Guo *et al. J Wireless Com Network* (2021) 2021:174
https://doi.org/10.1186/s13638-021-02006-1

EURASIP Journal on Wireless Communications and Networking

RESEARCH

Open Access

Designated server-aided revocable identity-based keyword search on lattice

Ying Guo[1], Fei Meng[2], Leixiao Cheng[3], Xiaolei Dong[4] and Zhenfu Cao[4,5,6*]

*Correspondence:
zfcao@sei.ecnu.edu.cn
[4] Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China
Full list of author information is available at the end of the article

Abstract
Public key encryption scheme with keyword search is a promising technique supporting search on encrypted data without leaking any information about the keyword. In real applications, it's critical to find an effective revocation method to revoke users in multi-user cryptosystems, when user's secret keys are exposed. In this paper, we propose the first designated server-aided revocable identity-based encryption scheme with keyword search (dSR-IBKS) from lattice. The dSR-IBKS model requires each user to keep just one private key corresponding with his identity and does not need to communicate with the key generation center or the server during key updating. We have proved that our scheme can achieve chosen keyword indistinguishability in the standard model. In particular, our scheme can designate a unique tester to test and return the search results, therefore no other entity can guess the keyword embedded in the ciphertext by generating search queries and doing the test by itself. We provide a formal security proof of our scheme assuming the hardness of the learning with errors problem on the standard model.

Keywords: Public key encryption, Identity-based, Keyword search, Server-aided, Lattice

1 Introduction

In the cloud computing scenarios, data should be encrypted at first before uploaded to the cloud server, otherwise the privacy of sensitive information could be exposed. In this case, new searchable techniques should be applied. Boneh et al. [1], initially constructed the public key encryption with keyword search (PEKS), in which ciphertext is encrypted with a ciphertext keyword and the public key of data receiver. Whiling searching for a ciphertext with specific keyword, data receiver generates and submits a search query, embedded with a target keyword, to the cloud. The cloud returns the matched ciphertext to the receiver, only if it contains the same keyword as the search query.

Derived from Boneh's work, different PEKS schemes had been proposed catering for various functionalities. For example, In Dong et al. [2] proposed the public key encryption with conjunctive field keyword search. In Boneh et al. [3] provided several searchable systems achieving comparison queries and arbitrary conjunctive queries on the ciphertext. Li et al. claimed that they proposed the first authenticated identity-based encryption with keyword search (IBKS) [4] and attribute-based encryption with keyword search (ABKS) [5–7] achieving both access control and keyword search. However, all the above schemes are proved secure under the hardness of the discrete logarithm problem. In order to resist against quantum attacks, many PEKS schemes based on lattices [8–13] have been proposed.

To the best of our knowledge, [10] is the first identity-based encryption scheme with keyword search from lattice assumption, which is a combination of identity-based encryption (IBE) and searchable encryption. One practical issue of IBE in real applications is to find an effective revocation method to revoke users in multi-user cryptosystems, because users may behave inappropriately or their secret keys may be compromised.

In Boneh et al. [14] proposed a revocation mechanism for IBE, in which the up-to-date revocation list is controlled by a trusted authority called Key Generation Center (KGC), who issues secret key $sk_{\mathsf{id}||\mathsf{t}}$ for each non-revoked user id in every time period t. In this mechanism, only non-revoked users can decrypt ciphertext bound to their identity and the same time slot (i.e., $\mathsf{id}||\mathsf{t}$). However, this approach is inefficient since the KGC have to generate $O(N-r)$ new secret keys in each time period, where N is the total number of users and r is the number of revoked users in time period t. That is, the workload of the KGC is proportional to the number of users N.

In Boldyreva et al. [15] proposed another revocation mechanism based on the tree-based revocation scheme of [16], and formalized the notion of revocable IBE (RIBE). In this mechanism, each user keeps $O(\log N)$ long-term secret keys and the KGC broadcasts $O(r\log(N/r))$ update keys for each time period t. Only non-revoked users can obtain their decryption keys from their long-term secret keys and the update keys. Compared with Boldyreval et al. [14], this mechanism significantly reduces the size of update key from linear (i.e., $O(N-r)$) to logarithmic (i.e., $O(r\log(N/r))$) in the number of users. However, this revocation mechanism still does not provide an efficient revocation for the following limitations: (1) it requires KGC to stay online regularly and all non-revoked users need to communicate with KGC and update their decryption keys periodically; (2) the sizes of both update keys (i.e., $O(r\log(N/r))$) and users' secret keys $O(\log N)$ are logarithmical in the number of users.

To overcome the two limitations in Boldyreva et al. [15], Qin et al. [17] proposed a novel RIBE system model called server-aided revocable IBE (SR-IBE) under the decisional bilinear diffie-hellman (DBDH) assumption. The server is assumed to be *untrusted* in the sense that it doesn't keep any secret data and only performs public storage and computation operations according to the system specification. In SR-IBE system model, no communication is required between users and the KGC during key update. In addition, although the size of update keys from the KGC to the server is still logarithmic (i.e. $O(r\log(N/r))$), the size of every user's private key is constant (i.e. $O(1)$) here instead of $O(\log N)$ in [15].

In addition, making use of the binary-tree data structure of [15, 18] proposed the first RIBE from lattice. Following the security model of [17], Nguyen et al. [19] considered selective-identity security and proposed the first SR-IBE from lattices. One application of SR-IBE is encrypted email supporting lightweight devices in which an email server plays the role of the untrusted server so that only non-revoked users can read their email messages.

2 Method

2.1 Preliminaries

2.1.1 Notations

For a binary string α and a positive integer n, let $|\alpha|$ denote its binary length and $[n]$ denote the set $\{1, \ldots, n\}$. If S is a finite set then $x \leftarrow S$ is the operation of choosing an element uniformly at random from S. For a probability distribution $\mathcal{D}$, $x \leftarrow \mathcal{D}$ denotes the operation of choosing an element according to $\mathcal{D}$. If α is either an algorithm or a set then $x \leftarrow \alpha$ is a simple assignment statement.

Let λ denote the security parameter. A function $f(\lambda)$ is *negligible*, denoted as $\mathsf{negl}(\lambda)$, if for every $c > 0$ there exists an λ_c such that $f(\lambda) < 1/\lambda^c$ for all $\lambda > \lambda_c$. An algorithm is probabilistic polynomial-time (PPT) computable if it is modeled as a probabilistic Turing machine whose running time is bounded by some polynomial function $\mathsf{poly}(\lambda)$. Two distribution ensembles $\{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $\{Y_\lambda\}_{\lambda\in\mathbb{N}}$ are *computationally indistinguishable*, if for any PPT algorithm D, and for sufficiently large λ, we have $|\Pr[D(X_\lambda) = 1] - \Pr[D(Y_\lambda) = 1]|$ is negligible in λ.

For a matrix $\mathbf{T} = [\mathbf{t}_1 \ldots \mathbf{t}_k] \in \mathbb{R}^{m\times k}$, let $\|\mathbf{T}\|$ denote the L_2 length of the longest column vector in $\mathbf{T}$, i.e., $\|\mathbf{T}\| := \max_i \|\mathbf{t}_i\|$ for $1 \le i \le k$; let $s_1(\mathbf{T})$ denote the largest singular value of $\mathbf{T}$, i.e., $s_1(\mathbf{T}) := \sup_{\mathbf{u}\in\mathbb{R}^k, \|\mathbf{u}\|=1} \|\mathbf{Tu}\|$. If $\mathbf{t}_1, \ldots, \mathbf{t}_k$ in $\mathbf{T}$ are linearly independent, let $\widetilde{\mathbf{T}} = [\tilde{\mathbf{t}}_1, \ldots, \tilde{\mathbf{t}}_k]$ denote the Gram-Schmidt orthogonalization of $\mathbf{T}$. For two matrices $\mathbf{X} \in \mathbb{R}^{n\times m}$ and $\mathbf{Y} \in \mathbb{R}^{m\times k}$, we have $\|\mathbf{X}\|, \|\mathbf{X}^\top\| \le s_1(\mathbf{X})$, and $s_1(\mathbf{XY}) \le s_1(\mathbf{X}) \cdot s_1(\mathbf{Y})$. For two matrices $\mathbf{X} \in \mathbb{R}^{n\times m_1}$ and $\mathbf{Y} \in \mathbb{R}^{n\times m_2}$, $[\mathbf{X} \mid \mathbf{Y}] \in \mathbb{R}^{n\times(m_1+m_2)}$ is the concatenation of the columns of $\mathbf{X}$ and $\mathbf{Y}$. For two matrices $\mathbf{X} \in \mathbb{R}^{n_1\times m}$ and $\mathbf{Y} \in \mathbb{R}^{n_2\times m}$, $[\mathbf{X}; \mathbf{Y}] \in \mathbb{R}^{(n_1+n_2)\times(m)}$ is the concatenation of the rows of $\mathbf{X}$ and $\mathbf{Y}$.

2.1.2 The binary tree data structure and the CS method

The complete subtree (CS) method was introduced by Naor et al. [16]. A binary tree along with the CS method is an efficient revocation mechanism, which has been widely used in systems [20–22]. To introduce this mechanism, we use the following notations: BT denotes a binary-tree; root denotes the root node of BT; θ denotes a node in the binary tree and η emphasizes that the node θ is a leaf node. The set $\mathsf{Path}(\mathsf{BT}, \eta_{\mathsf{ID}})$ stands for the collection of nodes on the path from the leaf η_{ID} to the root (including η_{ID} and the root). If θ is a non-leaf node then θ_ℓ, θ_r denote the left and right child of θ, respectively.

Before introducing the CS method, we describe the $\mathsf{KUNodes}$ algorithm [16] as follows. The KUNode algorithm takes as input a binary tree BT, a revocation list RL, and outputs a set of nodes Y, such that the subtrees with root $\theta \in \mathsf{Y}$ cover all leaves η_{ID} in $\mathsf{BT}\backslash\mathsf{RL}$ and do not cover any leaves η_{ID} for $\mathsf{ID} \in \mathsf{RL}$. The description of KUNode algorithm is as follows:

```
KUNodes(BT, RL):
    $X, Y \leftarrow \emptyset$;    $\forall \mathsf{ID} \in \mathsf{RL}$, add $\mathsf{Path}(\mathsf{BT}, \eta_{\mathsf{ID}})$ to $X$;
    $\forall \theta \in X$, if $\theta_\ell \notin X$ then add $\theta_\ell$ to $Y$, if $\theta_r \notin X$ then add $\theta_r$ to $Y$;
    If $Y = \emptyset$ then add $\mathsf{root}$ to $Y$, return $Y$;
```

We adopt the definition of the CS method given by Katsumata et al. [22], which consists of the following four algorithms:

$\mathsf{CS.Setup}(N) \rightarrow \mathsf{BT}$:	on input the number of users N, it outputs a binary tree BT with at least N and at most $2N$ leaves.
$\mathsf{CS.Assign}(\mathsf{BT}, \mathsf{ID}) \rightarrow (\eta_{\mathsf{ID}}, \mathsf{BT})$:	on input a binary tree BT and an identity ID, it randomly assigns the identity ID to a leaf node η_{ID}, to which no other identities have been assigned yet. Then, it outputs a leaf node η_{ID} and an "updated" binary tree BT.
$\mathsf{CS.Cover}(\mathsf{BT}, \mathsf{RL}) \rightarrow \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL})$:	on input a binary tree and a revocation list RL, it outputs a set of nodes $\mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL})$.
$\mathsf{CS.Match}(\mathsf{Path}(\mathsf{BT}, \eta_{\mathsf{ID}}), \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL})) \rightarrow \theta/\emptyset$:	on input $\mathsf{Path}(\mathsf{BT}, \eta_{\mathsf{ID}})$ and $\mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL})$, it outputs a node $\theta \in \mathsf{Path}(\mathsf{BT}, \eta_{\mathsf{ID}}) \cap \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL})$ if it exists. Otherwise, it outputs $\emptyset$.

2.1.3 Background on lattice

For any positive integers n, m and $q \geq 2$, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a vector $\mathbf{u} \in \mathbb{Z}_q^n$, we define the m-dimensional lattice $\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{z} = \mathbf{0}_n \mod q\}$ and $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{z} = \mathbf{u} \mod q\}$.

Let Λ be a lattice in $\mathbb{Z}^m$. For any vector $\mathbf{c} \in \mathbb{R}^m$ and any parameter $s \in \mathbb{R}_+$, define $\rho_{s,\mathbf{c}}(\mathbf{x}) = \exp(-\pi \frac{\|\mathbf{x}-\mathbf{c}\|^2}{s^2})$ and $\rho_{s,\mathbf{c}}(\Lambda) = \sum_{\mathbf{x}\in\Lambda} \rho_{s,\mathbf{c}}(\mathbf{x})$. The *discrete Gaussian distribution* over Λ with center $\mathbf{c}$ and Gaussian parameter s is $\mathcal{D}_{\Lambda,s,\mathbf{c}} = \frac{\rho_{s,\mathbf{c}(\mathbf{y})}}{\rho_{s,\mathbf{c}}(\Lambda)}$ for $\forall \mathbf{y} \in \Lambda$. If $\mathbf{c} = \mathbf{0}$, we conveniently use ρ_s and $\mathcal{D}_{\Lambda,s}$.

Lemma 1 ([23]) *Let Λ be an m-dimensional lattice. Let $\mathbf{T}$ be a basis for Λ, and suppose $\sigma \geq \|\widetilde{\mathbf{T}}\| \cdot \omega(\sqrt{\log m})$. Then* $\Pr[\|\mathbf{x}\| > \sigma\sqrt{m} : \mathbf{x} \leftarrow \mathcal{D}_{\Lambda,\sigma}] \leq \mathsf{negl}(m)$.

