A revocable identity-based encryption (IBE) scheme under Learning with Error (LWE) assumption from lattice is proposed, compared with the existing IBE schemes from lattices, two improvements are available in the new scheme. First, the revocation mechanism for identity of user (or thing) is added in the new scheme to manage the dynamic user (or thing) identity in the system, and make identity-based encryption scheme from lattice becomes more practical than ever before. Second, the security of the new scheme is based on Learning with Error (LWE) assumption from lattice, so our construction has a solid secure foundation. The scheme is proved to be secure against adaptive-ID attacks under LWE assumption from lattice in the standard model.

**INDEX TERMS** Internet of things, Information Security, Encryption, Lattices

**I. INTRODUCTION**

Many problems in the traditional public key cryptosystems may be avoided in the identity-based cryptography such as maintenance, updating and revocation of public key certificates. The identity-based cryptography has fundamentally changed the methods of management and operation of the traditional public key systems. The main difference between identity-based cryptography and public key cryptosystems is that the identity information of a user (or thing) can be used to uniquely identify the user (or thing) in identity-based cryptography, thus the public key of a user(or thing) can be derived from his identity, while in public key cryptography the public key of a user(or thing) has nothing to do with the identity of the user(or thing). So a public key certificate is needed to bind the public key and the identity of the user (or thing). Identity-based cryptography can be used in e-business, e-commerce and e-government, especially for internet of things, etc.

To ensure the security of cryptographic systems especially with a vast number of users (or things), the revocation mechanism is indispensable. A user’s (or thing’s) identity may be revoked for various reasons. Such as in the case that the user’s private key is leaked, or the user is no longer as legitimate user, or suppose that there is an employee who resigned from the company, his identity should be revoked from the system.

Identity revocation was first proposed by Boneh and Franklin in [2], in their scheme the key generation center (KGC) regularly generate private key for every user who hasn’t been revoked, and regularly update their private key, this makes KGC workload increase linearly as the number of users increases.

Boldyreva et al. [3] proposed a new method to reduce restrictions on the revocable IBE scheme, they define a framework and a security model for it, the model takes into account the entire adversary’s ability in standard security model of IBE, and improves the effectiveness of the previous scheme. By using of the combination of binary tree data structure and fuzzy IBE scheme which is proposed by Sahai and waters in [4], they construct a RIBE scheme in which the KGC workload increase logarithmically as the number of users increases, the security of the scheme is selective identity secure.
Libert and Vergnaud [5] improved the scheme of Boldyreva et al. [3]. The scheme reached a higher level of security, that is, adaptive identity secure.

As the lattice has become a more favorable tool for cryptographic systems, many lattice-based IBE scheme has been proposed so far, a reasonable explanation is that lattice-based encryption scheme is much easier and more effective to implement. Chen et al. [25] constructed lattice-based RIBE Scheme, by using following three tectonic blocks: IBE Scheme in [1] proposed by Agrawal, Boneh and Boyen, trapdoor from lattice in [9] proposed by Gentry, Peikert et al, and binary tree data structure for key update.

In 2013, Su et al. [13] proposed a more effective RIBE Scheme based on the accumulation, the scheme is of adaptive identity security, and the number of update key is of constant size, which is a great improvement for RIBE Schemes.

In 2014, Tsai et al. [15] proposed a public revocation mechanism and defined RHIBE scheme framework with public revocation mechanism. In addition, its security concept formalizes some possible threats. The paper proposed a specific RHIBE scheme by using of bilinear maps, the scheme is based on HIBE scheme proposed by Lewko and Waters in [18] with the help of public revocation mechanism. The scheme is constructed by using of the bilinear maps.

In 2016 Wang et al. [22] proposed an efficient lattice-based hierarchical IBE scheme. And zhang et al. [23] proposed hierarchical identity-based broadcast encryption scheme on lattices. Wang et al. [24] proposed a full secure IBE scheme with short public key size over lattices. An IBE scheme is also proposed in [27]. A lattice-based identity-based resplittable threshold encryption scheme is proposed in [28]. But in all these schemes the problem that a user may be revoked in the practice has not been properly addressed, so all the scheme is not perfect for practical use, only the scheme in [25] considered the identity revocation problem. But we used a different way to address the revocation problem.

**Our contribution**

- We use a random vector \( \mathbf{u} \) to connect the user identity with time, and the vector \( \mathbf{u} \) was randomly divided into two vectors \( \mathbf{u}_{id,1}, \mathbf{u}_{id,2} \) corresponding to the identity and time respectively. But for a revoked user, he/she cannot get the latest update key which evolves with time.

- The security of the new scheme is based on worst-case lattice problems named Learning with Error (LWE) problem, so our cryptographic construction has a solid secure foundation. As our scheme is based on the lattice so our scheme is much more efficient than that of in [18] which is based on the bilinear maps.

- An comparing of the performance of our scheme with the schemes of [22],[23],[24],[25],[27] and [28] is shown in Table I.

### Table I. COMPARISON OF PERFORMANCE

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Lattice based</th>
<th>Identity revocation</th>
<th>Identity based</th>
<th>Standard model</th>
<th>cipher-text connected with time</th>
</tr>
</thead>
<tbody>
<tr>
<td>[22]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>[23]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>[24]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>[25]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>[27]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>[28]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>our scheme</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Form the table I we can see that our new scheme and scheme in [25] is of full function. Compared with the scheme in [25] we use a revocation list to address the revocation of the user, while in [25] the authors use a different way of the binary tree revocation technique, so our revocation method is different from that in [25], and our revocation method is relatively much easier to management.

## II. PRELIMINARIES

### A. Integer Lattices

**Definition 1.** For a prime \( q \), a matrix \( \mathbf{A} \in \mathbb{Z}^{n \times m}_q \) and a vector \( \mathbf{u} \in \mathbb{Z}_q^m \), define

\[
\Lambda_{\mathbf{A}}(\mathbf{e}) := \{ \mathbf{e} \in \mathbb{Z}^m \text{ s.t. } \mathbf{Ae} = \mathbf{0} \pmod{q} \},
\]

\[
\Lambda_{\mathbf{u}}(\mathbf{e}) := \{ \mathbf{e} \in \mathbb{Z}^m \text{ s.t. } \mathbf{Ae} = \mathbf{u} \pmod{q} \}.
\]
**Theorem 1** ([16]). Let \( q \geq 2, n \geq 1 \) be integers and \( m = \lceil 2n \log q \rceil \). There is an algorithm \( \text{TrapGen}(q, n) \) that outputs a pair \((A \in \mathbb{Z}_q^{n \times m}, T, \in \mathbb{Z}_q^{m \times m})\), where \( T \) is a basis for \( \Lambda_q \) with \( \|T\| \leq O(\sqrt{n \log q}) \) and \( \|T\| \leq O(n \log q) \).

