Multi-Keyword Searchable Identity-Based Proxy Re-Encryption with Validity Period Control from Lattices[*]

Er-Shuo Zhuang, Kai-Him Lam, Ming-Feng Tsai, Wei-Cheng Hung, and Chun-I Fan[†]

National Sun Yat-sen University, Kaohsiung, Taiwan
{zhuanges, parktasi, will900329}@gmail.com, s93681147@yahoo.com.hk,
cifan@mail.cse.nsysu.edu.tw

Abstract

The rapid advancement of science and technology has led to the emergence of quantum computers, posing a significant threat to traditional encryption schemes. As a result, there is a pressing need to develop encryption methods that can withstand quantum attacks. This thesis proposes an IBE scheme with a time-invalidation mechanism to revoke expired ciphertext. What sets this scheme apart is its support for multi-keyword search, AND/OR operations and proxy encryption to delegate encryption workload to a proxy server, reducing the computing load on the user side. This IBE scheme eliminates the need for certificates, reducing computation and transmission costs. The proposed scheme's security is based on the D-RLWE assumption. We also validate the proposed scheme through implementation.

1 Introduction

The rise of cloud computing has heightened concerns regarding the privacy and security of sensitive data stored in third-party environments. While cloud services offer scalability and flexibility in data management, they also introduce risks such as unauthorized access, data breaches, and privacy violations. Traditional encryption techniques are inadequate for handling the complex access control demands inherent in collaborative data sharing. As a result, more advanced cryptographic solutions are required.

Functional encryption addresses this need by enabling fine-grained access control, allowing users to decrypt only specific data elements based on predefined privileges. This is particularly valuable in collaborative scenarios, such as cloud environments, where multiple users require access to different subsets of encrypted data. Functional encryption thereby enhances data confidentiality and minimizes unnecessary exposure.

To simplify key management, Shamir introduced identity-based encryption (IBE) in 1984 [10], allowing encryption keys to be derived directly from user identities. This approach eliminates the need for certificates and streamlines key distribution. Later, Blaze et al. proposed proxy re-encryption (PRE) [4], which addresses the inefficiencies of traditional encryption in data sharing. PRE enables ciphertexts to be re-encrypted for a new recipient without decrypting the original content, reducing the risk of data leakage and improving efficiency. Building upon this, Green and Ateniese introduced identity-based proxy re-encryption (IB-PRE) [6], combining the benefits of IBE and PRE to further enhance secure data sharing.

[*] Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. 22, December 16-18, 2025, Sapporo, Japan.

[†] Corresponding author

Searchable encryption (SE) addresses the challenge of querying encrypted data without requiring decryption. Traditional methods often necessitate downloading and decrypting entire datasets for local search, leading to high communication and computation overhead. Song, Wagner, and Perrig introduced the first SE scheme [11] to perform keyword searches over encrypted data securely. Subsequently, Boneh et al. proposed a public-key SE scheme [5], allowing a cloud server (CS) to search ciphertexts using trapdoors without learning the plaintext or the search query.

Data validity control further strengthens SE by enforcing temporal access policies. By restricting access to a predefined validity period, the system ensures that expired ciphertext becomes inaccessible or is automatically deleted. This not only reduces the risk of unauthorized access but also improves operational efficiency by minimizing the overhead of managing outdated information.

Lattice-based cryptography has emerged as a leading candidate for post-quantum cryptographic security, as it relies on hard problems believed to be resistant to quantum attacks. Its foundation includes the closest vector problem (CVP) and the shortest vector problem (SVP). The learning with errors (LWE) problem [9], which involves decoding noisy linear equations, underpins many lattice-based schemes and increases in difficulty with problem dimension. The ring-LWE (RLWE) problem [7], a variant of LWE that utilizes ring structures, offers improved efficiency in computation and storage.

This thesis proposes a novel scheme that embeds data validity periods directly into ciphertexts. By incorporating expiration times, data owners can control the lifecycle of their encrypted content, ensuring that outdated ciphertexts are automatically invalidated and access is restricted beyond their valid duration.

1.1 Contributions

We improve upon the scheme proposed by Zhuang and Fan [16] by replacing the underlying hardness assumption with ring learning with errors (RLWE), resulting in more efficient storage and computation. The proposed scheme has the following key features:

- Unlike Zhuang and Fan's work, our scheme adopts a polynomial representation based on RLWE. Although RLWE is relatively recent (introduced in 2010) and has limited literature, it offers several advantages:
  - The polynomial structure enables encryption of multiple bits simultaneously, reducing transmission overhead.
  - It allows the search algorithm to perform multiple verifications efficiently during the search phase.
  - Polynomial arithmetic accelerates computation; incorporating Montgomery reduction further improves the efficiency of polynomial multiplication.
- To the best of our knowledge, this is the first RLWE-based multi-keyword searchable encryption scheme that supports both AND and OR logic operations.
- The scheme integrates data validity control by embedding expiration timestamps into ciphertexts. This ensures automatic invalidation of expired data, allowing data owners (DOs) to manage the lifecycle of encrypted content more effectively.
- This work follows a parallel track of theoretical design and practical implementation. We validate the theoretical constructs through real-world implementation results, bridging the gap between algorithm design and deployment.

- The scheme introduces an operational procedure for Type-2 trapdoor functions, which offer superior performance compared to Type-1 trapdoors.
- The key generation center (KGC) is only involved during the user registration phase and does not participate in subsequent processes, thereby reducing communication and computation overhead as well as minimizing the KGC's exposure to network-based attacks.
- Proxy re-encryption (PRE) is employed to enable outsourced computation. Encryption and re-encryption operations can be delegated to the proxy server (PS), alleviating the computational load on DOs and lowering the hardware requirements for client devices.
- The security of the proposed scheme relies on the decisional RLWE (D-RLWE) assumption. We prove that it achieves both indistinguishability under chosen-plaintext attacks (IND-CPA) and indistinguishability under chosen-keyword attacks (IND-CKA). Due to the strict 20-page limit of the MobiSec 2025 submission format (EasyChair style), which includes appendices, the detailed security proofs are omitted in this version. The complete proofs are available upon request.

2 Related Works

This section reviews recent advancements in SE, PRE, and validity period control mechanisms.

Yang et al. proposed a conjunctive time-validity searchable encryption scheme tailored for e-healthcare cloud environments [13]. Their scheme enables data owners (DOs) to revoke a user's access rights through time validity controls. It also allows DOs to define specific access periods for different users, enabling fine-grained access control over sensitive medical data.

Zhang et al. introduced an identity-based searchable encryption scheme with a time validity feature embedded via a validity period searchable encryption (VPSE) mechanism [15]. Each encrypted document is tagged with a specific validity period, allowing searches only within the designated time window and preventing access outside that period. This approach enhances data freshness and access relevance. Furthermore, the proxy search server remains oblivious to the searched keywords, preserving user privacy and ensuring search confidentiality.

Yu et al. proposed a lattice-based public-key encryption with keyword search for e-Office systems [14]. Their scheme incorporates PRE with validity period control and supports dynamic revocation of user access. Time information is embedded directly into user keys, enabling time-bound delegation without relying on an external time server. This design reduces both overhead and potential security risks.