Lemma 2 ([23]) *Let $n, m, q > 0$ be positive integers with $m \geq 2n\lceil \log q \rceil$ and q a prime. Let σ be any positive real such that $\sigma \geq \omega(\sqrt{\log m})$. Then for $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ and $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sigma}$, the distribution of $\mathbf{u} = \mathbf{A}\mathbf{e} \mod q$ is statistically close to uniform over $\mathbb{Z}_q^n$. Furthermore, for a fixed $\mathbf{u} \in \mathbb{Z}_q^n$, the conditional distribution of $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sigma}$, given $\mathbf{A}\mathbf{e} = \mathbf{u} \mod q$ for a uniformly random $\mathbf{A}$ in $\mathbb{Z}_q^{n\times m}$* is *$\mathcal{D}_{\Lambda_q^{\mathbf{u}},\sigma}$ with all but negligible probability.*

We first recall two algorithms. The first in [23–25] generates a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ that is statistically close to uniform, together with a short trapdoor basis for the associated lattice $\Lambda_q^{\perp}(\mathbf{A})$. The second in [25] generates the basis for lattice $\Lambda_q^{\perp}(\mathbf{G})$, where $\mathbf{G}$ is what they called the primitive matrix.

Lemma 3 ([23–25]) *Let $n, m, q > 0$ be positive integers with $m \geq 2n\lceil \log q \rceil$ and q a prime. Then, we have:*

- *a PPT algorithm* $\mathsf{TrapGen}(n,m,q)$ *that outputs a pair* $(\mathbf{A},\mathbf{T_A}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}^{m\times m}$ *such that* $\mathbf{A}$ *is statistically close to uniform and* $\mathbf{T_A}$ is a basis for $\Lambda_q^{\perp}(\mathbf{A})$ *satisfying* $\|\widetilde{\mathbf{T_A}}\| \leq O(\sqrt{n\log q})$.
- *a fixed full rank matrix* $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ *such that the lattice* $\Lambda_q^{\perp}(\mathbf{G})$ *has a publicly known basis* $\mathbf{T_G} \in \mathbb{Z}^{m\times m}$ *with* $\|\widetilde{\mathbf{T_G}}\| \leq \sqrt{5}$.

We review some of the algorithms that allow one to securely delegate a trapdoor of a lattice to an arbitrary higher-dimensional extension given a short basis for the lattice. They can be obtained by combining corresponding results in [26, 27].

Lemma 4 ([26, 27]) *Let* $n, m, \bar{m}, q > 0$ *be positive integers with* $m > n$ *and* q *a prime. Then, there exist PPT algorithms as follows:*

$\mathsf{ExtRndLeft}(\mathbf{A},\mathbf{F},\mathbf{T_A},\sigma) \to \mathbf{T_{[A|F]}}$: *On input matrices* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{F} \in \mathbb{Z}_q^{n\times \bar{m}}$, *a basis* $\mathbf{T_A}$ of $\Lambda_q^{\perp}(\mathbf{A})$, *and a Gaussian parameter* $\sigma \geq \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log n})$, *outputs a matrix* $\mathbf{T_{[A|F]}} \in \mathbb{Z}^{(m+\bar{m})\times(m+\bar{m})}$ *distributed statistically close to* $(\mathcal{D}_{\Lambda_q^{\perp}([\mathbf{A}|\mathbf{F}]),\sigma})^{m+\bar{m}}$.

$\mathsf{ExtRndRight}(\mathbf{A},\mathbf{G},\mathbf{R},\mathbf{T_G},\sigma) \to \mathbf{T_{[A|AR+G]}}$: *On input full rank matrices* $\mathbf{A},\mathbf{G} \in \mathbb{Z}_q^{n\times m}$, a matrix $\mathbf{R} \in \mathbb{Z}_q^{m\times m}$, *a basis* $\mathbf{T_G}$ of $\Lambda_q^{\perp}(\mathbf{G})$, *and a Gaussian parameter* $\sigma \geq s_1(\mathbf{R}) \cdot \|\widetilde{\mathbf{T_G}}\| \cdot \omega(\sqrt{\log m})$, *outputs a matrix* $\mathbf{T_{[A|AR+G]}} \in \mathbb{Z}^{2m\times 2m}$ *distributed statistically close to* $(\mathcal{D}_{\Lambda_q^{\perp}([\mathbf{A}|\mathbf{AR}+\mathbf{G}]),\sigma})^{2m}$.

The following algorithms allow one to sample short vectors from a given lattice equipped with a short basis.

Lemma 5 ([25, 26]) *Let* $n, m, \bar{m}, q > 0$ *be positive integers with* $m \geq 2n\lceil\log q\rceil$ *and* q *a prime. Then, we have the following algorithms:*

$\mathsf{SamplePre}(\mathbf{A},\mathbf{T_A},\mathbf{u},\sigma) \to \mathbf{e}$: *on input a full rank matrix* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, *a basis* $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$ of $\Lambda_q^{\perp}(\mathbf{A})$, *a vector* $\mathbf{u} \in \mathbb{Z}_q^n$, *and a Gaussian parameter* $\sigma \geq \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log m})$, *it outputs a vector* $\mathbf{e} \in \mathbb{Z}^m$ *sampled from a distribution statistically close to* $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\sigma}$.

$\mathsf{SampleLeft}(\mathbf{A},\mathbf{F},\mathbf{u},\mathbf{T_A},\sigma) \to \mathbf{e}$: *On input a full rank matrix* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, *a matrix* $\mathbf{F} \in \mathbb{Z}_q^{n\times \bar{m}}$, *a vector* $\mathbf{u} \in \mathbb{Z}_q^n$, *a basis* $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$ *of* $\Lambda_q^{\perp}(\mathbf{A})$, *and a Gaussian parameter* $\sigma \geq \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log(m+\bar{m})})$, *it outputs a vector* $\mathbf{e} \in \mathbb{Z}^{m+\bar{m}}$ *sampled from a distribution statistically close to* $\mathcal{D}_{\Lambda_q^{\mathbf{u}}([\mathbf{A}|\mathbf{F}]),\sigma}$.

The learning with errors (LWE) assumption was introduced by Regev [28]. For integers n, m, a prime q, a real $\alpha \in (0,1)$ such that $\alpha q > 2\sqrt{n}$, and a PPT algorithm $\mathcal{A}$, the advantage for learning with errors problem $\mathsf{LWE}_{n,m,q,\mathcal{D}_{\mathbb{Z}^m,\alpha q}}$ of $\mathcal{A}$ is defined as $\left|\Pr[\mathcal{A}(\mathbf{A}, \mathbf{A}^\top \mathbf{s} + \mathbf{x}) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{v}) = 1]\right|$, where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\alpha q}$, $\mathbf{v} \leftarrow \mathbb{Z}_q^m$. We say that the LWE assumption holds if the above advantage is negligible for all PPT adversary $\mathcal{A}$.

2.1.4 Facts

Lemma 6 ([26, 29]) *Let* $\mathbf{R}$ *be a* $m \times k$ *matrix chosen at random from* $\{-1,1\}^{m\times k}$*, then there exists a universal constant* C *such that* $\Pr[s_1(\mathbf{R}) > C\sqrt{m+k}] < e^{-(m+k)}$.

Lemma 7 ([30]) *Suppose that* $m > (n+1)\log q + \omega(\log n)$ *and that* q *is a prime. Let* $\mathbf{A}, \mathbf{B}$ *be matrices chosen uniformly in* $\mathbb{Z}_q^{n\times m}$ *and let* $\mathbf{R}$ *be an* $m \times m$ *matrix chosen uniformly in* $\{-1,1\}^{m\times m} \mod q$*. Then, for all vectors* $\mathbf{w}$ *in* $\mathbb{Z}_q^m$*, the distribution of* $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^\top\mathbf{w})$ *is statistically close to the distribution of* $(\mathbf{A}, \mathbf{B}, \mathbf{R}^\top\mathbf{w})$.

Definition 1 *(FRD* [26]*)* Let n, q be positive integers with q a prime. We say that a function $\mathsf{H} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ is a full-rank difference (FRD) map if: for all distinct $\mathsf{ID}, \mathsf{ID}' \in \mathbb{Z}_q^n$, the matrix $\mathsf{H}(\mathsf{ID}) - \mathsf{H}(\mathsf{ID}') \in \mathbb{Z}_q^{n\times n}$ is full rank, and H is computable in polynomial time in $n \log q$.

2.2 Framework of dSR-IBKS

As shown in Fig. 1, our dSR-IBKS involves four parties: KGC, sender, recipient and server. Algorithms among the parties are as following:

$\mathsf{Setup}(1^\lambda) \to (\mathsf{pp}, \mathsf{sk}_{\mathsf{kgc}})$	is run by the KGC. It takes as input a security parameter λ and outputs a public parameter pp, and the KGC's secret key $\mathsf{sk}_{\mathsf{kgc}}$ (also called a master key). We assume that the system parameter params is contained in pp and pp is an implicit input of all other algorithms.
$\mathsf{PrivKG}(\mathsf{sk}_{\mathsf{kgc}}, \mathsf{ID}) \to \mathsf{Priv}_{\mathsf{ID}}$	is run by the KGC. It takes as input the KGC's secret key $\mathsf{sk}_{\mathsf{kgc}}$ and the recipient's identity ID, and outputs a private key $\mathsf{Priv}_{\mathsf{ID}}$ for the recipient. The private key must be sent to the recipient through a secure channel.
$\mathsf{ServKG}(\mathsf{sk}_{\mathsf{kgc}}, S_{\mathsf{ID}}) \to \mathsf{Serv}_{\mathsf{ID}}$	is run by the KGC. It takes as input the KGC's secret key $\mathsf{sk}_{\mathsf{kgc}}$ and the server's identity S_{ID}, and outputs a private key $\mathsf{Serv}_{\mathsf{ID}}$ for the server. The private key must be sent to the server through a secure

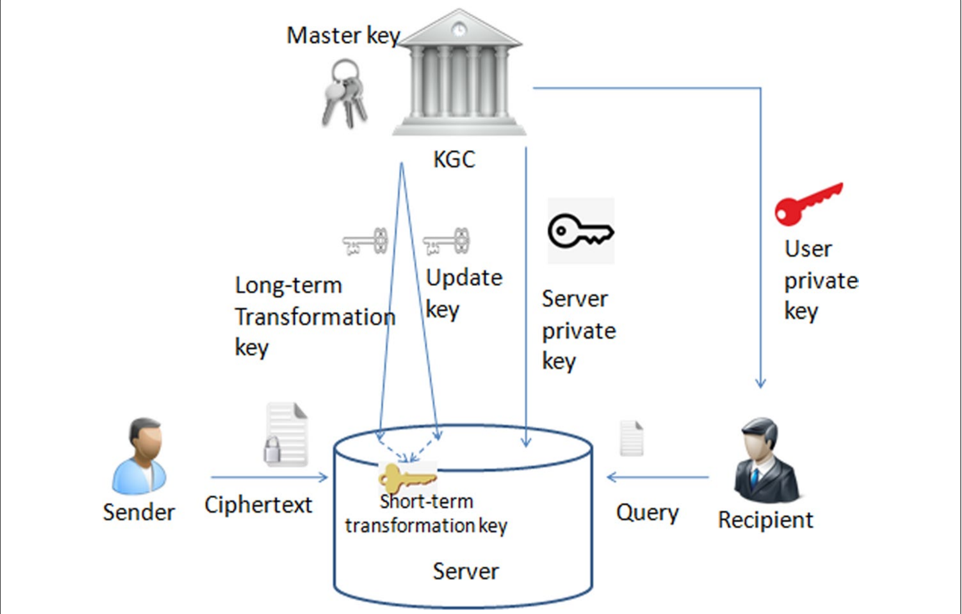

Fig. 1 System model of dSR-IBKS

channel.

$\mathsf{L-TranKG}(\mathsf{sk_{kgc}},\mathsf{ID}) \to (\mathsf{sk_{ID}},\mathsf{sk'_{kgc}})$ is run by the KGC. It takes as input the KGC's secret key $\mathsf{sk_{kgc}}$, an identity ID, and may update the KGC's secret key $\mathsf{sk_{kgc}}$. Then, it outputs a long-term transformation key $\mathsf{sk_{ID}}$ and the KGC's "updated" state $\mathsf{sk'_{kgc}}$. The long-term transformation key $\mathsf{sk_{ID}}$ is sent to the server through a public channel.

$\mathsf{UpdKG}(\mathsf{sk_{kgc}},\mathsf{t},\mathsf{RL_t}) \to (\mathsf{uk_t},\mathsf{sk'_{kgc}})$ is run by the KGC. It takes as input the KGC's secret key $\mathsf{sk_{kgc}}$, a time period t, a revocation list $\mathsf{RL_t}$, and may update the KGC's secret key $\mathsf{sk_{kgc}}$. Then, it outputs an update key $\mathsf{uk_t}$ and the KGC's "updated" state $\mathsf{sk'_{kgc}}$. The updated key $\mathsf{uk_t}$ is sent to the server through a public channel.

$\mathsf{S-TranKG}(\mathsf{sk_{ID}},\mathsf{uk_t}) \to \mathsf{e_{ID,t}}/\perp$ is run by the server. It takes as input a long-term transformation key $\mathsf{sk_{ID}}$ for identity ID, an update key $\mathsf{uk_t}$ for time period t, and outputs a short-term transformation key $\mathsf{e_{ID,t}}$ or the special symbol $\perp$ indicating that ID has been revoked.

$\mathsf{Enc}(\mathsf{ID},S_{\mathsf{ID}},\mathsf{t},\mathsf{kw}) \to \mathsf{ct_{ID,t,kw}}$ is run by the sender. It takes as input the recipient's identity ID, server's identity S_{ID}, a time period t, a ciphertext keyword kw, and outputs a ciphertext $\mathsf{ct}_{\mathsf{ID},\mathsf{t},\mathsf{kw}}$. The ciphertext is sent to the server.

$\mathsf{QueKG}(\mathsf{Priv}_{\mathsf{ID}}, \mathcal{S}_{\mathsf{ID}}, \mathsf{t}, \mathsf{kw}') \to \mathbf{q}_{\mathsf{ID},\mathsf{t},\mathsf{kw}}$ is run by the recipient himself. It takes as input his private key $\mathsf{Priv}_{\mathsf{ID}}$, the server's identity $\mathcal{S}_{\mathsf{ID}}$, a time period t and a target keyword kw', and outputs a search query $\mathbf{q}_{\mathsf{ID},\mathsf{t},\mathsf{kw}'}$.

