice denotes a basis that vectors from this basis are the smallest vectors of this lattice. If the norms of vectors from a basis are small enough, they can be deemed as a trapdoor basis, which are usually kept secret by its owner in the cryptographic applications.

In our scheme, more attention will be paid to a special form of integer lattices denoted as Modular Lattice. For a prime number q and a vector $y \in Z_q^n$, we define $\Lambda_q^\perp(A) = \{x \in Z_q^m, Ax = 0 \ (\mathrm{mod}\, q)\}$ and $\Lambda_q^y(A) = \{x \in Z_q^m, Ax = y \ (\mathrm{mod}\, q)\}$.

2.3 Discrete Gaussian Distribution

Gaussian functions and Gaussian distribution over lattice are widely used in lattice-based cryptography. The Gaussian function on R^n centered at c with the parameter $\sigma > 0$ can be defined as $\rho_{\sigma,c}(x) = \exp(-\pi \| x - c \|^2 / \sigma^2)$.

For a matrix $A \in Z_q^{n \times m}$, the discrete Gaussian distribution over lattice $\Lambda_q^\perp(A)$ is defined as $D_{\Lambda_q^\perp(A),\sigma}(x) = \rho_{\sigma,c}(x) / \rho_{\sigma,c}(\Lambda_q^\perp(A))$. $D_{\Lambda_q^\perp(A),\sigma}(x)$ can be regarded as a "conditional" distribution that is resulted from sampling $x \in R^n$ from a Gaussian distribution with the parameter σ and under the condition of $x \in \Lambda_q^\perp(A)$.

For a fixed vector $y \in Z_q^n$ in the span of A, it will be useful to define the coset of $\Lambda_q^\perp(A)$ as $\Lambda_q^y(A) = \{x \in Z_q^m, Ax = y(\mathrm{mod}\, q))\} = t + \Lambda_q^\perp(A)(\mathrm{mod}\, q)$, where t is an arbitrary

solution over Z of equation $Ax = y(\text{mod } q)$.

The discrete Gaussian distribution over lattice $\Lambda_q^y(A)$ is defined as:

$$D_{\Lambda_q^y(A),\sigma}(x) = \rho_{\sigma,c}(x) / \rho_{\sigma,c}(t + \Lambda_q^y(A)).$$

$D_{\Lambda_q^y(A),\sigma}(x)$ can also be regarded as a "conditional" distribution that is resulted from sampling $x \in R^n$ from a Gaussian distribution with the parameter σ and under the condition of $AX = y(\text{mod } q)$.

2.4 Hardness Problems

Definition 1: (Learning with Errors (LWE) Problem) For the parameters (n, m, q), $s \in Z_q^n$ and an error distribution χ over Z_q^m, $A_{s,\chi}$ is a distribution obtained by computing $\{A, A^T s + x(modq)\}$ where $A \in Z_q^{n\times m}$ is randomly chosen and errors vector x is chosen in accordance with the error distribution χ. Learning with errors (LWE) problem is defined as follows. Given a sample from $A_{s,\chi}$, output s with a noticeable probability. The decisional LWE problem is to distinguish $A_{s,\chi}$ from a uniform distribution.

Definition 2: (Small Integer Solution (SIS) Problem) For the parameters (n, m, q, β) and a matrix $A \in Z_q^{n\times m}$, the small integer solution (SIS) problem is to find a non-zero integer vector $v \in Z_q^m$ such that $\|v\| \le \beta$ and $Av = 0(\text{mod } q)$.

2.5 Trapdoors for Lattices, Pre-image Sampling and Basic Delegation Algorithm

Alwen *et al.* [24] has shown a basis sampling algorithm to sample a nearly uniform matrix $A \in Z_q^{n\times m}$ together with a relatively short basis B of $\Lambda_q^{\perp}(A)$.

Lemma 1: [24] Let n, q, m, be positive integers with $q \ge 2$ and $m \ge 6n \lg q$. There exists a PPT algorithm *TrapGen* that outputs a pair $(A \in Z_q^{n\times m}, B \in Z_q^{m\times m})$ such that A is statistically close to uniform on $Z_q^{n\times m}$ and B is a basis of $\Lambda_q^{\perp}(A)$ such that $\| B \| \le m \cdot \omega(\sqrt{\log m})$ with all but $n^{\omega(1)}$ probability.

Lemma 2: [17] Let n, q, m, be positive integers with $q \ge 2$ and $m \ge 2n \lg q$. There exists a PPT algorithm *SamplePre* such that on input of $A \in Z_q^{n\times km}$, a basis Bs for $\Lambda_q^{\perp}(A)$, a vector $y \in Z_q^n$ and an integer $r \ge \| \tilde{B} \| \cdot \omega(\sqrt{\log m})$, the distribution of the output of $e \leftarrow SamplePre(A, B, y, r)$ is with negligible statistical distance of $D_{\Lambda_q^y(A),r}$.