Zhuang and Fan presented a multi-keyword searchable identity-based PRE scheme from lattices using a tree-based access structure [16]. They highlighted potential vulnerabilities in prior schemes based on linear secret sharing (LSSS). Their approach employs PRE to reduce file-sharing overhead for DOs and leverages outsourcing to scale the system without increasing the burden on DOs. As the number of data users (DUs) grows, the proxy server absorbs the additional workload, keeping the DO's cost constant.

3 Preliminaries

This section introduces the preliminaries used in the proposed scheme. Lowercase symbols denote vectors, e.g., a, while uppercase symbols denote matrices, e.g., A. The notations a^{T} and A^{T} represent the transposes of vectors and matrices, respectively. The notation $[a_1 \mid a_2]$ denotes the concatenation of two vectors or two matrices. Let $\mathbb{R} = \mathbb{Z}[x]/\langle x^n + 1\rangle$ be the polynomial ring. Each element in $\mathbb{R}$ is a polynomial of degree at most $n-1$, where n is a power of 2. $\mathbb{R}_q$ denotes the set of polynomials modulo q, and $\mathbb{Z}_q$ denotes the set of integers modulo q.

3.1 Lattice

A lattice is formed by a set of linearly independent vectors. Its formal definition is as follows:

Definition 3.1. Given n linearly independent vectors $a_1, a_2, a_3, \ldots, a_n \in \mathbb{Z}^m$, a lattice Λ of dimension m is generated through the linear combinations of these vectors.

$$\Lambda = \zeta(a_1, a_2, a_3, \ldots, a_n) = \{\sum_{m=1}^{n} x_i a_i | x_i \in \mathbb{Z}\}$$

where n signifies its rank. The set of vectors $\{a_1, a_2, a_3, \ldots, a_n\}$ constitutes the basis of the lattice Λ.

Definition 3.2. Given a prime number q, a basis $A = \{a_1, a_2, a_3, \ldots, a_n\} \in \mathbb{Z}^{m\times n}$, and a random vector $u \in \mathbb{Z}_q^n$, the following are three common types of q-ary lattices:

$$\begin{aligned}\Lambda_q(A) &:= \{s \in \mathbb{Z}^m \mid A^{\mathrm{T}} e = s \mod q, e \in \mathbb{Z}^n\}\\ \Lambda_q^u(A) &:= \{s \in \mathbb{Z}^m \mid As = u \mod q\}\\ \Lambda_q^{\perp}(A) &:= \{s \in \mathbb{Z}^m \mid As = 0 \mod q\}\end{aligned}$$

3.2 The Ring Decisional Learning-With-Errors Problem

Given a polynomial ring $\mathbb{R}_q$ and two samplers O_s and O_Ψ defined as follows:

- O_s: A noisy pseudo-random sampler that outputs pairs $(a_i, a_i s + e_i) \in \mathbb{R}_q^2$, where $a_i \in \mathbb{R}_q$ is a uniformly random polynomial, $s \in \mathbb{R}_q$ is a fixed secret polynomial, and $e_i \in \mathbb{R}_\chi$ is a noise polynomial drawn from an error distribution χ.
- O_Ψ: A truly random sampler that outputs independent and uniformly random pairs $(a_i, v_i) \in \mathbb{R}_q^2$.

Definition 3.3. Ring Decisional Learning-With-Errors (R-DLWE) Problem
Given access to a set of samples generated by an oracle $O_?$ that is either O_s or O_Ψ, the goal is to determine whether $O_? = O_s$ or $O_? = O_\Psi$.

3.3 Lattice Trapdoors

Possessing a trapdoor for a particular lattice enables the efficient solution of certain otherwise hard problems. Two primary approaches for generating such trapdoors are based on the short integer solution (SIS) problem [1] and the learning with errors (LWE) problem [9], respectively. Micciancio and Peikert [8] proposed a more efficient construction of trapdoor functions for both LWE and RLWE. The scheme presented in this work adopts trapdoor functions based on the RLWE assumption.

For clarity, we focus on describing the lattice trapdoor functions in terms of their inputs and outputs. The detailed algorithms and internal computations of these trapdoor functions are omitted in this version. The complete algorithmic descriptions are available upon request.

Definition 3.4. Gadget vector g
A gadget vector $g = (1, 2, 4, \ldots, 2^{w-1})^{\mathrm{T}} \in \mathbb{R}_q^w$, where $w = \lceil \log_2 q \rceil$.

Definition 3.5. $(a, R) \leftarrow$ ***GenTrap***(a_0, h)
Input: A tag $h \in \mathbb{Z}_q$, and a vector $a_0 \in \mathbb{R}_q^{m'}$.
Output: $(a,\ R)$, where R is a trapdoor of a.

Definition 3.6. $(x) \leftarrow$ ***SampleD***(R, a, h, t, α)
Input: Given a vector $a \in \mathbb{R}_q^m$, a trapdoor R of a, a tag h, a target $t \in \mathbb{R}_q$ and a Gaussian parameter α.
Output: x, such that $ax = t$.

Definition 3.7. $(R') \leftarrow$ ***DelTrap***(a', R, h, α)
Input: Given $a' = [a|a_1] \in \mathbb{R}_q^m \times \mathbb{R}_q^w$, a tag h, a trapdoor R of a.
Output: R', where R' is a trapdoor of a'.

3.4 Tree-Based Access Structure

The proposed scheme requires the data owner (DO) to embed keywords into the ciphertext during the encryption phase. Users subsequently generate search tokens according to a specified search policy. The cloud server (CS) transmits the ciphertext only when the user's search token matches the embedded keywords.

The proposed scheme employs a tree-based structure to define complex access control policies. In this structure, the leaf nodes represent the values of the attributes (or the keywords), while the non-leaf nodes correspond to logical gates such as AND and OR. By combining these logical operations, the structure enables the construction of fine-grained access rules, ensuring that access to protected resources is granted only to authorized users.

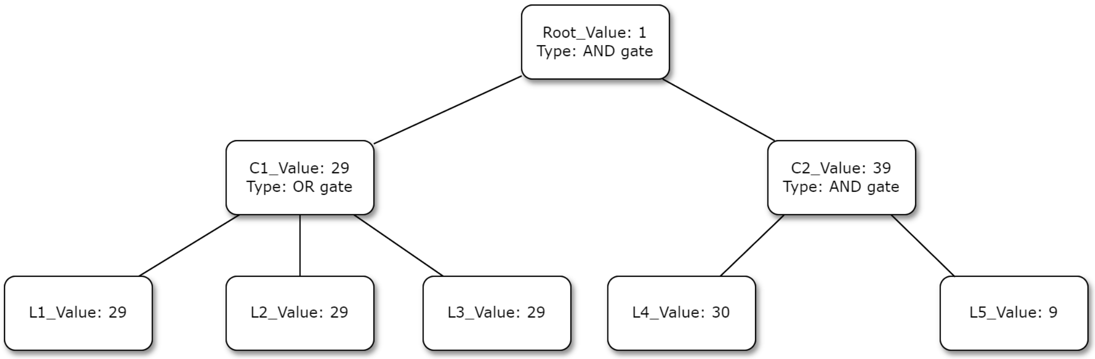

Figure 1: An Example for Tree-Based Structure

Figure 1 illustrates an example of a search policy defined as $\tau = (L1 \cup L2 \cup L3) \cap (L4 \cap L5)$. Let $q = 67$. The root node is assigned the value 1 and has two child nodes, denoted as C1 and C2. Since the root node represents an AND gate, the sum of the values of C1 and C2 must equal that of the root node modulo q.

