mathematics

MDPI

Article

Lattice-Based Revocable Certificateless Public Key Encryption for Team Score Orienteering

You Zhao [1], Mingyan Yan [2,*], Kaien Yan [2] and Juyan Li [2,*]

[1] College of Physical Education, Harbin University, Harbin 150090, China; zhaoy@hrbu.edu.cn
[2] College of Data Science and Technology, Heilongjiang University, Harbin 150080, China; 2232660@s.hlju.edu.cn
* Correspondence: 2232681@s.hlju.edu.cn (M.Y.); lijuyan@hlju.edu.cn (J.L.)

Abstract: Team score orienteering, a challenging and interesting sport, is gradually becoming known by the majority of sports enthusiasts. Integrating team score orienteering with the Internet can enhance the interactive experience for athletes. However, this integration increases the risk of the leakage of the athletes' information. In order to protect the privacy of athletes, it is necessary to employ encryption. Therefore, this paper proposes an efficient lattice-based revocable certificateless public key encryption (RCL-PKE) scheme with decryption key exposure resistance (DKER). The adoption of certificateless encryption not only avoids the complex certificate management required for traditional public key encryption, but also addresses the key escrow problem of identity-based encryption, thereby significantly ensuring data security and privacy. Furthermore, the revocable mechanism enables the organizing committee to flexibly manage the athletes' qualification for competitions, and DKER can effectively prevent the leakage of decryption keys, which further enhances data security. The constructed RCL-PKE scheme was proven to be IND-CPA secure under the learning with errors (LWE) assumption. Experiments indicated that the proposed RCL-PKE scheme had lower computation and communication costs, making it particularly suitable for team score orienteering.

Keywords: decryption key exposure resistance; revocable certificateless public key encryption; team score orienteering; lattice

MSC: 68P25; 94A60

Citation: Zhao, Y.; Yan, M.; Yan, K.; Li, J. Lattice-Based Revocable Certificateless Public Key Encryption for Team Score Orienteering. *Mathematics* **2024**, *12*, 1706. https://doi.org/10.3390/math12111706

Academic Editor: Antanas Cenys

Received: 4 May 2024
Revised: 25 May 2024
Accepted: 28 May 2024
Published: 30 May 2024

1. Introduction

Nowadays, the sports industry is closely integrated with the Internet. The advancement of the Internet and the development of big data technology have undoubtedly propelled the sports industry into a new phase. To further improve competition experience, team score orienteering can be combined with Internet technology to achieve the real-time tracking of athletes, score statistics, and other functions in the process of the competition. However, personal data involving athletes contain sensitive information, so it is necessary to encrypt these data to ensure their security.

Although public key encryption solves the key distribution problem of symmetric encryption, the introduction of public key certificates also brings complex certificate management. Shamir et al. [1] proposed an identity-based encryption (IBE), in which the user's identity information is used as a public key. This scheme simplifies the key management process and reduces the cost of using and managing public key certificates. Since the user's private key is entirely generated by a trusted third party, key escrow problems arise. Certificateless public key encryption (CL-PKE) merges the public key encryption with identity-based encryption (IBE), reducing certificate management overheads and resolving the key escrow problem, while improving the security and efficiency of encryption. Additionally, certificateless public key encryption (CL-PKE) can be combined with lattice

encryption to resist quantum computing attacks. Li et al. [2] applied a lattice-based efficient certificateless public key encryption to an EMR cloud storage system to ensure the security of big data. Therefore, a certificateless public key encryption scheme can be applied to team score orienteering, to more effectively protect athletes' personal and competition data. Moreover, considering elimination of athletes for foul play, a revocable mechanism is introduced. This paper constructs a lattice-based revocable certificateless public key encryption for team score orienteering, which can not only protect the security of athletes' sensitive data, but also solve the problems of key escrow and key revocation.

1.1. Research Contributions

For team score orienteering, this paper proposes a lattice-based revocable certificateless public key encryption (RCL-PKE) scheme with DKER based on the learning with errors (LWE) assumption. The contributions are as follows:

(1) A formal definition and IND-CPA security model of RCL-PKE are provided. The RCL-PKE scheme involves three participants, among which KGC can efficiently perform user revocation operations. In the IND-CPA security model, the DKER property is considered, which can resist decryption key leakage attacks.
(2) The first lattice-based RCL-PKE scheme is proposed, which not only has the DKER property, but is also resistant to quantum computing attacks. For three types of adversaries, the proposed RCL-PKE scheme proved to be IND-CPA-secure based on the LWE assumption.
(3) The proposed schemes were compared theoretically and simulated experimentally. Theoretical comparison showed that the proposed scheme is optimal in terms of computation, storage, and communication costs. Simulation results showed that the time required by the proposed scheme increased with the parameter n, but the trend was acceptable.
(4) In order to enrich and optimize appreciation and participation in the competition, team scoring orienteering is integrated with the Internet. The adoption of the RCL-PKE scheme not only strengthens the security of participants' data, but also solves the key escrow problem. In addition, the revocation mechanism allows the organizers to flexibly deprive an athlete of access rights, which improves the fairness of the competition.

1.2. Paper Organization

The rest of this paper is structured as follows: Section 2 describes the related work. Section 3 introduces the preliminaries required to construct the RCL-PKE scheme. Section 4 gives the system model and the security model. Section 5 illustrates the specific construction, correctness, and security of the RCL-PKE scheme. Section 6 analyzes the performance of the proposed RCL-PKE scheme. Section 7 explores a real application of the RCL-PKE scheme, and Section 8 gives the conclusions.

2. Related Work

Certificateless encryption (CLE) solves the certificate management problem of public key encryption and the key escrow problem of identity-based encryption. Adding a revocable mechanism to certificateless encryption scheme not only ensures the security of the data but also allows for more flexible control of the user access rights. In 2013, Shen et al. [3] proposed the first efficient and CCA2-secure revocable certificateless encryption (RCLE) scheme. However, in 2015, Tang et al. [4] found that Shen et al.'s [3] scheme is not secure and the revoked user can still decrypt the ciphertext. In the same year, Sun et al. [5] first proposed a scalable revocable certificateless encryption (RCLE) scheme, which can effectively prevent the threat of decryption key exposure (DKER) and thus ensure the security of encrypted data. Tsai et al. [6] first introduced a revocable certificateless public key encryption (RCL-PKE) scheme that provides a revocable mechanism using a public channel, while keeping the efficiency of encryption and decryption. In 2018, Sun et al. [7]

proposed an IND-CPA secure revocable certificateless encryption scheme under the BDH assumption, ensuring that a revoked user can no longer decrypt past ciphertexts using the previous private key. In 2020, Sun et al. [8] further proposed a revocable certificateless encryption scheme with ciphertext evolution that ensures each user retains only one decryption key. In addition, Zhang et al. [9] proposed a certificateless public data integrity detection scheme for user revocation in cloud storage environments, enhancing the security of cloud storage data and resisting chosen-message attacks. Ma et al. [10] proposed a RCL-PKE scheme with a semi-trusted cloud revocation agent that achieves uniqueness of public keys and flexibility of revocation. Then, in 2022, Tsai et al. [11] used outsourced revocation authority in a certificateless public key system to accomplish the task of revoking a user. In the same year, Tsai et al. [12] first introduced a revocable certificateless public key encryption with equivalence test (RCL-PKEET), which not only revokes illegal users but also maintains the validity of the equivalence test of existing certificateless encryption schemes. Tseng et al. [13] proposed a leakage-resilient revocable certificateless encryption with outsourced revocation authority (LR-RCLE-ORA) scheme for the first time, which revokes compromised users and resists side-channel attacks. Wang et al. [14] proposed a certificateless conditional privacy-preserving authentication (ISC-CPPA) scheme with a revocation mechanism, applying the scheme to ensure data security in the Internet of Vehicles (IoV), where the revocable mechanism can delete the data of a malicious vehicle. In 2023, Tseng et al. [15] first proposed a leakage-resilient anonymous multi-receiver outsourced revocable certificateless encryption (LRAMRORCLE) scheme, which implements the revocation function using an outsourced revocation authority. In 2024, Meng et al. [16] proposed a server-aided traceable and revocable attribute-based encryption (STR-ABKS) scheme based on keyword search.

Revocable encryption schemes, known for their high security and flexibility, are well-suited for data encryption and access control across various scenarios. In recent years, the research on revocable encryption schemes has made significant progress. In 2023, Guo et al. [17] proposed a lattice-based revocable attribute-based encryption (RL-ABE) scheme with a new revocation mechanism that avoids key leakage problem and applies the LWE assumption in a lattice, to resist quantum computing attacks. Additionally, Guo et al. [18] also proposed a new lattice-based traceable and revocable attribute-based encryption (LTR-ABE) scheme, featuring a revocation mechanism that does not require updating the key to achieve revocation. In 2024, Wen et al. [19] first introduced a lattice-based revocable ring signature scheme that can revoke the identity of a vehicle user at any time, enhancing its reliability and efficiency.

To address the problem of decryption key leakage from external attacks or user error, the concept of decryption key exposure resistance (DKER) was developed. DKER ensures that the leakage of a decryption key in any time period does not affect the confidentiality of the ciphertext in other time periods. Katsumata et al. [20] first proposed a revocable identity-based encryption (RIBE) scheme based on a lattice with DKER, which allows an adversary to make a decryption key leakage query. This means that if an adversary $\mathcal{A}$ attempts to obtain the decryption key $dk_{ID^*,t}$ at a particular point in time ($t \neq t^*$), then they must make a secret key reveal query on ID^*, which ID^* will be revoked before t^*. In 2023, Wang et al. [21] proposed a lattice-based RABE scheme applied in a electronic healthcare scenario, which has En-DKER to ensure the confidentiality and privacy of other ciphertexts. Huang et al. [22] proposed a lattice-based ciphertext-policy attribute-based encryption (CP-ABE) scheme with DKER for use in cloud file sharing. In 2024, Wang et al. [23] also introduced enhanced decryption key exposure resistance (En-DKER) in their RIBE scheme, which aims to ensure that the exposure of the decryption key in any time period does not compromise the confidentiality and anonymity of the ciphertext encrypted in different time periods. However, there is no revocable certificateless public key encryption scheme using a lattice. Table 1 presents a comparison of the lattice-based schemes mentioned above.

Table 1. Comparison of lattice-based schemes.

	[17]	[18]	[19]	[20]	[21]	[22]	[23]	Proposed Scheme
Certificateless	×	×	×	×	×	×	×	✓
Revocable	✓	✓	✓	✓	✓	✓	✓	✓
DKER	×	×	×	✓	✓	✓	✓	✓

3. Preliminaries

Lattice For positive integers n, m, and q. An m-dimensional lattice Λ on $\mathbb{Z}^m$ is a set $\left\{\sum_{i=1}^{m} x_i b_i | x_i \in \mathbb{Z}\right\}$ generated by B, where $B = (b_1, \cdots, b_m)$ is the basis of the lattice Λ. For a matrix A in $\mathbb{Z}_q^{n\times m}$ and a vector u in $\mathbb{Z}_q^n$, this lattice can be defined as $\Lambda_q^{\perp}(A) = \{x \in \mathbb{Z}^m : Ax \equiv 0 \bmod q\}$, $\Lambda_q^u(A) = \{x \in \mathbb{Z}^m : Ax \equiv u \bmod q\}$.