Note: The *SamplePre* algorithm enables one to efficiently sample from the distribution $D_{Z^m,r}$ for any $r \ge \omega(\sqrt{\log m})$, by taking B to be the standard basis [16].

Let $A \in Z_q^{n\times km}$ and denote $A = [A_1, \cdots, A_k]$, where $A_i \in Z_q^{n\times m}$. For $S \subseteq \{1, \cdots, k\}$, $S = \{i_1, \ldots, i_j\}$, denote A_S as $[A_{i_1}, \cdots, A_{i_j}]$. The following procedure is used to generate a short basis of $\Lambda_q^{\perp}(A)$ from a short basis of $\Lambda_q^{\perp}(A_S)$.

Theorem 1: [16] Let n, m, q, k be positive integers with $q \ge 2$ and $m \ge 2n \lg q$. There exist a PPT algorithm *SampleBasis* such that on input of $A \in Z_q^{n\times km}$, a set $S \subseteq \{1, \cdots, k\}$, a

basis B_S for $\Lambda_q^{\perp}(A_S)$ and an integer $L \geq \| \tilde{B}_S \| \cdot \sqrt{km} \cdot \omega(\sqrt{\log km})$, outputs $B \leftarrow$ *SampleBasis*(A, B_S, S, L), such that, for an overwhelming fraction of $A \in Z_q^{n \times km}$, B is a basis of $\Lambda_q^{\perp}(A)$ with $\| \tilde{B} \| \leq L$ (with overwhelming probability). Furthermore, up to a statistical distance, the distribution of the basis B only depends on A and L and does not depend on B_S and S.

Given a short basis B_S for $\Lambda_q^{\perp}(A_S)$, the *GenSamplePre* is used to sample the preimage of the function $f_A(e) = Ae (\text{mod } q)$. The output of algorithm is within the negligible statistical distance of $D_{\Lambda_q^{\perp}(A),r}$, where $r \geq \| \tilde{B}_S \| \cdot \omega(\sqrt{\log km})$. Let $S = [s]$ for some $s \in [k]$ and $S^c = [k] \setminus S$, where $[k] = \{1, \cdots, k\}$ and $[s] = \{1, \cdots, s\}$. The sampling algorithm *GenSample* (A, B_S, y, r) proceeds as follows.

1. Sample $e_{S^c} \in Z^{(k-s)m}$ from the distribution $D_{Z^{(k-s)m},r}$ and set $z = y - A_{S^c} e_{S^c}$. Parse e_{S^c} as $[e_{s+1}, \cdots, e_k]$. This defines e_i for each $i \in S^c$.
2. Run $e_S \leftarrow SamplePre(A_S, B_S, z)$ from Lemma 2 to sample a vector $e_S \in Z^{sm}$ from the distribution $D_{\Lambda_q^y(A),r}$. Parse e_S as $[e_1, \cdots, e_s] \in (Z^m)^s$. This defines e_i for each $i \in S$.
3. Output $e \in Z^{km}$ as $e = [e_1, \cdots, e_k]$.

Theorem 2: [16] Let n, q, m, k be positive integers with $q \geq 2$ and $m \geq 2n \lg q$. There exists a PPT algorithm *GenSamplePre* such that on input of $A \in Z_q^{n \times km}$, a set $S \subseteq \{1, \ldots, k\}$, a basis B_S for $\Lambda_q^{\perp}(A_S)$, a vector $y \in Z_q^n$ and an intger $r \geq \| \tilde{B}_S \| \cdot \omega(\sqrt{\log km})$, the distribution of the output of $e \leftarrow GenSamplePre(A, B_S, S, y, r)$ is within negligible statistical distance of $D_{\Lambda_q^y(A),r}$ for an overwhelming fraction of $A \in Z_q^{n \times km}$.

The algorithm $SampleBasis(A, B_S, S, L)$ works as follows. It draws $O((km)^2)$ samples by running $GenSamplePre(A, B_S, S, y, r)$ many times, where $y = 0, r = L / \sqrt{km}$. With the overwhelming probability, the samples contain linearly-independent vectors and with length at most $r \cdot \sqrt{km} = L$. The algorithm then applies the deterministic procedure from Lemma 2.1 in [16] to process the samples into a basis for $\Lambda_q^{\perp}(A)$ without increasing the length of their Gram-Schmidt vectors.+

3. SEARCHABLE ENCRYPTION SCHEME FROM LATTICES

3.1 System Model

The system under the study mainly consists of three entities: a data sender, a remote data storage server and a data user as shown in Fig. 1. The data sender has a collection of files to be outsourced. Keywords should be extracted from the documents, which are encrypted to secure the index by a public key before outsourcing. Files are encrypted and attached with a secure index. Then these encrypted documents are outsourced to the remote data server. Besides the file storage service, the data storage server also provides search service for the authorized users over the encrypted documents. Any user with a valid private key has the authority to search. The data server is deemed to be honest but serious. The server is responsible to map the searching query of the data user to a set of encrypted documents through a test algorithm. At the same time, the server is also curious about the plaintext content and the keywords contained in the search query and en-

crypted documents. To resist the eavesdropping, the queried keywords should be protected and hidden by the user's private key and transformed into a keyword trapdoor. In addition, the test algorithm at the server should be "blind". It means that the data server should find all encrypted documents that match the query without knowing the underlying plaintext keyword.

