Journal of Southeast University (English Edition) Vol. 39, No. 2, pp. 169 - 175 June 2023 ISSN 1003—7985

An LWE-based verifiable multi-keyword search scheme in cloud storage

Wang Pan Jiang Rui

(School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China)

Abstract: To solve the problems in public key encryption with the keyword search (PEKS) algorithm, a learning with errors based verifiable multi-keyword search (LWE-VMKS) scheme is proposed. Firstly, the LWE-VMKS scheme applies lattice-based algorithms to encrypt the keywords index to resist quantum computing attacks. Then, the LWE-VMKS scheme combines multiple keywords in a single search query to achieve a multi-keyword search. Subsequently, the LWE-VMKS scheme implements the lattice-based signatures and merges them to enable users to verify the correctness of the search result without decrypting the ciphertext. In addition, the scheme applies trapdoor functions to generate different keys for different data owners to withstand keyword guessing attacks(KGA). Finally, the LWE-VMKS scheme is formally proven to be secure against a quantum computing attack. It also realizes highly efficient multi-keyword searches, achieves verification for searched results, and is secure against KGA.

Key words: lattice-based cryptography; learning with errors (LWE); public key encryption with keyword search (PEKS); multi-keyword search; verifiability; keyword guessing attacks (KGA)

DOI: 10.3969/j.issn.1003-7985.2023.02.008

With the rapid development of network technology, the security of cloud computing is becoming increasingly significant. Many data owners or companies choose to store data in the cloud to reduce large management and storage costs, and legal users can then flexibly access the data via the Internet. However, cloud server providers are not always trustworthy and can become curious about the stored data. Some outsourced data are extremely sensitive, and as maintaining the privacy of these data is essential, owners are reluctant to store them in the cloud.

To solve this problem, data encryption is required prior to it being outsourced to a cloud server in real scenarios. However, this leads to the difficulty of data processing and involves searching over the cloud. In this respect, Boneh et al.[1] provided an asymmetric scheme known as public key encryption with keyword search (PEKS) to retrieve encrypted data. This scheme was novel, as it allowed the data owner to delegate search abilities to users while maintaining data encryption. The PEKS algorithm aroused great attention in the field of cloud computing, and a large number of schemes based on the PEKS algorithm have been proposed to date[2-3]. However, these schemes[2-3] cannot resist keyword guessing attacks (KGA), and therefore, are not fit for practical application in a cloud storage environment. Therefore, studies were conducted to enrich flexibility and security, and a multi-keyword search[4-5], a verifiable search[6-7], and an anti-KGA search[8] were proposed. Unfortunately, almost all PEKS schemes are based on the Diffie-Hellman problem or discrete Logarithm problem, which may result in data leakage if a quantum computing attack occurs. Therefore, determining how to resist a quantum computing attack has become important due to the developments in quantum computing technology.

The learning with errors (LWE)[9] problem is based on the lattice problem and was proposed to withstand quantum computing attacks. Researchers then began to apply the LWE problem to the PEKS scheme. With the help of the LWE problem, Zhang et al.[10], Yu et al.[11] and Behnia et al.[12] proposed LWE-based PEKS schemes to resist quantum computing attacks, and their schemes further improved the security associated with PEKS. Xu et al.[13] provided a multiple data owner scheme based on the LWE problem to expand the application scope of multiple data owners. However, the schemes proposed by these four studies could not achieve multi-keyword searches, which could lead to the problem of a single keyword search generating many irrelevant results with high computation costs. The schemes were also unable to resist KGA, which can lead to information leakage. Zhang et al.[14] then proposed a multi-keyword search scheme to enable more accurate search results and reduce the computation cost for the user; however, the scheme was not successful in finding all files containing the searched keywords. Zhang et al.[15] subsequently proposed an anti-KGA PEKS scheme, but this scheme could not realize multi-keyword

Received 2022-12-01, **Revised** 2023-03-03.

Biographies: Wang Pan (1998—), male, graduate; Jiang Rui (corresponding author), male, doctor, professor, R. Jiang@seu.edu.cn.

Foundation items: The National Natural Science Foundation of China (No. 61372103), the Natural Science Foundation of Jiangsu Province (No. BK20201265), the National Engineering Research Center of Classified Protection and Safeguard Technology for Cyber security (No. C21640-2).

Citation: Wang Pan, Jiang Rui. An LWE-based verifiable multi-keyword search scheme in cloud storage[J]. Journal of Southeast University (English Edition), 2023, 39(2): 169 - 175. DOI: 10.3969/j.issn.1003-7985.2023.02.008.

searches. Wang et al.[16] proposed a multi-keyword search scheme, but it had a high computation cost for the cloud server, and it could not resist KGA. In addition, none of the above-mentioned studies[10-16] enabled the correctness of the search results to be verified for the user prior to decrypting the ciphertext. Finally, Mei et al.[17] proposed a scheme that verified the correctness of the search results, but it was unable to resist KGA or achieve multi-keyword searches.

In short, none of the above schemes provide a secure, highly efficient, multi-keyword search within a cloud storage environment. In addition, nearly all of the schemes are unable to resist KGA. Furthermore, most of the existing schemes based on the lattice cannot verify the correctness of the search results without decrypting the ciphertext, and this is problematic for a cloud server that is untrustworthy or for a system encountering failure, which may return the wrong search results. To solve these problems, we propose an LWE-based verifiable multi-keyword search (LWE-VMKS) scheme.

1 Preliminaries

In this section, we provide the background to lattices and PEKS and some of the important algorithms that are applied in this paper.

1.1 Lattice

Definition 1 (lattice)[9] Given n linear independent basis vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n \in \mathbf{Z}^n$, let $\boldsymbol{B} = [\boldsymbol{b}_1 \quad \boldsymbol{b}_2 \quad \cdots \quad \boldsymbol{b}_n]$, the full-rank lattice Λ generated by $\boldsymbol{B}$ is the infinite periodic set as $\Lambda = \mathcal{L}(\boldsymbol{B}) = \left\{ \sum_{i=1}^{n} x_i \boldsymbol{b}_i \mid x_i \in \mathbf{Z}, 1 \leqslant i \leqslant n \right\}$.

Definition 2 (q-ary lattice)[9] For q prime, $\boldsymbol{A} \in \mathbf{Z}_q^{n \times m}$ and $\boldsymbol{u} \in \mathbf{Z}_q^n$, it is defined as $\Lambda_q^{\perp}(\boldsymbol{A}) = \{ \boldsymbol{e} \in \mathbf{Z}^m \mid \boldsymbol{A}\boldsymbol{e} = 0 \bmod q \}$ and $\Lambda_q^{u}(\boldsymbol{A}) = \{ \boldsymbol{e} \in \mathbf{Z}^m \mid \boldsymbol{A}\boldsymbol{e} = \boldsymbol{u} \bmod q \}$.