**B. Sampling Algorithms**

**SampleLeft** \((A, M, T, u, \sigma)\) ([1]). Let \( q > 2, m > n \). It takes as input matrices \( A \in \mathbb{Z}_q^{n \times m} \) and \( M \in \mathbb{Z}_q^{m \times m} \), a “short” basis \( T \) of \( \Lambda_q \), a vector \( u \in \mathbb{Z}_q^m \), and a Gaussian parameter \( \sigma > \|T\| / \sqrt{\log(m + m)} \). It outputs a vector \( e \in \mathbb{Z}_q^{m + m} \), which is distributed statistically with \( D_{\Lambda_q(I)}^{\|T\| / \sqrt{\log(m + m)}}, \) where \( F = (A \mid M) \). In particular \( e \in \Lambda_q^*(F_1) \).

**SampleRight** \((A, B, R, T, u, \sigma)\) ([1]). Let \( q > 2, m > n \). It takes as input matrices \( A, B, R \in \mathbb{Z}_q^{n \times m} \), where \( B \) is of rank \( n \), a uniform randomly selected matrix \( R \in \{-1, 1\}^{m \times m} \), a “short” basis \( T \) of \( \Lambda_q(B) \), a vector \( u \in \mathbb{Z}_q^n \), and a Gaussian parameter \( \sigma > \|T\| / \sqrt{\log(m + m)} \) where \( s = \|R\| = \sup_{1 \leq q \leq m} \|Rq\| \). It outputs a vector \( e \in \mathbb{Z}_q^m \), which is distributed statistically close to \( D_{\Lambda_q(I)}^{\|T\| / \sqrt{\log(m + m)}}, \) where \( F = (A \mid AR + B) \). In particular \( e \in \Lambda_q^*(F_2) \).

**Lemma 1** ([1], lemma 13). Suppose that \( m > (n + 1) \log q + \omega(\log n) \) and that \( q \) is a prime. Let \( A, B \) be matrices chosen uniformly in \( \mathbb{Z}_q^{n \times m} \) and let \( R \) be a \( m \times m \) matrix chosen randomly in \( \{-1, 1\}^{m \times m} \) mod \( q \). Then, for all vectors \( w \in \mathbb{Z}_q^m \), the two distributions of \( (A, AR, R^T w) \) and \( (A, B, R^T w) \) are statistically closed.

**Lemma 2** ([1], lemma 15). Let \( R \) be a randomly \( m \times m \) matrix in \( \{-1, 1\}^{m \times m} \). Then there is a constant \( C \) such that \( \Pr[s = \|R\| \geq C \sqrt{m}] \leq e^{-2m} \).

**Lemma 3** ([1], lemma 12). Let \( e \in \mathbb{Z}_q^n \) and \( y \in \mathbb{Z}_q^m \). Then the quantity \( e^T y \) regards the integer in \([0, q - 1]\) satisfies \( e^T y \leq \|e\|_2 \|q \alpha \| \sqrt{\log m} + \|e\|_2 m / 2 \) with all but negligible probability in \( m \).

**C. Abort-resistant hash functions AND ACRONYMS**

**Definition 2** ([1]). Let \( \mathcal{H} = \{H : X \rightarrow Y\} \) denote a family of hash functions, where \( 0 \in Y \). For a set of \( Q \) inputs \( \overline{X} = \{x_1, \ldots, x_Q\} \in X^Q \), the non-abort probability respective to \( J \) -place of \( \overline{X} \) is defined as \( \alpha(H) = \Pr[H \in H : E_j(H)] \).

Where \( E_j(H) \) is the event that given \( J \subseteq [1, \ldots, Q] \), \( (H(x_i) = 0, \forall i \in J) \land (H(x_i) \neq 0, \forall k \in [Q] \setminus J) \).

We call that \( H \) is \((Q, \alpha_{\min}, \alpha_{\max})\) abort-resistant in terms of \( J \) -place if for all \( \overline{X} = \{x_1, \ldots, x_Q\} \in X^Q \) with the condition \( \{x_i : i \in J\} \cap \{x_i : k \in [Q] \setminus J\} = \emptyset \), we have \( \alpha_j(\overline{X}) \in [\alpha_{\min}, \alpha_{\max}] \).

This paper uses the following definition of abort-resistant hash function family. For a prime \( q \), let \( (L_q) = \{L_q^k \mid k \geq 1\} \) and define the family \( \mathcal{H}_1 = \{H_{E_1} : (L_q^k)^* \rightarrow \mathbb{Z}_q^1 \} \) as \( H_{E_1}(id) = 1 + \sum_{s} h_{s} \cdot id = (h_1, \ldots, h_q) = (L_q^k)^*, h = (h_1, \ldots, h_q) \in \mathbb{Z}_q^q \). Then for the hash family \( \mathcal{H}_1 \) defined above satisfies:

- If \( J = \{i\} \), \( \mathcal{H}_1 \) is \((Q, 1 - (1 - 2^{-q}), \frac{1}{q^2})\) abort-resistant in terms of \( J \) -place.
- If \( J = \{i, j\} \), \( \mathcal{H}_1 \) is \((Q, 1 - \left(1 - 2^{-q}\right)^2, \frac{1}{q^2})\) abort-resistant in terms of \( J \) -place.

**Definition 3** (Decisional LWE problem, [9]). Consider a prime \( q \), a positive integer \( n \), and a distribution \( \chi \in \mathbb{Z}_q^* \), all public. A decisional \((\mathbb{Z}_q^n, \chi) - \text{LWE}\) problem instance consists of an unspecified challenge oracle \( O \), being, either, a noisy pseudo-random sampler \( O \) carrying some constant random secret key \( x \in \mathbb{Z}_q^* \), or, a truly sampler \( O \), whose behaviors are respectively as follows.

- \( O \) : Output noisy pseudo-random samples of the form \((w, v) = (w, w^T x + \chi) \in \mathbb{Z}_q^* \times \mathbb{Z}_q^* \), where, \( x \in \mathbb{Z}_q^* \) is a uniformly distributed persistent secret key that is invariant across invocations. \( \chi \) is a freshly generated ephemeral additive noise component with distribute \( \chi \), and \( w \in \mathbb{Z}_q^* \) is a fresh uniformly distributed vector revealed as part of the output.
\( \Omega \): Output truly random samples \((w,v) \in \mathbb{Z}_q^n \times \mathbb{Z}_q \), drawn in independently uniformly at random in the entire domain \( \mathbb{Z}_q^n \times \mathbb{Z}_q \).