Let Λ be an m-dimensional lattice, $c \in \mathbb{R}^m$ be any vector, and $\sigma \in \mathbb{R}$ be any positive parameter. Then, the Gaussian function is defined as $\rho_{\sigma,c} = \exp(-\pi\|x-c\|^2 \Big/ \sigma^2)$ centered at c on $\mathbb{R}^m$. For any $y \in \mathbb{R}^m$ over Λ, and the discrete Gaussian distribution $D_{\Lambda,\sigma,c}(y) = \frac{\rho_{s,c}(y)}{\rho_{\sigma,c}(\Lambda)}$ over Λ, where $\rho_{\sigma,c}(\Lambda) = \sum_{x\in\Lambda}\rho_{\sigma,c}(x)$ is the sum of $\rho_{\sigma,c}$ over Λ.

Definition 1. *For a prime q and a positive integer n, the full rank differences map can be defined as $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$. For all distinct $x, y \in \mathbb{Z}_q^n$, the matrix $H(x) - H(y) \in \mathbb{Z}_q^{n\times n}$ is full rank.*

Lemma 1 ([24,25]). *Let n, m, q be the integers and $n \geq 1, q \geq 2, m = \lceil 2n\log q\rceil$. There exists a PPT algorithm $TrapGen(q,n)$, which produces the output $(A \in \mathbb{Z}_q^{n\times m}, T_A \in \mathbb{Z}_q^{m\times m})$, where A is statistically close to a matrix in $\mathbb{Z}_q^{n\times m}$ and T_A is the basis of $\Lambda_q^{\perp}(A)$ in $\mathbb{Z}_q^{m\times m}$. T_A satisfies $\left\|\widetilde{T_A}\right\| \leq O(\sqrt{n\log q})$, $\|T_A\| \leq O(\sqrt{n\log q})$ with almost negligible probability in n. In particular, there exists a full rank gadget matrix G in $\mathbb{Z}_q^{n\times m}$, such that T_G is the basis of $\Lambda_q^{\perp}(G)$ in $\mathbb{Z}^{m\times m}$, where $\|T_G\|_{GS} \leq \sqrt{5}$.*

Lemma 2 ([20,26]). *For the positive integers n, m, t, q, and $m \geq 2n\lceil\log q\rceil$, there exists a PPT algorithm $SampleLeft(A, E, u, T_A, \sigma) \to c$, which takes as input a full rank matrix A in $\mathbb{Z}_q^{n\times m}$, a matrix E in $\mathbb{Z}_q^{n\times t}$, a vector u in $\mathbb{Z}_q^n$, a basis T_A on $\Lambda_q^{\perp}(A)$ and a Gaussian parameter $\sigma \geq \|T_A\|_{GS} \cdot \omega(\sqrt{\log m})$, then outputs a vector $c \in \mathbb{Z}^{m+t}$, where the distribution of c is statistically close to $D_{\Lambda_q^u([A|E]),\sigma}$.*

Lemma 3 ([20,26]). *Let n, m, t, q be the positive integers, and $m > n$. The PPT algorithms are as follows:*

$ExtRndLeft(A, E, T_A, \sigma) \to T_{[A|E]}$ is a random algorithm that takes as input a full rank matrix A in $\mathbb{Z}_q^{n\times m}$, a matrix E in $\mathbb{Z}_q^{n\times t}$, a basis T_A over $\Lambda_q^{\perp}(A)$ and a Gaussian parameter $\sigma \geq \|T_A\|_{GS} \cdot \omega(\sqrt{\log n})$, and outputs a matrix $T_{[A|E]} \in \mathbb{Z}^{(m+t)\times(m+t)}$ with probability distribution close to $\left(D_{\Lambda_q^u([A|E]),\sigma}\right)^{m+t}$.

$ExtRndRight(A, R, G, T_G, \sigma) \to T_{[A|AG+R]}$ is a random algorithm that takes as input full rank matrix A, G in $\mathbb{Z}_q^{n\times m}$, a basis T_R over $\Lambda_q^{\perp}(R)$, and a Gaussian parameter $\sigma \geq \|G\|_2 \cdot \|T_R\|_2 \cdot \omega(\sqrt{\log n})$, and outputs a matrix $T_{[A|AG+R]} \in \mathbb{Z}^{2m\times 2m}$ with probability distribution close to $\left(D_{\Lambda_q^u([A|AG+R]),\sigma}\right)^{2m}$.

Lemma 4 ([24,27]). *For a prime q and $m > (n+1)logq\omega(logn)$, randomly select the matrices $A, B \leftarrow \mathbb{Z}_q^{n\times m}, R \leftarrow \{-1,1\}^{m\times m} \bmod q$, and the vector $u \in \mathbb{Z}_q^m$, and the distribution of $(A, AR, R^T u)$ is statistically close to the distribution of $(A, B, R^T u)$.*

Theorem 1. *(Learning with Errors) For integers n, m, the prime q, and $\alpha \in (0,1)$ satisfies $\alpha q > 2\sqrt{n}$. The advantage of learning with error $LWE_{n,m,q,D_{Z^m,\alpha q}}$ for any PPT adversary $\mathcal{A}$ is the difference between $Pr\left[\mathcal{A}\left(A, A^T s + x\right) = 1\right]$ and $Pr[\mathcal{A}(A, u + x) = 1]$, where $A \leftarrow \mathbb{Z}_q^{m \times n}$, $s \leftarrow \mathbb{Z}_q^n$, $x \leftarrow \chi^m$, $u \leftarrow \mathbb{Z}_q^m$. The LWE assumption holds if the above advantage is negligible.*

Lemma 5 ([20])**.** *For an m-dimensional lattice Λ defined by a basis T, the Gaussian parameter $\sigma \geq \|T_A\|_{GS} \cdot \omega(\sqrt{\log m})$, and $x \leftarrow D_{\Lambda,\sigma}$, then the probability $\|x\|_2 > \sigma\sqrt{m}$ holds that is less than or equal to $negl(m)$.*

The Binary Tree Structure

The binary tree and complete subtree (CS) method can be efficiently used to update the key for unrevoked users, the key update process includes the following three steps, where the initial state ST is an empty binary tree BT, the root node is rt, the leaf node is L, and the non-leaf node is N. The path from any leaf node L to the root node rt is defined as $\mathrm{Path}(L)$.

(1) Key distribution: When the user registers, KGC randomly selects an empty leaf node L to store the identity of the user and assigns a different set of private keys sk to all nodes on Path (L). The state ST is then updated to reflect the new binary tree.

(2) Key revocation: For the revocation list RL, if the user is revoked at time t, KGC identifies the minimum subset of nodes S, which excludes any ancestor nodes of the revoked user before time t. The leaf nodes that have not been revoked have only one ancestor (or themselves) in the set S. KUNodes algorithm [24] is employed to find the minimum subset of nodes $S = \mathrm{KUNodes}(BT, RL, t)$. Firstly, input the binary tree BT, the revocation list RL, and time t. Secondly, traversing the binary tree BT, marking the ancestor nodes of the user in the revocation list RL as revoked up to time t. Identify the leaf nodes L and non-leaf nodes N that have not been revoked. Finally, output a subset of nodes S that the key requires to be updated.

(3) Key update: KGC publishes key update for all nodes in the subset of nodes S. The update state ST is the updated binary tree.

4. Formal Definition and Security Model

As shown in Figure 1, the RCL-PKE scheme contains three participants: the key generation center (KGC), the data owner, and the data user.

(1) KGC: This is responsible for generating public parameters for the system and partial private keys for the users using the master private key. It maintains the user information in the system, and produces the time update keys at time t according to the revocation list RL and state ST, and broadcasts them across the network.
(2) Data owner: Encrypts the personal information to generate the ciphertext by using the public key and time t disclosed by the data user.
(3) Data user: Generates his/her own decryption key using the time update key broadcast by the KGC and the private key set by himself/herself, and then decrypts the ciphertext data to access the data owner's information. If the data user is revoked by KGC before time t, he/she cannot generate their decryption key according to the time update key broadcast by KGC, thus failing to access the data owner's information.

The formal definition of the RCL-PKE scheme is provided based on Tsai et al. [6], Sun et al. [8], and Katsumata et al. [20]. In the formal definition, the time update key is not tied to the user's ID, but only to the time, which significantly reduces the workload for KGC.

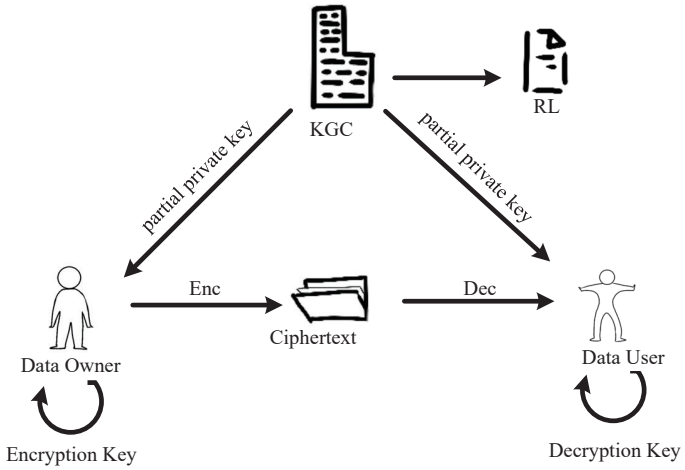

Figure 1. System model of RCL-PKE scheme.

4.1. Formal Definition of RCL-PKE

The RCL-PKE scheme consists of the following seven algorithms:

1. Setup$(\lambda, N) \rightarrow (pp, msk, RL, ST)$: Input security parameter λ and the total number of system users N. The KGC has the public parameters pp, retains the system master private key msk, the revocation list RL, and the state ST.
2. Extractppk$(pp, ID, msk, RL, ST) \rightarrow D_{ID}$: Input the public parameters pp, the user identity ID, the master private key msk, the revocation list RL, and the state ST. KGC generates the partial private key D_{ID} for the user ID and secretly sends it to the user ID.
3. Setkey$(pp, ID, D_{ID}) \rightarrow (sk_{ID}, pk_{ID})$: Input the public parameters pp, the user identity ID, and partial private key D_{ID}. User ID selects the secret value SV_{ID} and generates his/her own public key pk_{ID} and private key sk_{ID}. The public key pk_{ID} is publicized, while the private key sk_{ID} is kept private.
4. UpdateTK$(pp, t, msk, RL, ST) \rightarrow TK_t$: Input the public parameters pp, time t, the master private key msk, the revocation list RL, and the state ST. KGC outputs and broadcasts the time update key TK_t across the network.
5. Enc$(pp, ID, t, m, pk_{ID}) \rightarrow ct_{ID,t}$: Input the public parameters pp, the user identity ID, time t, the public key pk_{ID}, and message m. The data owner encrypts the message m to generate the ciphertext $ct_{ID,t}$ of user ID at time t.
6. GenDK$(pp, sk_{ID}, TK_t) \rightarrow dk_{ID,t}$: Input the public parameters pp, the private key sk_{ID}, and the time update key TK_t. The user ID generates his/her own decryption key $dk_{ID,t}$.
7. Dec$(pp, dk_{ID,t}, ct_{ID,t}) \rightarrow m$: Input the public parameters pp, the decryption key $dk_{ID,t}$, and the ciphertext $ct_{ID,t}$. The data user ID decrypts $ct_{ID,t}$ to obtain message m.