$\mathsf{Test}(\mathsf{ct}_{\mathsf{ID},\mathsf{t},\mathsf{kw}}, \mathsf{t}, \mathbf{e}_{\mathsf{ID},\mathsf{t}}, \mathbf{q}_{\mathsf{ID},\mathsf{t},\mathsf{kw}'}, \mathsf{Serv}_{\mathsf{ID}}) \to \{0,1\}$: On input a ciphertext $\mathsf{ct}_{\mathsf{ID},\mathsf{t},\mathsf{kw}}$, a time period t, and the short-term transformation key $\mathbf{e}_{\mathsf{ID},\mathsf{t}}$, recipient's query $\mathbf{q}_{\mathsf{ID},\mathsf{t},\mathsf{kw}'}$ and its own secret key $\mathsf{Serv}_{\mathsf{ID}}$, the server checks whether $\mathsf{kw} = \mathsf{kw}'$. If $\mathsf{kw} = \mathsf{kw}'$, server outputs 1 and otherwise outputs 0.

The *correctness* for a dSR-IBKS requires that for all security parameter $\lambda \in \mathbb{N}$, if ID is not revoked at time period t and if all parties follow the prescribed algorithms, then the Test algorithm outputs 1 only if $\mathsf{kw} = \mathsf{kw}'$.

2.3 Security model of dSR-IBKS

We considered the selective security, dSR-sID-CKA security, in which the adversary should announce the challenge identities $\mathsf{ID}^*, \mathcal{S}^*_{\mathsf{ID}}$, time period t^* and challenge keyword $(\mathsf{kw}^*_0, \mathsf{kw}^*_1)$ before the execution of algorithm Setup. Let SKList be a set that initially contains $(\mathsf{kgc}, \mathsf{sk}_{\mathsf{kgc}})$, and into which identity/long-term transformation key pairs $(\mathsf{ID}, \mathsf{sk}_{\mathsf{ID}})$ generated during the security game will be stored. From now on, whenever a new secret key $\mathsf{sk}_{\mathsf{ID}}$ is generated for ID or the secret key $\mathsf{sk}_{\mathsf{kgc}}$ is updated during the execution of $\mathsf{L-TranKG}$ or UpdKG, the challenger will store the pair $(\mathsf{ID}, \mathsf{sk}_{\mathsf{ID}})$ or update $(\mathsf{kgc}, \mathsf{sk}_{\mathsf{kgc}})$ in SKList, and we will not explicitly mention this addition/update. Also, set $\mathsf{PrivList}$ to a set that stores the identity/private key pair $(\mathsf{ID}, \mathsf{Priv}_{\mathsf{ID}})$ generated during the secure game. The challenger will store the pair $(\mathsf{ID}, \mathsf{Priv}_{\mathsf{ID}})$ in $\mathsf{PrivList}$ whenever a new private key $\mathsf{Priv}_{\mathsf{ID}}$ is generated for ID during the execution of PrivKG and we will not explicitly mention this addition as well. On this basis, the long-term transformation key generation and reveal queries are explicitly separated to describe situations where some $\mathsf{sk}_{\mathsf{ID}}$ has been generated but not revealed to an adversary. And, the same processing is done for private key generation and reveal queries. In addition, the "revoke" and "update key" queries in [17, 19] are merged into a single "revoke & update key" query, and the concept of the current time period t_{cu} coordinated with the adversary's "revoke & update key" query is introduced.

Definition 2 *(dSR-sID-CKA security)* Let $\mathcal{O}^{\mathsf{dSR\text{-}IBKS}}$ be the set of queries as follows:

Long-term Transformation Key Generation Query: upon a query ID from the adversary $\mathcal{A}$, where it is required that $(\mathsf{ID}, *) \notin \mathsf{SKList}$, the challenge $\mathcal{C}$ runs $(\mathsf{sk_{ID}}, \mathsf{sk'_{kgc}}) \leftarrow \mathsf{L-TranKG}(\mathsf{sk_{kgc}}, \mathsf{ID})$ and returns nothing to $\mathcal{A}$. We require that all identities ID appearing in the following queries be "activated" in the sense that $\mathsf{sk_{ID}}$ is generated via this query and hence $(\mathsf{ID}, \mathsf{sk_{ID}}) \in \mathsf{SKList}$.

Long-term Transformation Key Reveal Query: upon a query ID from $\mathcal{A}$, $\mathcal{C}$ finds $\mathsf{sk_{ID}}$ from SKList and returns it to $\mathcal{A}$.

Revoke [MYAMP Update Key Query:] upon a query RL (which represents the set of identities that are going to be revoked in the next time period) from $\mathcal{A}$, $\mathcal{C}$ checks whether $\mathsf{RL_{t_{cu}}} \subset \mathsf{RL}$. If not, it returns $\perp$; otherwise, it increments the current time period by $\mathsf{t_{cu}} \leftarrow \mathsf{t_{cu}} + 1$, sets $\mathsf{RL_{t_{cu}}} \leftarrow \mathsf{RL}$, runs $(\mathsf{uk_{t_{cu}}}, \mathsf{sk'_{kgc}}) \leftarrow \mathsf{UpdKG}(\mathsf{sk_{kgc}}, \mathsf{t_{cu}}, \mathsf{RL_{t_{cu}}})$ and returns $\mathsf{uk_{t_{cu}}}$ to $\mathcal{A}$.

Short-term Transformation Key Reveal Query: upon a query $(\mathsf{ID}, \mathsf{t})$ from $\mathcal{A}$, $\mathcal{C}$ checks whether conditions $\mathsf{t} \leq \mathsf{t_{cu}}$, $\mathsf{ID} \notin \mathsf{RL_t}$ and $(\mathsf{ID}, \mathsf{t}) \neq (\mathsf{ID}^*, \mathsf{t}^*)$ meet at the same time. If not, it returns $\perp$; otherwise, it finds $\mathsf{sk_{ID}}$ from SKList, runs $\mathsf{e_{ID,t}} \leftarrow \mathsf{S-TranKG}(\mathsf{sk_{ID}}, \mathsf{uk_t})$ and returns $\mathsf{e_{ID,t}}$ to $\mathcal{A}$.

Private Key Generation Query: upon a query ID (resp. $\mathcal{S}_{\mathsf{ID}}$)from $\mathcal{A}$, where it is required that $(\mathsf{ID}, *) \notin \mathsf{PrivList}$, $\mathcal{C}$ runs $\mathsf{Priv_{ID}} \leftarrow \mathsf{PrivKG}(\mathsf{sk_{kgc}}, \mathsf{ID})$ and returns nothing to $\mathcal{A}$ (resp. $\mathsf{Serv}_{\mathcal{S}_{\mathsf{ID}}} \leftarrow \mathsf{ServKG}(\mathsf{sk_{kgc}}, \mathcal{S}_{\mathsf{ID}})$). Similarly, we require that all identities ID appearing in the following queries be "activated" in the sense that $\mathsf{Priv_{ID}}$ is generated via this query and hence $(\mathsf{ID}, \mathsf{Priv_{ID}}) \in \mathsf{PrivList}$.

Private Key Reveal Query: upon a query ID (resp. $\mathcal{S}_{\mathsf{ID}}$) from $\mathcal{A}$, $\mathcal{C}$ finds $\mathsf{Priv_{ID}}$ (resp. $\mathsf{Serv}_{\mathcal{S}_{\mathsf{ID}}}$) from $\mathsf{PrivList}$ and returns it to $\mathcal{A}$.

Search Query: upon a query $(\mathsf{ID}, \mathsf{t}, \mathsf{kw})$ from $\mathcal{A}$, $\mathcal{C}$ finds $\mathsf{Priv_{ID}}$ from $\mathsf{PrivList}$, runs $\mathbf{q}_{\mathsf{ID,t,kw}} \leftarrow \mathsf{QueKG}(\mathsf{Priv_{ID}}, \mathsf{t}, \mathsf{kw})$ and returns $\mathbf{q}_{\mathsf{ID,t,kw}}$ to $\mathcal{A}$.

A dSR-IBKS scheme is dSR-sID-CKA secure if any PPT adversary $\mathcal{A}$ has negligible advantage in experiment $\mathsf{Exp}^{\mathsf{dSR-sID-CKA}}_{\mathcal{A},\mathcal{O}^{\mathsf{dSR\text{-}IBKS}}}(\lambda)$ as follows.

$$\boxed{\begin{array}{l}
\mathsf{Exp}^{\mathsf{dSR-sID-CKA}}_{\mathcal{A},\mathcal{O}^{dSR\text{-}IBKS}}(\lambda):\\
\mathsf{ID}^*, S^*_{\mathsf{ID}}, \mathsf{t}^*, \{\mathsf{kw}^*_0, \mathsf{kw}^*_1\} \leftarrow \mathcal{A};\\
(\mathsf{pp}, \mathsf{sk_{kgc}}) \leftarrow \mathsf{Setup}(1^\lambda);\ \mathsf{SKList} \leftarrow (\mathsf{kgc}, \mathsf{sk_{kgc}});\ (\mathsf{uk}_1, \mathsf{sk}'_{\mathsf{kgc}}) \leftarrow\\
\mathsf{UpdKG}(\mathsf{sk_{kgc}}, \mathsf{t_{cu}} = 1, \mathsf{RL}_1 = \emptyset);\\
b \leftarrow \{0,1\};\ \mathsf{ct}_{\mathsf{ID}^*,\mathsf{t}^*,\mathsf{kw}^*_b} \leftarrow \mathsf{Enc}(\mathsf{ID}^*, S^*_{\mathsf{ID}}, \mathsf{t}^*, \mathsf{kw}^*_b);\\
b' \leftarrow \mathcal{A}^{\mathcal{O}^{SR\text{-}IBKS}}(\mathsf{ct}_{\mathsf{ID}^*,\mathsf{t}^*,\mathsf{kw}^*_b});\\
\textit{Return 1 if } b' = b \textit{ and 0 otherwise.}
\end{array}}$$

The following restrictions are made in the experiments $\mathsf{Exp}^{\mathsf{dSR-sID-CKA}}_{\mathcal{A},\mathcal{O}^{\mathsf{dSR\text{-}IBKS}}}(\lambda)$:

1 If both $\mathsf{ServKG}(S^*_{\mathsf{ID}})$ and $\mathsf{PrivKG}(\mathsf{ID}^*)$ were queried, then $\mathsf{ID}^* \in \mathsf{RL_t}$ for some $\mathsf{t} \leq \mathsf{t}^*$.
2 If $\mathsf{PrivKG}(\mathsf{ID}^*)$ was not queried, then $\mathsf{QueKG}(\cdot)$ can not be queried on $(\mathsf{id}, \mathsf{kw}, \mathsf{t}) = (\mathsf{id}^*, \mathsf{kw}^*, \mathsf{t}^*)$.
3 If $\mathsf{ServKG}(S^*_{\mathsf{ID}})$ was queried and $\mathsf{ID}^* \notin \mathsf{RL}_{\mathsf{t}^*}$, then $\mathsf{PrivKG}(\mathsf{ID}^*)$ can not be queried.

The advantage of $\mathcal{A}$ in the experiment $\mathsf{Exp}^{\mathsf{dSR-sID-CKA}}_{\mathcal{A},\mathcal{O}^{\mathsf{dSR\text{-}IBKS}}}(\lambda)$ is defined as:

$$\mathsf{Adv}^{\mathsf{dSR-sID-CKA}}_{\mathcal{A},\mathcal{O}^{\mathsf{dSR\text{-}IBKS}}}(\lambda) = 2 \cdot \left| \Pr\left[\mathsf{Exp}^{\mathsf{dSR-sID-CKA}}_{\mathcal{A},\mathcal{O}^{\mathsf{dSR\text{-}IBKS}}}(\lambda) = 1\right] - \frac{1}{2} \right| \quad (1)$$

2.4 Our lattice-based dSR-IBKS scheme

In the following, we formally describe our dSR-IBKS scheme.

$\mathsf{Setup}(1^\lambda) \rightarrow (\mathsf{pp}, \mathsf{sk_{kgc}})$: On input a security parameter λ, the KGC proceeds as follows.

1 Choose positive integers n, N, q, k, m, s_0, s_1, a real α and a prime q, where $k = \lceil \log q \rceil$ and N is the maximal number of users that the system will support. Select an FRD map $\mathsf{H} : \mathbb{Z}_q^n \rightarrow \mathbb{Z}_q^{n\times n}$ (see Sect. 2.1.4). Let the identity space $\mathcal{ID} = \mathbb{Z}_q^n$, the time space $\mathcal{T} \subset \mathcal{ID} = \mathbb{Z}_q^n$, the keyword space $\mathcal{KW} \subset \mathcal{ID} = \mathbb{Z}_q^n$. Set the system parameter $\mathsf{params} = (n, N, q, k, m, s_0, s_1, \alpha, \mathsf{H}, \mathcal{ID}, \mathcal{T}, \mathcal{KW})$.
2 Generate three independent pairs $(\mathbf{A}, \mathbf{T_A})$, $(\mathbf{B}, \mathbf{T_B})$ and $(\mathbf{C}, \mathbf{T_C})$ by running $\mathsf{TrapGen}(n, m, q)$.
3 Select $\mathbf{u} \leftarrow \mathbb{Z}_q^n$ and $\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_1, \mathbf{B}_2, \mathbf{C}_1, \mathbf{C}_2 \leftarrow \mathbb{Z}_q^{n\times m}$.
4 Create a binary tree by running $\mathsf{BT_{kgc}} \leftarrow \mathsf{CS.Setup}(N)$.
5 Set the public parameter $\mathsf{pp} = (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{C}_1, \mathbf{C}_2, \mathbf{u}, \mathsf{params})$ and the KGC's secret

key $\mathsf{sk}_{\mathsf{kgc}} = (\mathbf{T_A}, \mathbf{T_B}, \mathbf{T_C}, \mathsf{BT}_{\mathsf{kgc}})$.

6 Output pp and $\mathsf{sk}_{\mathsf{kgc}}$.

$\mathsf{PrivKG}(\mathsf{sk}_{\mathsf{kgc}}, \mathsf{ID}) \to \mathsf{Priv}_{\mathsf{ID}}$: On input the KGC's secret key $\mathsf{sk}_{\mathsf{kgc}}$ and the recipient's identity ID, the KGC goes as follows:

1 Extend its basis by running

$$\mathbf{T}_{\mathbf{B}_{\mathsf{ID}}} \leftarrow \mathsf{ExtRndLeft}(\mathbf{B}, \mathbf{B}_1 + \mathsf{H}(\mathsf{ID})\mathbf{G}, \mathbf{T_B}, s_0), \tag{2}$$

where $\mathbf{B}_{\mathsf{ID}} = [\mathbf{B} \mid \mathbf{B}_1 + \mathsf{H}(\mathsf{ID})\mathbf{G}] \in \mathbb{Z}_q^{n\times 2m}$, $\mathbf{T}_{\mathbf{B}_{\mathsf{ID}}} \in \mathbb{Z}^{2m\times 2m}$.