The \((\mathbb{Z}_q, n, \chi) - LWE\) problem statement, or \(LWE\) for short, allows an unspecified number of queries to be made to the challenge oracle \( \Omega \), with no stated or prior bound.

We say that an algorithm \( \mathcal{A} \) decides the \((\mathbb{Z}_q, n, \chi) - LWE\) problem if \( \Pr[A^{\Lambda_1} = 1] - \Pr[A^{\Lambda_0} = 1] \) is non-negligible for a random \( x \in \mathbb{Z}_q^n \).

### III. Syntax of RIBE

#### A. Syntax of RIBE

Let \( \mathcal{M} \), \( \mathcal{T} \) and \( \mathcal{T} \) denote the message space, the identity space, and the total number of time period respectively. We treat time \( t \) as a discrete variable. More specifically, we divided the life time of the system into \( \mathcal{T} \) interval (or time period) labeled as \( 0, \ldots, \mathcal{T} - 1 \). A RIBE scheme is usually made up of seven probability polynomial time algorithms.

1. \( \text{setup}(1^k, N) \rightarrow (\mathbb{PP}, \mathbb{MK}, \mathbb{RL}, \mathbb{ST}) \) : Given the security parameter \( k \) with a maximal member number of users \( N \). It outputs a system public parameters \( \mathbb{PP} \), a master secret key \( \mathbb{MK} \), a user revocation list \( \mathbb{RL} \) (initially empty), and a system state \( \mathbb{ST} \).

2. \( \text{PriKeyGen}(\mathbb{PP}, \mathbb{MK}, \mathbb{id}, \mathbb{ST}) \rightarrow (\mathbb{SK}_{id}, \mathbb{ST}) \) : Given the public parameters \( \mathbb{PP} \), the master key \( \mathbb{MK} \), an identity \( \mathbb{id} \in \mathcal{I} \), and the state \( \mathbb{ST} \). It outputs a private key \( \mathbb{SK}_{id} \) and an updated state \( \mathbb{ST} \).

3. \( \text{KeyUpd}(\mathbb{PP}, \mathbb{MK}, t, \mathbb{RL}, \mathbb{ST}) \rightarrow \mathbb{KU}_t \) : Given the public parameters \( \mathbb{PP} \), the master key \( \mathbb{MK} \), a key update time \( t < \mathcal{T} \), a revocation list \( \mathbb{RL} \), and a state \( \mathbb{ST} \). It outputs a key update \( \mathbb{KU}_t \).

4. \( \text{DecKeyGen}(\mathbb{SK}_{id}, \mathbb{KU}_t) \rightarrow \mathbb{DK}_{id,t} \) : Given the public parameters \( \mathbb{PP} \), and key update \( \mathbb{KU}_t \), it outputs a decryption key \( \mathbb{DK}_{id,t} \) or a special symbol \( \bot \) indicating that \( \mathbb{id} \) was revoked.

5. \( \text{Enc}(\mathbb{PP}, \mathbb{id}, t, m) \rightarrow \mathbb{CT}_{id,t} \) : Given the public parameters \( \mathbb{PP} \), an identity \( \mathbb{id} \in \mathcal{I} \), an encryption time \( t < \mathcal{T} \), and a message \( m \in \mathcal{M} \). It outputs a cipher-text \( \mathbb{CT}_{id,t} \).

6. \( \text{Dec}(\mathbb{PP}, \mathbb{DK}_{id,t}, \mathbb{CT}_{id,t}) \rightarrow m \) : Given the public parameters \( \mathbb{PP} \), a cipher-text \( \mathbb{CT}_{id,t} \), a decryption key \( \mathbb{DK}_{id,t} \). It outputs a message \( m \).

7. \( \text{KeyRev}(\mathbb{id}, t, \mathbb{RL}, \mathbb{ST}) \rightarrow \mathbb{RL} \) : Given an identity \( \mathbb{id} \) to be revoked, a revocation time \( t \), a revocation list \( \mathbb{RL} \), and a state \( \mathbb{ST} \). It outputs an updated revocation list \( \mathbb{RL} \).

The consistency condition of RIBE requires that for all \( k \in \mathbb{N} \) and polynomial \( N \) (in \( k \)), all \( \mathbb{PP} \) and \( \mathbb{MK} \) output by \( \text{Setup}(1^k, N) \), all \( \mathbb{id} \in \mathcal{I} \), all \( \mathbb{SK}_{id} \) output by \( \text{PriKeyGen}(\mathbb{PP}, \mathbb{MK}, \mathbb{id}, \mathbb{ST}) \), all \( t < \mathcal{T} \) and \( \mathbb{RL} \), all \( \mathbb{KU}_t \) output by \( \text{KeyUpd}(\mathbb{PP}, \mathbb{MK}, t, \mathbb{RL}, \mathbb{ST}) \), all \( \mathbb{id}_t \), \( t \), and \( m \), all \( \mathbb{CT}_{id,t} \), \( \mathbb{SK}_{id} \) output by \( \text{Enc}(\mathbb{PP}, \mathbb{id}_t, t, m) \), need to meet the following conditions:

1. If \( \mathbb{id} \notin \mathbb{RL} \), we have \( \text{Dec}(\mathbb{PP}, \mathbb{DK}_{id,t}, \mathbb{CT}_{id,t}) \rightarrow m \).

2. If \( \mathbb{id} \in \mathbb{RL} \), \( \text{Dec}(\mathbb{PP}, \mathbb{SK}_{id}, \mathbb{KU}_t) \rightarrow \bot \) with overwhelming probability.

3. If \( (\mathbb{id} = \mathbb{id}_t) \land (t = t) \), we have \( \text{Dec}(\mathbb{PP}, \mathbb{DK}_{id,t}, \mathbb{CT}_{id,t}) \rightarrow m \).

4. If \( (\mathbb{id} \neq \mathbb{id}_t) \lor (t \neq t) \), \( \text{Dec}(\mathbb{PP}, \mathbb{DK}_{id,t}, \mathbb{CT}_{id,t}) \rightarrow \bot \) with overwhelming probability, where \( \bot \) means termination.

#### B. Security model of cipher-text indistinguishable against adaptive-ID and chosen plaintext attack (ind-ad-ID-CPA)

The security model of a RIBE scheme is carried out as a game between an adversary \( \mathcal{A} \) and a challenger, and the concrete process is as follows:

**Setup**: It is run to generate the public parameters \( \mathbb{PP} \), a master key \( \mathbb{MK} \), a revocation list \( \mathbb{RL} \) (initially empty), and a state \( \mathbb{ST} \) which include user information in the current system. Then \( \mathbb{PP} \) is send to the adversary \( \mathcal{A} \).