4.2. Security Model

Based on Tsai et al. [6], Sun et al. [8], and Katsumata et al. [20], the IND-CPA security under a choice of identity and time is considered, i.e., the adversary sends the identity and time (ID^*, t^*) of the challenge target to the challenger before the game begins. In the security game, the adversary can access the following oracle.

- $\mathcal{O}^{PPK}$—(partial private key oracle) The adversary inputs the user identity ID and the challenger $\mathcal{C}$ searches in the table Tb. If the partial private key D_{ID} exists, return to $\mathcal{A}$. Otherwise, $\mathcal{C}$ runs Extractppk$(pp, ID, msk, RL, ST) \rightarrow D_{ID}$, adds D_{ID} to Tb and returns to $\mathcal{A}$.
- $\mathcal{O}^{SV}$—(secret value oracle) The adversary inputs the user identity ID, and the challenger $\mathcal{C}$ searches in the table Tb. If the secret value SV_{ID} exists, returns SV_{ID} to $\mathcal{A}$. Otherwise, $\mathcal{C}$ runs Setkey$(pp, ID, D_{ID}) \rightarrow (sk_{ID}, pk_{ID})$, adds (sk_{ID}, pk_{ID}) to Tb, and returns SV_{ID} to $\mathcal{A}$.

- $\mathcal{O}^{PK}$—(public key oracle) The adversary inputs the user identity ID, and the challenger $\mathcal{C}$ searches in the table Tb. Then, $\mathcal{C}$ returns the public key pk_{ID} to $\mathcal{A}$, if pk_{ID} exists. Otherwise, $\mathcal{C}$ runs Setkey$(pp, ID, D_{ID}) \rightarrow (sk_{ID}, pk_{ID})$, adds (sk_{ID}, pk_{ID}) to Tb, and returns pk_{ID} to $\mathcal{A}$.
- $\mathcal{O}^{PKR}$—(public key replacement oracle) The adversary inputs the user identity ID and a new public key pk'_{ID}, and the challenger $\mathcal{C}$ replaces pk_{ID} in the table Tb with pk'_{ID}.
- $\mathcal{O}^{SK}$—(secret key oracle) The adversary inputs the user identity ID. If the private key sk_{ID} does not exist, the challenger $\mathcal{C}$ runs Setkey$(pp, ID, D_{ID}) \rightarrow (sk_{ID}, pk_{ID})$ and adds sk_{ID} to the table Tb.
- $\mathcal{O}^{SKR}$—(secret key reveal oracle) The adversary inputs the user identity ID, and the challenger $\mathcal{C}$ searches the corresponding sk_{ID} from the table Tb and returns it to $\mathcal{A}$.
- $\mathcal{O}^{TK}$—(revoke and key update oracle) The adversary inputs the revocation list RL of time t_{cu}. The challenger $\mathcal{C}$ searches in the table Tb and returns $TK_{t_{cu}}$ to the adversary if the time update key $TK_{t_{cu}}$ exists. Otherwise, $\mathcal{C}$ runs UpdateTK$(pp, t, msk, RL, ST) \rightarrow TK_t$, adds it to the table Tb, and returns it to $\mathcal{A}$.
- $\mathcal{O}^{DKR}$—(decryption key reveal oracle). The adversary inputs the identity and time (ID, t). The challenger $\mathcal{C}$ searches in the table Tb, and if the decryption key $dk_{ID,t}$ exists, returns it to $\mathcal{A}$. Otherwise, $\mathcal{C}$ runs GenDK$(pp, sk_{ID}, TK_t) \rightarrow dk_{ID,t}$, adds it to the table Tb, and returns it to $\mathcal{A}$.

To enhance the security of the scheme, the adversary is permitted to access the secret values of the user. Consequently, there are three types of adversaries in the security game: (1) An honest but curious KGC adversary $\mathcal{A}_I$, since $\mathcal{A}_I$ possesses the master private key and has access to the user's secret values, which can replace the public keys of all users except the target identity. (2) Adversary $\mathcal{A}_{II}$ is able to replace the public keys of all users, but does not initiate the private key query to ID^*. (3) Adversary $\mathcal{A}_{III}$ is able to replace the public keys of all users and initiates a private key query to ID^*.

The IND-CPA security of the RCL-PKE scheme is defined through a game between an adversary $\mathcal{A} \in \{\mathcal{A}_I, \mathcal{A}_{II}, \mathcal{A}_{III}\}$ and a challenger $\mathcal{C}$. A global parameter t_{cu} is defined with an initial value 1, which represents the "current time period" that controls the response of $\mathcal{C}$ to the query from $\mathcal{A}$.

Initialization. The adversary $\mathcal{A}$ sends the challenge identity/time (ID^*, t^*) to the challenger $\mathcal{C}$, the challenger $\mathcal{C}$ generates $(pp, msk, RL, ST) \leftarrow Setup(\lambda, N)$, $TK_1 \leftarrow UpdateTK(pp, msk, RL = \emptyset, ST, t_{cu} = 1)$ and then sends pp and TK_1 to $\mathcal{A}$. If $\mathcal{A} = \mathcal{A}_{II}$, send msk to $\mathcal{A}_{II}$. Otherwise, keep msk.

Phase 1. $\mathcal{A}$ has polynomial time access to the oracle $\mathcal{O}^{PPK}, \mathcal{O}^{SV}, \mathcal{O}^{PK}, \mathcal{O}^{PKR}, \mathcal{O}^{SK}, \mathcal{O}^{SKR}, \mathcal{O}^{TK}, \mathcal{O}^{DKR}$. The limitations are as follows:

(1) If $\mathcal{A} = \mathcal{A}_I$, $\mathcal{O}^{PKR}(ID^*)$ cannot be accessed and the secret value of ID^* cannot be queried.
(2) If $\mathcal{A} = \mathcal{A}_{II}$, the public key of ID^* is replaced with a valid public key, the partial private key of ID^* cannot be queried.
(3) If $\mathcal{A} = \mathcal{A}_{III}$, the partial private key of ID^* has been queried, ID^* must be revoked before time t^*.

Challenge. $\mathcal{A}$ sends two messages m_0, m_1 to $\mathcal{C}$, performing the following steps:

(1) If $\mathcal{A} = \mathcal{A}_{II}$, and the public key corresponding to ID^* is replaced with an invalid public key, the game ends with $\mathcal{A}$ failing.
(2) Return $\perp$, if $\mathcal{O}^{DKR}(ID^*, t^*)$ was queried.
(3) Otherwise, $\mathcal{C}$ chooses $b \leftarrow \{0, 1\}$, computes $ct^* \leftarrow Enc(pp, ID^*, t^*, m_b, pk_{ID^*})$, and returns ct^* to $\mathcal{A}$.

Phase 2 is the same as Phase 1

Guess. $\mathcal{A}$ outputs his/her guess b'.

If $b' = b$, then this shows that $\mathcal{A}$ wins the game. The advantage of $\mathcal{A}$ winning the game is defined as $\varepsilon = 2\left|\Pr(b' = b) - \frac{1}{2}\right|$. If ε is negligible for any PPT adversary $\mathcal{A}$, then the RCL-PKE scheme is IND-CPA secure.

Remark 1. *Since in the RCL-PKE scheme, the time update key TK_t is not tied to the user's identity ID but is broadcast across the network by the KGC, any user can receive TK_t. Therefore, if the adversary accesses the private key of ID^*, then ID^* must be revoked before t^*. Otherwise, the adversary obtains the decryption key of ID^*.*

Remark 2. *Similarly to the security model of [20], it is known that, since the security model defined in this paper contains a decryption key reveal query, the scheme captures the decryption key exposure resistance (DKER).*

5. RCL-PKE from Lattices

5.1. Construction

1. Setup$(\lambda, N) \to (pp, msk, RL, ST)$: Input the security parameter λ, the total number of system users N. The KGC performs the following operations:
   (1) Generate $(A, T_A), (\overline{A}, T_{\overline{A}}) \leftarrow \text{TrapGen}(1^n, 1^m, q)$
   (2) Randomly select $B_1, B_2 \leftarrow \mathbb{Z}_q^{n\times m}, \mu \leftarrow \mathbb{Z}_q^n$, and full rank differences map $\text{H} : \{0,1\}^* \to \mathbb{Z}_q^{n\times n}$
   (3) Select a complete binary tree BT containing at least N leaf nodes, such that RL is an initially empty set, and let the revocation list be RL.
   (4) Output the public parameter $pp = \{A, \overline{A}, H, B_1, B_2, \mu\}$, the master private key $msk = \{T_A, T_{\overline{A}}\}$, RL and BT.
2. Extractppk$(pp, ID, msk, RL, ST) \to D_{ID}$: Input pp, ID, msk, RL, ST. The KGC performs the following operations:
   (1) Randomly select an empty leaf node v in BT and store ID in v.
   (2) For any $\theta \in \text{Path}(v)$, if μ_θ does not exist, then randomly select $\mu_\theta \leftarrow \mathbb{Z}_q^n$ and store it in node θ. Sample $d_{ID}^\theta \leftarrow \text{SampleLeft}(A, E(ID), \mu_\theta, T_A, \sigma)$, where $[A|E(ID)]d_{ID}^\theta = \mu_\theta$.
   (3) Generate $T_{(\overline{A}|E(ID))} \leftarrow \text{ExtRndLeft}(\overline{A}, E(ID), T_{\overline{A}}, \sigma)$.
   (4) Output the partial private keys $D_{ID} = (\{d_{ID}^\theta\}_{\theta\in Path(v)}, T_{(\overline{A}|E(ID))})$ and ST.
3. Setkey$(pp, ID, D_{ID}) \to (sk_{ID}, pk_{ID})$: Input pp, D_{ID}. User ID selects $B \leftarrow \mathbb{Z}_q^{n\times m}, x \leftarrow \chi^n, e_1 \leftarrow \chi^m$, computes $b = B^T x + 2e_1$, and outputs $pk_{ID} = (b, B), sk_{ID} = (x, D_{ID})$.
4. UpdateTK$(pp, t, msk, RL, ST) \to TK_t$: Input pp, t, msk, RL, ST. The KGC performs the following operations:
   (1) For $\forall_\theta \in \text{KUNodes}(\text{BT}, \text{RL}, \text{t})$, if μ_θ does not exist, randomly pick $\mu_\theta \leftarrow \mathbb{Z}_q^n$ and store it in node θ. Sample $d_t^\theta \leftarrow \text{SampleLeft}(A, E(t), \mu - \mu_\theta, T_A, \sigma)$, where $[A|E(t)]d_t^\theta = \mu - \mu_\theta$.
   (2) Output $TK_t = (\{d_t^\theta\}_{\theta\in\text{KUNodes(BT,RL,t)}})$.
5. Enc$(pp, ID, t, m, pk_{ID}) \to ct_{ID,t}$: Input pp, t, m, and the public key pk for user ID. The user selects $R_i \leftarrow \{-1,1\}^{m\times m}, i = 1, 2, m, r \leftarrow \{0,1\}^m, s_1, s_2 \leftarrow \chi^n, e_2 \leftarrow \chi^m, e_3 \leftarrow \chi^m, e \leftarrow \chi^m$, and computes the

$$\begin{aligned} C_0 &= \mu^T(s_1 + s_2) + 2e + m + b^T r \\ C_1 &= Br \\ C_2 &= [A|E(ID)|E(t)]^T s_1 + 2[e_2, R_1^T e_2, R_2^T e_2] \\ C_3 &= [\overline{A}|E(ID)|E(t)]^T s_2 + 2[e_3, R_1^T e_3, R_2^T e_3] \end{aligned}$$

   Output the ciphertext $ct_{ID,t} = (C_0, C_1, C_2, C_3)$, where $E(ID) = B_1 + H(ID)G, E(t) = B_2 + H(t)G$, and G is the gadget matrix.
6. GenDK$(pp, sk_{\text{ID}}, TK_t) \to dk_{ID,t}$: Input $pp, sk_{ID} = (x, T_{(\overline{A}|E(ID))}, \{d_{ID,\theta}\}_{\theta\in I}), TK_t = (\{d_{t,\theta}\}_{\theta\in J})$.