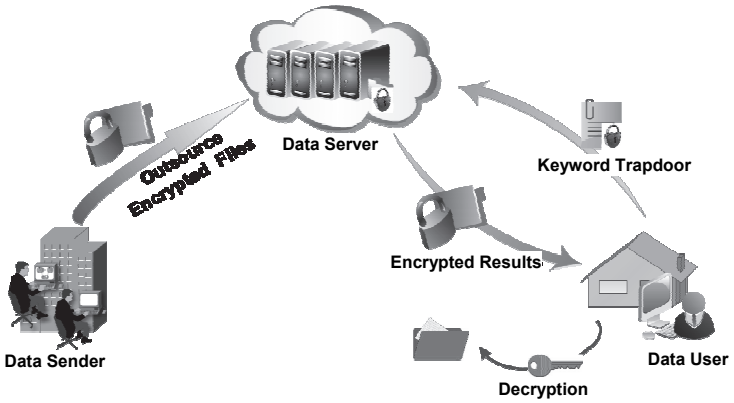

Fig. 1. System model.

The public key encryption scheme with keyword search consists of the following algorithms.

- *KeyGen*(κ) $\rightarrow$ (*pk*, *sk*): This algorithm takes an input of the security parameter κ and returns a public and secret key pair *pk*, *sk*.
- *PEKS*(*pk*, *KW*) $\rightarrow$ *CT*: This algorithm takes an input of the public key *pk* and the keyword *KW*, while it returns a ciphertext *CT* by *KW*.
- *Trapdoor*(*sk*, *KW*) $\rightarrow$ T_{KW}: This algorithm takes an input of the private key *sk* of the user and the keyword *KW*, while it outputs a trapdoor T_{KW}.
- *Test*(*pk*, *CT*, T_{KW}) $\rightarrow$ 1 or 0: This algorithm takes an input of the public key *pk* of the user, the ciphertext *CT* and the trapdoor T_{KW}, while it returns 1 if *KW* is included in *CT* and 0 otherwise.

3.2 Security Model

The semantic security of the scheme is defined as following to make sure that no information about the keyword *KW* will be leaked from $CT = PEKS(pk, KW)$. It is assumed that there exists an adversary $\mathcal{A}$ who is capable to obtain the trapdoor T_{KW} for any chosen. The adversary $\mathcal{A}$ is still unable to distinguish an encryption of a keyword KW_0 from an encryption of another keyword KW_1. Adversary $\mathcal{A}$ is considered to be successful if he wins the following interactive game.

- *KeyGen*: Challenger $\mathcal{C}$ runs the key generation algorithm *KeyGen*(κ) to generate (*pk*, *sk*) and sends adversary $\mathcal{A}$ the public parameters *pk*.
- *Phase* 1: Adversary $\mathcal{A}$ is able to adaptively ask the challenger $\mathcal{C}$ for the trapdoor T_{KW}

for any keyword $KW \in \{0,1\}^*$ of his choice.

- *Challenge*: When $\mathcal{A}$ decides that phase 1 is over, $\mathcal{A}$ sends the challenger $\mathcal{C}$ two keywords KW_1^*, KW_2^* on which he wants to be challenged. The restriction on the choice of KW_1^*, KW_2^* is that the trapdoors of $T_{KW_1^*}$ and $T_{KW_2^*}$ is not queried in phase 1. Then, the challenger $\mathcal{C}$ chooses a random $b \in \{0, 1\}$ and gives the attacker $\mathcal{A}$ a challenge ciphertext $CT^* = PEKS(pk, KW_b^*)$.
- *Phase* 2: Adversary $\mathcal{A}$ continues to issue trapdoor generation queries as in phase 1 with the constraint that the queried keyword $KW \neq KW_1, KW_2$.
- *Guess*: Finally, the adversary $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ and wins the game if $b' = b$.

If $\mathcal{A}$ somehow manages to guess the correct answer in the experiment above, then $\mathcal{A}$ wins the experiment and the scheme is not secure. We say that $\mathcal{A}$ has a guessing advantage ϵ (*i.e.*, the probability of $\mathcal{A}$ winning the experiment) is $Pr[b = b'] = 1/2 + \epsilon$.

Definition 4: A scheme is $(t, \varepsilon, q_T) - \text{IND} - \text{CKA}$ (indistinguishable against chosen keyword attack) secure if at all t time, adversaries making at most q_T trapdoor generation queries have advantage at most ϵ in winning the above game.