2 Output $\mathsf{Priv}_{\mathsf{ID}} = \mathbf{T}_{\mathbf{B}_{\mathsf{ID}}}$ as the private key of user ID.

3 Send $\mathsf{Priv}_{\mathsf{ID}}$ to the user ID through a secret channel.

$\mathsf{ServKG}(\mathsf{sk}_{\mathsf{kgc}}, \mathcal{S}_{\mathsf{ID}}) \to \mathsf{Serv}_{\mathsf{ID}}$: On input the KGC's secret key $\mathsf{sk}_{\mathsf{kgc}}$ and the designated server's identity $\mathcal{S}_{\mathsf{ID}} \in \mathcal{ID}$, the KGC proceeds as follows:

1 Sample $\mathbf{T}_{\mathbf{C}_{\mathcal{S}_{\mathsf{ID}}}} \leftarrow \mathsf{ExtRndLeft}(\mathbf{C}, \mathbf{C}_1 + \mathsf{H}(\mathcal{S}_{\mathsf{ID}})\mathbf{G}, \mathbf{T_C}, s_0)$, where $\mathbf{C}_{\mathcal{S}_{\mathsf{ID}}} = [\mathbf{C} \mid \mathbf{C}_1 + \mathsf{H}(\mathcal{S}_{\mathsf{ID}})\mathbf{G}] \in \mathbb{Z}_q^{n\times 2m}$.

2 Output the private key $\mathsf{Serv}_{\mathsf{ID}} = \mathbf{T}_{\mathbf{C}_{\mathcal{S}_{\mathsf{ID}}}} \in \mathbb{Z}^{2m\times 2m}$.

$\mathsf{L-TranKG}(\mathsf{sk}_{\mathsf{kgc}}, \mathsf{ID}) \to (\mathsf{sk}_{\mathsf{ID}}, \mathsf{sk}'_{\mathsf{kgc}})$: On input the KGC's secret key $\mathsf{sk}_{\mathsf{kgc}}$ and an identity $\mathsf{ID} \in \mathcal{ID}$, the KGC goes as follows:

1 Run $(\mathsf{BT}_{\mathsf{kgc}}, \eta_{\mathsf{ID}}) \leftarrow \mathsf{CS.Assign}(\mathsf{BT}_{\mathsf{kgc}}, \mathsf{ID})$.

2 For each $\theta \in \mathsf{path}(\mathsf{BT}_{\mathsf{kgc}}, \eta_{\mathsf{ID}})$, if $\mathbf{u}_{\mathsf{kgc},\theta}$ is undefined, pick $\mathbf{u}_{\mathsf{kgc},\theta} \leftarrow \mathbb{Z}_q^n$, update $\mathsf{sk}_{\mathsf{kgc}}$ by storing $\mathbf{u}_{\mathsf{kgc},\theta}$ in the node $\theta \in \mathsf{BT}_{\mathsf{kgc}}$. Sample $\mathbf{e}_{\mathsf{ID},\theta} \leftarrow \mathsf{SampleLeft}(\mathbf{A}, \mathbf{A}_1 + \mathsf{H}(\mathsf{ID})\mathbf{G}, \mathbf{T_A}, \mathbf{u}_{\mathsf{kgc},\theta}, s_1)$, where $\mathbf{A}_{\mathsf{ID}} = [\mathbf{A} \mid \mathbf{A}_1 + \mathsf{H}(\mathsf{ID})\mathbf{G}] \in \mathbb{Z}_q^{n\times 2m}$. Note that $\mathbf{e}_{\mathsf{ID},\theta} \in \mathbb{Z}^{2m}$ and $\mathbf{A}_{\mathsf{ID}} \cdot \mathbf{e}_{\mathsf{ID},\theta} = \mathbf{u}_{\mathsf{kgc},\theta}$.

3 Output $\mathsf{sk}_{\mathsf{ID}} = \left(\mathsf{path}(\mathsf{BT}_{\mathsf{kgc}}, \eta_{\mathsf{ID}}), (\mathbf{e}_{\mathsf{ID},\theta})_{\theta\in\mathsf{path}(\mathsf{BT}_{\mathsf{kgc}},\eta_{\mathsf{ID}})}\right)$ as the long-term transformation key and the updated secret key $\mathsf{sk}'_{\mathsf{kgc}}$.

4 Send $\mathsf{sk}_{\mathsf{ID}}$ to the server through a public channel.

$\mathsf{UpdKG}(\mathsf{sk}_{\mathsf{kgc}}, \mathsf{t}, \mathsf{RL}_{\mathsf{t}}) \to (\mathsf{uk}_{\mathsf{t}}, \mathsf{sk}'_{\mathsf{kgc}})$: On input the KGC's secret key $\mathsf{sk}_{\mathsf{kgc}}$, a time period $\mathsf{t} \in \mathcal{T}$, a revocation list RL_{t}, the KGC works as follows:

1 Run $\mathsf{KUNodes}(\mathsf{BT}_{\mathsf{kgc}}, \mathsf{RL}_{\mathsf{t}}) \leftarrow \mathsf{CS.Cover}(\mathsf{BT}_{\mathsf{kgc}}, \mathsf{RL}_{\mathsf{t}})$ and check whether $\mathbf{u}_{\mathsf{kgc},\theta}$ is defined for each $\theta \in \mathsf{KUNodes}(\mathsf{BT}_{\mathsf{kgc}}, \mathsf{RL}_{\mathsf{t}})$. If not, pick $\mathbf{u}_{\mathsf{kgc},\theta} \leftarrow \mathbb{Z}_q^n$, update $\mathsf{sk}_{\mathsf{kgc}}$ by storing $\mathbf{u}_{\mathsf{kgc},\theta}$ in node $\theta \in \mathsf{BT}_{\mathsf{kgc}}$. Sample $\mathbf{e}_{\mathsf{t},\theta} \leftarrow \mathsf{SampleLeft}(\mathbf{A}, \mathbf{A}_{\mathsf{t}}, \mathbf{T_A}, \mathbf{u} - \mathbf{u}_{\mathsf{kgc},\theta}, s_1)$, where $\mathbf{A}_{\mathsf{t}} = [\mathbf{A} \mid \mathbf{A}_2 + \mathsf{H}(\mathsf{t})\mathbf{G}] \in \mathbb{Z}_q^{n\times 2m}$. Note that $\mathbf{e}_{\mathsf{t},\theta} \in \mathbb{Z}^{2m}$ and $\mathbf{A}_{\mathsf{t}} \cdot \mathbf{e}_{\mathsf{t},\theta} = \mathbf{u} - \mathbf{u}_{\mathsf{kgc},\theta}$.

2 Output $\mathsf{uk}_{\mathsf{t}} = \left(\mathsf{KUNodes}(\mathsf{BT}_{\mathsf{kgc}}, \mathsf{RL}_{\mathsf{t}}), (\mathbf{e}_{\mathsf{t},\theta})_{\theta\in\mathsf{KUNodes}(\mathsf{BT}_{\mathsf{kgc}},\mathsf{RL}_{\mathsf{t}})}\right)$ as the update key and the (possibly) updated secret key $\mathsf{sk}'_{\mathsf{kgc}}$.

$\mathsf{S-TranKG}(\mathsf{sk_{ID}}, \mathsf{uk_t}) \to \mathbf{e}_{\mathsf{ID,t}}/\perp$: On input a long-term transformation key $\mathsf{sk_{ID}}$ for identity $\mathsf{ID} \in \mathcal{ID}$, an update key $\mathsf{uk_t}$ for time period $\mathsf{t} \in \mathcal{T}$, the server goes as follows:

1 Extract $\mathsf{path}(\mathsf{BT_{kgc}}, \eta_{\mathsf{ID}})$ in $\mathsf{sk_{ID}}$ and $\mathsf{KUNodes}(\mathsf{BT_{kgc}}, \mathsf{RL_t})$ in $\mathsf{uk_t}$ and run $\theta/\emptyset \leftarrow \mathsf{CS.Match}(\mathsf{path}(\mathsf{BT_{kgc}}, \eta_{\mathsf{ID}}), \mathsf{KUNodes}(\mathsf{BT_{kgc}}, \mathsf{RL_t}))$. If the output is $\emptyset$, output $\perp$. Otherwise, extract $\mathbf{e}_{\mathsf{ID},\theta}, \mathbf{e}_{\mathsf{t},\theta} \in \mathbb{Z}^{2m}$ in $\mathsf{sk_{ID}}, \mathsf{uk_t}$, respectively, and parse it as

$$\mathbf{e}_{\mathsf{ID},\theta} = [\mathbf{e}^{\mathsf{L}}_{\mathsf{ID},\theta} \mid \mathbf{e}^{\mathsf{R}}_{\mathsf{ID},\theta}], \quad \mathbf{e}_{\mathsf{t},\theta} = [\mathbf{e}^{\mathsf{L}}_{\mathsf{t},\theta} \mid \mathbf{e}^{\mathsf{R}}_{\mathsf{t},\theta}]. \tag{3}$$

where $\mathbf{e}^{\mathsf{L}}_{\mathsf{ID},\theta}, \mathbf{e}^{\mathsf{R}}_{\mathsf{ID},\theta}, \mathbf{e}^{\mathsf{L}}_{\mathsf{t},\theta}, \mathbf{e}^{\mathsf{R}}_{\mathsf{t},\theta} \in \mathbb{Z}^m$. Compute

$$\mathbf{e}_{\mathsf{ID,t}} = [\mathbf{e}^{\mathsf{L}}_{\mathsf{ID},\theta} + \mathbf{e}^{\mathsf{L}}_{\mathsf{t},\theta} \mid \mathbf{e}^{\mathsf{R}}_{\mathsf{ID},\theta} \mid \mathbf{e}^{\mathsf{R}}_{\mathsf{t},\theta}]. \tag{4}$$

Note that $\mathbf{e}_{\mathsf{ID,t}} \in \mathbb{Z}^{3m}$ and $[\mathbf{A} \mid \mathbf{A}_1 + \mathsf{H}(\mathsf{ID})\mathbf{G} \mid \mathbf{A}_2 + \mathsf{H}(\mathsf{t})\mathbf{G}] \cdot \mathbf{e}_{\mathsf{ID,t}} = \mathbf{u}$.

2 Output the short-term transformation key $\mathbf{e}_{\mathsf{ID,t}}$.

$\mathsf{Enc}(\mathsf{ID}, \mathcal{S}_{\mathsf{ID}}, \mathsf{t}, \mathsf{kw}) \to \mathsf{ct}_{\mathsf{ID,t,kw}}$: On input user identity $\mathsf{ID} \in \mathcal{ID}$, server identity $\mathcal{S}_{\mathsf{ID}} \in \mathcal{ID}$, a time period $\mathsf{t} \in \mathcal{T}$, a ciphertext keyword $\mathsf{kw} \in \mathcal{KW}$, the sender works as follows:

1 Set

$$\begin{aligned} \mathbf{A}_{\mathsf{ID,t}} &= [\mathbf{A} \mid \mathbf{A}_1 + \mathsf{H}(\mathsf{ID})\mathbf{G} \mid \mathbf{A}_2 + \mathsf{H}(\mathsf{t})\mathbf{G}] \in \mathbb{Z}_q^{n\times 3m} \\ \mathbf{B}_{\mathsf{ID,t,kw}} &= [\mathbf{B} \mid \mathbf{B}_1 + \mathsf{H}(\mathsf{ID})\mathbf{G} \mid \mathbf{B}_2 + \mathsf{H}(\mathsf{t})\mathbf{G} \mid \mathbf{B}_3 + \mathsf{H}(\mathsf{kw})\mathbf{G}] \in \mathbb{Z}_q^{n\times 4m} \\ \mathbf{C}_{\mathcal{S}_{\mathsf{ID}},\mathsf{t}} &= [\mathbf{C} \mid \mathbf{C}_1 + \mathsf{H}(\mathcal{S}_{\mathsf{ID}})\mathbf{G} \mid \mathbf{C}_2 + \mathsf{H}(\mathsf{t})\mathbf{G}] \in \mathbb{Z}_q^{n\times 3m}. \end{aligned} \tag{5}$$

2 Sample $\mathbf{s}, \mathbf{s}', \mathbf{s}'' \leftarrow \mathbb{Z}_q^n$, $x \leftarrow D_{\mathbb{Z},\alpha q}$, $\mathbf{x}, \mathbf{x}', \mathbf{x}'' \leftarrow \mathcal{D}_{\mathbb{Z}^m,\alpha q}$.

3 Choose $\mathbf{R}_{11}, \mathbf{R}_{12}, \mathbf{R}_{21}, \mathbf{R}_{22}, \mathbf{R}_{23}, \mathbf{R}_{31}, \mathbf{R}_{32} \leftarrow \{-1, 1\}^{m\times m}$.

4 Choose a random bit $b \in_R \{0, 1\}$ and set

$$\begin{aligned} &c_0 = \mathbf{u}^\top(\mathbf{s} + \mathbf{s}' + \mathbf{s}'') + x + \lfloor \frac{q}{2} \rfloor \cdot b, \quad \mathbf{c}_1 = \mathbf{A}^\top_{\mathsf{ID,t}}\mathbf{s} + \begin{bmatrix} \mathbf{x} \\ \mathbf{R}_{11}^\top \mathbf{x} \\ \mathbf{R}_{12}^\top \mathbf{x} \end{bmatrix}, \\ &\mathbf{c}_2 = \mathbf{B}^\top_{\mathsf{ID,t,kw}}\mathbf{s}' + \begin{bmatrix} \mathbf{x}' \\ \mathbf{R}_{21}^\top \mathbf{x}' \\ \mathbf{R}_{22}^\top \mathbf{x}' \\ \mathbf{R}_{23}^\top \mathbf{x}' \end{bmatrix}, \quad \mathbf{c}_3 = \mathbf{C}^\top_{\mathcal{S}_{\mathsf{ID}},\mathsf{t}}\mathbf{s}'' + \begin{bmatrix} \mathbf{x}'' \\ \mathbf{R}_{31}^\top \mathbf{x}'' \\ \mathbf{R}_{32}^\top \mathbf{x}'' \end{bmatrix}. \end{aligned} \tag{6}$$

5 Output the ciphertext $(\mathsf{ct}_{\mathsf{ID,t,kw}}, b)$, where $\mathsf{ct}_{\mathsf{ID,t,kw}} = (c_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) \in \mathbb{Z}_q \times \mathbb{Z}_q^{3m} \times \mathbb{Z}_q^{4m} \times \mathbb{Z}_q^{3m}$.