**Query 1**: \( \mathcal{A} \) may adaptively ask a polynomial of queries of the following oracles:

- The private key generation oracle \( \text{PriKeyGen}(\mathbb{id}) \), the challenger runs \( \text{PriKeyGen}(\mathbb{PP}, \mathbb{MK}, \mathbb{id}, \mathbb{ST}) \), the output private key \( \mathbb{SK}_{id} \) is send to the adversary \( \mathcal{A} \).

- The key update generation oracle \( \text{KeyUpd}(t) \), the challenger runs \( \text{KeyUpd}(\mathbb{PP}, \mathbb{MK}, t, \mathbb{RL}, \mathbb{ST}) \), the output key update \( \mathbb{KU}_t \) is send to the adversary \( \mathcal{A} \).

- The revocation oracle \( \text{KeyRev}(\mathbb{id}, t) \), the challenger runs \( \text{KeyRev}(\mathbb{id}, t, \mathbb{RL}, \mathbb{ST}) \), the output update \( \mathbb{RL} \) is send to the adversary \( \mathcal{A} \).

**Challenge**: The adversary \( \mathcal{A} \) outputs the target pair \((\mathbb{id}', t')\) and two selected messages \( m_0, m_1 \). The challenger
selects a random bit \( b \in \{0, 1\} \) and sends the ciphertext \( C' = \text{Enc}(PP, id', r', m_b) \) to \( A \).

Query 2: The adversary \( A \) continues to do queries as in the Query 1.

Guess: At the end of the game, the adversary outputs a guess bit \( b' \). If \( b' = b \), set return = 1, otherwise return = 0. The adversary’s advantage is defined as

\[
\text{Adv}_A^\text{RIBE}(\lambda) = \left| \mathbb{P}[\text{return} = 1] - \frac{1}{2} \right|
\]

An RIBE scheme is ind-aID-CPA secure if for all PPT adversaries \( A \) in the above ind-aID-CPA game, the adversary’s advantage \( \text{Adv}_A^\text{RIBE}(\lambda) \) is negligible.

### IV. OUR CONSTRUCTION

In this section, we give our RIBE scheme. The main idea of our construction is as follows: we use a random vector \( \mathbf{u} \) to connect the user identity with the time, the vector \( \mathbf{u} \) is randomly divided into two vectors \( \mathbf{u}_{id,1}, \mathbf{u}_{id,2} \) corresponding to the user identity and the time respectively. When the decryption key of the user is generated, two extracted short vectors corresponding to the above two vector \( \mathbf{u}_{id,1}, \mathbf{u}_{id,2} \) are generated, which are used as secret key. The first secret key \( \mathbf{SK}_{id} := \mathbf{e}_{id} \in \mathbb{Z}_q^{2m} \) is connected with user identity \( id \), the second secret key \( \mathbf{e}_{id} \) is connected with time, which is changed with time, and the encrypted cipher-text is also dependent on time. To achieve the revocation, key update mechanism are used, the second secret key is updated with time. But for revoked user, he cannot get the latest update key. Our scheme is forward back secure, that is, if a user is revoked at time \( t \), the user will lost the decryption ability for the ciphertext encrypted after time \( t \), but the user still can decrypt the ciphertext encrypted before the time \( t \). Following is our construction.

1. setup(1^n, \( N \)). Take as input a security parameter \( n \), and a maximal number of users \( N \), output system parameters \( q, m, \sigma, \alpha \) specified in section 4.3 below, where \( q \) is a prime. Let the identity of a user be denoted as \( id = \{b_1, \ldots, b_q\} \in \{-1,1\}^q \), and the lifetime of the system is divided into \( T \) time periods labeled 0,1,..,\( T-1 \), let time \( t \) denote the time period, and denote it as \( t = \{t_1, \ldots, t_q\} \in \{-1,1\}^q \), and \( 0 \leq t \leq T-1 \), then do the following steps:

   1) Use the algorithm TrapGen(\( q, n \)) to generate a uniformly random matrix \( \mathbf{A}_q \in \mathbb{Z}_q^{n \times m} \) with a basis \( T_{\alpha} \) for

\[
\mathbb{A}_q(A_b) \quad \text{such that} \quad \| \mathbf{F}_{\alpha} \| \leq O(\sqrt{n \log q}) \quad \text{and} \quad \| \mathbf{T}_{\alpha} \| \leq O(n \log q).
\]

   2) Select \( l_1, l_2, 1 \) uniformly randomly

   3) Select a uniformly random vector \( \mathbf{u} \in \mathbb{Z}_q^m \)

   4) Let RL be revocation list of revoked identity, and RL be an empty set initially. Let system state \( ST \) be a collection of some user information in the current system, and set \( ST = \emptyset \) initially.

   5) Output \( ST, RL \), the public parameters \( PP \), and the master key \( MK \),

\[
PP = \{ \mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_k, \mathbf{C}_1, \ldots, \mathbf{C}_k, \mathbf{B}_1, \mathbf{B}_2, \mathbf{u} \}, \quad MK = \{ T_{\alpha} \}
\]

   2. PriKeyGen(\( PP, MK, id, ST \)) : Take as input \( PP, MK \), an identity \( id = \{b_1, \ldots, b_q\} \in \{-1,1\}^q \), and system state \( ST \), then runs the following steps:

   1) If the identity \( id \) is not in state \( ST \), then randomly select \( \mathbf{u}_{id,1} \sim \mathbb{Z}_q^m \) and set \( \mathbf{u}_{id,2} = \mathbf{u} - \mathbf{u}_{id,1} \), and store the item \( (id, \mathbf{u}_{id,1}, \mathbf{u}_{id,2}) \) into the state \( ST \), that is, to update the current state as \( ST := ST \cup \{ (id, \mathbf{u}_{id,1}, \mathbf{u}_{id,2}) \} \). If the identity \( id \) is in the state \( ST \), retrieves the item \( (id, \mathbf{u}_{id,1}, \mathbf{u}_{id,2}) \) from the state \( ST \).

   2) Set \( \mathbf{A}_id = \mathbf{B}_1 + \sum_{i=1}^k b_i \mathbf{A}_i \in \mathbb{Z}_q^{n \times m} \), where \( id = \{b_1, \ldots, b_q\} \in \{-1,1\}^q \).

   3) Sample \( \mathbf{e}_{id} \sim \text{SampleLeft}(\mathbf{A}_0, \mathbf{A}_id, T_{\alpha}, \mathbf{u}_{id,1}, \sigma) \), the distribution of \( \mathbf{e}_{id} \) is statistically close to \( D_{\mathbf{SK}_{id}}^{rev}(\mathbf{e}_{id}) \), where \( \mathbf{F}_{id} = \mathbf{A}_0 \| \mathbf{A}_id \), and \( \mathbf{e}_{id} \in \mathbb{Z}_q^m \) satisfies \( \mathbf{F}_{id} \mathbf{e}_{id} = \mathbf{u}_{id,1}, \| \mathbf{e}_{id} \| \leq \sigma \sqrt{2m} \).