C1 corresponds to an OR gate, implying that each of its child nodes must carry the same value as C1. On the other hand, C2 represents an AND gate, and therefore the sum of its child nodes must equal C2 modulo q.

For a user's search to succeed (i.e., the keywords match), the following equation must be satisfied:

$$(L1 \cup L2 \cup L3) + L4 + L5 \equiv 1 \pmod{q}.$$

3.5 System Model

Figure 2 illustrates the proposed system model, which involves five distinct roles:

- **Key Generation Center (KGC)**: The KGC is responsible for initializing the system by generating public parameters and distributing cryptographic keys. It is assumed to be a fully trusted entity.
- **Cloud Server (CS)**: The CS stores ciphertexts and indexes, and performs search operations upon receiving search tokens. It is considered an honest-but-curious entity, meaning it follows protocols correctly but may attempt to infer sensitive information.
- **Proxy Server (PS)**: The PS handles the re-encryption phase. Upon receiving the re-encryption keys, ciphertexts, indexes, and data user identities from the data owner (DO), the PS re-encrypts the ciphertext and index accordingly. The re-encrypted data is then sent to the CS. The PS is regarded as a fully trusted entity.
- **Data Owner (DO)**: The DO possesses the original plaintext data and is responsible for generating re-encryption keys for authorized data users.
- **Data User (DU)**: The DU can generate search tokens based on their access rights and decrypt ciphertexts that match their authorized search queries.

The proposed scheme consists of the following nine algorithms:

- $(PP, MSK) \leftarrow$ ***Setup***(λ): Given a security parameter λ, the KGC generates the public parameters PP and the master secret key MSK.
- $(SK_{ID}) \leftarrow$ ***KeyGen***$(MSK,\ ID)$: Given a user identity ID and the master secret key MSK, the KGC generates the private key SK_{ID} corresponding to ID.
- $(RK_{ID_x \to ID_y}) \leftarrow$ ***Re-KeyGen***(ID_x, ID_y, SK_{ID_x}): Given the data owner's identity ID_x, the data user's identity ID_y, and the data owner's private key SK_{ID_x}, the data owner computes the re-encryption key $RK_{ID_x \to ID_y}$.
- $(C_{ID_x}, I_{ID_x}) \leftarrow$ ***Encrypt***(μ, ID_x, KW_{ID_x}): Given a message μ, the data owner's identity ID_x, and a set of keywords KW_{ID_x}, the data owner outputs the ciphertext C_{ID_x} and the corresponding index I_{ID_x}.
- $(C_{ID_y}, I_{ID_y}, ID_y) \leftarrow$ ***Re-Enc***$(C_{ID_x}, I_{ID_x}, ID_y,\ RK_{ID_x \to ID_y})$: Given the ciphertext C_{ID_x}, index I_{ID_x}, data user's identity ID_y, and the re-encryption key $RK_{ID_x \to ID_y}$, the proxy server computes the re-encrypted ciphertext C_{ID_y} and index I_{ID_y}.
- $(TK) \leftarrow$ ***TokenGen***(τ, SK_{ID}, ID): Given a keyword search policy τ, identity ID, and private key SK_{ID}, the data user generates a search token TK.

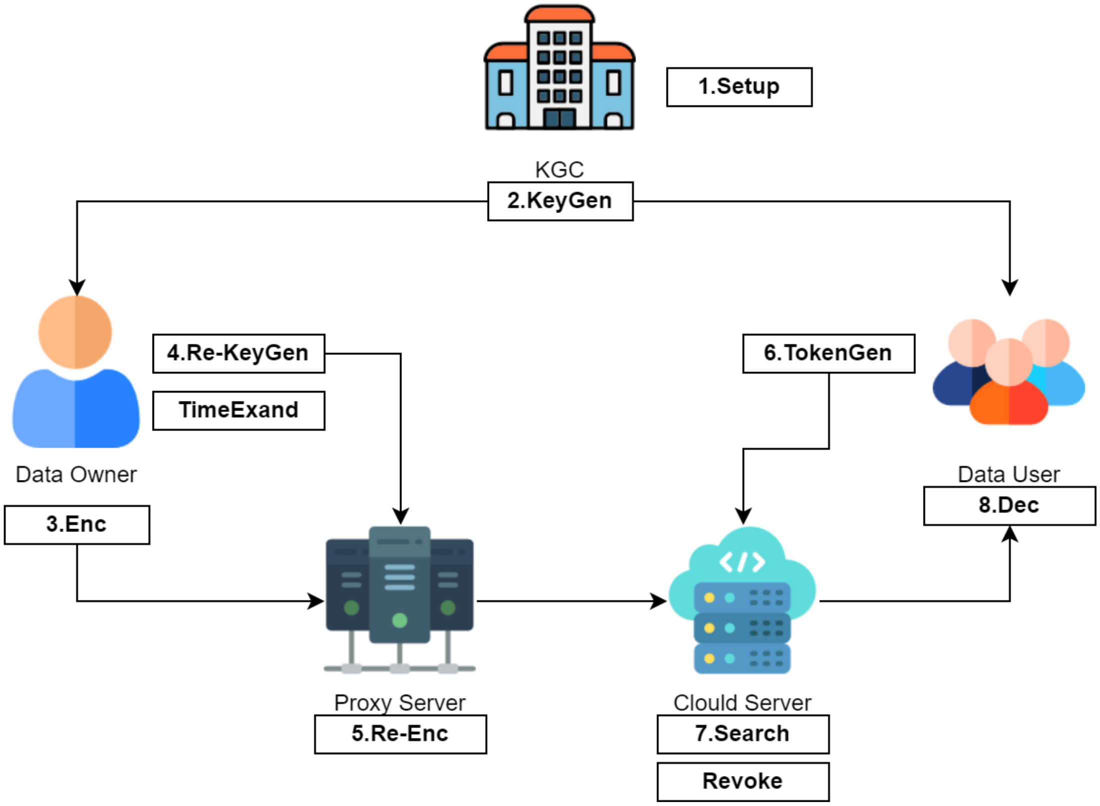

Figure 2: The System Model

- $(C_{ID}$ **or** $\perp) \leftarrow$ ***Search***$(TK, \tau, ID, C_{ID}, I_{ID})$: Given the search token TK, identity ID, ciphertext C_{ID}, and index I_{ID}, the cloud server checks whether TK matches I_{ID} and is within the valid time period. If both conditions hold, the ciphertext C_{ID} is returned; otherwise, the output is $\perp$.

- $(I_1^{ID_x}, I_2^{ID_x}) \leftarrow$ ***UpdateTime***$(I_1^{ID_x}, I_2^{ID_x}, p_0, p_1)$: Given index components $I_1^{ID_x}$ and $I_2^{ID_x}$, and time-related parameters p_0 and p_1, the cloud server updates the index to produce new values $(I_1^{ID_x}, I_2^{ID_x})$.

- $(\mu) \leftarrow$ ***Decrypt***(SK_{ID}, C_{ID}): Given the ciphertext C_{ID} and the private key SK_{ID}, the data user decrypts C_{ID} to recover the original message μ.

3.6 Security Model

The proposed scheme achieves ciphertext indistinguishability under chosen plaintext attacks (IND-CPA) and keyword search security under chosen keyword attacks (IND-CKA). Due to submission format constraints, detailed security proofs are omitted in this version; the complete proofs are available upon request.