(1) If $I \bigcap J = \phi$, then let $dk_{ID,t} = \perp$. If $I \bigcap J \neq \phi$, then for $\forall_\theta \in I \bigcap J$, let $d_{ID,\theta} = \begin{pmatrix} d_{ID}^{\theta,1} \\ d_{ID}^{\theta,2} \end{pmatrix}$, $d_{t,\theta} = \begin{pmatrix} d_t^{\theta,1} \\ d_t^{\theta,2} \end{pmatrix}$, and compute $d_{ID,t}^{\theta} = \begin{pmatrix} d_{ID}^{\theta,1} + d_t^{\theta,1} \\ d_{ID}^{\theta,2} \\ d_t^{\theta,2} \end{pmatrix}$.

(2) Sample $\overline{d_{ID,t}} \leftarrow \text{SampleLeft}\left[\overline{A}|E(ID), E(t), \mu, T_{\overline{A}|E(ID)}, \sigma\right]$, where $\left[\overline{A}|E(ID)|E(t)\right]\overline{d_{ID,t}} = \mu$.

(3) Output $dk_{ID,t} = (x, \overline{d_{ID,t}}, \left\{d_{ID,t}^{\theta}\right\}_{\theta \in I \cap J})$

7. $\text{Dec}(pp, dk_{ID,t}, ct_{ID,t}) \rightarrow m$:
Input $pp, dk_{ID,t}, ct_{ID,t}$, and compute $m = \left[C_0 - x^T C_1 - \overline{d_{ID,t}}^T C_3 - {d_{ID,t}^{\theta}}^T C_2\right] \bmod 2$.

For correct decryption, we can set $n = \lambda$, $m = 2n\lceil \log q \rceil$, $\chi = D_{z,\alpha_q}$, $\sigma = m\omega(\sqrt{\log m})$, $\alpha^2 < \frac{1}{32qm\omega(\sqrt{\log n})}$. See Appendix A for details.

Remark 3. *The proposed scheme can solve the problems of public key certificate management in the traditional public key infrastructure (PKI) and secret key escrow in identity-based encryption schemes. (1) Unlike the traditional PKI, users no longer need to verify the authenticity of public keys through certificates. In the proposed scheme, because the KGC knows the master private key* $msk = \left\{T_A, T_{\overline{A}}\right\}$*, the partial private keys* D_{ID} *of the user can be generated according to their identity ID, thus eliminating the dependence on the certificate. During encryption, the identity ID needs to be bound to the public key* $pk_{ID} = (b, B)$ *set by the user. This step no longer requires that the certificate maintained by the KGC matches the public key, which greatly simplifies the complexity of certificate management. (2) Although the KGC can generate partial private keys* D_{ID} *and time update keys* TK_t *for users, the KGC cannot obtain the user's decryption key* $Dk_{ID,t}$ *because the KGC does not know the secret value x set by the user, thus solving the key custody problem. (3) Users can be revoked, because the KGC maintains a revocation list RL and periodically updates and broadcasts the time update key* TK_t*. If a user is added to the revocation list RL at a specific time t, he/she will not be able to obtain a valid time update key* TK_t *for that time t to generate his/her own decryption key, thereby losing access. Therefore, the proposed RCL-PKE scheme significantly reduces the dependence on certificates and improves security.*

5.2. Security

In this section, the proposed scheme is proven to be secure with respect to each of the three types of adversaries, and thus it follows that the constructed RCL-PKE scheme is IND-CPA secure.

Theorem 2. *Let* $\mathcal{A} = \mathcal{A}_I$*, then the above RCL-PKE scheme is IND-CPA secure under the LWE assumption.*

Proof. $Game_0^{\text{I}}$. The game is the same as the secure game.

$Game_1^I$. The game is the same as $Game_0^I$, except that pk_{ID^*} is generated differently. In $Game_1^I$, randomly select $b_{ID}^* \leftarrow \mathbb{Z}_q^m$ to replace the original b_{ID}.

Since $\mathcal{A} = \mathcal{A}_I$ possesses a master private key msk, $\mathcal{A}$ can generate his/her own partial private key D_{ID} and the time update key TK_t. However, $\mathcal{A}$ cannot access the secret value of ID^*. Under the LWE assumption, (b_{ID}, B_{ID}) is computationally indistinguishable from (b_{ID}^*, B_{ID}), so $Game_0$ is indistinguishable from $Game_1$.

$Game_2^I$. The game is the same as $Game_1^I$, except that the challenge ciphertexts are generated in different ways, randomly selecting $b \leftarrow \{0, 1\}$, $\alpha \leftarrow \mathbb{Z}_q$, $\beta \leftarrow \mathbb{Z}_q^n$. Compute the challenge ciphertext $C_0 = \mu^T(s_1 + s_2) + 2e + \alpha + m_b$, $C_1 = \beta$ and C_2, C_3 are the same as $Game_1$.

From the leftover hash lemma, we see that α is statistically indistinguishable from $b_{ID}^* r$, β is statistically indistinguishable from $B_{ID} r$, and thus $Game_1$ is indistinguishable from $Game_2$, because α is a random uniform distribution on $\mathbb{Z}_q$, and is independent of other

ciphertext elements. Therefore, the adversary's advantage in winning $Game_2$ is negligible. Finally, the theorem holds. □

Theorem 3. *Let $\mathcal{A} = \mathcal{A}_{II}$, then the above RCL-PKE scheme is IND-CPA secure under the LWE assumption.*

Proof. $Game_0^{II}$. The game is the same as the secure game.

$Game_1^{II}$. The game is the same as $Game_0^{II}$ except that B_1, B_2 are generated differently. In $Game_1^{II}$, $\mathcal{C}$ selects $R_j^* \leftarrow \{0,1\}^{m\times m}, j = 1,2$, computes $B_1 = \overline{A}R_1^* - H(ID^*)G$, $B_2 = \overline{A}R_2^* - H(t^*)G$, and retains R_j^*.

From the leftover hash lemma, the advantage of $\mathcal{A}$ in distinguishing between $Game_0^{II}$ and $Game_1^{II}$ is negligible.

$Game_2^{II}$. The game is the same as $Game_1^{II}$, except that $\overline{A}$ is generated differently. In $Game_2^{II}$, $\mathcal{C}$ randomly selects $\overline{A} \leftarrow \mathbb{Z}_q^{n\times m}$. Since $\mathcal{C}$ does not possess the trapdoor $T_{\overline{A}}$, and $\mathcal{C}$ needs to simulate the items generated by $T_{\overline{A}}$ in $Game_1^{II}$, such as $T_{\left[\overline{A}|E(ID)\right]}, ID \neq ID^*$ and $\overline{d_{ID,t}}$, where $(ID,t) \neq (ID^*,t^*), t \leq t_{cu}, ID \notin RL_t$.

Since $\left[\overline{A}|E(ID)\right] = \left[\overline{A}|\overline{A}R_1^* + (H(ID) - H(ID^*))G\right]$, if $ID \neq ID^*$, $\mathcal{C}$ can use T_G and $ExtRndRight$ algorithms to obtain $T_{\left[\overline{A}|\overline{A}R_1^* + (H(ID)-H(ID^*))G\right]}$, then it can use $SampleLeft$ algorithms to obtain $\overline{d_{ID,t}}$.

$Game_3^{II}$. This game is the same as $Game_2^{II}$, except that the ciphertexts are generated in different ways. In $Game_3^{II}$, $\mathcal{C}$ selects $b \leftarrow \{0,1\}, s_1, s_2 \leftarrow \chi^n, x \leftarrow \chi, \overline{x} \leftarrow \chi^m, e_2 \leftarrow \chi^m$. Let $\alpha = \mu^T s_2 + 2x, \beta = \overline{A}^T s_2 + 2\overline{x}$, and computes

$$\begin{aligned} C_0 &= \mu^T s_1 + \alpha + b^T r + m_b \\ C_1 &= Br \\ C_2 &= [A|E(ID^*)|E(t^*)]^T s_1 + \left[e_2, R_1^{*T} e_2, R_2^{*T} e_2\right] \\ C_3 &= \left[\beta, R_1^{*T}\beta, R_2^{*T}\beta\right] \end{aligned}$$

Output the ciphertext $ct^* = (C_0, C_1, C_2, C_3)$

Because in $Game_2^{II}$

$$\begin{aligned} C_3 &= \left[\overline{A}|E(ID^*)|E(t^*)\right]^T s_2 + 2\left[e_3, R_1^{*T} e_3, R_2^{*T} e_3\right] \\ &= \left[\overline{A}|\overline{A}R_1^{*T}|\overline{A}R_2^{*T}\right]^T s_2 + 2\left[e_3, R_1^{*T} e_3, R_2^{*T} e_3\right] \\ &= \left[\overline{A}s_2 + 2e_3, R_1^{*T}(\overline{A}s_2 + 2e_3), R_2^{*T}(\overline{A}s_2 + 2e_3)\right] \end{aligned}$$

The advantage of the adversary in distinguishing between $Game_2^{II}$ and $Game_3^{II}$ is negligible.

$Game_4^{II}$. The game is the same as $Game_3^{II}$, except that the ciphertext is generated differently. In $Game_4^{II}$, $\mathcal{C}$ selects $b \leftarrow \{0,1\}, s_1 \leftarrow \chi^n, \omega \leftarrow Z_q, W \leftarrow Z_q^m, x \leftarrow \chi, e_2 \leftarrow \chi^m$. Let $\alpha = \omega + 2x$, $\beta = W$, and C_0, C_1, C_2, C_3 are the same as $Game_3^{II}$.

From LWE, the advantage of the adversary in distinguishing between $Game_3^{II}$ and $Game_4^{II}$ is negligible. Since α is a random uniform distribution on $\mathbb{Z}_q$, and is independent of other ciphertext elements. Therefore, the adversary's advantage of winning $Game_4^{II}$ is negligible. Finally, the theorem holds. □

Theorem 4. *Let $\mathcal{A} = \mathcal{A}_{III}$, then the above RCL-PKE scheme is IND-CPA secure under the LWE assumption.*

Proof. $Game_0^{III}$. The game is the same as the secure game. $Game_1^{III}$. The game is the same as $Game_0^{III}$, except that B_1, B_2 are generated differently. In $Game_1$, $\mathcal{C}$ randomly selects $R_j^* \leftarrow \{0,1\}^{m\times m}, j = 1,2$, and lets $B_1 = AR_1^* - H(ID^*)G, B_2 = AR_2^* - H(t^*)G$.