3.3 Proposed Scheme

In this subsection, we design a lattices based public key encryption with semantic keyword search scheme. In order to realize the semantic keyword search function, we utilize WordNet [25] to construct semantic keyword set. WordNet is a large lexical database of English created by Princeton University. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Utilizing WordNet, a keyword KW can be extended to its synonym set $\{KW, s_1, ..., s_n\}$, in which $s_1, ..., s_n$ are the synonyms of keyword KW. In this scheme, we re-arrange the synonym set to its lexicographical order and denote it as Υ_{KW}.

Let $k, m, n, q, t > 0$ be integers with $q \geq 2$, $m \geq 6n\lg q$. The parameter functions $L(k)$, $r(k)$, $\alpha(k)$ are defined as follows:

◊ The size of user's secret basis: $L(k) \geq L \cdot m^{k/2} \cdot \omega(\log^{k/2} m)$, $L \geq m \cdot \omega(\log n)$;
◊ Gaussian parameter for generating the short basis: $r(k) \geq L(k-1) \cdot \omega(\sqrt{\log m})$;
◊ Gaussian parameter for adding noise to ciphertext:

$$\alpha(k) \geq 1/(r(k) \cdot \sqrt{km+1} \cdot \omega(\sqrt{\log n})).$$

- *KeyGen*(κ) $\rightarrow$ (pk, sk): This algorithm takes the security parameter κ as an input, chooses a hash function $H_1 : \{0,1\}^* \rightarrow Z_q^{n \times m}$ and a random $v = (v_1, \cdots, v_{2t}) \in Z_q^{n \times 2t}$ and generates $A_0 \in Z_q^{n \times m}$ with a short basis $B_0 \in Z_q^{m \times m}$ ($||B_0|| < L$) for $\Lambda_q^{\perp}(A_0)$ according to the *TrapGen* algorithm shown in Lemma 1. It returns the public and secret key pair pk = (A_0, v), $sk = B_0$.
- *PEKS*(pk, KW) $\rightarrow$ CT: This algorithm takes the public key pk and the keyword $KW \in \{0, 1\}^*$ as inputs. It firstly extends KW to its lexicographic order synset Υ_{KW}. Then, the algorithm computes $A_{KW} = H_1(\Upsilon_{KW}) \in Z_q^{n \times m}$ and $Q_{KW} = [A_0 \,||\, A_{KW}] \in Z_q^{n \times 2m}$, randomly chooses $u \in Z_q^n$, $x_1 \leftarrow \chi$, $x_2 \leftarrow \chi^{2t}$, $\chi = \psi_{\alpha(k+1)}$, computes $p = Q_{KW}^T u + x_1 \in Z_q^{2m}$ and ran-

domly selects $\sigma, \beta \in \{0,1\}^t$. For $1 \le j \le 2t$, let $b_j = bit_j(\sigma \| \beta) \in \{0,1\}^t$ be the jth bit of $\sigma \| \beta$. It computes $c = v^T u + x_2 + (\sigma \| \beta) \cdot \lfloor q/2 \rfloor \in Z_q^{2t}$ and outputs $CT = (p, c, Q_{KW}, \sigma)$.

- *Trapdoor*$(sk, KW) \rightarrow T_{KW}$: This algorithm takes the private key sk of the user and the keyword KW as inputs, computes $A_{KW} = H_1(Y_{KW}) \rightarrow Z_q^{n \times m}$ and $Q_{KW} = [A_0 \| A_{KW}] \in Z_q^{n \times 2m}$ and generates a short basis for $\Lambda_q^{\perp}(Q_{KW})$ as *SampleBasis*$(Q_{KW}, B_0, S_0 = \{1\}, L(1)) \rightarrow B_{KW}, \| \tilde{B}_{KW} \| \le L(1)$, according to theorem 1. It then outputs the trapdoor $T_{KW} = B_{KW}$.
- *Test*$(pk, CT, T_{KW}) \rightarrow 1$ or 0: This algorithm takes public key pk of the user, the ciphertext CT and the trapdoor T_{KW} as inputs and generates $e_j \leftarrow GenSamplePre(Q_{KW}, T_{KW}, v_j, r(k+1)) \in Z^{2m}$, according to Theorem 2, where e_j is distributed according to $D_{\Lambda^{v_j}(Q_{KW}), r(k+1)}$. let $c_j = bit_j(c) \in Z_q$ be the jth element of c. For $1 \le j \le 2t$, it computes $b'_j = c_j - e_j^T p \in Z_p$. Let $\gamma_j = 0$ if $b_{j'}$ is closer to 0 than to $\lfloor q/2 \rfloor \in Z_q$. Otherwise, $\gamma_j = 1$. If $[\gamma_1, \cdots \gamma_t]$ equals to σ, the algorithm returns 1 to indicate that the KW is included in CT. It outputs 0, otherwise.