4 Construction

The proposed scheme can be divided into nine algorithms, which are ***Setup***, ***KeyGen***, ***Re-KeyGen***, ***Encrypt***, ***Re-Enc***, ***TokenGen***, ***Search***, ***Decrypt***, and ***UpdateTime***. Table 1 lists the parameters used in the proposed scheme.

Table 1: The Notations

Notation	Meaning
PP	Public parameters
λ	A security parameter
k_i	The ith keyword
ID	User's identity
KW	System keyword set
l	Number of keywords
a	A vector
a^{T}	The transpose of a
H_0, H_1, H_2	Hash functions
SK_{ID}	The private key of ID
q	A prime number
w	$w = \lceil \log_2 q \rceil$
TK	A search token
τ	An access policy
T_{τ,k_i}	A search keyword in τ
μ	Plaintext
P_0	Start Time
P_1	End Time
p_0	Modified Start Time
p_1	Modified End Time

4.1 Setup

$(PP, MSK) \leftarrow$ ***Setup***(λ): Given a security parameter λ, KGC performs the following steps to generate the master secret key MSK and PP.

1. Select a Gaussian distribution χ, a Gaussian parameter σ, a prime q, three values n, m', l, a keyword set $KW = \{k_i\}_{1 \leq i \leq l}$, a random vector $a_0 \in \mathbb{R}_q^{m'}$, an integer $\mathbf{h} \in \mathbb{Z}_q$, a random polynomial $\mathbf{u} \in \mathbb{R}_q$. All computations work over a polynomial ring $\mathbb{R}_q = \mathbb{Z}_q[x]/(x^n + 1)$. Note that n should be a power of 2.

2. Set $w = \lceil \log_2 q \rceil$, $m = m' + w$ and a gadget vector $g = (1, 2, 4, \ldots, 2^{w-1})^{\mathrm{T}} \in \mathbb{R}_q^w$.

3. Select three hash functions $H_0 : \{0,1\}^* \rightarrow \mathbb{R}_q^w, H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q$, and $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_q$.

4. Generate a vector $a = [a_0^{\mathrm{T}} | -a_0^{\mathrm{T}} R + hg^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}_q^m$ and a trapdoor R by ***GenTrap*** function defined in 3.5.

5. Publish the public parameters $PP = (\chi, \sigma, q, n, m, l, w, KW = \{k_i\}_{1 \leq i \leq l}, a,$
$H_0, H_1, H_2, u)$ and set $MSK = R$.

4.2 KeyGen

$(SK_{ID}) \leftarrow$ ***KeyGen***$(MSK,\ ID)$: KGC performs the following steps to compute the user's private key SK_{ID}.

1. Compute $a_{ID} = H_0(ID) \in \mathbb{R}_q^w$.
2. Set $a'_{ID} = [a|a_{ID}]$.
3. Compute $h_{ID} = H_1(ID) \in \mathbb{Z}_q$.
4. Generate R_{ID} using ***DelTrap***$(R, a'_{ID}, h_{ID}, \sigma)$ as defined in definition 3.7, such that $a^{\mathrm{T}} R_{ID} = h_{ID} g^{\mathrm{T}} - a_{ID}^{\mathrm{T}}$, where R_{ID} is the trapdoor for $\Lambda_q^{\perp}(a'_{ID})$.
5. Generate x_{ID} using ***SampleD***$(R_{ID}, a'_{ID}, h_{ID}, u, \sigma)$ as defined in definition 3.6, where $a'_{ID} x_{ID} = u$.
6. Return $SK_{ID} = \{R_{ID}, x_{ID}\}$ as user's private key.

4.3 Re-KeyGen

$(RK_{ID_x \to ID_y}) \leftarrow$ ***Re-KeyGen***(ID_x, ID_y, SK_{ID_x}): DO can compute re-encryption keys by using its SK_{ID}.

1. Compute $a_{ID_x} = H_0(ID_x) \in \mathbb{R}_q^w$.
2. Set $a'_{ID_x} = [a|a_{ID_x}] \in \mathbb{R}_q^{m+w}$.
3. Compute $a_{ID_y} = H_0(ID_y) \in \mathbb{R}_q^w$.
4. Set $a'_{ID_y} = [a|a_{ID_y}] \in \mathbb{R}_q^{m+w}$.
5. Compute $h_{ID_x} = H_1(ID_x) \in \mathbb{Z}_q$.
6. Generate $R_{ID_x \to ID_y}$ using ***SampleD***$(R_{ID_x}, a'_{ID_x}, h_{ID_x}, a'_{ID_y}, \sigma)$ as defined in definition 3.6, where $a'_{ID_x} R_{ID_x \to ID_y} = a'_{ID_y}$.
7. Return $RK_{ID_x \to ID_y} = R_{ID_x \to ID_y}^{\mathrm{T}}$.

4.4 Encrypt

$(C_{ID_x}, I_{ID_x}) \leftarrow$ ***Encrypt***(μ, ID_x, KW_{ID_x}): DO encrypt a message $\mu = \mu_0 x^0 + \mu_1 x^1 + \ldots + \mu_{n-1} x^{n-1} \in \mathbb{R}_q$ with a keyword set $KW_{ID_x} \subseteq KW = \{k_i\}_{1 \le i \le l}$ and its ID.

1. Set two secret polynomials P_0 and P_1.
2. Compute $a_{ID_x} = H_0\left(ID_x\right) \in \mathbb{R}_q^w$.
3. Set $a'_{ID_x} = [a \mid a_{ID_x}] \in \mathbb{R}_q^{m+w}$.
4. Sample noises $e_1, e_3 \in \mathbb{R}_\chi$.
5. Sample noises $e_2, e_{4,1}, e_{4,2}, \ldots e_{4,l} \in \mathbb{R}_\chi^{m+w}$.

```
6. Choose two secret polynomials $s, v \in \mathbb{R}_q$.
7. Compute $C_1^{ID_x} = us + \mu \left\lfloor \frac{q}{2} \right\rfloor + e_1 \in \mathbb{R}_q$.
8. Compute $C_2^{ID_x} = a'_{ID_x} s + e_2 \in \mathbb{R}_q^{m+w}$.
9. Set the ciphertext $C_{ID_x} = \left\{ C_1^{ID_x}, C_2^{ID_x} \right\}$.
10. Compute $I_1^{ID_x} = uv + e_3 + P_0 \cdot \frac{q}{t} \in \mathbb{R}_q$.
11. Compute $I_2^{ID_x} = P_1 \cdot \frac{q}{t} \in \mathbb{R}_q$.
12. For $i = 1$ to $l$,
13.     if $k_i \in KW_{ID_x}$:
14.         Compute $h_{k_i} = H_2(k_i) \in \mathbb{Z}_q$.
15.         Compute $I_{3,k_i}^{ID_x} = a'^{\mathrm{T}}_{ID_x} h_{k_i} v + e_{4,i} \in \mathbb{R}_q^{m+w}$.
16.     else:
17.         Generate a random vector $I_{3,k_i}^{ID_x} \in \mathbb{R}_q^{m+w}$.
18. Set the index $I_{ID_x} = \left\{ I_1^{ID_x}, I_2^{ID_x}, \left\{ I_{3,k_i}^{ID_x} \right\}_{1 \le i \le l} \right\}$.
19. Return $(C_{ID_x}, I_{ID_x})$ and send it to PS.
```