From the leftover hash lemma, $Game_0^{III}$ is indistinguishable from $Game_1^{III}$.

$Game_2^{III}$. The game is the same as $Game_1^{III}$, except that the binary tree BT is generated differently and the leaf nodes are selected differently. The challenger creates an empty binary tree BT, then chooses a random leaf node η_{ID^*} to place ID^*, and finally sends BT to $\mathcal{A}$.

Because the creation of BT is only a conceptual manner, and the storage leaf position of ID^* is hidden from $\mathcal{A}$, so $\mathcal{A}$ cannot distinguish between $Game_1^{III}$ and $Game_2^{III}$.

$Game_3^{III}$. The game is the same as $Game_2^{III}$, except that the storage generation of μ_θ in BT for some nodes V is different. Since $\mathcal{A} = \mathcal{A}_{III}$, $\mathcal{A}$ accesses the private key of ID^*, ID^* must be revoked before the time of t^*. It is known that $S_{path} \bigcap S_{TK,t^*} = \phi$, where $S_{path} = Path(BT, V_{ID^*})$, $S_{TK,t^*} = \mathrm{KUNodes(BT, RL, t^*)}$. When $\mathcal{A}$ initiates a D_{ID^*} (or TK_{t^*}) query, $\mathcal{C}$ selects $d_{ID^*}^\theta \leftarrow D_{Z^2,\sigma}$ (or $d_{t^*}^\theta \leftarrow D_{Z^2,\sigma}$), and computes $[\mathrm{A}|E(ID)]d_{ID^*}^\theta = \mu_\theta, \theta \in S_{path}$ (or $\mu - [A|E(ID)]d_{\mathrm{t}^*}^\theta = \mu_\theta, \theta \in S_{TK,t^*}$). The corresponding μ_θ is stored in the node v and retains $d_{ID^*}^\theta$ ($d_{t^*}^\theta$).

From Lemma 3, $Game_2^{III}$ is indistinguishable from $Game_3^{III}$.

$Game_4^{III}$. The game is the same as $Game_3^{III}$, except that A is generated differently. In $Game_4^{III}$, randomly select $A \leftarrow \mathbb{Z}_q^{n\times m}$, so $\mathcal{C}$ does not possess T_A. When $\mathcal{A}$ initiates $\mathcal{O}^{PPK}(ID)$ and $\mathcal{O}^{TK}(t)$ queries, if $ID = ID^*$ or $t = t^*$, return d_{ID}^θ stored in $Game_3$.

If $ID \neq ID^*$,there is $[A|E(ID)] = [A|B_1 + H(ID)G] = \left[A|AR_1^* + (H(ID) - H(ID^*))G\right]$, utilizing T_G to run the *SampleRight* algorithm to get d_{ID}^θ, and then using the *ExRndRight* algorithm to get $T_{\left[\mathrm{A}|AR_1^*+(H(ID)-H(ID^*))G\right]}$.

Similarly, if $\mathrm{t} \neq \mathrm{t}^*$, there is $[A|E(t)] = [A|AR_2^* + (H(t) - H(t^*))G]$, using T_G to run the *ExRndRight* algorithm to get $T_{\left[A|AR_2^*+(H(t)-H(t^*))G\right]}$.

From Lemma 3, $Game_3^{III}$ is indistinguishable from $Game_4^{III}$.

$Game_5^{III}$. The game is the same as $Game_4^{III}$, except that the ciphertext is generated differently. In $Game_5^{III}$, $\mathcal{C}$ selects $b \leftarrow \{0,1\}$, $s_1, s_2 \leftarrow \chi^n, x, \leftarrow \chi, \overline{x} \leftarrow \chi^m, e_3 \leftarrow \chi^m$. Let $\alpha = \mu^T s_1 + 2x, \beta = A^T s_1 + 2\overline{x}$, and compute

$$\begin{aligned} C_0 &= \mu^T s_2 + \alpha + b^T r + m_b \\ C_1 &= Br \\ C_2 &= \left[\beta, R_1^{*T}\beta, R_2^{*T}\beta\right] \\ C_3 &= \left[\overline{A}|E(ID^*)|E(t^*)\right]^T s_2 + \left[e_3, R_1^{*T}e_3, R_2^{*T}e_3\right] \end{aligned}$$

Output the ciphertext $ct^* = (C_0, C_1, C_2, C_3)$.

Because in $Game_4^{III}$

$$\begin{aligned} C_2 &= [A|E(ID^*)|E(t^*)]^T s_2 + 2\left[e_2, R_1^{*T}e_2, R_2^{*T}e_2\right] \\ &= \left[A|AR_1^{*T}|\mathrm{A}R_2^{*T}\right]^T s_2 + 2\left[e_2, R_1^{*T}e_2, R_2^{*T}e_2\right] \\ &= \left[As_2 + 2e_2, R_1^{*T}(\mathrm{A}s_1 + 2e_2), R_2^{*T}(As_1 + 2e_2)\right] \end{aligned}$$

From the leftover hash lemma, the advantage of an adversary in distinguishing between Game_4^{III} and $Game_5^{III}$ is negligible.

$Game_6^{III}$. The game is the same as $Game_5^{III}$, except that the ciphertext is generated differently. In $Game_6^{III}$, $\mathcal{C}$ selects $b \leftarrow \{0,1\}$, $s_2 \leftarrow \chi^n, \omega \leftarrow \mathrm{Z}_q, W \leftarrow \mathrm{Z}_q^n, x \leftarrow \chi, e_3 \leftarrow \chi^m$. Let $\alpha = \omega + 2x$, $\beta = W$, and C_0, C_1, C_2, C_3 are the same as $Game_3^{II}$.

The advantage of the adversary in distinguishing between $Game_5^{III}$ and $Game_6^{III}$ is negligible using the LWE assumption. Since α is a random uniform distribution on $\mathbb{Z}_q$ and is independent of other ciphertext elements. Therefore, the adversary's advantage of winning $Game_6^{III}$ is negligible. Finally, the theorem holds. □

Remark 4. *Since the ciphertext $CT_{ID,t}$ of user ID is not only associated with its public key pk_{ID}, but also related to a specific time t, if the user ID wants to decrypt the ciphertext $CT_{ID,t}$, user ID must obtain the decryption key $DK_{ID,t}$ corresponding to the time t. The decryption key is generated using the user's private key SK_{ID} and the time update key TK_t at time t. Only when the user ID is not revoked at time t can he obtain the time update key TK_t at time t, generate the decryption key $DK_{ID,t}$, and decrypt the ciphertext $CT_{ID,t}$. In our security model, the adversary may access the decryption key reveal oracle, so the scheme has the DKER property, which guarantees that even if the user's decryption key is disclosed at a certain time, the user's private key cannot be calculated from it. Therefore, the decryption key of the other time cannot be calculated; that is, the security of the ciphertext encrypted in the other time cannot be affected. Therefore, our scheme ensures both forward and backward security: even if the adversary obtains the private key of the user ID at time t, he cannot decrypt the ciphertext before (backward secure) or after (forward secure) time t.*

6. Performance

In this section, the proposed RCL-PKE scheme is compared in terms of space and computational costs with Wang et al.'s [21] scheme and Huang et al.'s [22] scheme through theoretical analysis. Subsequently, the performance of the constructed scheme is further evaluated using simulation experiments.

6.1. Theoretical Evaluation

Currently, there are no RCL-PKE schemes using a lattice, and the proposed scheme is the first lattice-based revocable certificateless public key encryption. Both the revocable attribute-based encryption (RABE) schemes proposed by Wang et al. [21] and Huang et al. [22] and the proposed scheme are lattice-based revocable encryption schemes. In the following, the proposed scheme is compared in terms of space and computational costs with those of Wang et al. [21] and Huang et al. [22].

6.1.1. Space Costs

The space costs are compared in terms of private key size, decryption key size, and ciphertext size. The schemes proposed by Wang et al. [21] and Huang et al. [22] encrypt 1-bit plaintext at a time, while the proposed scheme encrypts m-bit plaintext at a time. For comparison purposes, the average 1-bit of the proposed scheme is computed. Meanwhile, let $k = \log n$, $q = 2^{\sqrt{n}}$, and $m = 2n\lceil \log q \rceil$, $i \in [l]$, $\theta \in \mathrm{KUNodes}(\mathrm{RL_t})$. In scheme [22], N is the number of attribute authorities, l_s is the total number of attributes, where $N = 3$, and $l_s = 1$.

As shown in Table 2, when encrypting 1-bit plaintext, the private key size of the proposed scheme is $\frac{n}{m} + 2n + 2$, while Wang et al.'s [21] scheme has a private key size of $3m \times m$, Huang et al.'s [22] scheme has a private key size of $\frac{1}{2}(l_s + 3N + 1) \times m$. Similarly, the decryption key size of the proposed scheme is $\frac{n}{m} + 6$, Wang et al.'s [21] scheme has a decryption key size of $4m \times k$, Huang et al.'s [22] scheme has a decryption key size of $\frac{1}{2}(l_s + 3N) \times m$. Additionally, the ciphertext size of the proposed scheme is $\frac{n}{m} + 7$, Wang et al.'s [21] scheme has a ciphertext size of $2m + m \times i + m \times \theta + k$, Huang et al.'s [22] scheme has a ciphertext size of $(l_s + N) \times m$. Therefore, it can be concluded that the space costs of the proposed scheme are smaller than those of Wang et al. [21] and Huang et al. [22].

Table 2. Comparison of space costs.

	Wang et al. [21]	**Huang et al. [22]**	**Proposed Scheme**
plaintext size	1	1	1
private key size	$3m \times m$	$\frac{1}{2}(l_s + 3N + 1) \times m$	$\frac{n}{m} + 2n + 2$
decryption key size	$4m \times k$	$\frac{1}{2}(l_s + 3N) \times m$	$\frac{n}{m} + 6$
ciphertext size	$2m + m \times i + m \times \theta + k$	$(l_s + N) \times m$	$\frac{n}{m} + 7$

Due to the fact that the ciphertext size of Wang et al.'s [21] scheme is related to the attribute and node sets, while the proposed scheme does not involve attributes, only the private key size and decryption key size are compared. As shown in Table 3, we compared the proposed scheme with [21,22] for private key size and decryption key size under different security parameters n. Figures 2 and 3 demonstrate that the proposed scheme has a significant advantage in terms of space costs.

Table 3. Specific comparison of space costs.