1.2 Learning with errors (LWE)

Definition 3 (LWE)[9] For a security parameter λ, an integer $q \geqslant 2$, a secret vector $\boldsymbol{s}$, and a dimension $n \in \mathbf{Z}_q^n$, χ is a distribution over $\mathbf{Z}$. The LWE problem is used to distinguish two distributions, one distribution is $[\boldsymbol{a}_i \mid b_i]$, which is uniformly sampled from $\mathbf{Z}_q^{n+1}$ (where $\boldsymbol{a}_i \in \mathbf{Z}_q^n$ and $b_i \in \mathbf{Z}_q$), and the other distribution samples $[\boldsymbol{a}_i \mid b_i]$ by sampling $\boldsymbol{a}_i \in \mathbf{Z}_q^n$ uniformly, $e_i \in \chi$, and set $b_i = \boldsymbol{a}_i \cdot \boldsymbol{s} + e_i$. The LWE assumption is that this problem is infeasible.

1.3 Discrete Gaussians

Definition 4 (discrete Gaussian distribution)[9] For a vector, $\boldsymbol{c} \in \mathbf{Z}^m$ and $\sigma \in \mathbf{R}$, $\rho_{\sigma,c} = \exp\left(-\pi \frac{\| \boldsymbol{x} - \boldsymbol{c} \|}{\sigma^2} \right)$ denotes an n-dimensional probability density function of a Gaussian distribution with center $\boldsymbol{c}$, and variance σ^2. For a given lattice, $\sigma > 0$ is a parameter, and $D_{\Lambda,\sigma,c}(\boldsymbol{x}) = \rho_{\sigma,c}(\boldsymbol{x}) / \rho_{\sigma,c}(\Lambda)$ denotes the discrete Gaussian distribution in the lattice, $\Lambda(\boldsymbol{A})$, with center $\boldsymbol{c}$, and variance σ^2, where $\boldsymbol{x} \in \Lambda$.

1.4 Trapdoor functions

TrapGen $(\overline{\boldsymbol{A}}, \boldsymbol{Q})$[18] Given a random matrix $\overline{\boldsymbol{A}} \in \mathbf{Z}_q^{n \times \bar{m}}$ for some $\bar{m} \geqslant 1$, and an invertible matrix $\boldsymbol{Q} \in \mathbf{Z}_q^{n \times n}$, the function TrapGen$(\overline{\boldsymbol{A}}, \boldsymbol{Q})$ outputs a key pair $\{ \boldsymbol{A} \in \mathbf{Z}_q^{n \times m}, \boldsymbol{T} \in \mathbf{Z}_q^{m \times m} \}$, where $\boldsymbol{T}$ is the basis of $\Lambda^{\perp}(\boldsymbol{A})$.

NewBasisDel $(\boldsymbol{A}, \boldsymbol{T}_A, \sigma)$[19] Given a matrix $\boldsymbol{A} \in \mathbf{Z}_q^{n \times m}$, a short basis, and $\boldsymbol{T}_A \in \Lambda^{\perp}(\boldsymbol{A})$, a parameter $\sigma \geqslant \| \tilde{\boldsymbol{T}}_A \| \sqrt{mn \log q} \omega((\log m)^{5/2})$, the function NewBasisDel$(\boldsymbol{A}, \boldsymbol{T}_A, \sigma)$ outputs a matrix $\boldsymbol{R}$, and a short basis $\boldsymbol{T}_B$ of $\boldsymbol{B}$, where $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{R}^{-1}$.

LeftSample$(\boldsymbol{A}, \boldsymbol{A}_1, \boldsymbol{T}_A, \boldsymbol{u}, \sigma)$[20] Given a matrix $\boldsymbol{A} \in \mathbf{Z}_q^{n \times m}$, a matrix $\boldsymbol{A}_1 \in \mathbf{Z}_q^{n \times m_1}$, a matrix $\boldsymbol{T}_A \in \mathbf{Z}_q^{m \times m}$, which is a short basis of $\Lambda^{\perp}(\boldsymbol{A})$, a vector $\boldsymbol{u} \in \mathbf{Z}_q^n$, and a Gaussian parameter $\sigma \geqslant (\| \tilde{\boldsymbol{T}}_A \| \omega(\sqrt{\log(m + m_1)}))$, the function LeftSample$(\boldsymbol{A}, \boldsymbol{A}_1, \boldsymbol{T}_A, \boldsymbol{u}, \sigma)$ outputs $\boldsymbol{f} \in \mathbf{Z}^{m+m_1}$ to satisfy $[\boldsymbol{A} \mid \boldsymbol{A}_1]\boldsymbol{f} = \boldsymbol{u} (\bmod p)$ and $\| \boldsymbol{f} \| \leqslant \sigma \sqrt{m + m_1}$.

Lemma 1[13] For a key pair $\boldsymbol{V}_{pk} \in \mathbf{Z}_{2q}^{m \times n}$ and $\boldsymbol{V}_{sk} \in \mathbf{Z}_{2q}^{n \times m}$, and an appropriate big integer τ, an anti-quantum computing attack algorithm samples $\boldsymbol{z} \in \mathbf{Z}_q^m$ and $\boldsymbol{y} \in \{-1, 0, 1\}^n$ with probability $1 \Big/ \left[X \exp\left(-\frac{\| \boldsymbol{V}_{sk}\boldsymbol{y} \|^2}{2\sigma^2} \right) \right] \cdot \cosh\left(\frac{\langle \boldsymbol{z}, \boldsymbol{V}_{sk}\boldsymbol{y} \rangle}{\sigma^2} \right)$, where $\sigma = \tau \| \boldsymbol{V}_{sk}\boldsymbol{y} \|$ and $X = \exp(1 + 1/(2\tau^2))$. Furthermore, $\boldsymbol{z}$ satisfies $\| \boldsymbol{z} \| \leqslant B_2 = 1.4\sqrt{m}\sigma$ and $\| z_i \| < q/4$, where $\| \boldsymbol{z}_i \|$ is the i-th norm of vector $\boldsymbol{z}$.

Lemma 2[9] Let $\alpha \in (0,1)$ be a real number and q a prime number, such that $\alpha q > 2\sqrt{m}$. Assume that an efficient algorithm exists that can solve the LWE problem with a Gaussian distribution with mean α. An efficient quantum algorithm then also exists to solve the worst case of the SVP and CVP problems.

Lemma 3[9] Let n be a prime number and $\boldsymbol{\eta} \in \mathbf{Z}_q^n$, such that $\boldsymbol{\eta} \cdot \mathbf{Z}_q^n$ is coprime with q, then the function $\boldsymbol{\eta}: \mathbf{Z}_q^n \rightarrow \mathbf{Z}_q^n$ is defined as $\boldsymbol{\eta}(\boldsymbol{\alpha}_i) = \boldsymbol{\eta} \cdot \boldsymbol{\alpha}_i$ to induce an isomorphism from $\mathbf{Z}_q^n$ to $\mathbf{Z}_q^n$.