4.5 Re-Enc

$(C_{ID_y}, I_{ID_y}, ID_y) \leftarrow$ ***Re-Enc***$(C_{ID_x}, I_{ID_x}, ID_y, RK_{ID_x \to ID_y})$: PS re-encrypts C_{ID_x} and I_{ID_x} with DU's ID_y, and the re-encryption key $RK_{ID_x \to ID_y}$.

```
1. Sample noises $e'_2, e'_{4,1}, e'_{4,2}, \ldots e'_{4,l} \in \mathbb{R}_\chi^{m+w}$.
2. Set $C_1^{ID_y} = C_1^{ID_x}$.
3. Compute $C_2^{ID_y} = R^{\mathrm{T}}_{ID_x \to ID_y} C_2^{ID_x} + e'_2$.
4. Set $C_{ID_y} = \left\{ C_1^{ID_y}, C_2^{ID_y} \right\}$.
5. Set $I_1^{ID_y} = I_1^{ID_x}$.
6. Set $I_2^{ID_y} = I_2^{ID_x}$.
7. For $i = 1$ to $l$,
8.     Compute $I_{3,k_i}^{ID_y} = R^{\mathrm{T}}_{ID_x \to ID_y} I_{3,k_i}^{ID_x} + e'_{4,i}$.
9. Set $I_{ID_y} = \left\{ I_1^{ID_y}, I_2^{ID_y}, \left\{ I_{3,k_i}^{ID_y} \right\}_{1 \le i \le l} \right\}$.
10. Return $\left( C_{ID_y}, I_{ID_y} \right), ID_y$ and send it to CS.
```

4.6 TokenGen

$(TK) \leftarrow$ ***TokenGen***$(\tau,\ SK_{ID},\ ID)$: DU can compute a search token with a keyword-search policy τ, and its SK_{ID}.

```
1. Compute $a_{ID} = H_0(ID) \in \mathbb{R}_q^w$.
2. Set $a'_{ID} = [a|a_{ID}] \in \mathbb{R}_q^{m+w}$.
3. Compute $h_{ID} = H_1(ID) \in \mathbb{Z}_q$.
4. According to $\tau$, generate values $\{T_{\tau,k_i} \in \mathbb{Z}_q\}_{k_i \in \tau}$, (the method is defined in section 2.7).
5. for $i = 1$ to $l$,
6.     if $k_i \in \tau$:
7.         Compute $h_{k_i} = H_2(k_i) \in \mathbb{Z}_q$.
8.         Compute the inverse of $h_{k_i}$ to get $h_{k_i}^{-1}$.
9.         Compute $T'_{k_i} = h_{k_i}^{-1} T_{\tau,k_i} u$.
10.        Generate $w_{k_i}$ using SampleD$(R_{ID}, a'_{ID}, h_{ID}, T'_{k_i}, \sigma)$ as defined in definition 3.6, where $a'_{ID} w_{k_i} = T'_{k_i}$.
11. Set the search token $TK = (\tau, \{w_{k_i}\}_{k_i \in \tau})$ and send $(TK,\ ID)$ to CS.
```

4.7 Search

$(C_{ID}$ **or** $\perp) \leftarrow$ ***Search***$(TK,\ \tau,\ ID,\ C_{ID},\ I_{ID})$: CS can check whether TK matches I_{ID} within a validity period between p_0 and p_1.

```
1. Set $z = 0$.
2. For $i = 1$ to $l$,
3.     if $k_i \in \{k'_1, k'_2, \dots, k'_o\}$:
4.         compute $z = z + w_{k_i}^{\top} I_{3,k_i}^{ID}$.
5. Compute $f_1 = |z - I_1^{ID} + P_i \cdot \frac{q}{t}| + |z - (I_1^{ID} + I_2^{ID}) + P_i \cdot \frac{q}{t}\ |$.
6. Compute $f_2 = I_2^{ID}$.
7. Compute $f = |\ f_1 - f_2\ |$.
8. For $j = 0$ to $n - 1$,
9.     if $f_j > \frac{q}{2t}$, return $\perp$.
10. Return $C_{ID}$.
```

4.8 UpdateTime

$(I_1^{ID_x}, I_2^{ID_x}) \leftarrow$ ***UpdateTime***$(I_1^{ID_x}, I_2^{ID_x}, p_0, p_1)$: If DO wants to update the validity period of the index, DO sends two values (p_0,p_1) to CS. Then, CS can update the validity period of the index.

1. Update $I_1^{ID_x} = I_1^{ID_x} - p_0 \cdot \frac{q}{t}$.

2. Update $I_2^{ID_x} = I_2^{ID_x} + p_1 \cdot \frac{q}{t}$.

3. Return $(I_1^{ID_x}, I_2^{ID_x})$.

4.9 Decrypt

$(\mu) \leftarrow$ ***Decrypt***(SK_{ID}, C_{ID}): DU can decrypt the ciphertext C_{ID} with its SK_{ID}.

```
1. Compute $\mu' = |C_1^{ID} - C_2^{ID^{\mathrm{T}}} x_{ID}|$.
2. For $j = 0$ to $n-1$,
3.     if $\mu'_j < \frac{q}{4}$, set $\mu_j = 0$; otherwise, set $\mu_j = 1$.
4. Return $\mu$.
```

4.10 Decryption Correctness

$$\begin{aligned}\mu' &= \left|C_1^{ID} - C_2^{ID^{\mathrm{T}}} x_{ID}\right| \\ &= \left|\left(us + \mu \left\lfloor \frac{q}{2} \right\rfloor + e_1\right) - (x_{ID} a_{ID} s + x_{ID} e_2)\right| \\ &= \left|us + \mu \left\lfloor \frac{q}{2} \right\rfloor + e_1 - us - x_{ID} e_2\right| \\ &= |\mu \left\lfloor \frac{q}{2} \right\rfloor + \underbrace{e_1 - x_{ID} e_2}_{error\ term}|.\end{aligned}$$

To ensure the accuracy of the decryption results, the absolute value of each coefficient of the error term must be less than $\frac{q}{4}$.

4.11 Search Correctness

$$\begin{aligned}
f_1 &= |z - I_1^{ID} - P_i \cdot \frac{q}{t}| + |z - (I_1^{ID} - I_2^{ID}) - P_i \cdot \frac{q}{t}\ | \\
&= |\ \sum_{i=1}^{n} w_{k'_i}^{\mathrm{T}} I_{3,k'_i}^{ID} - I_1^{ID} - P_i \cdot \frac{q}{t}\ | + |\ \sum_{i=1}^{n} w_{k'_i}^{\mathrm{T}} I_{3,k'_i}^{ID} \\
&- I_2^{ID} - P_i \cdot \frac{q}{t}\ | \\
&=\ |\sum_{i=1}^{n} (w_{k'_i}^{\mathrm{T}} a_{ID}^{'T} h_{k'_i}^{'T} v + w_{k'_i}^{\mathrm{T}} e_{4,i'}) - (uv + e_3 - P_0 \cdot \frac{q}{t}) \\
&- P_i \cdot \frac{q}{t}| + |\sum_{i=1}^{n} (w_{k'_i}^{\mathrm{T}} a_{ID}^{'T} h_{k'_i}^{'T} v + w_{k'_i}^{\mathrm{T}} e_{4,i'}) \\
&\quad - (uv + e_3 - P_0 \cdot \frac{q}{t} - P_1 \cdot \frac{q}{t}) - P_i \cdot \frac{q}{t}| \\
&= |\sum_{i=1}^{n} uT_{\tau,k'_i} h_{k'_i}^{-1^{\mathrm{T}}} h_{k'_i}^{T} v + \sum_{i=1}^{n} w_{k'_i}^{\mathrm{T}} e_{4,i'} - uv - e_3 \\
&\quad + \frac{q}{t} P_0 - P_i \cdot \frac{q}{t}| + \sum_{i=1}^{n} uT_{\tau,k'_i} h_{k'_i}^{-1^{\mathrm{T}}} h_{k'_i}^{T} v| \\
&\quad + |\sum_{i=1}^{n} w_{k'_i}^{\mathrm{T}} e_{4,i'} - uv - e_3 + \frac{q}{t} P_0 - \frac{q}{t} P_1 - P_i \cdot \frac{q}{t}|
\end{aligned}$$