Scheme		Private Key Size	Decryption Key Size
Wang et al. [21]	$n = 64$	34.8000 KB	0.2720 KB
	$n = 128$	278.3800 KB	0.8970 KB
	$n = 256$	2227.0500 KB	2.9000 KB
	$n = 512$	17,816.4300 KB	9.2300 KB
	$n = 1024$	143,557.7600 KB	28.8000 KB
Huang et al. [22]	$n = 64$	0.2070 KB	0.1880 KB
	$n = 128$	0.5850 KB	0.5320 KB
	$n = 256$	1.6600 KB	1.5100 KB
	$n = 512$	4.6800 KB	4.2600 KB
	$n = 1024$	13.2500 KB	12.0400 KB
Average 1 bit of Proposed Scheme	$n = 64$	0.0159 KB	0.0007 KB
	$n = 128$	0.0315 KB	0.0007 KB
	$n = 256$	0.0627 KB	0.0007 KB
	$n = 512$	0.1250 KB	0.0007 KB
	$n = 1024$	0.2480 KB	0.0007 KB

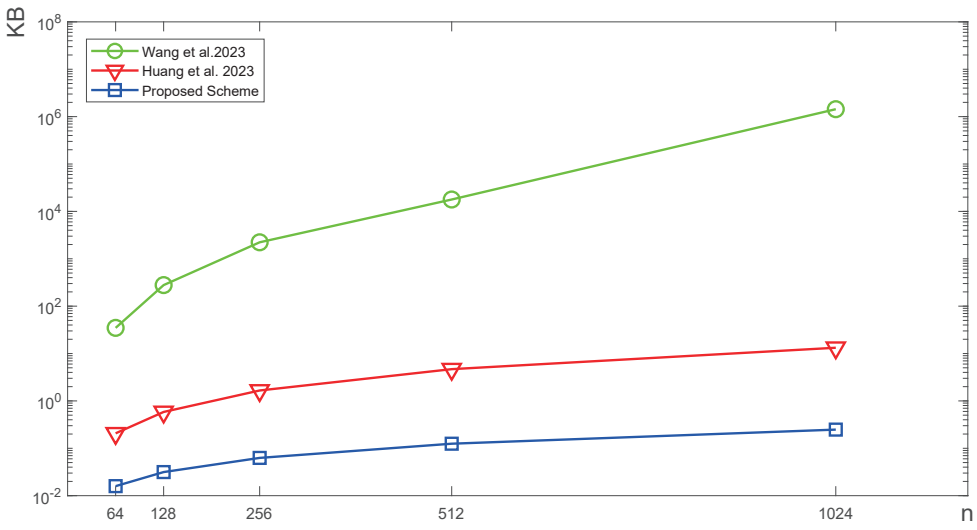

Figure 2. Comparison of private key size with [21,22].

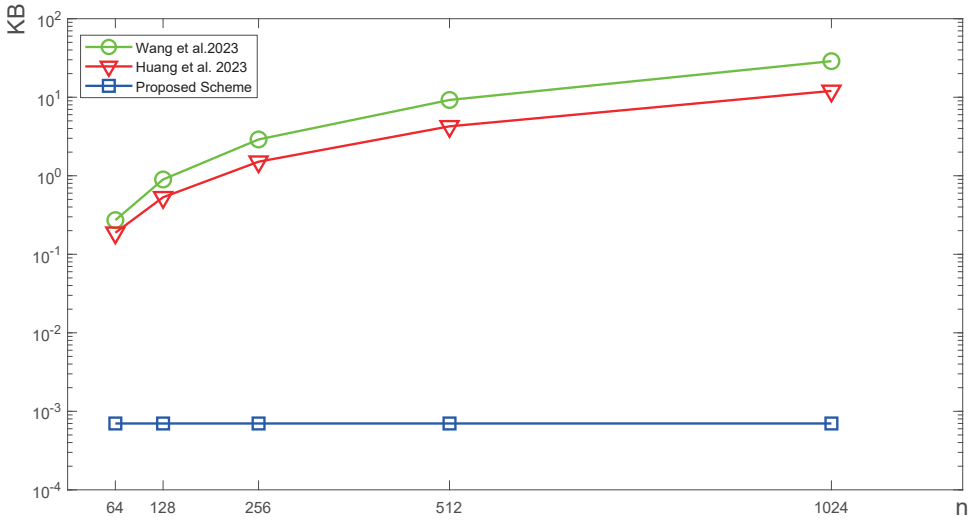

Figure 3. Comparison of decryption key size with [21,22].

6.1.2. Computation Costs

Let T_{sp} and T_{sl} denote the time costs of running algorithms $SamplePre$ and $SampleLeft$, respectively. T_{mul} denotes the time costs of multiplication operation in $\mathbb{Z}_q$. Users can operate the hash function and the matrix addition operation offline, so the computational costs of both can be ignored. According to the the running time of every algorithm of [2], Table 4 shows the average time of ten operations of these algorithms.

Table 4. The running time of every algorithm.

n	$SamplePre$ (ms)	$SampleLeft$ (ms)	T_{mul} (ms)
64	178	181	0.03
128	558	563	0.05

For comparison, let $k = \log n$, $q = 2^{\sqrt{n}}$, and $m = 2n\lceil \log q \rceil = 2n^{1.5}$. In scheme [22], let $m = n\lceil \log q \rceil = n^{1.5}$, $N = 3$, and $l_s = 1$. As illustrated in Table 5, when n = 64, Wang et al.'s [21] scheme requires $8n^2T_{mul} \approx 0.98$ s for executing the Enc algorithm, and $(n \times k + 14n^{2.5})T_{mul} + T_{sp} \approx 18.51$ s for the GenDK algorithm. Huang et al.'s [22] scheme requires $(7n^{1.5} + 2n^3)T_{mul} \approx 15.83$ s for executing the Enc algorithm, and $2n^{1.5}T_{mul} + T_{sl} + 2T_{sp} \approx 0.56$ s for the GenDK algorithm. While the proposed scheme requires $(9n + 8n^{1.5})T_{mul} \approx 0.14$ s for executing the Enc algorithm, and $6n^{1.5}T_{mul} + T_{sl} \approx 0.27$ s for the GenDK algorithm. When n = 128, Wang et al.'s [21] scheme requires $8n^2T_{mul} \approx 6.55$ s for executing the Enc algorithm, and $(n \times k + 14n^{2.5})T_{mul} + T_{sp} \approx 130.32$ s for the GenDK algorithm. Huang et al.'s [22] scheme requires $(7n^{1.5} + 2n^3)T_{mul} \approx 210.22$ s for executing the Enc algorithm, and $2n^{1.5}T_{mul} + T_{sl} + 2T_{sp} \approx 1.82$ for the GenDK algorithm. While the proposed scheme requires $(9n + 8n^{1.5})T_{mul} \approx 0.63$ s for executing the Enc algorithm, and $6n^{1.5}T_{mul} + T_{sl} \approx 0.99$ s for the GenDK algorithm. In Figures 4 and 5, the comparison demonstrates that the constructed scheme also has a significant advantage in computation cost over the schemes of Wang et al. [21] and Huang et al. [22].

Table 5. Specific comparison of computation costs.

Schemes	Enc	GenDK
Wang et al. [21]	$8n^2T_{mul}$	$(n \times k + 14n^{2.5})T_{mul} + T_{sp}$
Huang et al. [22]	$(7n^{1.5} + 2n^3)T_{mul}$	$2n^{1.5}T_{mul} + T_{sl} + 2T_{sp}$
Average 1 bit of Our scheme	$(9n + 8n^{1.5})T_{mul}$	$6n^{1.5}T_{mul} + T_{sl}$

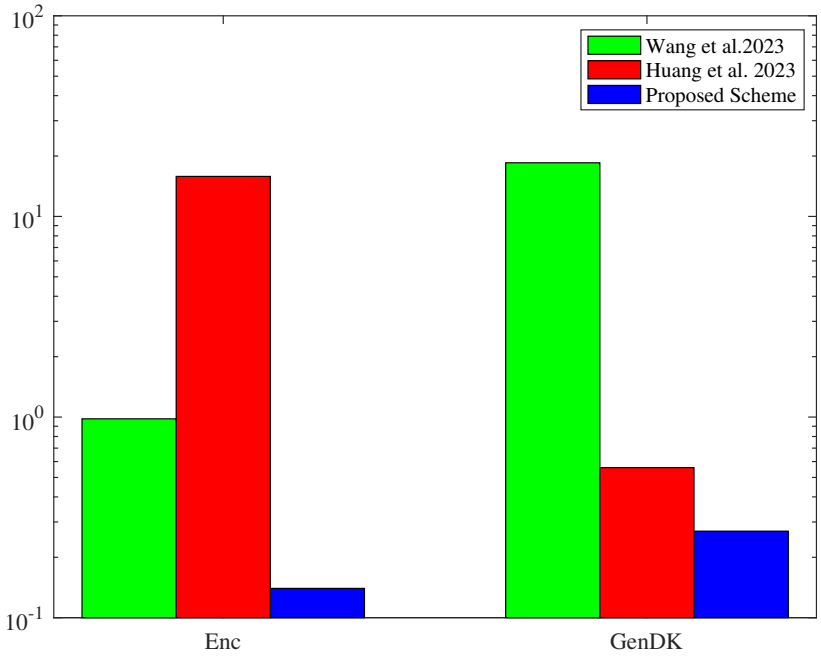

Figure 4. The running time of these two algorithms when $n = 64$ compared with [21,22].

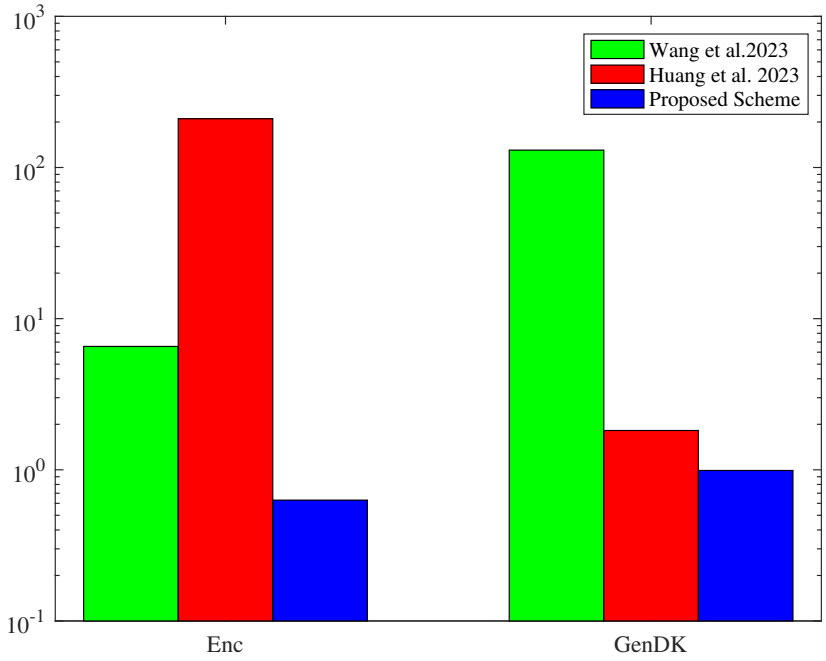

Figure 5. The running time of these two algorithms when $n = 128$ compared with [21,22].

6.2. Simulation Experiments

The proposed RCL-PKE scheme was run on a Ubuntu laptop with an 12th Gen Intel(R) Core(TM) i9-12900H 2.50 GHz CPU and 16 GB of memory. For better portability, the program was implemented using the NTL library and C++ language. In addition, a single leaf node we used as an example to implement the code. Table 6 shows the specific running time of the proposed scheme.