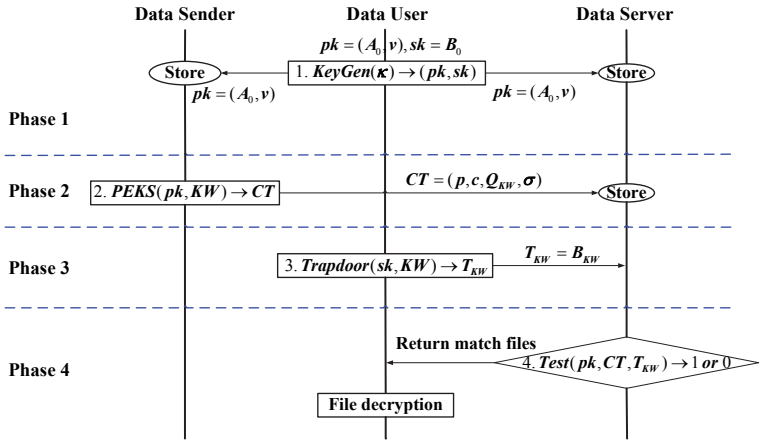

Fig. 2. Workflow of the scheme.

The workflow of the proposed scheme is shown in Fig. 2. The first key generation scheme is run by data user and generates public and private keys. The public key is then distributed to the data sender and the data server. The private key is kept secret by the data user. In the second phase, the data sender encrypts documents and the extracted keywords using the received public key. All of them will be sent to the data server. In the third phase, the data user wants to retrieve all the encrypted documents that contain certain keywords. A keyword trapdoor is generated with the user's private key and transmitted to the data server. In the last phase, the data server runs a test algorithm and returns all the matching files to the user. Those files can be decrypted by the user's private key. The encryption and decryption operations on the documents are free to be selected by the user and the data sender without specified in this scheme because a lot of symmetric encryption and decryption scheme can be selected to complete the work. The outsourced documents will be encrypted with a symmetric key. Then, the symmetric key is encapsulated within the user's public key. In this paper, we focus on exploring the way to carry out the keyword search operation on the encrypted files. The proposed scheme has constant communication and computation overheads.

4. ANALYSIS OF THE PROPOSED SCHEME

4.1 Parameters and Correctness

The correctness of the proposed scheme is inherited from the choice of parameters combined with Theorem 2.3 shown in [16] and the properties of the trapdoor functions [17].

- Since $e_j \leftarrow GenSamplePre(Q_{KW}, T_{KW}, v_j, r(k+1)) \in Z^{2m}$, we have $Q_{KW}e_j = v_j$ for $j = 1, \ldots, 2t$.
- If CT is a legally constructed ciphertext for keyword KW, then $p = (Q_{KW})^T u + x_1$, $c = v^T u + x_2 + (\sigma \| \beta) \cdot \lfloor q/2 \rfloor$.
- It is easy to find

$$\begin{aligned} b_{j'} &= c_j - e_j^T p \\ &= (v_j^T u + x_{2,j} + b_j \cdot \lfloor q/2 \rfloor) - e_j^T (Q_{KW}^T u + x_1) \\ &= (v_j^T u + x_{2,j} + b_j \cdot \lfloor q/2 \rfloor) - (e_j^T Q_{KW}^T u + e_j^T x_1) \\ &= (v_j^T u + x_{2,j} + b_j \cdot \lfloor q/2 \rfloor) - [(Q_{KW} e_j)^T u + e_j^T x_1] \\ &= (v_j^T u + x_{2,j} + b_j \cdot \lfloor q/2 \rfloor) - (v_j^T u + e_j^T x_1) \\ &= b_j \cdot \lfloor q/2 \rfloor + (x_{2,j} - e_j^T x_1). \end{aligned}$$

- Since $v_j^T = (Q_{KW} e_j)^T$. The formula $(x_{2,j} - e_j^T x_1)$ is the error term.

Due to the fact that $\| e_j \| \le \lambda\sqrt{m} = \sigma m\omega(\sqrt{logm})$ and Lemma 19 in [14], we have

$$\begin{aligned} | x_{2,j} - e_j^T x_1 | &\le \| e_j \| (q\alpha\omega(\sqrt{logm}) + \sqrt{m}/2) \\ &\le \sigma m\omega(\sqrt{logm})(q\alpha\omega(\sqrt{logm}) + \sqrt{m}/2) \\ &\le \sigma q\alpha m\omega(logm) + \sigma(m)^{3/2}\omega(\sqrt{logm}) \end{aligned}$$

In order to make the scheme work correctly, we need the following requirements.

- Algorithm *TrapGen* is able to operate, then $m \ge 6nlogq$.
- The error term $| x_{2,j} - e_j^T x_1 |$ should no more than $q/5$ ($i.e.\ \sigma q\alpha m\omega(\log m) + \sigma(m)^{3/2}\omega(\sqrt{\log m}) \le q/5$).
- Regev's LWE reduction can operate, then $q > 2\sqrt{n}/\alpha$.