   4) Output \( \mathbf{SK}_{id} = \mathbf{e}_{id} \in \mathbb{Z}_q^m \) as the secret key of the user with identity \( id \).

3. KeyUpd(\( PP, MK, t, RL, ST \)) : On input \( PP, MK \), the current time \( t = \{t_1, \ldots, t_q\} \in \{-1,1\}^q \), revocation list \( RL \), and state \( ST \), then runs the following steps:

   1) Set \( \mathbf{C}_t = \mathbf{B}_2 + \sum_{i=1}^{t_i} t_i \mathbf{C}_i \in \mathbb{Z}_q^{n \times m} \), where \( t = \{t_1, \ldots, t_q\} \in \{-1,1\}^q \).

   2) Define \( R \) as the set of revoked users before time \( t \). Here \( R \) is constructed from revocation list RL as follows, for all \( (id', t') \in RL \) and \( t' < t \), add \( id' \) to \( R \).
3) For all $id \not\in R$, retrieves $(id, u_{id,1,1}, u_{id,2,2})$ from the state ST, sample $e_{id,1} \leftarrow \text{SampleLeft}(A_0, C, T_{A_0}, u_{id,2,2}, \sigma)$, the distribution of $e_{id,1}$ is statistically close to $D_{A_0}^{(u_{id,1,1},C)}$, where $F_t = A_0 \| C_t$, and satisfies $F_t e_{id,1} = u_{id,2,2}$, $\| e_{id,1} \| \leq \sigma \sqrt{2m}$.

4) Output update keys $K_{U_{id,1}} := e_{id,1} \in \mathbb{Z}_q^{2m}$ at time $t$, and send it to the user with identity $id \not\in R$.

4. Encrypt($PP, id, t, \tilde{m}$) : On input $PP$, an identity $id = \{b_1, \cdots, b_t\} \in \{-1,1\}^t$, time $t = \{t_1, \cdots, t_l\} \in \{-1,1\}^l$, and a message $\tilde{m} \in \{0,1\}$, then runs following steps:

1) Parse $F_{t,1} := (A_0 \| \| A_1, C_1) \in \mathbb{Z}_q^{2m + n}$ where $A_{id} = B_1 + \sum_{i=1}^t b_i A_i \in \mathbb{Z}_q^{2m}$, $C = B_2 + \sum_{i=1}^t t_i C_i \in \mathbb{Z}_q^{2m}$.

2) Choose a random vector $s \leftarrow \mathbb{Z}_q^n$.

3) Choose $R_{id,1} = \sum_{i=1}^t b_i R_{id,i} \in \{-1,1, \cdots, l_1 \}^{n\times m}$.

4) Choose noise vectors $

\sigma_x \leftarrow \mathbb{Z}_q, \sigma_y \leftarrow \mathbb{Z}_q^n,$

set $z_1 = R_{id,1}^T y \in \mathbb{Z}_q^n, z_2 = R_{id,1}^T y \in \mathbb{Z}_q^n$, and set cipher-text
\[ c_0 \leftarrow u^T s + x + \tilde{m} \frac{q}{2} \in \mathbb{Z}_q^n, e_1 \leftarrow F_{t,1}^T s + z_1 \in \mathbb{Z}_q^n. \]

5) Output cipher-text $CT_{id,1} := (c_0, e_1) \in \mathbb{Z}_q \times \mathbb{Z}_q^{2m}$.

5. Decrypt($PP, CT_{id,1}, DK_{id,1}$) : On input $PP$, a decryption key $DK_{id,1} = \{SK_{id} = e_{id,1} \in \mathbb{Z}_q^{2m}, KU_{id,1} = e_{id,2} \in \mathbb{Z}_q^{2m}\}$, and a cipher-text $CT_{id,1}$, if the decryption key and the cipher-text correspond to different identities and times, then output terminator $\perp$, otherwise run as following steps:

1) Parse $e_1$ as $c_{1,0}, c_{1,1},$, where $c_{1,i} \in \mathbb{Z}_q^n, i = 0,1,2$.

2) Compute $w = c_0 - e_{id,1}^T [c_{1,0}, c_{1,1}, e_{id,2}] \in \mathbb{Z}_q^n$.

3) Compare $w$ and $\left\lfloor \frac{q}{2} \right\rfloor$, if $w - \frac{q}{2} \geq \frac{q}{4}$, then output $\tilde{m} = 1$, otherwise output $\tilde{m} = 0$.

6. KeyRev($id, t, RL$) : On input an identity $id = \{b_1, \cdots, b_t\} \in \{-1,1\}^t$, time $t = \{t_1, \cdots, t_l\} \in \{-1,1\}^l$, revocation list $RL$, the algorithm add $(id, t)$ to revocation list $RL$.

D. Correctness proof

The correctness of the scheme is as follows:
\[ w = c_0 - e_{id,1}^T \begin{bmatrix} c_{1,0} \\ c_{1,1} \\ e_{id,2} \end{bmatrix} - \frac{q}{2} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = u^T s + x + \tilde{m} \frac{q}{2} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} - e_{id,1}^T \begin{bmatrix} y \\ z_1 \\ z_2 \end{bmatrix} \]
\[ = u^T s + x + \tilde{m} \frac{q}{2} \begin{bmatrix} q \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} \end{bmatrix} - e_{id,1}^T \begin{bmatrix} y \\ z_1 \\ z_2 \end{bmatrix} \]
\[ = \tilde{m} \frac{q}{2} + x - e_{id,1}^T \begin{bmatrix} y \\ z_1 \\ z_2 \end{bmatrix}, \]

In the following we will prove that under suitable selecting of the system parameter, the above error term will satisfy
\[ |x - e_{id,1}^T \begin{bmatrix} y \\ z_1 \\ z_2 \end{bmatrix}| \leq \frac{q}{5}. \]

This ensures that our decryption algorithm is correct.

Theorem 2. The boundary of error term $x - e_{id,1}^T \begin{bmatrix} y \\ z_1 \\ z_2 \end{bmatrix}$ in the above correctness proofs is $\alpha q (1 + o(\sqrt{\log m})) O((l_1, l_1, \sigma m) + 1)$.