Table 6. Running time of RCL-PKE scheme.

n	Setup (s)	Extractppk (s)	SetKey (s)	UpdateTK (s)	Enc (s)	GenDK (s)	Dec (s)
64	0.54	14.85	1.35	14.92	0.52	14.36	0.13
128	4.19	113.83	2.95	115.01	1.38	112.09	0.55

As shown in Table 6, in the proposed scheme, when n = 64, the Setup algorithm ran in about 0.54 s, the Extractppk algorithm spent about 14.85 s, the time cost of the SetKey algorithm was about 1.35 s, the UpdateTK algorithm ran in about 14.92 s, the Enc algorithm ran in about 0.52 s, the GenDK algorithm required about 14.36 s, and the Dec algorithm was completed in about 0.13 s. When n = 128, the Setup algorithm ran in about 4.19 s, the Extractppk algorithm spent about 113.83 s, the SetKey algorithm took about 2.95 s, the UpdateTK algorithm ran in about 115.01 s, the Enc algorithm ran in about 1.38 s, the GenDK algorithm required about 112.09 s, and the Dec algorithm was completed in about 0.55 s. The specific trend of the algorithm's running time is illustrated in Figure 6. As can be seen from Figure 6, the running time required by the algorithms involved in the proposed scheme increased as the security parameter n increased. When n was doubled, the increase in the time required by the algorithm was not significant and the trend in the change was acceptable.

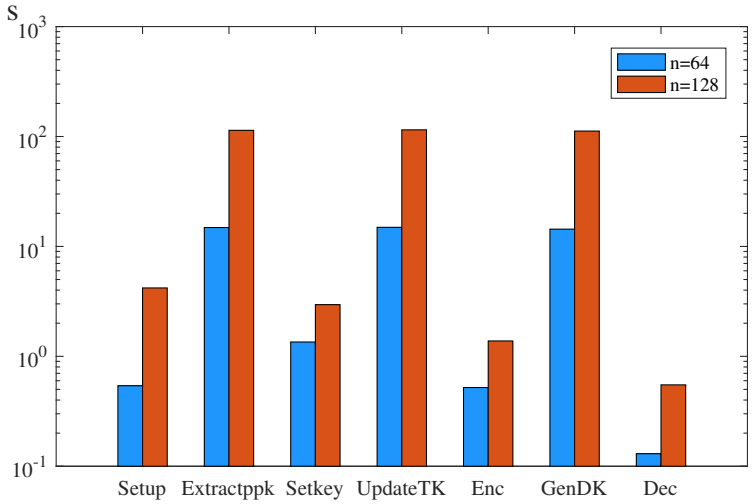

Figure 6. The trends in running time of RCL-PKE scheme.

7. Real Application

In team score orienteering, numerous checkpoints (divided into compulsory and optional points) are scattered on a map, and each point assigns a different score. Compulsory checkpoints are those that all athletes on the team must reach, and optional checkpoints the athletes work together to accomplish. With three to four athletes as a team, athletes use maps and compasses to locate these checkpoints within the specified time, and the team with the highest total score wins.

Currently, the checkpoints in orienteering are electronic, with no internet connection. Athletes wear non-contact bracelets to sign in at each electronic checkpoint, and their scores are obtained after reaching the finish line. However, there are three problems with this model. Firstly, there is no network link between the checkpoints. Team score orienteering requires cooperation between teams to obtain valid checkpoint information and help other team members to know that he/she has visited a certain checkpoint. Secondly, it is challenging to track the precise path of the athletes during the competition, and there will be errors in the position of the point signer and the map labeling, which diminishes the viewer experience and hinders the live broadcasting of promotional events. Finally, the athletes and the coaches can not observe competition data for analysis, such as running posture and physical distribution.

For the above three issues, the RCL-PKE scheme was applied to the team score orienteering, as shown in Figure 7. Team score orienteering involves three entities: the organizing committee, athletes, and coaches/referees. The organizing committee, as the key generator center, is responsible for managing the global parameters and revoking violating athletes. Athletes, as the data owners, encrypt their data (such as checkpoint information, running posture, and physical distribution) and upload them to the cloud server. Coaches/referees/athletes, as the data users, are responsible for downloading the ciphertexts from the cloud server for decryption and analyzing the decrypted data. The specific implementation steps are as follows:

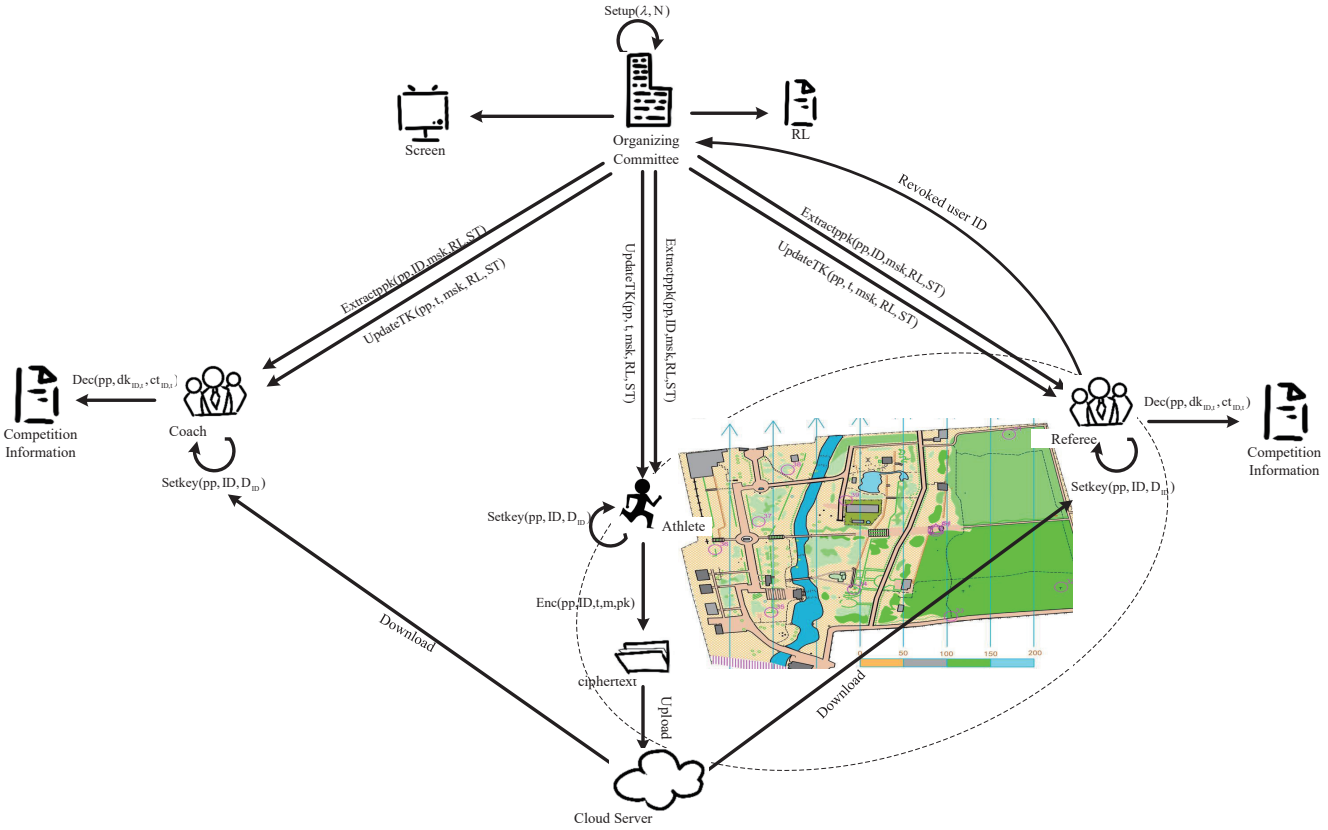

Figure 7. The RCL-PKE scheme in team score orienteering.

Initialization phase: (1) Athletes wear the sensors that are linked to their personal information and check whether the sensors are working properly. (2) The organizing committee executes the Setup(λ, N) algorithm to generate the public parameters, master private keys, revocation lists, and state.

Registration phase: (3) The user (athlete/coach/referee) registers with the organizing committee, which executes the Extractppk(pp, ID, msk, RL, ST) algorithm to generate the partial private key for the user. (4) The user (athlete/coach/referee) chooses a secret value and executes the Setkey(pp, ID, D_{ID}) to generate his/her own public/private key pair.

Competition phase: (5) Referees are responsible for supervising the behavior of the athletes, and if any violations are found, they will immediately report to the organizing committee. (6) The organizing committee disqualifies the fouling athletes from the competition based on this feedback and places them on the revocation list, which is notified to all the teams and spectators through the broadcasting and the big screen. (7) The organizing committee generates the updated time key based on the algorithm of UpdateTK(pp, t, msk, RL, ST).

Encryption phase: (8) When the athlete arrives at the checkpoint, the sensor with encryption/decryption function encrypts the athlete's personal information and checkpoint

information according to the $\mathrm{Enc}(pp, ID, t, m, pk)$ algorithm. (9) The generated ciphertext is then uploaded and stored on the cloud server.

Decryption phase: (10) Coach/referee/athlete generates a decryption key according to $\mathrm{GenDK}(pp, sk_{ID}, TK_t)$ algorithm. (11) The coach/referee/athlete downloads the ciphertext from the cloud server and executes $\mathrm{Dec}(pp, dk_{ID,t}, ct_{ID,t})$ algorithm to obtain the athlete's information.

Data use phase: (12) Athletes decrypt and receive valid checkpoint information from other athletes for rational route planning. Coaches decrypt the data received from athletes to provide real-time guidance. The organizing committee decrypts the data and updates the results and ranking to the big screen in real time, so that the audience can keep track of the progress of the competition at any time.

Applying the RCL-PKE scheme to team score orienteering not only ensures the security of athletes' data but also solves the key escrow problem caused by the organizing committee generating keys for the users separately.

8. Conclusions

This paper proposed a lattice-based revocable certificateless public key encryption (RCL-PKE) scheme with decryption key exposure resistance (DKER), which ensures that the leakage of the decryption key in any time period does not compromise the confidentiality of the ciphertexts in other time periods. Furthermore, it was proven for three types of adversaries that the proposed scheme is IND-CPA secure under the LWE assumption. Compared with other lattice-based revocable schemes, the proposed RCL-PKE scheme has a higher efficiency in the revocation mechanism. Therefore, the scheme is more suitable for being applied in team score orienteering. In future work, we aim to further improve the efficiency of the scheme based on Ring-LWE combined with online/offline and other techniques.