Lemma 4[21] Let Λ be a lattice, σ be an isomorphism mapping $\mathbf{Z}_q^n$ to lattice Λ, and $r \geqslant \sqrt{2}\mu_{\varepsilon}(\Lambda)$ for some negligible ε, for $\boldsymbol{z}$ is distributed by $D_{\Lambda,r}$, χ is distributed by $D_{r'}$ with $r' \geqslant r \| \boldsymbol{x} \|$, then the distribution of $\boldsymbol{z} \cdot \boldsymbol{x} + \chi$ is within a negligible statistical distance of the Gaussian distribution D_r, where $r_i^2 = r^2 \mid \sigma_i(\boldsymbol{x}) \mid^2 + (r')^2$.

Lemma 5[18] The inhomogeneous small integer solution problem (ISIS) is defined as follows: given an inte-

ger, q, a matrix, $\boldsymbol{A} \in \mathbf{Z}_q^{n\times m}$, a real β, and a vector $\boldsymbol{u} \in \mathbf{Z}_q^n$, the integer vector $\boldsymbol{f} \in \mathbf{Z}_q^m$ can be found so that $\boldsymbol{Af} = \boldsymbol{u} \bmod q$ and $\| \boldsymbol{f} \| \leqslant \beta$. The ISIS assumption is that this problem is infeasible.

2 Proposed LWE-VMKS Scheme

2.1 System model

Our LWE-VMKS scheme comprises of four entities, as shown in Fig. 1. The key generation center (KGC) generates and manages the search of key pairs and verifies them. Data owners possess the data files with encrypted keyword indexes and upload the ciphertexts, encrypted keyword signatures to the cloud server. The cloud server stores the ciphertext and helps the users to find the ciphertext. Each user can retrieve the data files and verify whether the searched results are correct.

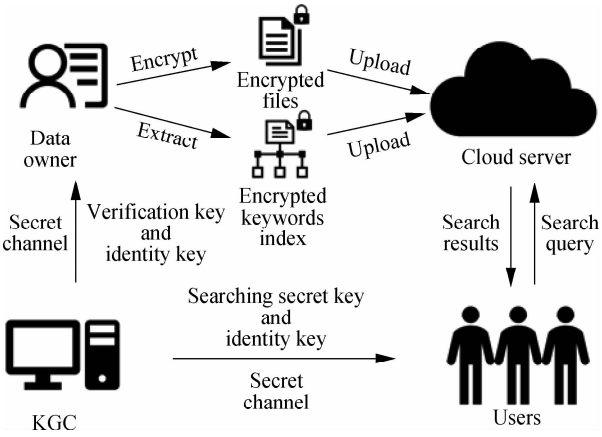

Fig. 1 System model

2.2 Threat model

We define the threat model as follows. The data owners and KGC are trusted. They will not leak secret information.

The cloud server is semi-trusted, which means that the cloud server may correctly perform all the appropriate operations within the scheme, but it may be curious about the secret data.

Online attackers are malicious, which means that they may eavesdrop on the information transmitted within the network, try to obtain private data files, and illegally obtain search abilities. In addition, online attackers may launch keyword guessing attacks to obtain the keywords searched by the user.

2.3 Details of our LWE-VMKS scheme

2.3.1 System initialization

The KGC chooses a modulus q, lattice dimension parameter n, $m = n\log q$, and an error distribution χ appropriately for LWE. The KGC then selects a random matrix $\overline{\boldsymbol{A}} \in \mathbf{Z}_q^{n\times \bar{m}}$ for $\bar{m} \geqslant 1$, and an invertible matrix $\boldsymbol{Q} \in \mathbf{Z}_q^{n\times n}$. The KGC then runs TrapGen$(\overline{\boldsymbol{A}}, \boldsymbol{Q})$ to generate a matrix as the master public key $\boldsymbol{M}_{pk} = \boldsymbol{A}_0$, and a matrix as the master secret key $\boldsymbol{M}_{sk} = \boldsymbol{T}_{A_0}$.

The KGC then chooses $\sigma_1 \geqslant \| \tilde{\boldsymbol{T}}_A \| \sqrt{mn\log q}\, \omega((\log m)^{5/2})$, $\sigma_2 > \| \boldsymbol{T}_{A_0} \| \omega(\sqrt{\log(2m)})$, a random vector $\boldsymbol{u} \in \mathbf{Z}_q^n$, a random matrix $\boldsymbol{D} \in \mathbf{Z}_q^{1\times m}$, and hash functions $\boldsymbol{H}_1 : \{0,1\}^l \to \mathbf{Z}_q^n$ and $\boldsymbol{H}_2 : \{0,1\}^* \to \{-1,0,1\}^n$. The KGC also sets the keyword $\boldsymbol{K}_{ws} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_N\}$, where $\boldsymbol{w}_j \in \{0,1\}^l$.

Finally, the KGC sets $p_{\mathrm{p}} = \{n, m, q, \boldsymbol{A}_0, \boldsymbol{H}_1, \boldsymbol{H}_2, \boldsymbol{u}, \boldsymbol{D}, \sigma_1, \sigma_2\}$ as public parameters and maintains the privacy of $\boldsymbol{M}_{sk} = \boldsymbol{T}_{A_0}$.

2.3.2 Key generation

First, the KGC generates the search key pairs as follows: when a data owner, i, joins the system, he/she sends his identity to the KGC. After receiving the identity, the KGC verifies whether the data owner is legal. If i is legal, the KGC implements NewBasisDel$(\boldsymbol{A}_0, \boldsymbol{T}_{A_0}, \sigma_1)$ to generate an identity key $\boldsymbol{R}_i \in \mathbf{Z}_q^{m\times m}$ for i and a short basis $\boldsymbol{T}_{A_i} \in \mathbf{Z}_q^{m\times m}$ from $\boldsymbol{A}_i = \boldsymbol{A}_0\boldsymbol{R}_i^{-1} \in \mathbf{Z}_q^{n\times m}$.

The KGC then generates the signature key pairs as follows: it randomly chooses $\boldsymbol{A}' \in \mathbf{Z}_q^{n\times(m-n)}$ and $\boldsymbol{B}' \in \mathbf{Z}_q^{(m-n)\times n}$, and then computes verification master key pairs as $\boldsymbol{V}_{pk} = [\,2\boldsymbol{A}' \mid 2\boldsymbol{A}'\boldsymbol{B}' \bmod q + q\boldsymbol{I}_n\,] \in \mathbf{Z}_{2q}^{n\times m}$ and $\boldsymbol{V}_{sk} = \begin{bmatrix} \boldsymbol{B}' \\ -\boldsymbol{I}_n \end{bmatrix} \in \mathbf{Z}_{2q}^{m\times n}$.