We set the parameters (q, m, σ, α) as follows to satisfy the requirements (assume $n^\delta > \lceil \log q \rceil = O(\log n)$). Using the similar techniques in [15], we set $m = 6n^{1+\delta}$, $q = m^{2.5} \cdot \omega(\log n)$, $\sigma = m \cdot \omega(\sqrt{logn})$ and $\alpha = 1/(m^2\omega(\sqrt{logn}))$. Due to m is an integer and q is a prime number, we should round up m to the nearest larger integer and q to the nearest larger prime number.

4.2 Comparison

In this subsection, this scheme is compared with the existing lattice based searcha-

ble encryption schemes [22, 23] in Table 1. Due to the serious problems mentioned in introduction, the scheme in [21] is not included in the comparison. We can easily find that these three schemes all have constant size of public keys, ciphertext and trapdoor.

The difference is that this scheme could support semantic keyword search. If the user queries keyword "revoke", this system will return all confidential files that contain keywords semantically approximate to "revoke" (for instance, "withdraw", "abolish", "recall", "repeal", "rescind"). Semantic keyword search is a very useful function in real application. However, the schemes in [22, 23] can only support exact keyword search.

To conclude, the suggested scheme has the same security and efficiency level as the schemes in [22, 23]. Furthermore, semantically keyword search function is realized in this scheme.

Table 1. Comparison of lattice based searchable encryption scheme.

Scheme	Synonym query	Lattice-based	Public key size	Private key size	Trapdoor size	Security
[22]	No	Yes	O(1)	O(1)	O(1)	ROM
[23]	No	Yes	O(1)	O(1)	O(1)	ROM
Ours	**Yes**	**Yes**	**O(1)**	**O(1)**	**O(1)**	**ROM**

ROM: random oracle model

4.3 Security Analysis

In this subsection, we prove that the security of the scheme can be reduced to the hardness of LWE problem. In the formal security proof, the adversary is deemed as a semi-trusted data server or a vicious outside attacker. For both of them, public key can be obtained and the private keys of authorized users are unknown. In the key generation phase of the proof, the public key is sent to adversary and the private key of user is kept secret.

The analysis of unauthorized data user is different. The "unauthorized data user" means that the adversary is a user in the system but not a legal member to access the data that do not belong to him. He can search on his own documents. The encrypted files that belong to others should not be queried by him. This adversary has his own private key. But he cannot succeed to make a legal keyword trapdoor for others' documents. We give out a separate security analysis of unauthorized data user.

4.3.1 Formalized proof

Theorem 3: Let $q \geq 5r(2)(m+1)$ and $m \geq 2n \lg q$. The proposed scheme is secure against chosen keyword attacks assuming $LWE_{q,\chi}$ is intractable, where $\chi = \psi_{\alpha(2)}$.

***Proof*:** Suppose there is an adversary $\mathcal{A}$ breaking our scheme with advantage $Adv_{\mathcal{A}}(\kappa)$. We now construct a challenger $\mathcal{C}$ that has with advantage $Adv_{\mathcal{C}}^{LWE}(\kappa)$ in solving the $LWE_{q,\chi}$ problem, where $Adv_{\mathcal{C}}^{LWE}(\kappa) \geq Adv_{\mathcal{A}}(\kappa) / Q_{H_1} - negl$, Q_{H_1} is the total number of queries on H_1.

KeyGen: Challenger $\mathcal{C}$ runs the key generation algorithm $KeyGen(\kappa)$ to generate (pk, sk). Challenger $\mathcal{C}$ first obtains $2m+1$ samples $(a_j, b_j) \in Z_q^n \times Z_q$ $(1 \le j \le 2m+1)$ from the LWE oracle, in which all $a_j \in Z_q^n$ are randomly chosen and either all $b_j \in Z_q$ are also randomly selected or all equal to $a_j^T s + x_j$. In the above sampling process, $s \in Z_q^n$ is a uniform secret and x_j is the independent Gaussian Noise that is selected in accordance with error distribution χ. Then $\mathcal{C}$ denotes the LWE samples $(a_j, b_j) \in Z_q^n \times Z_q$ $(1 \le j \le 2m)$ as $(A_i^*, p_i^*) \in Z_q^{n \times m} \times Z_q^m$ $(0 \le i \le 1)$ and $(v^*, c^*) = (a_{2m+1}, b_{2m+1}) \in Z_q^n \times Z_q$. The public key is $pk = (A_0, v) = (A_0^*, v^*)$. The secret key sk is set as the short basis for $\Lambda_q^{\perp}(A)$ which is unknown to adversary $\mathcal{A}$. Then, the public key pk is sent adversary $\mathcal{A}$.

Phase 1: Attacker $\mathcal{A}$ adaptively issues the following queries.