Proof: to prove the bounds of the above error term, for simplicity, denote $e_1 := e_{id,1}$, and parse $e_{id,1} = \begin{bmatrix} e_{id,1,1} \\ e_{id,1,2} \end{bmatrix}, e_1 = \begin{bmatrix} c_{1,0} \\ c_{1,1} \end{bmatrix},$ where $e_{id,1,1,1}, e_{id,1,2,1}, c_{1,0}, c_{1,1} \in \mathbb{Z}_q^n$. By the LeftSample algorithm we get $\| e_{id,1} \| \leq \sigma \sqrt{2m}$, $\| e_1 \| \leq \sigma \sqrt{2m}$, so $\| e_{id,1} \|, \| e_1 \|, \| e_{id,1} \|, \| e_1 \| \leq \sigma \sqrt{2m}$. Because of $R_{id} = \sum_{i=1}^l b_i R_i$, we have $\| R_{id,1} \| \leq \sum_{i=1}^l b_i R_i \| \leq l_1 O(\sqrt{m})$ by lemma 2, and similarly $\| R_{id,2} \| \leq l_1 O(\sqrt{m})$, so $\| R_{id,1} + R_{id,2} \| \leq \| e_{id,1} \| + \| R_{id,2} \| \leq \sigma \sqrt{2m} (1 + O(l_1, \sqrt{m})) \leq O(l_1, \sigma m)$.
\[ \|e_{i,t} + R_{e_{i,t}}\| \leq \|e_{i,t}\| + \|R_{e_{i,t}}\| \leq \sigma \sqrt{2m(1 + O(l_2 / \sqrt{m}))} \leq O(l_2 \sigma m) \]

By using lemma 3, and note that \( q > 2 \sqrt{m / \alpha} \), we have
\[
\begin{align*}
&\left| x - e_{i,t}^T \begin{bmatrix} y \\ z_i \end{bmatrix} - e_{d_{i,t}}^T \begin{bmatrix} y \\ z_i \end{bmatrix} \right| \\
&\leq |x| + \|e_{d_{i,t}} + R_{e_{d_{i,t}}}\| y + \|e_{i,t} + R_{e_{i,t}}\| y \\
&\leq 1 + q\alpha q(\log m) + \|e_{d_{i,t}} + R_{e_{d_{i,t}}}\| (\sqrt{m / 2} + q\alpha q(\log m)) \\
&\quad + \|e_{i,t} + R_{e_{i,t}}\| (\sqrt{m / 2} + q\alpha q(\log m)) \\
&= 1 + q\alpha q(\log m) + \|e_{d_{i,t}} + R_{e_{d_{i,t}}}\| + \|e_{i,t} + R_{e_{i,t}}\| (\sqrt{m / 2} + q\alpha q(\log m)) \\
&\leq (1 + \|e_{d_{i,t}} + R_{e_{d_{i,t}}}\| + \|e_{i,t} + R_{e_{i,t}}\| (\sqrt{m / 2} + q\alpha q(\log m)) \\
&\leq \left( O(l_1 + l_2) \sigma m \right) + 1 \left( \alpha q(1 + \alpha q(\log m)) O(l_1 + l_2) \sigma m + 1 \right)
\end{align*}
\]

So, the boundary of the error term is \( \alpha q(1 + \alpha q(\log m))(O(l_1 + l_2) \sigma m + 1) \).

E. Parameters setup
In order to guarantee the correctness of the scheme, the parameters should satisfy:
The error term in the decryption algorithm should less than \( q / 5 \), that is,
\[
\alpha < \frac{1}{5} \left( 1 + (\alpha q(\log m))(O(l_1 + l_2) \sigma m + 1) \right)^{-1}.
\]

TrapGen Algorithm must satisfy \( m > 2n \log q \).

For LeftSample Algorithm and RightSample Algorithm, \( \sigma \) is large enough, and satisfies
\[
\sigma > \max(l_1, l_2)m \log \left( \sqrt{m / \alpha} \right) = \max(l_1, l_2)m \log \left( \sqrt{m / \alpha} \right)
\]
Regev’s reduction process must meet \( q > 2 \sqrt{m / \alpha} \).

To meet the above requirements, the setting parameters are as follows:
1. \( m > 2n \log q \).
2. \( \sigma > max(l_1, l_2)m \log \left( \sqrt{m / \alpha} \right) \).
3. Noise parameter
\[
\alpha < \frac{1}{5} \left( 1 + (\alpha q(\log m))(O(l_1 + l_2) \sigma m + 1) \right)^{-1}.
\]
4. Modulus \( \theta \) is a prime and satisfies
\[
q > 10 \sqrt{m(1 + (\alpha q(\log m))(O(l_1 + l_2) \sigma m + 1))}.
\]

F. Security proof
Theorem 3. Our construction is ind-aID-cPA secure under the \((\mathbb{Z}_q, n, \overline{\Phi}_q) - \text{LWE}\) assumption.

Proof: Our proof proceeds with 4 games, the first game is the same as the ind-aID-CPA game in section 3.2 and the adversary has no advantage in last game under the \((\mathbb{Z}_q, n, \overline{\Phi}_q) - \text{LWE}\) assumption.

As long as no probabilistic polynomial time adversary \( A \) is able to distinguish any two games. In game \( i \), let \( W_i \) denote the event that the adversary wins the game. That is, the adversary guesses the challenge bit correctly, where \( i = 1, 2, 3, 4 \).

Game 0. The game is the ind-aID-CPA game that defines in section 3.2. The challenger selects \( l_1 + l_2 + 2 \) random matrices and generates public parameter PP and master key MK. In challenge phase, the challenger sets the challenge ciphertext \( CT^* = \text{Enc}(PP, i^{d_{i,t}}, t^*, m_h) \). Set \( R_i' \in \{-1,1\}^{m \times n} \) for \( i = 1, \ldots, l_1 + l_2 \) that denote \( l_1 + l_2 \) temporary random matrices in creating ciphertext \( CT^* \).

Game 1. In this game, the challenger changes the way of generating \( A_i, C_j \) in the public parameters, and selects \( l_1 + l_2 \) random scalar \( h_i \in \mathbb{Z}_q \) for \( i \in \{1, \ldots, l_1 + l_2\} \), then the challenger generates \( A_0, B_j, B_i \) like game 0.

For \( i \in \{1, \ldots, l_1\}, j \in \{1, \ldots, l_2\} \), we construct matrices \( A_i, C_j \) as follow:
\[
\begin{align*}
A_i & \leftarrow A_0 \cdot R_i' + h_i \cdot B_i \in \mathbb{Z}_q^{m \times n} \\
C_j & \leftarrow A_0 \cdot R_j' + h_j \cdot B_i \in \mathbb{Z}_q^{m \times n}
\end{align*}
\]
(1)

The remained of the game is unchanged. Note the matrices \( R_i' \in \{-1,1\}^{m \times n} \) are the same as in game 0 for \( i = 1, \ldots, l_1 + l_2 \).