Finally, the KGC sends $\{\boldsymbol{T}_{A_i}, \boldsymbol{R}_i, \boldsymbol{K}_{ws}\}$ to the users who have access abilities to search the files belonging to i through a secure channel, sends $\{\boldsymbol{R}_i, \boldsymbol{K}_{ws}, \boldsymbol{V}_{sk}\}$ to each i through the secure channel, and publishes $\boldsymbol{V}_{pk}$ publicly.

2.3.3 Encrypted keyword index generation

The data owner i first applies the hash function $\boldsymbol{H}_1$ to map $\boldsymbol{w}_j$ to $\boldsymbol{H}_1(\boldsymbol{w}_j) \in \mathbf{Z}_q^n$, where $1 \leqslant j \leqslant N$. The data owner i then selects $\boldsymbol{e}_0 \in \mathbf{Z}_q^n$ and $\boldsymbol{e}_{1,0,k} \in \mathbf{Z}_q^n$ from the $1 \leqslant k \leqslant m$ error distribution, selects a random matrix $\boldsymbol{B}_i \in \mathbf{Z}_q^{n\times n}$, and then calculates $\boldsymbol{C}_0 = \boldsymbol{B}_i\boldsymbol{u} + \boldsymbol{e}_0 \in \mathbf{Z}_q^n$, $\boldsymbol{C}_{1,0} = \boldsymbol{B}_i\boldsymbol{A}_i + \boldsymbol{E}_{1,0} \in \mathbf{Z}_q^{n\times m}$, where $\boldsymbol{E}_{1,0} = [\,\boldsymbol{e}_{1,0,1} \mid \boldsymbol{e}_{1,0,2} \mid \cdots \mid \boldsymbol{e}_{1,0,\mathrm{m}}\,]^{\mathrm{T}}$. For each file that i possesses, he/she checks if the file contains $\boldsymbol{w}_j$. If so, he/she sets $\boldsymbol{M}_j = \boldsymbol{H}_1(\boldsymbol{w}_j)\boldsymbol{D} \in \mathbf{Z}_q^{n\times m}$; Otherwise, he/she selects a random matrix, $\boldsymbol{M}_j \in \mathbf{Z}_q^{n\times m}$. Next, i selects $\boldsymbol{e}_{1,j,k} \in \mathbf{Z}_q^n$ for the $1 \leqslant k \leqslant m$ error distribution and then computes $\boldsymbol{C}_{1,j} = \boldsymbol{B}_i\boldsymbol{M}_j + \boldsymbol{E}_{1,j} \in \mathbf{Z}_q^{n\times m}$, where $\boldsymbol{E}_{1,j} = [\,\boldsymbol{e}_{1,j,1} \mid \boldsymbol{e}_{1,j,2} \mid \cdots \mid \boldsymbol{e}_{1,j,m}\,]^{\mathrm{T}} \in \mathbf{Z}_q^{n\times m}$.

Finally, i generates the encrypted keyword index, $\boldsymbol{C} = \{\boldsymbol{C}_0 = \boldsymbol{B}_i\boldsymbol{u} + \boldsymbol{e}_0 \in \mathbf{Z}_q^n, \boldsymbol{C}_{1,0} = \boldsymbol{B}_i\boldsymbol{A}_i + \boldsymbol{E}_{1,0} \in \mathbf{Z}_q^{n\times m}, \boldsymbol{C}_{1,j} = \boldsymbol{B}_i\boldsymbol{M}_j + \boldsymbol{E}_{1,j} \in \mathbf{Z}_q^{n\times m}\}$.

2.3.4 Signature generation

If the data file does not contain $\boldsymbol{w}_j$, then i selects a random $\boldsymbol{y}_j \in \{-1,0,1\}^n$ and $\boldsymbol{z}_j \in \mathbf{Z}_q^m$, and generates the signature as $\boldsymbol{s}_{\mathrm{ig}} = \{\boldsymbol{z}_j, \boldsymbol{y}_j\}$.

If the data file contains $\boldsymbol{w}_j$, then i first randomly selects $\boldsymbol{x}_j \in \mathbf{Z}_q^m$ and then computes $\boldsymbol{y}_j = \boldsymbol{H}_2(\boldsymbol{V}_{pk}\boldsymbol{R}_i\boldsymbol{x}_j \bmod 2q, \boldsymbol{r}_{i,1},$

$\boldsymbol{r}_{i,2},\cdots,\boldsymbol{r}_{i,m},\boldsymbol{w}_j) \in \{-1,0,1\}^n$, where $\boldsymbol{r}_{i,j}$ donates the j-th column of $\boldsymbol{R}_i$. The data owner i then selects a random bit, $b \in \{0,1\}$, calculates $\boldsymbol{z}_j = \boldsymbol{x}_j + (-1)^b \boldsymbol{R}_i^{-1}\boldsymbol{V}_{sk}\boldsymbol{y}_j \in \boldsymbol{Z}_q^m$, selects a big enough integer τ to calculate $\sigma = \tau \| \boldsymbol{R}_i^{-1}\boldsymbol{V}_{sk}\boldsymbol{y}_i \|$, $X = \exp(1 + 1/(2\tau^2))$, and implements reject sampling to generate $\boldsymbol{s}_{ig} = (\boldsymbol{z}_j, \boldsymbol{y}_j)$ with probability $1 \Big/ \left[X\exp\left(-\frac{\| \boldsymbol{R}_i^{-1}\boldsymbol{V}_{sk}\boldsymbol{y}_j \|^2}{2\sigma^2} \right) \cosh\left(\frac{\langle \boldsymbol{z}_j, \boldsymbol{R}_i^{-1}\boldsymbol{V}_{sk}\boldsymbol{y}_j \rangle}{\sigma^2} \right) \right]$.

Finally, i sends the ciphertext of the data files with signature $\boldsymbol{s}_{ig} = \{\boldsymbol{z}_j, \boldsymbol{y}_j\}$ and encrypted keyword index $\boldsymbol{C} = \{\boldsymbol{C}_0, \boldsymbol{C}_{1,0}, \boldsymbol{C}_{1,j}\}$ to the cloud server.

2.3.5 User request generation

First, i decides a keyword set $K_U = \{\boldsymbol{w}_{U_1}, \boldsymbol{w}_{U_2}, \cdots, \boldsymbol{w}_{U_k}\}$ containing a keyword k that they want to search, where $\boldsymbol{U} = \{U_1, U_2, \cdots, U_k\}$ and $1 \leqslant k \leqslant N$. Subsequently, i computes $\boldsymbol{H}_1(\boldsymbol{w}_{U_j})(1 \leqslant j \leqslant k)$ and obtains $\boldsymbol{u}, \boldsymbol{D}$, and σ_2 from p_p, and then calculates $\boldsymbol{A}_i = \boldsymbol{A}_0\boldsymbol{R}_i^{-1}$ and applies LeftSample$(\boldsymbol{A}_i, \sum_{j=1}^{k}\boldsymbol{H}_1(\boldsymbol{w}_{U_j})\boldsymbol{D}, \boldsymbol{T}_{A_i}, \boldsymbol{u}, \sigma_2)$ to generate $\boldsymbol{f} \in \boldsymbol{Z}_q^{2m}$, where $\boldsymbol{f}$ satisfies $\left[\boldsymbol{A}_i \,\middle|\, \sum_{j=1}^{k}\boldsymbol{H}_1(\boldsymbol{w}_{U_j})\boldsymbol{D} \right]\boldsymbol{f} = \boldsymbol{u} \bmod q$.