- **Hash queries:** On the ith hash query from the adversary $\mathcal{A}$ on keyword KW_i, challenger $\mathcal{C}$ runs the algorithm *TrapGen* to generate $A_i \in Z_q^{n \times m}$ and the corresponding trapdoor basis $B_i \in Z_q^{m \times m}$, in which A_i is statically selected to uniform over $Z_q^{n \times m}$. Then, challenger $\mathcal{C}$ returns A_i to $\mathcal{A}$ and stores the tuple $\langle KW_i, A_i, B_i \rangle$ in the list $\mathcal{H}_1$.
- **Extract queries:** On the trapdoor generation query from the adversary $\mathcal{A}$ on keyword KW, it is assumed that $\mathcal{A}$ has already made a hash query on KW_i. If the tuple $\langle KW_i, A_i, B_i \rangle$, is included in the list $\mathcal{H}_1$, challenger $\mathcal{C}$ computes a properly distributed basis B_{KW_i} corresponding to $A_{KW_i} = [A_0 \| A_i]$ by running $B_{KW_i} \leftarrow SampleBasis(A_{KW_i}, B_i, S_0 = \{1\}, L(1))$. If the generation is successful, then $\mathcal{C}$ returns B_{KW_i}. Otherwise, $\mathcal{C}$ aborts.

Challenge: When $\mathcal{A}$ decides that phase 1 is over, $\mathcal{A}$ sends the challenger $\mathcal{C}$ two keywords KW_1^*, KW_2^* on which he wants to be challenged. The restriction on the choice of KW_1^*, KW_2^* is that the trapdoors of $T_{KW_1^*}$ and $T_{KW_2^*}$ has not been queried in phase 1. Then, the challenger $\mathcal{C}$ chooses a random $b \in \{0,1\}$ and gives the attacker $\mathcal{A}$ a challenge ciphertext $CT^* = PEKS(pk, KW_b^*) = (p^*, c, Q_{KW_b^*}, \sigma^*)$, where $p^* = (p_0^*, p_1^*)$, $c = c^* + (\sigma^* \| b^*) \cdot \lfloor q/2 \rfloor$, $Q_{KW_b^*} = [A_0^* \| A_1^*]$, $\sigma^* \in \{0,1\}^t$ and $b^* \in \{0,1\}$ are randomly chosen.

Phase 2: Adversary $\mathcal{A}$ continues to issue trapdoor generation queries as in phase 1 with the constraint that the queried keyword $KW \neq KW_1, KW_2$.

Guess: Finally, the adversary $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$, $\mathcal{C}$ returns genuine if $b' = b^*$ or random if $b' \neq b^*$ as its answer for the LWE problem.

The distribution of the challenge ciphertext is statistically close to the real security environment because $Q_{KW_b^*}$, p^* and c^* are all constructed using the LWE instances. The challenge ciphertext C^* will have the same distribution as in the LWE game if LWE instances are genuine. If LWE instances are random elements, so will be the elements in C^*. From the above proof, we can see that if adversary $\mathcal{A}$ is able to break this scheme, then $\mathcal{A}$ could also break the LWE problem.

4.3.2 Malicious data user

The proposed scheme could prevent attacks unauthorized data subscribers. It is capable to deals with the problem of the unwarrantable data search by vicious users. For the purpose of searching on the encrypted files, the keyword should be hidden with the help of the secret key. However, only the legitimate data user has a proper secret key.

Any keyword trapdoor generated by an unauthorized data user cannot be tested to be successful. The fake trapdoor and the encrypted keyword ciphertext will not match each other. The hidden keyword retrieval request is only conform to the encrypted keywords if the trapdoor is also encrypted with the identical secret key. The mismatch will prevent the malicious data user from further illegal operations on the encrypted files. Even if the malicious data user conspires with the data server, the actions will not help the illegal user to increase the probability to produce a valid keyword trapdoor. The collusion will not bring about any useful information to them either.

5. CONCLUSIONS

In this paper, we have proposed a novel searchable encryption scheme which has its security functionality based on the lattice assumption rather than the bilinear map related assumption. The proposed scheme, motivated by the lattice basis delegation method, will incur a constant communication and computation overhead. We have also performed the security analysis on the proposed scheme. It has been proved to be secure against the chosen keyword attacks. It can be deemed as a candidate primitive that will still be secure for future cloud even in post-quantum age.

ACKNOWLEDGEMENTS

This research is supported by National Natural Science Foundation of China (614-02112, 61472307, 61472309).