Author Contributions: Conceptualization and methodology, Y.Z. and J.L.; writing—original draft, M.Y.; software and validation, M.Y. and K.Y.; writing—review and editing, Y.Z. and J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Decrypt the ciphertext $ct_{ID,t}$. Because

$$\begin{aligned}
&[A|E(ID)|E(t)]d^{\theta}_{ID,t} \\
&= [A|E(ID)|E(t)]\begin{pmatrix} d^{\theta,1}_{ID} + d^{\theta,1}_{t} \\ d^{\theta,2}_{ID} \\ d^{\theta,2}_{t} \end{pmatrix} \\
&= A(d^{\theta,1}_{ID} + d^{\theta,1}_{t}) + E(ID)d^{\theta,2}_{ID} + E(t)d^{\theta,2}_{t} \\
&= [A|E(ID)]\begin{bmatrix} d^{\theta,1}_{ID} \\ d^{\theta,2}_{ID} \end{bmatrix} + [A|E(t)]\begin{bmatrix} d^{\theta,1}_{t} \\ d^{\theta,2}_{t} \end{bmatrix} \\
&= [A|E(ID)]d^{\theta}_{ID} + [A|E(t)]d^{\theta}_{t} \\
&= \mu_{\theta} + \mu - \mu_{\theta} \\
&= \mu
\end{aligned}$$

we have,

$$\begin{aligned}
&C_0 - x^T C_1 - \overline{d_{ID,t}}^T C_3 - (d^{\theta}_{ID,t})^T C_2 \\
&= \mu^T s_1 + \mu^T s_2 + 2e + b^T r + m \\
&- X^T B r - \overline{d_{ID,t}}^T ([\overline{A}|E(ID)|E(t)]^T s_2 + 2e_2) \\
&- (d^{\theta}_{\mathrm{ID,t}})^T ([A|E(ID)|E(t)]^T s_1 + 2[e_2, R_1^T e_2, R_2^T e_2]) \\
&= \mu^T s_1 + \mu^T s_2 + 2e + X^T B r + 2e_1^T r + m - X^T B r \\
&- \left[[\overline{A}|E(ID)|E(t)] \overline{d_{ID,t}} \right]^T s_2 - 2\overline{d_{ID,t}}^T [e_3, R_1^T e_3, R_2^T e_3] \\
&- \left[[A|E(ID)|E(t)] d^{\theta}_{\mathrm{ID,t}} \right]^T s_1 - 2(d^{\theta}_{ID,t})^T [e_2, R_1^T e_2, R_2^T e_2] \\
&= m + \underbrace{2e + 2e_1^T r - 2\overline{d_{ID,t}}^T \left[e_3, R_1^T e_3, R_2^T e_3 \right] - 2(d^{\theta}_{ID,t})^T \left[e_2, R_1^T e_2, R_2^T e_2 \right]}_{2\Delta} \\
&= m + 2\Delta
\end{aligned}$$

Therefore, when $|2\Delta| < \frac{q}{2}$, the decryption is correct.
The RCL-PKE scheme must meet the following conditions:

(1) Algorithm *TrapGen* requests $m \geq 2n\lceil \log q \rceil$
(2) Algorithm *SamplePre* requests $\sigma \geq \left\| \widetilde{T_A} \right\| \omega(\sqrt{\log m})$
(3) Algorithm *ExtRndLeft* requests $\sigma \geq \left\| \widetilde{T_A} \right\| \omega(\sqrt{\log n})$
(4) Leftover hash lemma requests $m > (n+1)\lceil \log q \rceil + \omega(\sqrt{\log n})$
(5) Lwe requests $\alpha q > 2\sqrt{n}$

Therefore, let $n = \lambda$, $m = 2n\lceil \log q \rceil$, $\chi = D_{z,\alpha_q}$, $\sigma = m\omega(\sqrt{\log m})$, $\alpha^2 < \frac{1}{32qm\omega(\sqrt{\log n})}$.
Since $\left\| d_{ID,t} \right\| \leq \sigma(\sqrt{3m})$, $\left\| d^{\theta}_{ID,t} \right\| \leq 2\sigma(\sqrt{2m})$, we obtain

$$\begin{aligned}
&|2\Delta| = 2\left| e + e_1^T r - d_{ID,t}^T [e_3, R_1^T e_3, R_2^T e_3] - {d^{\theta}_{ID,t}}^T [e_2, R_1^T e_2, R_2^T e_2] \right| \\
&\leq 2|e| + 2\left| e_1^T r \right| + 2\left| d_{ID,t}^T [e_3, R_1^T e_3, R_2^T e_3] \right| + 2\left| {d^{\theta}_{ID,t}}^T [e_2, R_1^T e_2, R_2^T e_2] \right| \\
&< 2\sigma\omega(\sqrt{\log n}) + 2\sqrt{m}\sigma\omega(\sqrt{\log n}) + 2\left\| d_{ID,t}^T \right\| 2\sqrt{m}\sigma\omega(\sqrt{\log n}) \\
&+ 2\left\| {d^{\theta}_{ID,t}}^T \right\| 2\sqrt{m}\sigma\omega(\sqrt{\log n}) \\
&\leq 2\sigma\omega(\sqrt{\log n}) \left[1 + \sqrt{m} + \left\| d_{ID,t}^T \right\| \sqrt{m} + 2\left\| {d^{\theta}_{ID,t}}^T \right\| \sqrt{m} \right] \\
&\leq 2\sigma\omega(\sqrt{\log n}) \left[1 + \sqrt{m} + \sigma(\sqrt{3m})\sqrt{m} + 4\sigma(\sqrt{2m})\sqrt{m} \right] \\
&\leq 2\sigma\omega(\sqrt{\log n}) \left[1 + \sqrt{m} + \sigma m(\sqrt{3}) + 4\sigma m\sqrt{2} \right] \\
&< 16\sigma^2 m\omega(\sqrt{\log n}) \\
&< \frac{16\sigma^2 m\omega(\sqrt{\log n})}{32qm\omega(\sqrt{\log n})} \\
&< \frac{q}{2}
\end{aligned}$$

References

1. Shamir, A. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology: Proceedings of CRYPTO 84 4*; Springer: Berlin/Heidelberg, Germany, 1985; pp. 47–53.
2. Li, J.; Yan, M.; Peng, J.; Huang, H.; Abd El-Latif, A. A lattice-based efficient certificateless public key encryption for big data security in clouds. *Future Gener. Comput. Syst.* **2024**, *158*, 255–266. [CrossRef]
3. Shen, L.; Zhang, F.; Sun, Y. Efficient revocable certificateless encryption secure in the standard model. *Comput. J.* **2014**, *57*, 592–601. [CrossRef]
4. Tang, Y.; Chow, S.; Liu, J. Comments on 'Efficient revocable certificateless encryption secure in the standard model'. *Comput. J.* **2015**, *58*, 779–781. [CrossRef]
5. Sun, Y.; Zhang, F.; Shen, L.; Deng, R. Efficient revocable certificateless encryption against decryption key exposure. *IET-Form. Secur.* **2015**, *9*, 158–166. [CrossRef]
6. Tsai, T.; Tseng, Y. Revocable certificateless public key encryption. *IEEE Syst. J.* **2013**, *9*, 824–833. [CrossRef]

7. Sun, Y.; Zhang, F.; Fu, A. Revocable certificateless encryption with ciphertext evolution. In Proceedings of the Information Security and Privacy: 23rd Australasian Conference, ACISP 2018, Wollongong, NSW, Australia, 11–13 July 2018; pp. 741–749.
8. Sun, Y.; Zhang, F.; Fu, A.; Xia, Z. CCA-Secure and Revocable Certificateless Encryption with Ciphertext Evolution. *Int. J. Found. Comput. Sci.* **2020**, *31*, 175–191. [CrossRef]
9. Zhang, Y.; Zhang, T.; Xu, S.; Xu, G.; Zheng, D. Revocable and certificateless public auditing for cloud storage. *Sci. China Inf. Sci.* **2020**, *63*, 1. [CrossRef]
10. Ma, M.; Shi, G.; Shi, X.; Su, M.; Li, F. Revocable certificateless public key encryption with outsourced semi-trusted cloud revocation agent. *IEEE Access* **2020**, *8*, 148157–148168. [CrossRef]
11. Tsai, T.; Tseng, Y.; Huang, S. Equality Test of Ciphertexts in Certificateless Public Key Systems with an Outsourced Revocation Authority. In Proceedings of the 2022 IEEE 11th Global Conference on Consumer Electronics (GCCE), Osaka, Japan, 18–21 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 36–37.
12. Tsai, T.; Lin, H.; Tsai, H. Revocable certificateless public key encryption with equality test. *Inf. Technol. Control* **2022**, *51*, 638–660. [CrossRef]
13. Tseng, Y.; Huang, S.; Tsai, T.; Chuang, Y.; Hung, Y. Leakage-resilient revocable certificateless encryption with an outsourced rev-ocation authority. *Informatica* **2022**, *33*, 151–179. [CrossRef]
14. Wang, Y.; Liu, Y.; Tian, Y. ISC-CPPA: Improverd-Security Certificateless Conditional Privacy-Preserving Authentication Scheme With Revocation. *IEEE Trans. Veh. Technol.* **2022**, *71*, 12304–12314. [CrossRef]
15. Tseng, Y.; Chien, H.; Hung, R.; Tsai, T. Leakage-Resilient Anonymous Multi-Receiver Outsourced Revocable Certificateless En-cryption. In Proceedings of the 2023 5th International Conference on Computer Communication and the Internet (ICCCI), Fujisawa, Japan, 23–25 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 127–132.
16. Meng, F.; Cheng, L. STR-ABKS: Server-Aided Traceable and Revocable Attribute-Based Encryption With Keyword Search. *IEEE Internet Things J.* **2024**, *11*, 12649–12659. [CrossRef]
17. Guo, L.; Wang, L.; Ma, X.; Ma, Q. A New Revocable Attribute Based Encryption on Lattice. In *International Conference on Provable Security*; Springer Nature: Cham, Switzerland, 2023; pp. 309–326.
18. Guo, L.; Wang, L.; Ma, X.; Zhang, X. New Traceable and Revocable Attribute Based Encryption on Lattices. In Proceedings of the 2023 International Conference on Networking and Network Applications (NaNA), Qingdao, China, 18–21 August 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 359–364.
19. Wen, J.; Bai, L.; Yang, Z.; Zhang, H.; Wang, H.; He, D. LaRRS: Lattice-based revocable ring signature and its application for VANETs. *IEEE Trans. Veh. Technol.* **2024**, *73*, 739–753. [CrossRef]
20. Katsumata, S.; Matsuda, T.; Takayasu, A. Lattice-based revocable (hierarchical) IBE with decryption key exposure resistance. *Theor. Comput. Sci.* **2020**, *809*, 103–136. [CrossRef]
21. Wang, Q.; Li, J.; Wang, Z.; Zhu, Y. Revocable-Attribute-Based Encryption with En-DKER from Lattices. *Mathematics* **2023**, *11*, 4986. [CrossRef]
22. Huang, B.; Gao, J.; Li, X. Efficient lattice-based revocable attribute-based encryption against decryption key exposure for cloud file sharing. *J. Cloud Comput.* **2023**, *12*, 37. [CrossRef] [PubMed]
23. Wang, Q.; Huang, H.; Li, J.; Yuan, Q. Revocable IBE with En-DKER from Lattices: A Novel Approach for Lattice Basis Delegation. In *European Symposium on Research in Computer Security*; Springer Nature: Cham, Switzerland, 2024; pp. 66–85.
24. Chen, J.; Lim, H.; Ling, S.; Wang, H.; Nguyen, K. Revocable identity-based encryption from lattices. In Proceedings of the Information Security and Privacy: 17th Australasian Conference, ACISP 2012, Wollongong, NSW, Australia, 9–11 July 2012; pp. 390–403.
25. Micciancio, D.; Peikert, C. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 700–718.
26. Agrawal, S.; Boneh, D.; Boyen, X. Efficient lattice (H) IBE in the standard model. In Proceedings of the Cryptology—EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, France, 30 May–3 June 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 553–572.
27. Dodis,Y.; Ostrovsky, R.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.* **2008**, *38*, 97–139. [CrossRef]