Finally, i sends the search query $\boldsymbol{f}$ and $\{U_j\}$ to the cloud server, where $1 \leqslant j \leqslant k$.

2.3.6 Ciphertext search

First, the cloud server finds the encrypted keyword index $\boldsymbol{C} = \{\boldsymbol{C}_0, \boldsymbol{C}_{1,0}, \boldsymbol{C}_{1,j}\}$ stored in the cloud server and applies $\boldsymbol{f}$ and $\{U_j\}$ uploaded by i to compute $\boldsymbol{\Omega} = \boldsymbol{C}_0 - \left[\boldsymbol{C}_{1,0} \,\middle|\, \sum_{j=1}^{k}\boldsymbol{C}_{1,U_j} \right]\boldsymbol{f}$, where $1 \leqslant j \leqslant n$.

The cloud server then verifies if $\Omega_j < q/4$, where Ω_j is the j-th norm of $\boldsymbol{\Omega}$. If so, the server returns true, which means that the ciphertext of data files contains all the searched keywords queried by the user. The cloud server then sends the ciphertext with signatures $\boldsymbol{s}_{ig} = \{\boldsymbol{z}_j, \boldsymbol{y}_j\}$ to the user. Otherwise, the cloud server returns $\perp$.

2.3.7 Verification

After receiving the ciphertext of data files with corresponding signatures $\boldsymbol{s}_{ig} = \{\boldsymbol{z}_j, \boldsymbol{y}_j\}$ from the cloud server, i checks whether the following equations hold:

$$\| \boldsymbol{z}_j \| \leqslant 1.4\sqrt{m}\sigma \tag{1}$$

$$\| z_{j,i} \| < \frac{q}{4} \qquad 1 \leqslant i \leqslant m \tag{2}$$

$$\boldsymbol{y}_j = \boldsymbol{H}_2((\boldsymbol{V}_{pk}\boldsymbol{R}_i\boldsymbol{z}_j + q\boldsymbol{y}_j) \bmod 2q, \boldsymbol{r}_{i,1}, \boldsymbol{r}_{i,2}, \cdots, \boldsymbol{r}_{i,m}, \boldsymbol{w}_{U_j}) \tag{3}$$

where $z_{j,i}$ is the i-th element of vector $\boldsymbol{z}_j$.

If all three equations hold, i accepts the ciphertext of the data files; otherwise, i rejects the ciphertext.

2.4 Correctness

Theorem 1 Our LWE-based verifiable multi-keyword search (LWE-VMKS) scheme is correct.

Proof First, we prove the correctness of the search phase.

The computation result in the ciphertext search phase for the cloud server is as follows:

$$\boldsymbol{\Omega} = \left(\boldsymbol{C}_0 - \left[\boldsymbol{C}_{1,0} \,\middle|\, \sum_{j=1}^{k}\boldsymbol{C}_{1,U_j} \right]\boldsymbol{f} \right) \bmod q =$$

$$\left(\boldsymbol{B}_i\boldsymbol{u} + \boldsymbol{e}_0 - \left[\boldsymbol{B}_i\boldsymbol{A}_i + \boldsymbol{E}_{1,0} \,\middle|\, \sum_{j=1}^{k}(\boldsymbol{B}\boldsymbol{H}_1(\boldsymbol{w}_{U_j})\boldsymbol{D} + \boldsymbol{E}_{1,U_j}) \right]\boldsymbol{f} \right) \bmod q =$$

$$\left(\boldsymbol{B}_i\boldsymbol{u} - \boldsymbol{B}_i\left[\boldsymbol{A}_i \,\middle|\, \sum_{j=1}^{k}\boldsymbol{H}_1(\boldsymbol{w}_{U_j})\boldsymbol{D} \right]\boldsymbol{f} + \boldsymbol{e}' \right) \bmod q$$

where $\boldsymbol{e}' = \boldsymbol{e}_0 - \left[\boldsymbol{E}_{1,0} \,\middle|\, \sum_{j=1}^{k}\boldsymbol{E}_{1,U_j} \right]\boldsymbol{f}$.

Note that $\boldsymbol{f}$ satisfies $\left[\boldsymbol{A}_i \,\middle|\, \sum_{j=1}^{k}\boldsymbol{H}_1(\boldsymbol{w}_{U_j})\boldsymbol{D} \right]\boldsymbol{f} = \boldsymbol{u} \bmod q$. Therefore, the cloud server can be obtained as $\boldsymbol{\Omega} = (\boldsymbol{B}_i\boldsymbol{u} - \boldsymbol{B}_i\boldsymbol{u} + \boldsymbol{e}') \bmod q = \boldsymbol{e}' \bmod q$.

Let e'_j denote the j-th element of $\boldsymbol{e}'$. According to the definition of the LeftSample, $\boldsymbol{f}$ satisfies $\| \boldsymbol{f} \| \leqslant \sigma_2\sqrt{2m}$. Let $e_{0,j}$ denote the j-th element of $\boldsymbol{e}_0$, $\boldsymbol{e}_{1,0,j}$ denote the j-th row of $\boldsymbol{E}_{1,0}$, and $\boldsymbol{e}_{1,U_j,j}$ denote the j-th row of $\boldsymbol{E}_{1,U_j}$. Since $\boldsymbol{e}_{0,j}$, $\boldsymbol{e}_{1,0,j}$, and $\boldsymbol{e}_{1,U_j,j}$ are selected with the error distribution, we know $\boldsymbol{e}_{0,j} \leqslant q\alpha\omega\sqrt{\log m}$, $\boldsymbol{e}_{1,0,j} \leqslant q\alpha\sqrt{m}\omega\sqrt{\log m}$, and $\boldsymbol{e}_{1,U_j,j} \leqslant q\alpha\sqrt{m}\omega\sqrt{\log m}$. We then have

$$\| e'_j \| = \left\| e_{0,j} - \left[\boldsymbol{e}_{1,0,j} \,\middle|\, \sum_{j=1}^{k}\boldsymbol{e}_{1,U_j,j}) \right]\boldsymbol{f} \right\| \leqslant$$

$$\| e_{0,j} \| + \left\| \left[\boldsymbol{e}_{1,0,j} \,\middle|\, \sum_{j=1}^{k}\boldsymbol{e}_{1,U_j,j}) \right]\boldsymbol{f} \right\| \leqslant$$