$$\begin{aligned}
f_1 = &|\sum_{i=1}^{n} uT_{\tau,k'_i} v + \sum_{i=1}^{n} w_{k'_i}^{\mathrm{T}} e_{4,i'} - uv - e_3 + \frac{q}{t} P_0 \\
&\quad - P_i \cdot \frac{q}{t}| + |\sum_{i=1}^{n} uT_{\tau,k'_i} v + \sum_{i=1}^{n} w_{k'_i}^{\mathrm{T}} e_{4,i'} - uv - e_3 \\
&\quad + \frac{q}{t} P_0 - \frac{q}{t} P_1 - P_i \cdot \frac{q}{t}| \\
= &|\ uv + \sum_{i=1}^{n} w_{k'_i}^{\mathrm{T}} e_{4,i'} - uv - e_3 + \frac{q}{t} P_0 - P_i \cdot \frac{q}{t}\ |+ \\
&|\ uv + \sum_{i=1}^{n} w_{k'_i}^{\mathrm{T}} e_{4,i'} - uv - e_3 + \frac{q}{t} P_0 - \frac{q}{t} P_1 - P_i \cdot \frac{q}{t}\ | \\
= &|\sum_{i=1}^{n} w_{k'_i}^{\mathrm{T}} e_{4,i'} - e_3 + \frac{q}{t} P_0 - P_i \cdot \frac{q}{t}| \\
&\quad + |\sum_{i=1}^{n} w_{k'_i}^{\mathrm{T}} e_{4,i'} - e_3 + \frac{q}{t} P_0 - \frac{q}{t} P_1 - P_i \cdot \frac{q}{t}|
\end{aligned}$$

$$f_2\ = I_2^{ID}$$

$$
\begin{aligned}
f &= |\ f_1 - f_2\ | \\
&= ||\sum_{i=1}^{n} w_{k_i'}^{\mathrm{T}} e_{4,i'} - e_3 + \frac{q}{t}P_0 - P_i \cdot \frac{q}{t}| \\
&\quad + |\sum_{i=1}^{n} w_{k_i'}^{\mathrm{T}} e_{4,i'} - e_3 + \frac{q}{t}P_0 - \frac{q}{t}P_1 - P_i \cdot \frac{q}{t}| - |\frac{q}{t}P_1|| \\
&= \underbrace{|\sum_{i=1}^{n} 2w_{k_i'}^{\mathrm{T}} e_{4,i'} - 2e_3\ |}_{error\ term}
\end{aligned}
$$

To ensure the accuracy of the search results, the absolute value of each coefficient of the error term must be less than $\frac{q}{4}$.

5 Security Proof

We provide security proofs for both ciphertext indistinguishability under chosen-plaintext attacks (IND-CPA) and keyword privacy under chosen-keyword attacks (IND-CKA).

In the IND-CPA game, the adversary is allowed to adaptively query encryption oracles before submitting two challenge messages. The proof proceeds through a series of hybrid games, each replacing portions of the ciphertext with random values. Any non-negligible advantage in distinguishing between these hybrids can be reduced to solving the decisional ring-LWE (D-RLWE) problem. The overall structure of the IND-CPA proof is illustrated in Fig. 5, which outlines the transition between hybrid games and the reduction process.

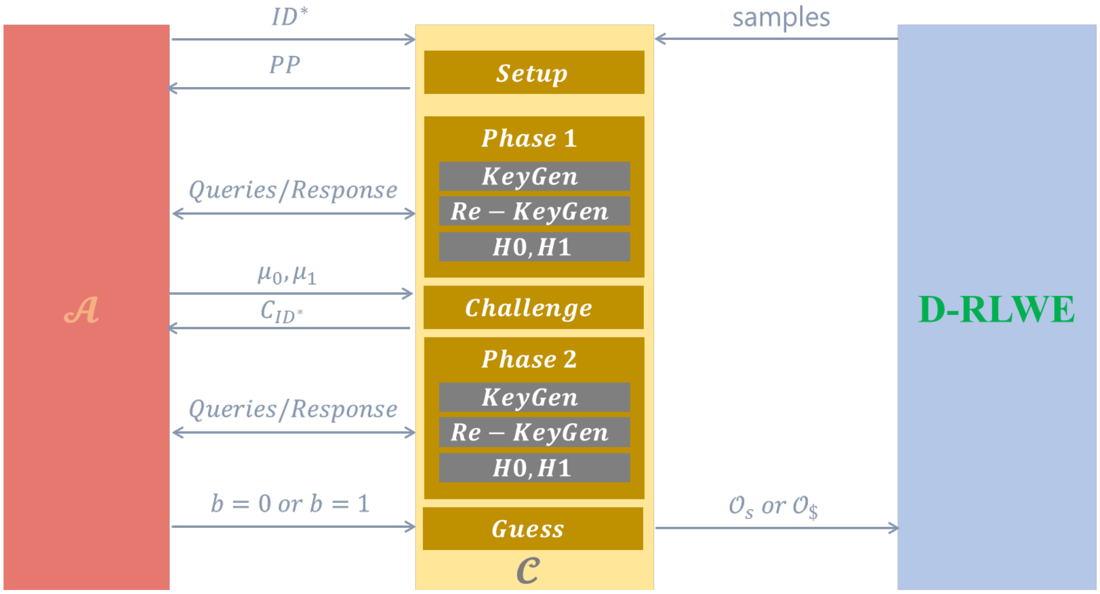

Figure 3: The IND-CPA Game

For IND-CKA security, the reduction constructs a simulator that replaces keyword ciphertexts with uniformly random elements while maintaining consistency across queries. Under the D-RLWE assumption, no efficient adversary can distinguish these simulated tokens from real ones, ensuring keyword privacy.

To comply with the submission format, the full formal proofs are omitted in this version; a complete version containing all details will be made available upon request.

6 Comparison

This section shows the comparisons between the proposed scheme and other schemes in several aspects: features, transmission cost, and computation cost. For fairness, we set the message of each scheme as one single bit. The feature comparison is shown in Table 2. The proposed scheme covers all the evaluated functions, including SE, multi-keyword search, PRE, validity period control, and quantum resistance. Table 3 shows the notations used for comparison.