Next, we prove that game 0 and game 1 is statistically indistinguishable by using lemma 1. Note that in game 1 the matrices \( R_i' \in \{-1,1\}^{m \times n} \) are used only for construction of \( A_i, C_j \), \( CT^* \), and error terms \( z_1 \leftarrow (R_{d_0})^T y, z_2 \leftarrow (R_i')^T y \), where \( R_{d_0} = \sum_{h_i} h_i R_i' \), \( R_i' = \sum_{h_i} t_i R_{d_{h_i}} \). Let \( R' = (R_1' \cdots R_{l_1 + l_2}') \in \mathbb{Z}_q^{m \times (l_1 + l_2)m} \), by using lemma 1, the distribution \((A_0, A_0 R_{i'}^*, (R_i')^T y)\) is statistically close to the distribution \((A_0, A_0 A_0, \cdots, A_0 A_0 A_0, \cdots, A_0 A_0 A_0, \cdots, A_0 R_{i_n}^*, (R_i')^T y)\), where \( A_i, C_j \) is random and uniform for \( i \in \{l_1\}, j \in \{l_2\} \).

Next, let \( z = (z_1, z_2) \), by using lemma 1, the distribution \((A_0, A_0 R_{i_1}^*, \cdots, A_0 R_{i_n}^*, A_0 R_{i_n+1}^*, \cdots, A_0 R_{i_n+1}^*, \cdots, z)\)
is statistically the same as the distribution \( (A_0, A_1', \cdots A_j', C_1', \cdots C_h', z) \). Then in the stand of the adversary’s view, the matrices \( A_i, R_i \) are statistically the same as uniform and independent of \( z \), therefore \( A_i, C_j \) as defined in the above equation (1) are close to uniform. This means that from the adversary’s view, they are random and independent uniform matrix, as in game 0. Thus proves \( \text{Pr}[W_0] = \text{Pr}[W_1] \).

**Game 2.** Game 2 is the same as game 1 except that an abort event is added. Let \( Q^{(d)} \) be the queries number of private key, and \( |T| \) be the queries number of update key. We suppose that adversary \( A \) queries the update key for all time \( t < |T| \), and we assume \( (t_1, \cdots, t_\gamma) \) is the time queries tuple, which is listed in an increasing order. The challenger of game 2 does as follows:

1) The setup phase is the same as in game 1, the difference is that the challenger also selects two hash functions \( H_1 \in \mathcal{H}_1, H_2 \in \mathcal{H}_2 \), and keeps them to itself.

2) The challenger replies to the update key queries and submits the challenge cipher-text the same as in game 1.

3) In the game 2, the challenger randomly selects \( i' \in [|T|] \)

4) In the game 2, the challenger guesses one of the two following adversary types:

**Type-1 adversary:** \( A \) makes private key query for the challenged identity \( id^* \), so \( id^* \) should be revoked before time \( t' \).

**Type-2 adversary:** \( A \) does not query private key for the challenged identity \( id^* \).

5) The probability that the challenger guesses correctly is \( \frac{1}{2} \), if the guess is Type-1, the Challenger randomly selects \( j' \in [Q^{(d)}] \), and suppose that adversary’s \( j' \)-th private key query is for \( id^* \), the probability that the challenger guesses it correctly is \( \frac{1}{2} \).

6) Let \( Q = Q^{(d)} + |T| \), in the final phase, the adversary outputs a guess \( r' \in \{0, 1\} \). The challenger behaves as following:

a) \( q_i = (t_i, \cdots, t_{\gamma}), q_j = (id_1, \cdots, id_{Q^{(d)}}) \), where \( (t_1, \cdots, t_\gamma) \) is queried time tuple and \( (id_1, \cdots, id_{Q^{(d)}}) \) is the identity tuple respectively. Set

\[
J_b = \begin{cases} \{i'\}, & \text{if } A \text{ is type-2, } b=2; \\ \{i', j'\}, & \text{if } A \text{ is type-1, } b=1. \\
\end{cases}
\]

b) Abort check: on input \( q_1, q_2 \), checks if \( H_i \) satisfies \( E_{A_i}(H_1), E_{A_j}(H_1) \), and checks if \( H_2 \) satisfies \( E_{A_i}(H_2) \). If any condition does not satisfies, the challenger rewrites \( r' \) with a new random bit in \( \{0, 1\} \) and aborts the games. Note that the adversary does not known \( H_1, H_2 \), so he does not known if the game ends with an abort.

c) Artificial abort: the challenger flips a bit \( \Gamma \in \{0, 1\} \) with \( \text{Pr}[\Gamma = 1] = \gamma(q_i), i = 1, 2 \), where the detail analysis and the definition of function \( \gamma() \) are referenced to [1]. If \( \Gamma = 1 \), the challenger rewrites \( r' \) with a new random bit in \( \{0, 1\} \) and aborts the games [1].

Note that the abort condition is decided by hash functions \( H_1, H_2 \), and they are independent of the adversary’s view. For queries tuple \( q_i, i = 1, 2 \), let \( \epsilon(q_i) \) be the probability that an abort event does not happens. Let \( \epsilon_{\text{max}}^{(b)}, \epsilon_{\text{min}}^{(b)} \) are scalars such that \( \epsilon(q_i) \in [\epsilon_{\text{min}}^{(b)}, \epsilon_{\text{max}}^{(b)}] \) for the case \( A \) is guesses type-b, \( b = 1, 2 \). Set

\[
\Delta_b = \begin{cases} 
\frac{1}{2 |T|}, & b = 2 \\
\frac{1}{2 |T| Q^{(d)}}, & b = 1 
\end{cases}
\]

The above equation denotes the probability that the challenger is correctly guess the type of adversary.

If there is no artificial abort, then \( \epsilon_{\text{min}}^{(b)} = \Delta_b \cdot \alpha_{h_{\text{min}}}, \epsilon_{\text{max}}^{(b)} = \Delta_b \cdot \alpha_{h_{\text{max}}} \), by lemma 4

\[
\epsilon_{\text{max}}^{(b)} - \epsilon_{\text{min}}^{(b)} = \left\{ \begin{array}{ll}
(2Q / q^4 - Q^2 / q^4), & b = 1 \\
2Q / q^4 - b = 2 &
\end{array} \right.
\]

This is non-negligible and fails to lead to a good lower bound on \( \text{Pr}[W_2] - \frac{1}{2} \). Thus we apply Water’s approach. With the artificial abort, \( \epsilon_{\text{max}}^{(b)} - \epsilon_{\text{min}}^{(b)} < \epsilon_{\text{min}}^{(b)} \text{Pr}[W_2] - \frac{1}{2} \) and therefore

\[
\text{Pr}[W_2] - \frac{1}{2} \geq \frac{1}{2} \text{Pr}[W_2] - \frac{1}{2} \frac{1}{2} (\epsilon_{\text{max}}^{(b)} - \epsilon_{\text{min}}^{(b)}) \geq \frac{1}{2} \frac{1}{2} (\epsilon_{\text{max}}^{(b)} - \epsilon_{\text{min}}^{(b)}) \text{Pr}[W_1] - \frac{1}{2}
\]

The above proof uses lemma 5.