$$\| e_{0,j} \| + \left(\| \boldsymbol{e}_{1,0,j} \| + \left\| \sum_{j=1}^{k}\boldsymbol{e}_{1,U_j,j} \right\| \right) \| \boldsymbol{f} \| \leqslant$$

$$q\alpha\omega\sqrt{\log m} + (k+1)q\alpha\sqrt{m}\omega\sqrt{\log m}\sigma_2\sqrt{2m} =$$

$$(1 + k\sigma m + m\sigma_2\sqrt{2})q\alpha\omega\sqrt{\log m}$$

Hence, if we set $\alpha < (4(1 + k\sigma m + m\sigma\sqrt{2})\omega \cdot \sqrt{\log m})^{-1}$, according to Lemma 2 with $\alpha q > 2\sqrt{m}$, we know that $q > 8(\sqrt{m} + k\sigma m\sqrt{m} + \sqrt{2}\sigma m\sqrt{m})\alpha\omega\sqrt{\log m}$. We can then obtain $\| e'_j \| \leqslant (1 + k\sigma_2 m + m\sigma_2 \cdot \sqrt{2})q\alpha\omega\sqrt{\log m} < q/4, j \in [1, m]$ and ensure that $\Omega_j < q/4$. The cloud server can thus correctly retrieve the ciphertext.

We then prove the correctness of the verification phase.

In the verification phase, according to Lemma 1, $\boldsymbol{z}_j$ satisfies $\| \boldsymbol{z}_j \| \leqslant B_2 = 1.4\sqrt{m}\sigma$, and $\| z_{j,i} \| < q/4$. We can then prove that $\boldsymbol{y}_j = \boldsymbol{H}_2(\boldsymbol{V}_{pk}\boldsymbol{R}_i\boldsymbol{z}_j + q\boldsymbol{y}_j \bmod 2q, \boldsymbol{r}_{i,1}, \boldsymbol{r}_{i,2}, \cdots, \boldsymbol{r}_{i,m}, \boldsymbol{w}_{U_j})$ as follows:

$$\boldsymbol{H}_2((\boldsymbol{V}_{pk}\boldsymbol{R}_i\boldsymbol{z}_j + q\boldsymbol{y}_j) \bmod 2q, \boldsymbol{r}_{i,1}, \boldsymbol{r}_{i,2}, \cdots, \boldsymbol{r}_{i,m}, \boldsymbol{w}_{U_j}) =$$

$$\boldsymbol{H}_2((\boldsymbol{V}_{pk}\boldsymbol{R}_i(\boldsymbol{x}_j + (-1)^b\boldsymbol{R}_i^{-1}\boldsymbol{V}_{sk}\boldsymbol{y}_j + q\boldsymbol{y}_j) \bmod 2q, \boldsymbol{r}_{i,1}, \boldsymbol{r}_{i,2}, \cdots, \boldsymbol{r}_{i,m}, \boldsymbol{w}_{U_j}) = \boldsymbol{H}_2((\boldsymbol{V}_{pk}\boldsymbol{R}_i\boldsymbol{x}_j + (-1)^b \cdot [2\boldsymbol{A}' \mid 2\boldsymbol{A}'\boldsymbol{B}' \bmod q + q\boldsymbol{I}_n][\boldsymbol{B}' \quad -\boldsymbol{I}_n]^{\mathrm{T}}\boldsymbol{y}_j + q\boldsymbol{y}_j)$$

$$\bmod 2q, \boldsymbol{r}_{i,1}, \boldsymbol{r}_{i,2}, \cdots, \boldsymbol{r}_{i,m}, \boldsymbol{w}_{U_j}) =$$
$$\boldsymbol{H}_2((\boldsymbol{V}_{pk}\boldsymbol{R}_i\boldsymbol{x}_j + (-1)^b q\boldsymbol{y}_j + q\boldsymbol{y}_j) \bmod 2q,$$
$$\boldsymbol{r}_{i,1}, \boldsymbol{r}_{i,2}, \cdots, \boldsymbol{r}_{i,m}, \boldsymbol{w}_{U_j}) =$$
$$\boldsymbol{H}_2((\boldsymbol{V}_{pk}\boldsymbol{R}_i\boldsymbol{x}_j) \bmod 2q, \boldsymbol{r}_{i,1}, \boldsymbol{r}_{i,2}, \cdots, \boldsymbol{r}_{i,m}, \boldsymbol{w}_{U_j}) = \boldsymbol{y}_j$$

Obviously, if the signature contains the same keywords, $\boldsymbol{w}_{U_j}$, for user searching, Eqs. (1) to (3) will hold. Therefore, once the cloud server provides a wrong search result, the user can check the correctness of the search result, and the correctness of the verification phase can be ensured.

3 Security Analysis

3.1 Formal proof

In this section, we formally prove that our LWE-VMKS scheme can resist quantum computing attacks, support a high-efficiency multi-keyword search, realize verification for the search results, and support KGA resistance.

Theorem 2 Our LWE-VMKS scheme can realize quantum computing attack resistance.

Proof We demonstrate the number field for the encrypted keywords index, $\boldsymbol{C}_{1,j} = \boldsymbol{B}_i\boldsymbol{M}_j + \boldsymbol{E}_{1,j} \in \boldsymbol{Z}_q^{n\times m}$, which is identical to the LWE problem and meets the number field requirement of the LWE problem. Similarly, $\boldsymbol{C}_0 = \boldsymbol{B}_i\boldsymbol{u} + \boldsymbol{e}_0 \in \boldsymbol{Z}_q^n$ and $\boldsymbol{C}_1 = \boldsymbol{B}_i\boldsymbol{A}_i + \boldsymbol{E}_{1,0} \in \boldsymbol{Z}_q^{n\times m}$ also meet the number of field requirements of the LWE problem.

Therefore, the encrypted keyword index $\boldsymbol{C} = \{\boldsymbol{C}_0, \boldsymbol{C}_{1,0}, \boldsymbol{C}_{1,j}\}$ can resist a quantum computing attack.

Theorem 3 Our LWE-VMKS scheme can achieve a secure multi-keyword search with high efficiency.

Proof According to Lemma 5, only the user i who possesses the secret key $\boldsymbol{T}_{A_i}$, can successfully generate a short vector, $\boldsymbol{f}$, thus satisfying $\left[\boldsymbol{A}_i \middle| \sum_{j=1}^{k} \boldsymbol{H}_1(\boldsymbol{w}_{U_j})\boldsymbol{D}\right]\boldsymbol{f} = \boldsymbol{u} \bmod q$. Therefore, attackers cannot search the ciphertext.

In addition, only one trapdoor function is implemented in the user request generation phase, and this does not increase the computation cost.

Theorem 4 Our LWE-VMKS scheme can securely verify the correctness of the search results.