$\mathsf{QueKG}(\mathsf{Priv_{ID}}, \mathcal{S}_{\mathsf{ID}}, \mathsf{t}, \mathsf{kw}') \to \mathbf{q}_{\mathsf{ID,t,kw}'}$: On input the private key of recipient $\mathsf{Priv_{ID}} = \mathbf{T}_{\mathbf{B}_{\mathsf{ID}}}$, the server's identity $\mathcal{S}_{\mathsf{ID}}$, a time period t and the target keyword kw', the algorithm does the follows:

```
1  Sample $\mathbf{g}_{\mathsf{ID,t,kw}} \leftarrow \mathsf{SampleLeft}(\mathbf{B}_{\mathsf{ID}}, \mathbf{B}_2 + \mathsf{H}(\mathsf{t})\mathbf{G}, \mathbf{B}_3 + \mathsf{H}(\mathsf{kw}')\mathbf{G}, \mathbf{T}_{\mathbf{B}_{\mathsf{ID}}}, \mathbf{u}, s')$.
   Note that $\mathbf{g}_{\mathsf{ID,t,kw}'} \in \mathbb{Z}^{4m}$ and
   $[\mathbf{B} \mid \mathbf{B}_1 + \mathsf{H}(\mathsf{ID})\mathbf{G} \mid \mathbf{B}_2 + \mathsf{H}(\mathsf{t})\mathbf{G} \mid \mathbf{B}_3 + \mathsf{H}(\mathsf{kw}')\mathbf{G}] \cdot \mathbf{g}_{\mathsf{ID,t,kw}'} = \mathbf{u}$. We set
   $\mathbf{g}_{\mathsf{ID,t,kw}'} = [\mathbf{g}_B \parallel \mathbf{g}_{\mathsf{ID}} \parallel \mathbf{g}_{\mathsf{t}} \parallel \mathbf{g}_{\mathsf{kw}'}]$.
2  Compute $\mathbf{g} = [\mathbf{g}_B \parallel \mathbf{g}_{\mathsf{ID}}]$.
3  Randomly choose $\theta \leftarrow \mathbb{Z}_q^n$ and compute
   $D = [\mathbf{C} \mid \mathbf{C}_1 + \mathsf{H}(S_{\mathsf{ID}})\mathbf{G}]^\top \theta + [\mathbf{g}_{\mathsf{t}} \parallel \mathbf{g}_{\mathsf{kw}'}]$
4  Output the search query $\mathbf{q}_{\mathsf{ID,t,kw}'} = (\mathbf{g}, D)$.
```

$\mathsf{Test}(\mathsf{ct}_{\mathsf{ID,t,kw}}, b, \mathsf{t}, \mathbf{e}_{\mathsf{ID,t}}, \mathbf{q}_{\mathsf{ID,t,kw}'}, \mathsf{Serv}_{\mathsf{ID}}) \rightarrow \{0,1\}$: On input a ciphertext $(\mathsf{ct}_{\mathsf{ID,t,kw}}, b)$, a time period t, and the short-term transformation key $\mathbf{e}_{\mathsf{ID,t}}$, recipient's query $\mathbf{q}_{\mathsf{ID,t,kw}'}$ and its own secret key $\mathsf{Serv}_{\mathsf{ID}}$ the server works as follows:

```
1  Compute $\mathbf{T}_{\mathbf{C}_{S_{\mathsf{ID}}}}^\top \cdot D \mod q = 0 + \mathbf{T}_{\mathbf{C}_{S_{\mathsf{ID}}}}^\top [\mathbf{g}_{\mathsf{t}} \parallel \mathbf{g}_{\mathsf{kw}'}] \mod q = \mathbf{T}_{\mathbf{C}_{S_{\mathsf{ID}}}}^\top [\mathbf{g}_{\mathsf{t}} \parallel \mathbf{g}_{\mathsf{kw}'}]$.
   Since both $\mathbf{T}_{\mathbf{C}_{S_{\mathsf{ID}}}}$ and $[\mathbf{g}_{\mathsf{t}} \parallel \mathbf{g}_{\mathsf{kw}'}]$ is sufficiently small, therefore the above equation holds
   for an overwhelming probability.
2  Recover $[\mathbf{g}_{\mathsf{t}} \parallel \mathbf{g}_{\mathsf{kw}'}]$ as $[\mathbf{g}_{\mathsf{t}} \parallel \mathbf{g}_{\mathsf{kw}'}] = (\mathbf{T}_{\mathbf{C}_{S_{\mathsf{ID}}}}^\top)^{-1} \cdot \mathbf{T}_{\mathbf{C}_{S_{\mathsf{ID}}}}^\top \cdot D$ and
   $\mathbf{g}_{\mathsf{ID,t,kw}'} = [\mathbf{g} \parallel \mathbf{g}_{\mathsf{t}} \parallel \mathbf{g}_{\mathsf{kw}'}] = [\mathbf{g}_B \parallel \mathbf{g}_{\mathsf{ID}} \parallel \mathbf{g}_{\mathsf{t}} \parallel \mathbf{g}_{\mathsf{kw}'}]$.
3  Sample $\mathbf{s}_{\mathsf{ID,t}} \leftarrow \mathsf{SampleLeft}(\mathbf{C}_{S_{\mathsf{ID}}}, \mathbf{C}_2 + \mathsf{H}(\mathsf{t})\mathbf{G}, \mathbf{T}_{\mathbf{C}_{S_{\mathsf{ID}}}}, \mathbf{u}, s_1)$. Note that
   $\mathbf{s}_{\mathsf{ID,t}} \in \mathbb{Z}^{3m}$ and $[\mathbf{C} \mid \mathbf{C}_1 + \mathsf{H}(S_{\mathsf{ID}})\mathbf{G} \mid \mathbf{C}_2 + \mathsf{H}(\mathsf{t})\mathbf{G}] \cdot \mathbf{s}_{\mathsf{ID,t}} = \mathbf{u}$.
4  Compute $c_0' = c_0 - \mathbf{e}_{\mathsf{ID,t}}^\top \mathbf{c}_1 - \mathbf{g}_{\mathsf{ID,t,kw}'}^\top \mathbf{c}_2$.
5  Output the partially decrypted ciphertext $\mathsf{ct}_{\mathsf{ID,t}}' = (c_0', \mathbf{c}_3) \in \mathbb{Z}_q \times \mathbb{Z}_q^{3m}$.
6  Compute $c = c_0' - \mathbf{s}_{\mathsf{ID,t}}^\top \mathbf{c}_3 \in \mathbb{Z}_q$.
7  Compare $c$ and $\lfloor \frac{q}{2} \rfloor$ by treating them as integers in $\mathbb{Z}$, output $b' = 1$ in case
   $|c - \lfloor \frac{q}{2} \rfloor| < \lfloor \frac{q}{4} \rfloor$ and $b' = 0$ otherwise.
8  If $b' = b$, the algorithm outputs 1 and 0 otherwise.
```

2.5 Correctness

Lemma 8 *Assume* $O\Big(\big(\alpha + \alpha s_1 m \cdot \omega(\sqrt{\log m})\big) \cdot q\Big) \le q/5$ *hold with overwhelming probability, then the above SR-IBKS scheme has negligible test error, if* $\mathsf{kw} = \mathsf{kw}'$.

Proof Since ID is non-revoked at time period t, there exists one node $\theta \in \mathsf{path}(\mathsf{BT}_{\mathsf{kgc}}, \eta_{\mathsf{ID}}) \cap \mathsf{KUNodes}(\mathsf{BT}_{\mathsf{kgc}}, \mathsf{RL}_{\mathsf{t}})$. If $\mathsf{kw} = \mathsf{kw}'$, we have $(\mathbf{e}_{\mathsf{ID,t}}, \mathbf{g}_{\mathsf{ID,t,kw}})$ such that

$$\begin{aligned} &[\mathbf{A} \mid \mathbf{A}_1 + \mathsf{H}(\mathsf{ID})\mathbf{G} \mid \mathbf{A}_2 + \mathsf{H}(\mathsf{t})\mathbf{G}] \cdot \mathbf{e}_{\mathsf{ID,t}} = \mathbf{u}, \\ &[\mathbf{B} \mid \mathbf{B}_1 + \mathsf{H}(\mathsf{ID})\mathbf{G} \mid \mathbf{B}_2 + \mathsf{H}(\mathsf{t})\mathbf{G} \mid \mathbf{B}_3 + \mathsf{H}(\mathsf{kw})\mathbf{G}] \cdot \mathbf{g}_{\mathsf{ID,t,kw}} = \mathbf{u}. \end{aligned} \tag{7}$$

where $\mathbf{e}_{\mathsf{ID,t}} = [\mathbf{e}_{\mathsf{ID},\theta}^{\mathsf{L}} + \mathbf{e}_{\mathsf{t},\theta}^{\mathsf{L}} \mid \mathbf{e}_{\mathsf{ID},\theta}^{\mathsf{R}} \mid \mathbf{e}_{\mathsf{t},\theta}^{\mathsf{R}}]$. During the Test algorithm performed by the server, we have

$$\begin{aligned}
c_0' =& c_0 - \mathbf{e}_{\mathsf{ID,t}}^\top \mathbf{c}_1 - \mathbf{g}_{\mathsf{ID,t,kw}}^\top \mathbf{c}_2 \\
=& \mathbf{u}^\top(\mathbf{s}+\mathbf{s}'+\mathbf{s}'') + x + \lfloor \frac{q}{2} \rfloor \cdot b - \left(\mathbf{u}^\top \mathbf{s} + \mathbf{e}_{\mathsf{ID,t}}^\top \begin{bmatrix} \mathbf{x} \\ \mathbf{R}_{11}^\top \mathbf{x} \\ \mathbf{R}_{12}^\top \mathbf{x} \end{bmatrix} \right) \\
& - \left(\mathbf{u}^\top \mathbf{s}' + \mathbf{g}_{\mathsf{ID,t,kw}}^\top \begin{bmatrix} \mathbf{x} \\ \mathbf{R}_{21}^\top \mathbf{x}' \\ \mathbf{R}_{22}^\top \mathbf{x}' \\ \mathbf{R}_{23}^\top \mathbf{x}' \end{bmatrix} \right) \\
=& \mathbf{u}^\top \mathbf{s}'' + x + \lfloor \frac{q}{2} \rfloor \cdot b - \mathbf{e}_{\mathsf{ID,t}}^\top \begin{bmatrix} \mathbf{x} \\ \mathbf{R}_{11}^\top \mathbf{x} \\ \mathbf{R}_{12}^\top \mathbf{x} \end{bmatrix} - \mathbf{g}_{\mathsf{ID,t,kw}}^\top \begin{bmatrix} \mathbf{x} \\ \mathbf{R}_{21}^\top \mathbf{x}' \\ \mathbf{R}_{22}^\top \mathbf{x}' \\ \mathbf{R}_{23}^\top \mathbf{x}' \end{bmatrix}.
\end{aligned} \tag{8}$$

The server's decryption key is of the form $\mathsf{dk}_{\mathsf{ID,t}} = \mathbf{s}_{\mathsf{ID,t}}$ such that

$$[\mathbf{C} \mid \mathbf{C}_1 + \mathsf{H}(S_{\mathsf{ID}})\mathbf{G} \mid \mathbf{C}_2 + \mathsf{H}(\mathsf{t})\mathbf{G}] \cdot \mathbf{s}_{\mathsf{ID,t}} = \mathbf{u}. \tag{9}$$

During the Test algorithm performed by the server, we have

$$\begin{aligned}
c =& c_0' - \mathbf{s}_{\mathsf{ID,t}}^\top \mathbf{c}_3 \\
=& \lfloor \frac{q}{2} \rfloor \cdot b + \underbrace{x - \mathbf{e}_{\mathsf{ID,t}}^\top \begin{bmatrix} \mathbf{x} \\ \mathbf{R}_{11}^\top \mathbf{x} \\ \mathbf{R}_{12}^\top \mathbf{x} \end{bmatrix} - \mathbf{g}_{\mathsf{ID,t,kw}}^\top \begin{bmatrix} \mathbf{x} \\ \mathbf{R}_{21}^\top \mathbf{x}' \\ \mathbf{R}_{22}^\top \mathbf{x}' \\ \mathbf{R}_{23}^\top \mathbf{x}' \end{bmatrix} - \mathbf{s}_{\mathsf{ID,t}}^\top \begin{bmatrix} \mathbf{x} \\ \mathbf{R}_{31}^\top \mathbf{x}'' \\ \mathbf{R}_{32}^\top \mathbf{x}'' \end{bmatrix}}_{:=z(\text{"noise"})}.
\end{aligned}$$

If we set the parameters appropriately, by Lemmas 1 and 5 , we know that

$$\|\mathbf{e}_{\mathsf{ID,t}}\| \le 2 \cdot \sqrt{3m} \cdot s_1, \quad \|\mathbf{g}_{\mathsf{ID,t,kw}}\| \le \sqrt{4m} \cdot s_1, \quad \|\mathbf{s}_{\mathsf{ID,t}}\| \le \sqrt{3m} \cdot s_1. \tag{10}$$

By Lemmas 1, 5 and 6 , we have

$$\begin{aligned}
&\|[\mathbf{I} \mid \mathbf{R}_{11} \mid \mathbf{R}_{12}] \cdot \mathbf{e}_{\mathsf{ID,t}}\| \le (1 + 2 \cdot \sqrt{2m}) \cdot \sqrt{3m} \cdot s_1 \le O(s_1 m), \\
&\|[\mathbf{I} \mid \mathbf{R}_{21} \mid \mathbf{R}_{22} \mid \mathbf{R}_{23}] \cdot \mathbf{g}_{\mathsf{ID,t,kw}}\| \le O(s_1 m), \quad \|[\mathbf{I} \mid \mathbf{R}_{31} \mid \mathbf{R}_{32}] \cdot \mathbf{s}_{\mathsf{ID,t}}\| \le O(s_1 m).
\end{aligned} \tag{11}$$

By Lemma 1 and the standard tail bound in [23], we have

$$\begin{aligned}
\|([\mathbf{I} \mid \mathbf{R}_{11} \mid \mathbf{R}_{12}] \cdot \mathbf{e}_{\mathsf{ID,t}})^\top \cdot \mathbf{x}\| &\le \|[\mathbf{I} \mid \mathbf{R}_{11} \mid \mathbf{R}_{12}] \cdot \mathbf{e}_{\mathsf{ID,t}}\| \cdot \alpha q \cdot \omega(\sqrt{\log m}), \\
&\le O(s_1 m) \cdot \alpha q \cdot \omega(\sqrt{\log m}).
\end{aligned} \tag{12}$$

S $\|([\mathbf{I} \mid \mathbf{R}_{21} \mid \mathbf{R}_{22} \mid \mathbf{R}_{23}] \cdot \mathbf{g}_{\mathsf{ID,t,kw}})^\top \cdot \mathbf{x}\|$, $\|([\mathbf{I} \mid \mathbf{R}_{31} \mid \mathbf{R}_{32}] \cdot \mathbf{s}_{\mathsf{ID,t}})^\top \cdot \mathbf{x}\| \le O(s_1 m) \cdot \alpha q \cdot \omega(\sqrt{\log m})$,
.