REFERENCES

1. D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proceedings of International Conference on Theory and Applications of Cryptographic Techniques*, Vol. 3027, 2004, pp. 506-522.
2. C. Gu, Y. Zhu, and Y. Zhang, "Efficient public key encryption with keyword search schemes from pairings," in *Proceedings of International Conference on Information Security and Cryptology*, Vol. 4990, 2007, pp. 372-383.
3. B. Long, D. Gu, N. Ding, and H. Lu, "On improving the performance of public key encryption with keyword search," in *Proceedings of International Conference on Cloud and Service Computing*, 2012, pp. 143-147.
4. Z. Chen, C. Wu, D. Wang, and S. Li, "Conjunctive keywords searchable encryption with efficient pairing, constant ciphertext and short trapdoor," in *Proceedings of International Conference on Intelligence and Security Informatics*, Vol. 7299, 2012, pp. 176-189.
5. Z. Liu, Z. Wang, X. Cheng, C. Jia, and K. Yuan, "Multi-user searchable encryption with coarser-grained access control in hybrid cloud," in *Proceedings of International Conference on Emerging Intelligent Data and Web Technologies*, 2013, pp. 249-255.
6. C. Orencik and E. Savas, "Efficient and secure ranked multi-keyword search on encrypted cloud data," in *Proceedings of Extending Database Technology Workshops*,

2012, pp. 186-195.
7. C. Lee, H. Yang, B. Lee, and D. Won, "A novel privacy-enhanced AMI system using searchable and homomorphic encryption techniques," in *Proceedings of International Conference on Convergence and Hybrid Information Technology*, 2012, pp. 608-617.
8. M. Wen, R. Lu, J. Lei, H. Li, X. Liang, and X. S. Shen, "SESA: an efficient searchable encryption scheme for auction in emerging smart grid marketing," *Security and Communication Networks*, 2013, Wiley Online Library, DOI: 10.1002/sec.699.
9. G. Hsieh and R. J. Chen, "Design for a secure interoperable cloud-based personal health record service," in *Proceedings of International Conference on Cloud Computing Technology and Science*, 2012, pp. 472-479.
10. M. Li, S. Yu, N. Cao, and W. Lou, "Authorized private keyword search over encrypted data in cloud computing," in *Proceedings of International Conference on Distributed Computing Systems*, 2011, pp. 383-392.
11. O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of Symposium on Theory of Computing*, 2005, pp. 84-93.
12. C. Gentry, S. Halevi, and V. Vaikuntanathan, "A simple BGN-type cryptosystem from LWE," in *Proceedings of International Conference on Theory and Applications of Cryptographic Techniques*, Vol. 6110, 2010, pp. 506-522.
13. C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem," in *Proceedings of Symposium on Theory of Computing*, 2009, pp. 333-342.
14. S. Agrawal, D. Boneh, and X. Boyen, "Efficient lattice (H)IBE in the standard model," in *Proceedings of International Conference on Theory and Applications of Cryptographic Techniques*, Vol. 6110, 2010, pp. 553-572.
15. S. Agrawal, D. Boneh, and X. Boyen, "Lattice basis delegation in fixed dimension and shorter-ciphtertext hierarchical IBE," in *Proceedings of Cryptology Conference*, Vol. 6223, 2010, pp. 98-115.
16. D. Cash, D. Hofheinz, and E. Kiltz, "How to delegate a lattice basis," http://eprint.iacr.org/2009/351.
17. C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of Symposium on Theory of Computing*, 2008, pp. 197-206.
18. S. Goldwasser, Y. Kalai, C. Peikert, and V. Vaikuntanathan, "Robustness of the learning with errors assumption," in *Proceedings of International Conference on Innovations in Computer Science*, 2010, pp. 230-240.
19. S. D. Gordon, J. J. Katz, and V. Vaikuntanathan, "A group signature scheme from lattice assumptions," in *Proceedings of International Conference on Theory and Application of Cryptology and Information Security*, Vol. 6477, 2010, pp. 395-412.
20. W. Jin and J. Bi, "Lattice-based identity-based broadcast encryption," http://eprint.iacr.org/2010/288.
21. J. Zhang, B. Deng, and X. Li, "Learning with error based searchable encryption scheme," *Journal of Electronics* (*China*), Vol. 29, 2012, pp. 473-476.
22. C. Gu, Y. Guang, Y. Zhu, and Y. Zheng, "Public key encryption with keyword search from lattices," *International Journal of Information Technology*, Vol. 19, 2013, pp. 1-10.
23. C. Hou, F. Liu, H. Bai, and L. Ren, "Public key encryption with keyword search from lattice," in *Proceedings of the 8th International Conference on P2P*, *Parallel*,

Grid, *Cloud and Internet Computing*, 2013, pp. 336-339.
24. J. Alwen and C. Peikert, "Generating shorter bases for hard random lattices," *Theory of Computing Systems*, Vol. 48, 2011, pp. 535-553.
25. WordNet Documentation [EB/OL], http://wordnet.princeton.edu/wordnet/documentation/.

Yang Yang (杨旸) is a Lecture in the College of Mathematics and Computer Science at Fuzhou University, China. Her research interests are in the area of information security and privacy protection.

Maode Ma (马懋德) is an Associate Professor at School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His interests are in the area of wireless networks, mobile computing, security and privacy.