Proof According to Lemma1, only the data owner i who possesses the verification key, $\boldsymbol{V}_{sk} \in \boldsymbol{Z}_{2q}^{n\times m}$, can generate correct $\boldsymbol{s}_{ig} = \{\boldsymbol{z}_j, \boldsymbol{y}_j\}$. After i receives the search result from the cloud server, he/she verifies the correctness of the search result and checks whether Eqs. (1) and (2) hold. Only when the equations hold, can i continue verification.

Then, as proven in Theorem 1, Eq. (3) only holds if the signature contains the same keywords. If the cloud returns wrong results, i rejects them.

Theorem 5 Our LWE-VMKS scheme can resist KGA.

Proof As proven in Theorem 2, attackers cannot obtain $\boldsymbol{R}_i$, even if they apply a quantum computing attack. Therefore, attackers cannot generate $\boldsymbol{C}_{1,0}$ to launch KGA.

3.2 Security goal comparison

In this section, we compare the security goals of our LWE-VMKS scheme with those of other schemes[12,15-16] in Tab. 1. The comparison is made to illustrate the multiple security goals of our developed schemes: resisting a quantum computing attack, enabling a multi-keyword search, conducting a verifiable search for results, and resisting KGA.

Tab. 1 Security goal comparison

Scheme	Quantum computing attack resistance	Multi-keyword search	Verifiable search for result	KGA resistance
Ref. [12]	√	×	×	×
Ref. [15]	√	×	×	√
Ref. [16]	√	√	×	×
LWE-VMKS	√	√	√	√

4 Performance Analysis

The performance environment of our LWE-VMKS scheme is shown in Fig. 1. The entire system runs on a distributed Internet environment with a 20 MB bandwidth. The cloud server is set in an Ali cloud server, which runs 64 bit CentOS 8.2 with a four-core CPU and 8 GB RAM. For the KGC, data owners and users are set on a desktop with an Intel Core i5-9500T CPU containing six cores rated at 2.20 GHz with 8 GB of memory.

4.1 Storage cost comparison

In this section, we compare the storage costs of our LWE-VMKS scheme with those in previous studies[12,15-16]. The storage costs are mainly decided by the encrypted keyword index size and signature size, and the results are provided in Tab. 2. We set the parameters of the LWE-based schemes as $n = 128$, $q = 257$, $m = 6n\lfloor \log q \rfloor = 1\,024$. It is of note that the schemes in previous studies[12, 15-16] do not support a verifiable search.

Tab. 2 Storage cost comparison

Scheme	Encrypted keyword index	Signature size
Ref. [12]	$(mn + n)\log q$	
Ref. [15]	$(2mn + n)\log q$	
Ref. [16]	$(mn + m + n)\log q$	
LWE-VMKS	$mn\log q$	$2m\log q$

4.2 Comparison of computation costs

In this section, we compare the computation costs of our LWE-VMKS scheme with those of previous studies[12,15-16], and the results are given in Tab. 3.

Tab. 3 Computation costs of keyword search

Scheme	Keyword encryption cost	User request cost	Search cost	Verification cost
Ref. [12]	$(m^2n + n^2 + n)t_{mul}$	$2 \mid W \mid t_s$	$mn \mid W \mid t_{mul}$	
Ref. [15]	$(2mn^2 + n^2)t_{mul}$	$n \mid W \mid t_s$	$2mn \mid W \mid t_{mul}$	
Ref. [16]	$3(mn + n^2)t_{mul}$	nt_s	$(\mid W \mid + 2)mnt_{mul}$	
LWE-VMKS	mn^2t_{mul}	t_s	$2mnt_{mul}$	$(mn + n)t_{mul}$

Let t_s denote the cost of one trapdoor function operation, t_{mul} denote the time cost of multiplication of two numbers, and $\mid W \mid$ denote the number of searched keywords. The computation cost for one keyword search in each phase is shown in Tab. 3.

4.2.1 Comparison of computation costs of keyword encryption

In this section, we compare the computation costs of keyword encryption in our LWE-VMKS scheme with those in previous studies[12,15-16], and the results are shown in Tab. 3.

The computation cost of keyword encryption in our LWE-VMKS scheme is less than that in Refs. [12, 15]: the computation costs in Refs. [12, 15] are 134 234 240t_{mul} and 33 570 816t_{mul}, respectively, according to $(m^2n + n^2 + n)t_{mul}$ and $(2mn^2 + n^2)t_{mul}$, respectively, while the computation cost of our LWE-VMKS scheme is 16 793 600t_{mul}. However, the computation cost of keyword encryption in our LWE-VMKS scheme is higher than that in Ref. [16]: the computation cost in Ref. [16] is 442 368 t_{mul} according to $3(mn + n^2)t_{mul}$, while the computation cost in our LWE-VMKS scheme is 16 793 600t_{mul}. However, Ref. [16] does not support verification for searched results, and it cannot realize KGA resistance.

4.2.2 Comparison of computation costs of user request

In this section, we compare the computation cost of the user request for our LWE-VMKS scheme with that of the schemes in Refs. [12,15-16], and the results are shown in Tab. 3. The computation cost of the user request of our LWE-VMKS scheme is lower than that of Refs. [12,15-16], as $2 \mid W \mid t_s > t_s$, $128 \mid W \mid t_s > t_s$, and $128t_s > t_s$, respectively.

4.2.3 Comparison of computation costs of keyword search

In this section, we compare the computation cost of conducting a keyword search in our LWE-VMKS scheme with that of the schemes in Refs. [12,15-16]. The results are shown in Fig. 2, where the x-axis denotes the number of searched keywords, and the y-axis denotes the logarithmic of the computation cost.

The computation cost of a keyword search in our LWE-VMKS scheme is lower than that in Refs. [12,15-16], when $\mid W \mid > 2$, $\mid W \mid > 1$, and at all times, respectively.

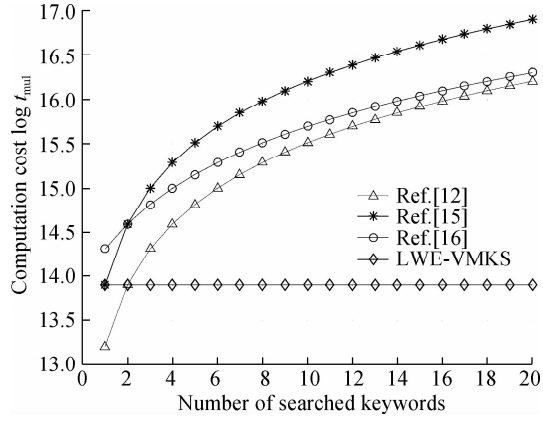

Fig. 2 Comparison of computational costs

5 Conclusions

1) Our LWE-VMKS scheme can resist quantum computing attacks and multi-keyword searches, verify the correctness of the searched result, and support KGA resistance.