6.1 Properties Comparison

Table 2 compares the functionalities of the proposed scheme with those of existing schemes. The PKE scheme proposed by Yu *et al.* [14] combines SE and validity period control based on the LWE problem and provides IND-CKA security. The IBE scheme proposed by Zhang *et al.* [15] supports multi-keyword search, validity period control, and quantum resistance but lacks a formal security proof. In contrast, our proposed scheme covers all evaluated functions, including SE, multi-keyword search, PRE, validity period control, and quantum resistance, demonstrating significant advantages in both functionality and security.

Moreover, the proposed scheme utilizes the RLWE problem, which offers an advantage in storage cost compared to Zhuang and Fan's scheme [16]. Wang *et al.* [12] and the proposed scheme both rely on RLWE; however, Wang *et al.*'s scheme does not support multi-keyword search.

Table 2: Feature Comparison

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11
[12]	Y	Y	N	None	N	N	N	Y	CPA, CKA	Y	RLWE
[14]	N	Y	N	None	Y	Y	N	Y	CKA	Y	LWE
[15]	Y	Y	N	None	N	Y	N	Y	Not Given	Y	LWE
[16]	Y	Y	Y	AND, OR	Y	N	N	Y	CPA, CKA	Y	LWE
Ours	Y	Y	Y	AND, OR	Y	Y	Y	Y	CPA, CKA	Y	RLWE

F1: Certificatelessness, F2: Searchable Encryption, F3: Multi-Keyword, F4: Operater, F5: PRE, F6: Validity Period Control, F7: Time Update Able, F8: Quantum Resistance, F9: Security Proof, F10: Offline KGC, F11: Underlying Hard Problem.

6.2 Transmission Cost

Table 4 presents the parameter settings used for the RLWE and LWE comparisons. The parameter settings for RLWE and LWE are based on Bert *et al.* [3] and Albrecht *et al.* [2], respectively. For fairness, we compare the average transmission cost of a single-bit message in each scheme. The results are summarized in Table 5.

Table 3: Comparison Parameters

λ	A security level
n	The number of components in a vector
m	The count of vectors comprising a basis
q	A prime number
σ	A Gaussian parameter
W	Total keywords in the system
N	Number of keywords in a search token

Table 4: Parameter Settings

Hard Problem	λ	n	m	q	σ	W	N
RLWE	40	512	50	2^{31}	3	4	4
LWE	40	128	3202	2^{24}	3	4	4

- Ciphertext size: Our proposed scheme is better than other schemes in terms of size. Specifically, based on RLWE, the proposed scheme has a much smaller ciphertext size of 1581 bits compared to the LWE schemes.
- Token size: Similarly, our proposed scheme also excels in token size, with 466.596 bits. In contrast, the LWE schemes require over 76000 bits.

6.3 Computation Cost

The proposed scheme is based on RLWE, which relies on polynomial operations. In contrast, three LWE-based schemes [14–16] use matrix operations. Because the underlying operations differ, it is challenging to compare the computational costs of these schemes directly based on parameters. Moreover, although Wang *et al.*'s scheme [12] is RLWE-based, it only supports single-keyword search. The proposed scheme employs a more complex architecture to enable multi-keyword search with AND and OR operations. Therefore, compared with Wang *et al.*'s scheme, the proposed scheme naturally incurs higher computational costs across various algorithmic components.

We implemented both the proposed scheme and Zhuang and Fan's scheme [16] to compare their computation and storage costs. Both schemes were executed on Kali Linux version 2024.1, using a 13th Gen Intel(R) Core(TM) i5-13500 processor with 32GB DDR5 RAM.

The proposed scheme incurs higher computational cost during the ***Setup*** phase compared to Zhuang and Fan's scheme [16]. In contrast, the ***KeyGen*** phase exhibits lower computational cost. Given the non-recurring nature of the ***Setup*** phase, its impact on overall performance is expected to be minimal.

Table 6 and Figure 4 show that the proposed scheme achieves faster total encryption time (0.00703 + 0.02830 s) compared to Zhuang and Fan's scheme (0.02978 + 0.0887 s). The difference in message size results in a notable advantage in encryption cost per unit message.

6.4 Storage Cost

Table 7 and Figure 5 show that the proposed scheme reduces storage costs, particularly in key management. Compared with Zhuang and Fan's scheme [16], the proposed scheme reduces the

Table 5: Transmission Cost Comparison

Scheme	Hard Problem	Ciphertext size	Token size
Wang *et al.* [12]	RLWE	1612	1550
Yu *et al.* [14]	LWE	76872	76848
Zhang *et al.* [15]	LWE	76992	76848
Zhuang and Fan [16]	LWE	76872	76848
Ours	RLWE	1581	1550

Unit: bit.

Table 6: Computation Cost Comparison - CPU Time

	[16]	Ours	factor
Setup	0.0777	20.72869	0.00375
KeyGen	1883.73133	908.55589	2.07
ReKeyGen	4821.86115	5710.31802	0.84
Encrypt	0.02978	0.00703125	4.23
Index	0.0887	0.02830	3.14
Re-Encrypt	1.30846	1597.12162	0.00082
TokenGen	2.93950	210.73126	0.0139
Search	0.00069	0.02304	0.03
Decrypt	0.00024	0.00756	0.032

Unit: second.

storage cost of private keys by a factor of 26.35. Similarly, the storage cost for re-encryption keys is reduced by a factor of 4.52.

Regarding ciphertext storage, the proposed scheme achieves a more efficient size, reducing the required storage by a factor of 9.67. Additionally, the storage cost for the index is lowered by a factor of 27.85. Overall, the proposed scheme provides a more efficient and scalable solution for storage compared to Zhuang and Fan's scheme [16].

Table 7: Storage Cost Comparison

	[16]	Ours	Factor
MPK	8089.6	442.5	18.28
MSK	56.9	2867.2	0.0198
Private Key	250982.4	9523.2	26.35
Re-Encryption Key	220160	48742.4	4.52
Ciphertext	104.4	10.8	9.67
Index	629.6	22.6	27.85

Unit: kilobyte (KB).

7 Conclusion

Although lattice-based schemes offer strong security, LWE-based constructions often suffer from large key and ciphertext sizes, limiting their suitability for resource-constrained environments.