Thus, the noise z can be bounded with overwhelming probability as follows:

$$\begin{aligned}\|z\| &\leq |x| + \|\mathbf{e}_{\mathsf{ID,t}}^{\top}\begin{bmatrix}\mathbf{x}\\ \mathbf{R}_{11}^{\top}\mathbf{x}\\ \mathbf{R}_{12}^{\top}\mathbf{x}\end{bmatrix}\| + \|\mathbf{g}_{\mathsf{ID,t,kw}}^{\top}\begin{bmatrix}\mathbf{x}\\ \mathbf{R}_{21}^{\top}\mathbf{x}\\ \mathbf{R}_{22}^{\top}\mathbf{x}\\ \mathbf{R}_{23}^{\top}\mathbf{x}\end{bmatrix}\| + \|\mathbf{s}_{\mathsf{ID,t}}^{\top}\begin{bmatrix}\mathbf{x}\\ \mathbf{R}_{31}^{\top}\mathbf{x}\\ \mathbf{R}_{32}^{\top}\mathbf{x}\end{bmatrix}\|,\\ &\leq \alpha q + O(s_1 m)\cdot \alpha q\cdot \omega(\sqrt{\log m}),\\ &= O\Big(\Big(\alpha + \alpha s_1 m\cdot\omega(\sqrt{\log m})\Big)\cdot q\Big).\end{aligned} \tag{13}$$

Since $O\Big(\Big(\alpha + \alpha s_1 m\cdot\omega(\sqrt{\log m})\Big)\cdot q\Big) \leq q/5$ holds with overwhelming probability, so $\|z\| \leq q/5$ holds with overwhelming probability. If $b = 1$, then $|c - \lfloor\frac{q}{2}\rfloor| = \|z\| < \lfloor\frac{q}{4}\rfloor$ with overwhelming probability; If $b = 0$, then $|c - \lfloor\frac{q}{2}\rfloor| = |z - \lfloor\frac{q}{2}\rfloor| > \lfloor\frac{q}{4}\rfloor$ with overwhelming probability. So we prove this lemma. □

2.6 Parameter selection and efficiency

In this section, we provide an example of parameter selection of our dSR-IBKS scheme. Note that we need to ensure that

- the "noise" term in the above section is less than $q/5$ with overwhelming probability (i.e., $O\Big(\Big(\alpha + \alpha s_1 m\cdot\omega(\sqrt{\log m})\Big)\cdot q\Big) \leq q/5$ by Lemma 8).
- algorithm TrapGen operated as specified (i.e., $m \geq 2n\lceil\log q\rceil$ by Lemma 3).
- algorithms SampleLeft and ExtRndLeft works as specified (i.e., $s_0 \geq O(\sqrt{n\log q})\cdot\omega(\sqrt{\log m}), s_1 \geq s_0\sqrt{m}\cdot\omega(\sqrt{\log m})$ by Lemmas 1, 3, 4 and 5).
- algorithm ExtRndRight works as specified (i.e., $s > \sqrt{m}\cdot\omega(\sqrt{\log m})$).
- the hardness assumption of LWE applies (i.e., $\alpha q > 2\sqrt{n}$).

According to the above restrictions, we can set the parameters of our SR-IBKS as follows:

$$\begin{aligned}&n = O(\lambda),\quad N = \mathsf{poly}(\lambda),\quad m = O(n\log q),\quad s_0 = \sqrt{m}\cdot\omega(\sqrt{\log n}), s_1 = m\cdot\omega(\log n),\\ &\alpha = m^{-2}\cdot\omega((\log n)^{\frac{3}{2}})^{-1},\quad q = n^{\frac{1}{2}}\cdot m^2\cdot\omega((\log n)^{\frac{3}{2}}).\end{aligned} \tag{14}$$

2.7 Security analysis

Theorem 1 *The SR-IBKS scheme is dSR-sID-CPA secure assuming the hardness of the* $\mathsf{LWE}_{n,m+1,q,\chi}$ *problem, where* $\chi = \mathcal{D}_{\mathbb{Z}^{m+1},\alpha q}$.

Proof Let $\mathcal{A}$ be a PPT adversary who succeeds in breaking the dSR-sID-CPA security of our SR-IBKS scheme with advantage $\mathsf{Adv}^{\mathsf{dSR-sID-CPA}}_{\mathcal{A},\mathcal{O}^{\mathsf{SR\text{-}IBks}}}(\lambda) = \epsilon$, let ID^* and S^*_{ID} be the challenge identities, t^* be the challenge time period, and kw be the two challenge keywords. Observe that the strategy taken by $\mathcal{A}$ can be further divided into three types of strategies that are mutually exclusive as follows.

Type-**I**: $\mathcal{A}$ issues private key reveal query on both ID^* and S^*_{ID}, and the long-term transformation key reveal query on ID^*. In this case, the challenge identity ID^* must be revoked before the challenge time t^*.

Type-**II**: $\mathcal{A}$ does not issue private key reveal query on the challenge identity ID^*, and $\mathcal{A}$ can issue any keyword search query on the tuple $(\mathsf{id}, \mathsf{t}, \mathsf{kw})$ with the restriction that $(\mathsf{id}, \mathsf{t}, \mathsf{kw}) \neq (\mathsf{id}^*, \mathsf{t}^*, \mathsf{kw}^*_0)$ and $(\mathsf{id}, \mathsf{t}, \mathsf{kw}) \neq (\mathsf{id}^*, \mathsf{t}^*, \mathsf{kw}^*_1)$.

Type-**III**: $\mathcal{A}$ does not issue private key reveal query on the S^*_{ID}.

As $\mathcal{A}$ always follows one of the above strategies, we only need to show that the advantage of $\mathcal{A}$ is negligible regardless of the strategy taken by $\mathcal{A}$. We proceed with a sequence of games where the first game is identical to the dSR-sID-CPA game from Definition 2 and the adversary has no advantage in the last game. We will show that $\mathcal{A}$ cannot distinguish between the games, which will prove that the adversary has negligible advantage in winning the original dSR-sID-CPA game and will complete our proof. □

Lemma 9 *The advantage of an adversary $\mathcal{A}_1$ using the Type-**I** strategy is negligible assuming the hardness of the* $\mathsf{LWE}_{n,m+1,q,\chi}$, where $\chi = \mathcal{D}_{\mathbb{Z}^{m+1},\alpha q}$.

Proof We define the sequence of games as follows:

$\mathsf{Game}_{\mathbf{I}-0}$: This is the original dSR-sID-CPA game from Definition 2.

$\mathsf{Game}_{\mathbf{I}-1}$: In this game, we change the way that the challenger generates $\mathbf{A}_1, \mathbf{A}_2$ in the public parameters. The $\mathsf{Game}_{\mathbf{I}-1}$ challenger samples $\mathbf{R}^*_{11}, \mathbf{R}^*_{12} \leftarrow \{-1, 1\}^{m \times m}$ at the setup phase and sets $\mathbf{A}_1, \mathbf{A}_2$ as

$$\mathbf{A}_1 = \mathbf{A}\mathbf{R}^*_{11} - \mathsf{H}(\mathsf{ID}^*)\mathbf{G}, \quad \mathbf{A}_2 = \mathbf{A}\mathbf{R}^*_{12} - \mathsf{H}(\mathsf{t}^*)\mathbf{G}. \tag{15}$$

The challenger keeps the matrices $\mathbf{R}^*_{11}, \mathbf{R}^*_{12}$ as a part of $\mathsf{sk}_{\mathsf{kgc}}$ and the remainder of the game is unchanged.

We now show that $\mathsf{Game}_{\mathbf{I}-0}$ is statistically indistinguishable from $\mathsf{Game}_{\mathbf{I}-1}$. Note that in $\mathsf{Game}_{\mathbf{I}-1}$ the matrices $\mathbf{R}^*_{11}, \mathbf{R}^*_{12}$ are used only in the construction of $\mathbf{A}_1, \mathbf{A}_2$ and in the construction of the challenge ciphertext where $\mathbf{z}_1 \leftarrow (\mathbf{R}^*_{11})^\top \mathbf{x}$, $\mathbf{z}_2 \leftarrow (\mathbf{R}^*_{12})^\top \mathbf{x}$. By Lemma 7, the distribution $(\mathbf{A}, \mathbf{A}\mathbf{R}^*_{11}, \mathbf{z}_1)$ and $(\mathbf{A}, \mathbf{A}\mathbf{R}^*_{12}, \mathbf{z}_2)$ are statistically close to the distribution $(\mathbf{A}, \mathbf{A}'_1, \mathbf{z}_1)$ and $(\mathbf{A}, \mathbf{A}'_2, \mathbf{z}_2)$ respectively, where $\mathbf{A}'_1, \mathbf{A}'_2$ are uniform $\mathbb{Z}_q^{n \times m}$ matrices. Hence, $\mathbf{A}_1, \mathbf{A}_2$ in $\mathsf{Game}_{\mathbf{I}-0}$ and $\mathsf{Game}_{\mathbf{I}-1}$ are indistinguishable.

$\mathsf{Game}_{\mathbf{I}-2}$: In this game, we change how we assign ID^* to the binary tree $\mathsf{BT}_{\mathsf{kgc}}$. Recall in the previous game, the challenger assigned ID^* to some random leaf η_{ID^*} of $\mathsf{BT}_{\mathsf{kgc}}$ when $\mathcal{A}_1$ issued a long-term transformation key query on ID^*. In this game, the $\mathsf{Game}_{\mathbf{I}-2}$ challenger chooses a random leaf in $\mathsf{BT}_{\mathsf{kgc}}$ to assign ID^* before providing $\mathcal{A}_1$ the public parameter pp. When $\mathcal{A}_1$ submitted a long-term transformation key query on some ID, if $\mathsf{ID} = \mathsf{ID}^*$, then the challenger proceeds with $\mathsf{L-TranKG}$ as if $(\mathsf{BT}_{\mathsf{kgc}}, \eta_{\mathsf{ID}^*}) \leftarrow \mathsf{CS.Assign}(\mathsf{BT}_{\mathsf{kgc}}, \mathsf{ID}^*)$ and otherwise it assigns ID to some random leaf of $\mathsf{BT}_{\mathsf{kgc}}$ that is not η_{ID^*}. Since the random assignment of ID^* made by the challenger is statistically hidden from $\mathcal{A}_1$, $\mathsf{Game}_{\mathbf{I}-1}$ and $\mathsf{Game}_{\mathbf{I}-2}$ are indistinguishable.

$\mathsf{Game}_{\mathbf{I}-3}$: In this game, we change the challenger so he does not have to use the trapdoor $\mathbf{T}_{\mathbf{A}}$ when generating short vectors as follows: $(\mathbf{e}_{\mathsf{ID},\theta})_{\theta\in\mathsf{path}(\mathsf{BT}_{\mathsf{kgc}},\eta_{\mathsf{ID}})}$ in $\mathsf{sk}_{\mathsf{ID}}$, $(\mathbf{e}_{\mathsf{t},\theta})_{\theta\in\mathsf{KUNodes}(\mathsf{BT}_{\mathsf{kgc}},\mathsf{RL}_{\mathsf{t}})}$ in uk_{t}, and $\mathsf{e}_{\mathsf{ID},\mathsf{t}}$. To achieve this goal, we modify when and how the vectors $\mathbf{u}_{\mathsf{kgc},\theta}$ for each node $\theta\in\mathsf{BT}_{\mathsf{kgc}}$ is chosen. By the definition of the Type-**I** strategy, ID^* must be revoked before time period t^*. Hence, we have $\mathsf{path}(\mathsf{BT}_{\mathsf{kgc}},\eta_{\mathsf{ID}^*})\cap\mathsf{KUNodes}(\mathsf{BT}_{\mathsf{kgc}},\mathsf{RL}_{\mathsf{t}^*})=\emptyset$ by the property of the CS scheme.

Whenever $\mathcal{A}_1$ issues a long-term transformation key query, a revoke & update key query, or a short-term transformation key reveal query, the $\mathsf{Game}_{\mathbf{I}-3}$ challenger generates the vectors $\mathbf{u}_{\mathsf{kgc},\theta}$ for each node $\theta\in\mathsf{BT}_{\mathsf{kgc}}$ as follows:

- If $\theta\in\mathsf{path}(\mathsf{BT}_{\mathsf{kgc}},\eta_{\mathsf{ID}^*})$, then it samples $\mathbf{e}_{\mathsf{ID}^*,\theta}\leftarrow\mathcal{D}_{\mathbb{Z}^{2m},s}$, sets $\mathbf{A}_{\mathsf{ID}^*}\cdot\mathbf{e}_{\mathsf{ID}^*,\theta}=\mathbf{u}_{\mathsf{kgc},\theta}$, stores $\mathbf{u}_{\mathsf{kgc},\theta}$ in the node θ and keeps $\mathbf{e}_{\mathsf{ID}^*,\theta}$ secret.
- If $\theta\notin\mathsf{path}(\mathsf{BT}_{\mathsf{kgc}},\eta_{\mathsf{ID}^*})$, then it samples $\mathbf{e}_{\mathsf{t}^*,\theta}\leftarrow\mathcal{D}_{\mathbb{Z}^{2m},s}$, sets $\mathbf{A}_{\mathsf{t}^*}\cdot\mathbf{e}_{\mathsf{t}^*,\theta}=\mathbf{u}_{\mathsf{kgc},\theta}$ by implicitly setting $\mathsf{t}_{\mathsf{cu}}=\mathsf{t}^*$, stores $\mathbf{u}_{\mathsf{kgc},\theta}$ in the node θ and keeps $\mathbf{e}_{\mathsf{t}^*,\theta}$ secret.