2) The performance analysis demonstrates that our scheme provides superior performance compared with other schemes.

3) Our scheme is not flexible enough to provide an accurate search result for the user.

References

[1] Boneh D, Crescenzo G D, Ostrovsky R, et al. Public key encryption with keyword search [C]// *Proc EUROCRYPT*. Berlin, Germany: Springer-Verlag, 2004: 506-522.

[2] Zhang B, Zhang F G. An efficient public key encryption with conjunctive-subset keywords search [J]. *Journal of Network and Computer Applications*, 2011, **34**(1): 262-267. DOI: 10.1016/j.jnca.2010.07.007.

[3] Wang S P, Zhang D, Zhang Y L, et al. Efficiently revocable and searchable attribute-based encryption scheme for mobile cloud storage [J]. *IEEE Access*, 2018, **6**: 30444-30457. DOI: 10.1109/ACCESS.2018.2846037.

[4] Park D J, Kim K, Lee P J. Public key encryption with conjunctive field keyword search [C]//*Proceedings of the 5th International Conference on Information Security Applications*. New York: ACM, 2004: 73-86. DOI: 10.1007/978-3-540-31815-6_7.

[5] Liu X Q, Yang G M, Susilo W, et al. Privacy-preserving multi-keyword searchable encryption for distributed systems [J]. *IEEE Transactions on Parallel and Distributed Systems*, 2021, **32**(3): 561-574. DOI: 10.1109/TPDS.2020.3027003.

[6] Miao Y B, Ma J F, Liu X M, et al. VMKDO: Verifiable multi-keyword search over encrypted cloud data for dynamic data-owner[J]. *Peer-to-Peer Networking and Applications*, 2018, **11**(2): 287 - 297. DOI: 10.1007/s12083-016-0487-7.

[7] Miao Y B, Tong Q Y, Deng R H, et al. Verifiable searchable encryption framework against insider keyword-guessing attack in cloud storage[J]. *IEEE Transactions on Cloud Computing*, 2022, **10**(2): 835 - 848. DOI: 10.1109/TCC.2020.2989296.

[8] Zhang Y, Xu C X, Ni J B, et al. Blockchain-assisted public-key encryption with keyword search against keyword guessing attacks for cloud storage[J]. *IEEE Transactions on Cloud Computing*, 2021, **9**(4): 1335 - 1348. DOI: 10.1109/TCC.2019.2923222.

[9] Regev O. On lattices, learning with errors, random linear codes, and cryptography[C]//*Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*. Baltimore, MD, USA, 2005: 84 - 93. DOI: 10.1145/1060590.1060603.

[10] Zhang X J, Xu C X. Trapdoor security lattice-based public-key searchable encryption with a designated cloud server[J]. *Wireless Personal Communications*, 2018, **100**(3): 907 - 921. DOI: 10.1007/s11277-018-5357-6.

[11] Yu X L, Xu C G, Xu L. Lattice-based searchable encryption with keywords revocable and bounded trapdoor exposure resistance[J]. *IEEE Access*, 2019, **7**: 43179 - 43189. DOI: 10.1109/ACCESS.2019.2908202.

[12] Behnia R, Ozmen M O, Yavuz A A. Lattice-based public key searchable encryption from experimental perspectives[J]. *IEEE Transactions on Dependable and Secure Computing*, 2020, **17**(6): 1269 - 1282. DOI: 10.1109/TDSC.2018.2867462.

[13] Xu L, Yuan X L, Steinfeld R, et al. Multi-writer searchable encryption: An LWE-based realization and implementation[C]//*Proceedings of the* 2019 *ACM Asia Conference on Computer and Communications Security*. Auckland, New Zealand, 2019: 122 - 133. DOI: 10.1145/3321705.3329814.

[14] Zhang X J, Huang C, Gu D W, et al. BIB-MKS: Post-quantum secure biometric identity-based multi-keyword search over encrypted data in cloud storage systems[J]. *IEEE Transactions on Services Computing*, 2023, **16**(1): 122 - 133. DOI: 10.1109/TSC.2021.3112779.

[15] Zhang X J, Xu C X, Wang H X, et al. FS-PEKS: Lattice-based forward secure public-key encryption with keyword search for cloud-assisted industrial Internet of Things[J]. *IEEE Transactions on Dependable and Secure Computing*, 2021, **18**(3): 1019 - 1032. DOI: 10.1109/TDSC.2019.2914117.

[16] Wang P, Xiang T, Li X G, et al. Public key encryption with conjunctive keyword search on lattice[J]. *Journal of Information Security and Applications*, 2020, **51**: 102433. DOI: 10.1016/j.jisa.2019.102433.

[17] Mei L, Xu C G, Xu L, et al. Verifiable identity-based encryption with keyword search for IoT from lattice[J]. *Computers, Materials and Continua*, 2021, **68**(2): 2299 - 2314.

[18] Micciancio D, Peikert C. Trapdoors for lattices: Simpler, tighter, faster, smaller[J]. *EUROCRYPT* 2012. *Lecture Notes in Computer Science*, 2012, **7237**: 700 - 718.

[19] Gentry C, Peikert C, Vaikuntanathan V. Trapdoors for hard lattices and new cryptographic constructions[C]//*Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*. Victoria, British Columbia, Canada, 2008: 197 - 206. DOI: 10.1145/1374376.1374407.

[20] Agrawal S, Boyen X, Vaikuntanathan V, et al. Functional encryption for threshold functions (or fuzzy IBE) from lattices[C]//*International Workshop on Public Key Cryptography*. Berlin, Germmany, 2012: 280 - 297. DOI: 10.1007/978-3-642-30057-8_17.

[21] Lyubashevsky V, Peikert C, Regev O. On ideal lattices and learning with errors over rings[C]// *International Conference on Theory and Applications of Cryptographic Techniques*. Riviera, French, 2010: 1 - 23.

云上基于错误学习的可验证多关键词搜索方案

汪 攀 蒋 睿

(东南大学网络空间安全学院, 南京 210096)

摘要:为了解决公钥加密关键字搜索(PEKS)算法面临的多种难题,提出了一种基于错误学习的可验证多关键字搜索方案(LWE-VMKS).该方案利用格加密的算法生成关键字索引,搜索查询和签名,以抵抗量子计算攻击.该方案将单个搜索查询中多个关键字合并,实现多关键字搜索.该方案结合基于格的签名,保证用户可以在不解密密文的情况下验证搜索结果的正确性.另外,该方案应用陷门函数为不同的数据所有者生成不同的密钥,从而抵抗关键词猜测攻击(KGA).最后,形式化证明了所提出的方案是安全的,能够实现高效的多关键词搜索并实现搜索结果的验证,并且能抵抗 KGA.

关键词:格基加密;错误学习;公钥加密关键字搜索(PEKS);多关键词搜索;可验证;关键词猜测攻击(KGA)

中图分类号:TN915.08