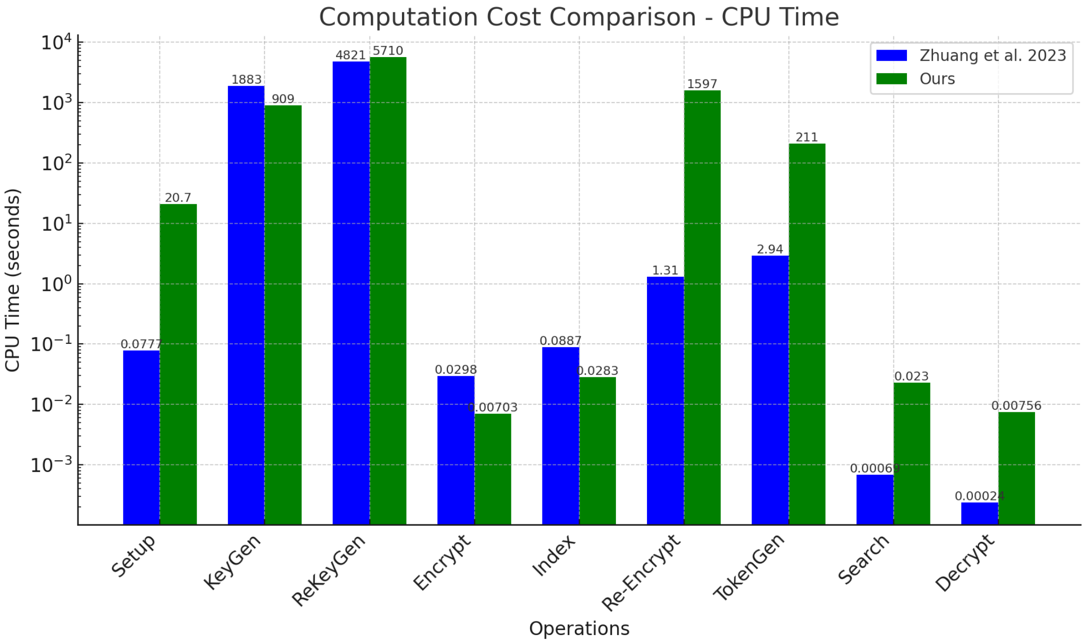

Figure 4: Computation Cost Comparison - CPU Time

RLWE-based approaches address this limitation by leveraging ring structures to reduce ciphertext size while preserving security.

Building upon Zhuang and Fan's scheme [16], we propose MKS-IBPRE, which achieves reduced key and cipertext sizes, improved encryption efficiency, and integrates a time-validity control mechanism for enhanced practicality.

The proposed scheme was implemented and evaluated through comprehensive comparisons with [16]. Results demonstrate notable improvements in storage and computational efficiency. Moreover, the inherent structure of RLWE enables batch encryption of multiple bits, significantly enhancing performance for large-scale data processing.

While the scheme performs well, its computational cost remains substantial due to operations such as modular exponentiation. Future work will explore hardware acceleration techniques, including the use of field-programmable gate arrays (FPGAs), to further improve efficiency. Additionally, we aim to refine the granularity of time control and improve search efficiency by simplifying the underlying algorithm or employing more effective data structures.

8 Acknowledgments

This work was partially supported by the National Science and Technology Council (NSTC) of Taiwan under grants 114-2634-F-110-001-MBK and 114-2221-E-110-040. It also was supported by the MediaTek Advanced Research Center under the research contract MTKC-2024-1133. Additional financial support was provided by the Information Security Research Center at National Sun Yat-sen University in Taiwan and the Intelligent Electronic Commerce Research

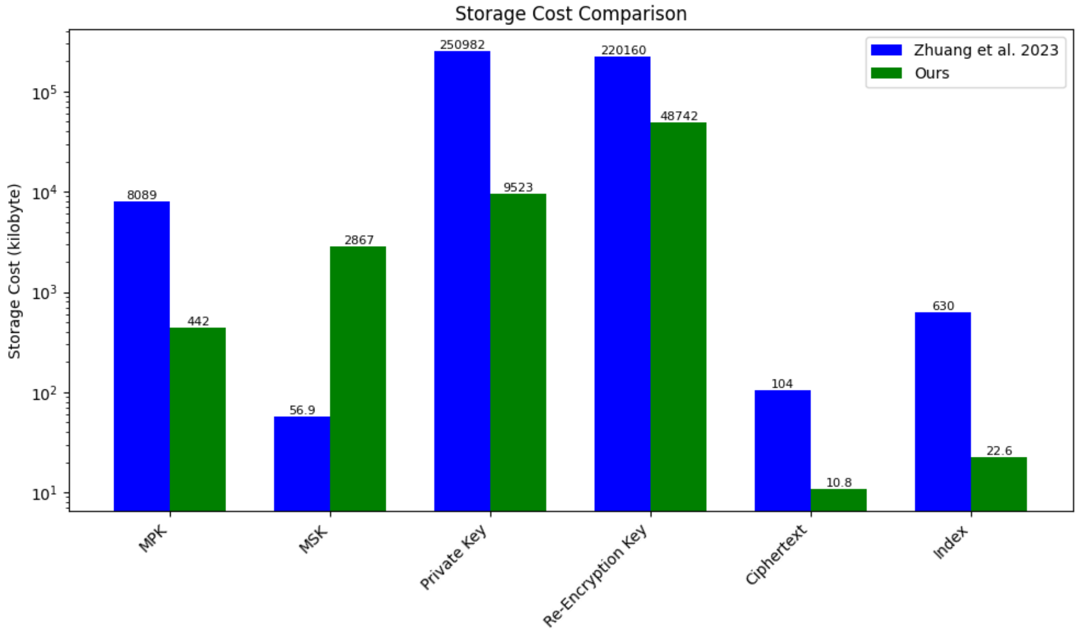

Figure 5: Storage-cost-comparison

Center from the Featured Areas Research Center Program through the Framework of the Higher Education Sprout Project by the Ministry of Education in Taiwan.

References

[1] Miklós Ajtai. Generating hard instances of the short basis problem. In *Automata, Languages and Programming: 26th International Colloquium, ICALP'99 Prague, Czech Republic, July 11–15, 1999 Proceedings 26*, pages 1–9. Springer, 1999.

[2] Martin R Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.

[3] Pauline Bert, Pierre-Alain Fouque, Adeline Roux-Langlois, and Mohamed Sabt. Practical implementation of ring-SIS/LWE based signature and IBE. In *Post-Quantum Cryptography: 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings 9*, pages 271–291. Springer, 2018.

[4] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In *International conference on the theory and applications of cryptographic techniques*, pages 127–144. Springer, 1998.

[5] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23*, pages 506–522. Springer, 2004.

[6] Matthew Green and Giuseppe Ateniese. Identity-based proxy re-encryption. In *Applied Cryptography and Network Security: 5th International Conference, ACNS 2007, Zhuhai, China, June 5-8,*

2007. Proceedings 5, pages 288–306. Springer, 2007.

[7] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29*, pages 1–23. Springer, 2010.

[8] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 700–718. Springer, 2012.

[9] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Journal of the ACM (JACM)*, volume 56, pages 1–40. ACM New York, NY, USA, 2009.

[10] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology: Proceedings of CRYPTO 84 4*, pages 47–53. Springer, 1985.

[11] Dawn Xiaoding Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *Proceeding 2000 IEEE symposium on security and privacy. S&P 2000*, pages 44–55. IEEE, 2000.

[12] Fenghe Wang, Han Xiao, Junquan Wang, Ye Wang, and Chengliang Cao. Efficient secure channel free identity-based searchable encryption schemes with privacy preserving for cloud storage service. *Journal of Systems Architecture*, page 103089, 2024.

[13] Yang Yang and Maode Ma. Conjunctive keyword search with designated tester and timing enabled proxy re-encryption function for e-health clouds. *IEEE Transactions on Information Forensics and Security*, 11(4):746–759, 2015.

[14] Xiaoling Yu, Chungen Xu, Bennian Dou, and Yuntao Wang. Multi-user search on the encrypted multimedia database: lattice-based searchable encryption scheme with time-controlled proxy re-encryption. *Multimedia Tools and Applications*, 80:3193–3211, 2021.

[15] En Zhang, Yingying Hou, and Gongli Li. A lattice-based searchable encryption scheme with the validity period control of files. *Multimedia Tools and Applications*, 80:4655–4672, 2021.

[16] Er-Shuo Zhuang and Chun-I Fan. Multi-keyword searchable identity-based proxy re-encryption from lattices. *Mathematics*, 11(18):3830, 2023.