Then, if $\mathcal{A}_1$ issued a long-term transformation key query on $\mathsf{ID}\neq\mathsf{ID}^*$, the $\mathsf{Game}_{\mathbf{I}-3}$ challenger runs $\mathsf{ExtRndRight}(\mathbf{A},\mathbf{G},\mathbf{R}^*_{11},\mathbf{T}_{\mathbf{G}},s_0)$ to create the trapdoor $\mathbf{T}_{\mathbf{A}_{\mathsf{ID}}=[\mathbf{A}|\mathbf{A}_1+\mathsf{H}(\mathsf{ID})\mathbf{G}]=[\mathbf{A}|\mathbf{AR}^*_{11}+(\mathsf{H}(\mathsf{ID})-\mathsf{H}(\mathsf{ID}^*))\mathbf{G}]}$ and samples the short vectors $(\mathbf{e}_{\mathsf{ID},\theta})_\theta$ by running $\mathsf{SamplePre}(\cdot)$ with trapdoor $\mathbf{T}_{[\mathbf{A}|\mathbf{AR}^*_{11}+(\mathsf{H}(\mathsf{ID})-\mathsf{H}(\mathsf{ID}^*))\mathbf{G}]}$ and Gaussian parameter s_1. Otherwise, if $\mathsf{ID}=\mathsf{ID}^*$, the challenger simply returns $(\mathbf{e}_{\mathsf{ID}^*,\theta})_\theta$ which he has already generated without using $\mathbf{T}_{\mathbf{A}}$. Moreover, if the counter t_{cu} on which $\mathcal{A}_1$ queried the revoke & key update query is not t^*, then the $\mathsf{Game}_{\mathbf{I}-3}$ challenger runs $\mathsf{ExtRndRight}(\mathbf{A},\mathbf{G},\mathbf{R}^*_{12},\mathbf{T}_{\mathbf{G}},s_0)$ to create $\mathbf{T}_{[\mathbf{A}|\mathbf{AR}^*_{12}+(\mathsf{H}(\mathsf{t})-\mathsf{H}(\mathsf{t}^*))\mathbf{G}]}$ and samples the short vectors $(\mathbf{e}_{\mathsf{t},\theta})_{\theta\in\mathsf{KUNodes}(\mathsf{BT}_{\mathsf{kgc}},\mathsf{RL}_{\mathsf{t}})}$ by running $\mathsf{SamplePre}(\cdot)$ with trapdoor $\mathbf{T}_{[\mathbf{A}|\mathbf{AR}^*_{12}+(\mathsf{H}(\mathsf{t})-\mathsf{H}(\mathsf{t}^*))\mathbf{G}]}$ and Gaussian parameter s_1. Otherwise, if $\mathsf{t}_{\mathsf{cu}}=\mathsf{t}^*$, the challenger simply returns $(\mathbf{e}_{\mathsf{t}^*,\theta})_\theta$ which he has already created without using $\mathbf{T}_{\mathbf{A}}$. Furthermore, if $\mathcal{A}_1$ issued a short-term transformation key reveal query on $(\mathsf{ID},\mathsf{t})\neq(\mathsf{ID}^*,\mathsf{t}^*)$, the challenger simply creates $\mathbf{e}_{\mathsf{ID},\mathsf{t}}$ from combing $(\mathbf{e}_{\mathsf{ID},\theta})_\theta$ and $(\mathbf{e}_{\mathsf{t},\theta})_\theta$. Since $\mathsf{path}(\mathsf{BT}_{\mathsf{kgc}},\eta_{\mathsf{ID}^*})\cap\mathsf{KUNodes}(\mathsf{BT}_{\mathsf{kgc}},\mathsf{RL}_{\mathsf{t}^*})=\emptyset$, the procedure described above is well-defined. Finally, according to Lemmas 2, 4 and 5 , the distribution of the short vectors given to $\mathcal{A}_1$ are distributed statistically close to those of the previous game. Therefore, $\mathsf{Game}_{\mathbf{I}-2}$ and $\mathsf{Game}_{\mathbf{I}-3}$ are indistinguishable.

$\mathsf{Game}_{\mathbf{I}-4}$: In this game we change how $\mathbf{A}$ is sampled. We generate $\mathbf{A}$ as a random matrix in $\mathbb{Z}_q^{n\times m}$ instead of generating it by running $\mathsf{TrapGen}$. By Lemma 3, $\mathsf{Game}_{\mathbf{I}-3}$ and $\mathsf{Game}_{\mathbf{I}-4}$ are indistinguishable.

$\mathsf{Game}_{\mathbf{I}-5}$: In this game, we change the way the challenge ciphertext $\mathbf{ct}_{\mathsf{ID}^*,\mathsf{t}^*,\mathsf{kw}^*_b}$ is created. In this game, when the $\mathsf{Game}_{\mathbf{I}-5}$ challenger is issued a challenge query on $\{\mathsf{kw}^*_0,\mathsf{kw}^*_1\}$ by $\mathcal{A}_1$, the challenger chooses a random bit $b\leftarrow\{0,1\}$, samples $\nu\leftarrow\mathbb{Z}_q$, $\mathbf{v}\leftarrow\mathbb{Z}_q^n$, $\mathbf{s}',\mathbf{s}''\leftarrow\mathbb{Z}_q^n$, $\mathbf{x}',\mathbf{x}''\leftarrow\mathcal{D}_{\mathbb{Z}^m,\alpha q}$, $\mathbf{R}_{21},\mathbf{R}_{22},\mathbf{R}_{23},\mathbf{R}_{31},\mathbf{R}_{32}\leftarrow\{-1,1\}^{m\times m}$ and sets

Table 1 Comparison with other lattice-based PEKS schemes

Scheme	Standard model	Keyword guessing resistance	User revocation	Server-aided
[8]	×	×	×	×
[9]	√	×	×	×
[11]	√	√	×	×
[12]	×	×	×	×
[10]	×	√	×	×
Ours	√	√	√	√

$$c_0 = v + \mathbf{u}^\top(\mathbf{s}' + \mathbf{s}'') + \lfloor \frac{q}{2} \rfloor \cdot b, \quad \mathbf{c}_1 = \begin{bmatrix} \mathbf{v} \\ (\mathbf{R}^*_{11})^\top \mathbf{v} \\ (\mathbf{R}^*_{12})^\top \mathbf{v} \end{bmatrix},$$
$$\mathbf{c}_2 = \mathbf{B}^\top_{\mathsf{ID}^*,\mathsf{t}^*,\mathsf{kw}^*_b}\mathbf{s}' + \begin{bmatrix} \mathbf{x}' \\ \mathbf{R}^\top_{21}\mathbf{x}' \\ \mathbf{R}^\top_{22}\mathbf{x}' \\ \mathbf{R}^\top_{23}\mathbf{x}' \end{bmatrix}, \quad \mathbf{c}_3 = \mathbf{C}^\top_{S^*_{\mathsf{id}},\mathsf{t}^*}\mathbf{s}'' + \begin{bmatrix} \mathbf{x}'' \\ \mathbf{R}^\top_{31}\mathbf{x}'' \\ \mathbf{R}^\top_{32}\mathbf{x}'' \end{bmatrix}. \tag{16}$$

Finally, the challenger outputs the challenge ciphertext as $\mathbf{ct}_{\mathsf{ID}^*,\mathsf{t}^*,\mathsf{kw}^*_b} = (c_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$. Since v is distributed uniformly at random over $\mathbb{Z}_q$ and independently of all other terms, the probability of $\mathcal{A}_1$ guessing whether $b = 0$ or $b = 1$ is exactly 1/2. In the following, we only need to show that $\mathsf{Game}_{\mathbf{I}-4}$ and $\mathsf{Game}_{\mathbf{I}-5}$ are indistinguishable assuming the hardness of the $\mathsf{LWE}_{n,m+1,q,\chi}$ to complete the proof. To this end, we use $\mathcal{A}_1$ to construct a LWE adversary $\mathcal{B}_1$ as follows:

$\mathcal{B}_1$ is given the problem instance of LWE as $(\bar{\mathbf{A}}, \bar{\mathbf{v}}) \in \mathbb{Z}_q^{n\times(m+1)} \times \mathbb{Z}_q^{(m+1)}$ and aims to distinguish whether $\bar{\mathbf{v}} = \bar{\mathbf{A}}^\top \mathbf{s} + \bar{\mathbf{x}}$ for some $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\bar{\mathbf{x}} \leftarrow \mathcal{D}_{\mathbb{Z}^{m+1},\alpha q}$ or $\bar{\mathbf{v}} \leftarrow \mathbb{Z}_q^{(m+1)}$. Let the first column of $\bar{\mathbf{A}}$ be $\mathbf{u}^* \in \mathbb{Z}_q^n$ and the remaining columns be $\mathbf{A}^* \in \mathbb{Z}_q^{n\times m}$. Let the first element of $\bar{\mathbf{v}}$ be v and the remaining elements be $\mathbf{v}$. Now, $\mathcal{B}_1$ sets $(\mathbf{A}, \mathbf{u}) = (\mathbf{A}^*, \mathbf{u}^*)$ and proceeds the setup as the $\mathsf{Game}_{\mathbf{I}-3}$ challenger. Furthermore, whenever $\mathcal{A}_1$ issues a query, $\mathcal{B}_1$ works as the $\mathsf{Game}_{\mathbf{I}-3}$ challenger and answers them without $\mathbf{T}_\mathbf{A}$. To generate the challenge ciphertext, $\mathcal{B}_1$ chooses $b \leftarrow \{0, 1\}$ and generate the challenge ciphertext as in Eq. (16) using v, $\mathbf{v}$, and returns it to $\mathcal{A}_1$. Let b' denote the output of $\mathcal{A}_1$, then $\mathcal{B}_1$ outputs 1 if $b' = b$ and 0 otherwise. Note that if $(\bar{\mathbf{A}}, \bar{\mathbf{v}})$ is a valid LWE sample, i.e., $\bar{\mathbf{v}} = \bar{\mathbf{A}}^\top \mathbf{s} + \bar{\mathbf{x}}$ for some $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, then the view of $\mathcal{A}_1$ is the same as that of $\mathsf{Game}_{\mathbf{I}-4}$. Otherwise, i.e., $\bar{\mathbf{v}} \leftarrow \mathbb{Z}_q^{(m+1)}$, it is the same as that of $\mathsf{Game}_{\mathbf{I}-5}$. Therefore, $\mathsf{Game}_{\mathbf{I}-4}$ and $\mathsf{Game}_{\mathbf{I}-5}$ are indistinguishable assuming the hardness of the $\mathsf{LWE}_{n,m+1,q,\chi}$, where $\chi = \mathcal{D}_{\mathbb{Z}^{m+1},\alpha q}$. □

Lemma 10 *The advantage of an adversary $\mathcal{A}_2$ using the Type-***II** *strategy is negligible assuming the hardness of the* $\mathsf{LWE}_{n,m+1,q,\chi}$, where $\chi = \mathcal{D}_{\mathbb{Z}^{m+1},\alpha q}$.

Proof The outline of this proof is essentially the same as that of Lemma 9. The difference is that in this proof, we modify the challenger so that he is able to simulate the game without $\mathbf{T}_\mathbf{B}$.

$\mathsf{Game}_{\mathbf{II}-0}$: This is the original dSR-sID-CPA game from Definition 2.

$\mathsf{Game}_{\mathbf{II}-1}$: In this game, we change the way that the challenger generates $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$ in the public parameters. The $\mathsf{Game}_{\mathbf{II}-1}$ challenger samples $\mathbf{R}_{21}^*, \mathbf{R}_{22}^*, \mathbf{R}_{23}^* \leftarrow \{-1,1\}^{m\times m}$ at the setup phase and sets $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$ as

$$\mathbf{B}_1 = \mathbf{B}\mathbf{R}_{21}^* - \mathsf{H}(\mathsf{ID}^*)\mathbf{G}, \quad \mathbf{B}_2 = \mathbf{B}\mathbf{R}_{22}^* - \mathsf{H}(\mathsf{t}^*)\mathbf{G}, \quad \mathbf{B}_3 = \mathbf{B}\mathbf{R}_{23}^* - \mathsf{H}(\mathsf{kw}_b^*)\mathbf{G} \tag{17}$$

The challenger keeps the matrices $\mathbf{R}_{21}^*, \mathbf{R}_{22}^*, \mathbf{R}_{23}^*$ as a part of $\mathsf{sk}_{\mathsf{kgc}}$ and the remainder of the game is unchanged. Similar to $\mathsf{Game}_{\mathbf{II}-1}$ in Lemma 9, by Lemma 7, $\mathbf{B}_1, \mathbf{B}_2$ in $\mathsf{Game}_{\mathbf{II}-0}$ and $\mathsf{Game}_{\mathbf{II}-1}$ are indistinguishable, hence $\mathsf{Game}_{\mathbf{II}-0}$ is indistinguishable from $\mathsf{Game}_{\mathbf{II}-1}$.

$\mathsf{Game}_{\mathbf{II}-2}$: In this game, we change the challenger so he does not have to use the trapdoor $\mathbf{T}_\mathbf{B}$ when generating $\mathbf{T}_{\mathbf{B}_{\mathsf{ID}}=[\mathbf{B}|\mathbf{B}_1+\mathsf{H}(\mathsf{ID})\mathbf{G}]=[\mathbf{B}|\mathbf{B}\mathbf{R}_{21}^*+(\mathsf{H}(\mathsf{ID})-\mathsf{H}(\mathsf{ID}^*))\mathbf{G}]}$. To this end, we modify the way $\mathbf{T}_{\mathbf{B}_{\mathsf{ID}}}$ and $\mathbf{g}_{\mathsf{ID,t,kw}}$ are sampled. By the definition of the Type-**II** strategy, if $\mathsf{ID} \neq \mathsf{ID}^*$, the $\mathsf{Game}_{\mathbf{II}-2}$ challenger runs $\mathsf{ExtRndRight}(\mathbf{B}, \mathbf{G}, \mathbf{R}_{21}^*, \mathbf{T}_\mathbf{G}, s_0)$ to create $\mathbf{T}_{\mathbf{B}_{\mathsf{ID}}}$. By Lemma 4, the distribution of $\mathbf{T}_{\mathbf{B}_{\mathsf{ID}}}$ given to $\mathcal{A}_2$ is distributed statistically close to that of the previous game. Furthermore, when $(\mathsf{kw}, \mathsf{t}) \neq (\mathsf{kw}_0^*, \mathsf{t}^*)$ and $(\mathsf{kw}, \mathsf{t}) \neq (\mathsf{kw}_1^*, \mathsf{t}^*)$, then if $\mathsf{ID} \neq \mathsf{ID}^*$, the challenger samples $\mathbf{g}_{\mathsf{ID,t,kw}}$ by running $\mathsf{SampleLeft}(\cdot)$ with $\mathbf{T}_{\mathbf{B}_{\mathsf{ID}}}$ and Gaussian parameter s_1. Otherwise, $\mathsf{t} \neq \mathsf{t}^*$ or $\mathsf{kw} \neq \mathsf{kw}_b^*$, the challenger first creates the trapdoor $\mathbf{T}_{\mathbf{B}_\mathsf{t}=[\mathbf{B}|\mathbf{B}_2+\mathsf{H}(\mathsf{t})\mathbf{G}]=[\mathbf{B}|\mathbf{B}\mathbf{R}_{22}^*+(\mathsf{H}(\mathsf{t})-\mathsf{H}(\mathsf{t}^*))\mathbf{G}]}$ or $\mathbf{T}_{\mathbf{B}_{\mathsf{kw}}=[\mathbf{B}|\mathbf{B}_2+\mathsf{H}(\mathsf{kw})\mathbf{G}]=[\mathbf{B}|\mathbf{B}\mathbf{R}_{23}^*+(\mathsf{H}(\mathsf{kw})-\mathsf{H}(\mathsf{kw}_b^*))\mathbf{G}]}$, then samples $\mathbf{g}_{\mathsf{ID,t,kw}}$. Finally, according to Lemmas 2, 4 and 5 , the distribution of $\mathbf{g}_{\mathsf{ID,t,kw}}$ given to $\mathcal{A}_2$ is distributed statistically close to that of the previous game. Therefore, $\mathsf{Game}_{\mathbf{II}-1}$ and $\mathsf{Game}_{\mathbf{II}-2}$ are indistinguishable.

$\mathsf{Game}_{\mathbf{II}-3}$: In this game we change how $\mathbf{B}$ is sampled. We generate $\mathbf{B}$ as a random matrix in $\mathbb{Z}_q^{n\times m}$ instead of generating it by running $\mathsf{TrapGen}$. By Lemma 3, $\mathsf{Game}_{\mathbf{II}-2}$ and $\mathsf{Game}_{\mathbf{II}-3}$ are indistinguishable.

$\mathsf{Game}_{\mathbf{II}-4}$: This game is the same as $\mathsf{Game}_{\mathbf{I}-5}$ Therefore, $\mathsf{Game}_{\mathbf{II}-3}$ and $\mathsf{Game}_{\mathbf{II}-4}$ are indistinguishable assuming the hardness of the $\mathsf{LWE}_{n,m+1,q,\chi}$, where $\chi = \mathcal{D}_{\mathbb{Z}^{m+1},\alpha q}$. □

Lemma 11 *The advantage of an adversary $\mathcal{A}_3$ using the Type-***III** *strategy is negligible assuming the hardness of the* $\mathsf{LWE}_{n,m+1,q,\chi}$, where $\chi = \mathcal{D}_{\mathbb{Z}^{m+1},\alpha q}$.

Proof The outline of this proof is essentially the same as that of Lemmas 9 and 10. The difference is that in this proof, we modify the challenger so that he is able to simulate the game without $\mathbf{T}_\mathbf{C}$, so we omit the detail of the proof of this lemma.

According to Lemmas 9, 10 and 11 , we can conclude that the SR-IBKS scheme is dSR-sID-CPA secure, which completes the proof of Theorem 1. □

3 Result and discussion

3.1 Discussion

One weaknesses of the current lattice-based IBKS schemes is that the system does not have the ability to revoke user's authority. This property does not satisfy certain application scenarios. For example, when a user leaves his job or loses the key, his secret key must be revoked, otherwise the privacy of the encrypted data will be exposed, which means that he can still search for the target encrypted files in the system. Moreover, the keyword space is supposed to be large enough like super-polynomial, but in the real application, keywords are often chosen from a relatively small space. In this case, any outside malicious user or tester can guess the keywords containing in the ciphertext by keyword guessing attack. Lattice-based PEKS schemes like [8, 9] are all vulnerable to such kinds of attack, since given a ciphertext, the adversary can generate the search query by itself and then run the Test algorithm to guess keyword in the ciphertext.

3.2 Result

Motivated by the above observation, in this paper, we propose the first designated server-aided revocable identity-based encryption scheme with keyword search (dSR-IBKS) from lattice as shown in Fig. 1. Specifically, the dSR-IBKS model requires each user to keep just one private key and does not need to keep communicating with the key generation center in order to update his secret key when another is revoked. This property is much applicable for resource-limited end users. In addition, our scheme designates a unique tester to test and return the search results, which makes it resist the keyword guess attack of external adversary. In other words, any other entities cannot guess the keyword embedded in the ciphertext by generating search queries and doing the text by itself. We prove our scheme achieving chosen keyword security under the hardness of the learning with errors (LWE) problem.

In Table 1, we compare our scheme with other PEKS schemes from lattices. From Table 1, we can clearly observe that our scheme is the first lattice-based server-aided revocable IBKS scheme supporting keyword guessing resistance in the standard model.

4 Conclusion

In this paper, we propose the first designated server-aided revocable identity-based encryption scheme with keyword search (dSR-IBKS) from lattice. In our scheme, recipient doesn't need to keep in touch with KGC to update his secret key when some one is revoked by the KGC. Moreover, our scheme designates a unique cloud server to conduct the Test algorithm. So any other adversaries cannot launch the keyword guessing attack on the ciphertext by generating search queries and doing the test by itself.

Acknowledgements
Not applicable.

Authors' contributions
YG, FM, LC, XD, ZC are five main authors of this work. All of them contributed to the scheme design and analysis. All authors read and approved the final manuscript.

Funding
This work was supported in part by the National Key Research and Development Program of China (Grant No. 2020YFA0712300), in part by the National Natural Science Foundation of China (Grant No. 61632012), in part by the Peng Cheng Laboratory Project of Guangdong Province (Grant No. PCL2018KP004).

Declarations

Competing interests
All authors declare that no competing interests exist in this paper.

Author details
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. [2]School of Mathematics, Shandong University, South Shanda Road, No.27, Jinan 250100, China. [3]School of Mathematical Sciences, Fudan University, Handan Road, No.220, Shanghai 200433, China. [4]Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China. [5]Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen 518055, China. [6]Institute of Intelligent Science and Technology, Tongji University, Shanghai 200092, China.

Received: 28 December 2020 Accepted: 16 May 2021
Published online: 06 September 2021

References
1. D. Boneh, G.D. Crescenzo, R. Ostrovsky, G. Persiano, Public key encryption with keyword search. EUROCRYPT **2004**, 506–522 (2004)
2. D.J. Park, K. Kim, P.J. Lee, Public key encryption with conjunctive field keyword search. WISA **2004**, 73–86 (2004)
3. D. Boneh, B. Waters, Conjunctive, subset, and range queries on encrypted data. TCC **2007**, 535–554 (2007)
4. H. Li, Q. Huang, J. Shen, G. Yang, W. Susilo, Designated-server identity-based authenticated encryption with keyword search for encrypted emails. Inf. Sci. **481**, 330–343 (2019)
5. Q. Zheng, S. Xu, G. Ateniese, VABKS: verifiable attribute-based keyword search over outsourced encrypted data. INFOCOM **2014**, 522–530 (2014)
6. W. Sun, S. Yu, W. Lou, Y.T. Hou, H. Li, Protecting your right: Verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud. IEEE Trans. Parallel Distrib. Syst. **27**(4), 1187–1198 (2016)
7. F. Meng, L. Cheng, M. Wang, ABDKS: attribute-based encryption with dynamic keyword search in fog computing. Front. Comput. Sci. **15**, 155810 (2021). https://doi.org/10.1007/s11704-020-9472-7
8. C. Hou, F. Liu, H. Bai and L. Ren, "Public-Key Encryption with Keyword Search from Lattice," 2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 2013, pp. 336–339. https://doi.org/10.1109/3PGCIC.2013.57
9. B. Rouzbeh, O.M. Ozgur, Y.A. Altay, Lattice-based public key searchable encryption from experimental perspectives. IEEE Trans. Dependable Secure Comput. **PP**, 1 (2018)
10. X. Zhang, C. Xu, L. Mu, J. Zhao, Identity-based encryption with keyword search from lattice assumption. China Commun. **15**(4), 164–178 (2018)
11. Y. Mao, X. Fu, C. Guo, G. Wu, Public key encryption with conjunctive keyword search secure against keyword guessing attack from lattices. Trans. Emerg. Telecommun. Technol. **30**(11), e3531 (2019)
12. P. Wang, T. Xiang, X. Li, H. Xiang, Public key encryption with conjunctive keyword search on lattice. J. Inf. Secur. Appl. **51**, 102433 (2020)
13. J. Li, M. Ma, J. Zhang, S. Fan, S. Li, Attribute-based keyword search from lattices, in Scrypt (2019), (2019) pp. 66–85
14. D. Boneh, M.K. Franklin, Identity-based encryption from the weil pairing, in Advances in Cryptology—CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19–23, 2001, Proceedings, (2001), pp. 213–229. https://doi.org/10.1007/3-540-44647-8_13
15. A. Boldyreva, V. Goyal, V. Kumar, Identity-based encryption with efficient revocation, in Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008, Alexandria, Virginia, USA, October 27-31, 2008, (2008), pp. 417–426. https://doi.org/10.1145/1455770.1455823
16. D. Naor, M. Naor, J.B. Lotspiech, Revocation and tracing schemes for stateless receivers. IACR Cryptol. ePrint Arch. **2001**, 59 (2001)
17. B. Qin, R.H. Deng, Y. Li, S. Liu, Server-aided revocable identity-based encryption, in Computer Security—ESORICS 2015—20th European Symposium on Research in Computer Security, Vienna, Austria, September 21-25, 2015, Proceedings, Part I, (2015), pp. 286–304. https://doi.org/10.1007/978-3-319-24174-6_15
18. J. Chen, H.W. Lim, S. Ling, H. Wang, K. Nguyen, Revocable identity-based encryption from lattices, in Information Security and Privacy—17th Australasian Conference, ACISP 2012, Wollongong, NSW, Australia, July 9–11, 2012. Proceedings, (2012), pp. 390–403. https://doi.org/10.1007/978-3-642-31448-3_29
19. K. Nguyen, H. Wang, J. Zhang, Server-aided revocable identity-based encryption from lattices, in Cryptology and Network Security—15th International Conference, CANS 2016, Milan, Italy, November 14–16, 2016, Proceedings, (2016), pp. 107–123. https://doi.org/10.1007/978-3-319-48965-0_7
20. A. Boldyreva, V. Goyal, V. Kumar, Identity-based encryption with efficient revocation. IACR Cryptol. ePrint Arch. **2012**, 52 (2012)
21. J.M.G. Nieto, M. Manulis, D. Sun, Fully private revocable predicate encryption, in Information Security and Privacy—17th Australasian Conference, ACISP 2012, Wollongong, NSW, Australia, July 9–11, 2012. Proceedings, (2012), pp. 350–363. https://doi.org/10.1007/978-3-642-31448-3_26
22. S. Katsumata, T. Matsuda, A. Takayasu, Lattice-based revocable (hierarchical) IBE with decryption key exposure resistance, in Public-Key Cryptography—PKC 2019—22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Beijing, China, April 14–17, 2019, Proceedings, Part II, (2019), pp. 441–471. https://doi.org/10.1007/978-3-030-17259-6_15

23. C. Gentry, C. Peikert, V. Vaikuntanathan, Trapdoors for hard lattices and new cryptographic constructions, in Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17–20, 2008, (2008), pp. 197–206. https://doi.org/10.1145/1374376.1374407
24. J. Alwen, C. Peikert, Generating shorter bases for hard random lattices. Theory Comput. Syst. **48**(3), 535–553 (2011). https://doi.org/10.1007/s00224-010-9278-3
25. D. Micciancio, C. Peikert, Trapdoors for lattices: simpler, tighter, faster, smaller, in Advances in Cryptology—EUROCRYPT 2012—31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15–19, 2012. Proceedings, (2012), pp. 700–718. https://doi.org/10.1007/978-3-642-29011-4_41
26. S. Agrawal, D. Boneh, X. Boyen, Efficient lattice (H)IBE in the standard model, in Advances in Cryptology—EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco/French Riviera, May 30–June 3, 2010. Proceedings, (2010), pp. 553–572. https://doi.org/10.1007/978-3-642-13190-5_28
27. D. Cash, D. Hofheinz, E. Kiltz, C. Peikert, Bonsai trees, or how to delegate a lattice basis. J. Cryptol. **25**(4), 601–639 (2012). https://doi.org/10.1007/s00145-011-9105-2
28. O. Regev, On lattices, learning with errors, random linear codes, and cryptography, in Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22–24, 2005, (2005). pp. 84–93. https://doi.org/10.1145/1060590.1060603
29. A.E. Litvak, A. Pajor, M. Rudelson, N. Tomczak-Jaegermann, Smallest singular value of random matrices and geometry of random polytopes. Adv. Math. **195**(2), 491–523 (2005)
30. Y. Dodis, R. Ostrovsky, L. Reyzin, A.D. Smith, Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. SIAM J. Comput. **38**(1), 97–139 (2008). https://doi.org/10.1137/060651380

Publisher's Note