**Lemma 5 ([1], lemma 28).** For type-b adversary, we have that

\[
\text{Pr}[W_2] - \frac{1}{2} \geq \frac{1}{2} \text{Pr}[W_2] - \frac{1}{2} \frac{1}{2} (\epsilon_{\text{max}}^{(b)} - \epsilon_{\text{min}}^{(b)})
\]
Therefore, we have
\[
\begin{align*}
\Pr[W_2] - \frac{1}{2} & \geq \frac{1}{8q^2T} \left( \Pr[W_1] - \frac{1}{2} \right), \quad b = 1 \\
\Pr[W_2] - \frac{1}{2} & \geq \frac{1}{8qT} \left( \Pr[W_1] - \frac{1}{2} \right), \quad b = 2
\end{align*}
\]
From the above, Game 1 and game 2 can be distinguished with the negligible advantages.

**Game 3.** We now give a new method to select \(A_0, B_1, B_2 \) in game 2.

The challenger randomly generates \( A_0 \) in \( Z_q^{n\times m} \), but generates \( B_1, B_2 \) by using of TrapGen algorithm such that \( B_1, B_2 \) are random matrices in \( Z_q^{n\times m} \) and the challenger owns trapdoors \( T_{A_0} \) \( T_{B_1} \) for \( \Lambda_q^+(B_1) \) \( \Lambda_q^+(B_2) \) respectively. The construction of \( A_i, C_j \) is the same as game 2 for \( i = 1, \cdots , l_1 \), \( j = 1, \cdots , l_2 \), as follows:

\[
\begin{align*}
A_i & \leftarrow A_0 \cdot R_i + h_i \cdot B_i \in Z_q^{n \times m} \\
C_j & \leftarrow A_0 \cdot R_{i,j} + h_{i,j} \cdot B_j \in Z_q^{n \times m}
\end{align*}
\]

**The query of the private key.** To reply a private key query for \( id = \{b_1, \cdots , b_i \} \in \{-1,1\}^i \) the challenger uses the trapdoor \( T_{A_0} \) and then runs the following steps:

1) If the identity \( id \) is not in state ST, then uniformly random select \( u_{id,2} \leftarrow Z_q^l \), set \( u_{id,1} = u - u_{id,1} \), and store \((id, u_{id,1}, u_{id,2})\) in state ST, if the identity \( id \) is in state ST, retrieval it \((id, u_{id,1}, u_{id,2})\) from ST.

2) Construct \( R_{id} = \sum_{j=1}^{l_1} h_{i,j} R_{i,j} \in Z_q^{n \times m}, h_d' = 1 + \sum_{j=1}^{l_2} h_{i,j} \in Z_q \) in the equation (1), next set

\[
A_{id} = B_1 + \sum_{i=1}^{l_1} h_i A_i = A_0 R_{id} + h_d B_1 \in Z_q^{n \times m}.
\]

Note that \( h_d = H_{b_0}(id) \), where \( H_{b_0} \) is the hash function in \( H_{\text{valid}} \) defined by \( h_i = (h_{i,1}, \cdots , h_{i,l_2}) \).

3) If \( h_{id} = 0 \), abort, and the adversary outputs a bit \( r' \in \{0,1\} \) randomly just as in game 2. Else, go to the next step.

4) Sample \( e_{id} \leftarrow \text{SampleRight}(A_{id}, h_{id} B_1, R_{id}, T_{B_1}, u_{id,1}, \sigma) \in Z_q^{2m} \), the distribution of \( e_{id} \) is statistically close to \( D_{\Lambda_q^+(B_1), \sigma} \), where \( F_{id} = A_0 \parallel A_{id} \), and \( \| e_{id} \| \leq \sigma \sqrt{2m} \).

5) Output SK_{id} = e_{id} \in Z_q^{2m} of identity \( id \) and updated state ST.

**The query of the update key.** To respond an update key query for \( id = \{b_1, \cdots , b_i \} \in \{-1,1\}^i \) the challenger b uses the trapdoor \( T_{B_1} \) and then runs the following steps:

1) Define \( R \) as a set consisting of revoked users in time \( t = \{t_1, \cdots , t_{l_2}\} \in \{-1,1\}^l \), that is, for any \( t' \leq t \) if exist \((id', t') \in R \), then add \((id', t')\) to \( R \).

2) Construct \( R_i' = \sum_{j=1}^{l_1} f_i R_{i,j} \in Z_q^{n \times m}, h_i' = 1 + \sum_{j=1}^{l_2} h_{i,j} \in Z_q \) in the equation (1), next set

\[
C_i = B_2 + \sum_{j=1}^{l_2} f_i C_{i,j} = A_0 R_i' + h_i B_1 \in Z_q^{n \times m}.
\]

Note that \( h_i = H_{b_0}(t_i) \), where \( H_{b_0} \) is the hash function in \( H_{\text{valid}} \) defined by \( h_i = (h_{i,1}, \cdots , h_{i,l_2}) \).

3) If \( h_i = 0 \), abort, and the adversary outputs a bit \( r' \in \{0,1\} \) randomly just as in game 2. Else, go to the next step.

4) For all \( id \not\in R \), retrieval \((id, u_{id,1}, u_{id,2})\) from ST, Sample \( e_{id,1} \leftarrow \text{SampleRight}(A_{id}, h_{id} B_2, R_i', T_{B_1}, u_{id,1}, \sigma) \), the distribution of \( e_{id,1} \) is statistically close to \( D_{\Lambda_q^+(B_1), \sigma} \), where \( F_i = A_0 [C_i \parallel \sigma] \), and \( \| e_{id,1} \| \leq \sigma \sqrt{2m} \).

5) Output update keys \( KU_{id} = e_{id,1} \in Z_q^{2m} \) in time \( t \).

For the rest part, the game 3 is the same with the game 2.

In the challenge phase, the challenger do the respond as follows.

a) If adversary is type-2, the challenger examines whether \( t' = \{t_1, \cdots , t_{l_2}\} \in \{-1,1\}^l \) satisfies \( h_i' = 1 + \sum_{j=1}^{l_2} h_{i,j} = 0 \). If not, the challenger aborts.

b) If adversary is type-1, the challenger examines whether \( t' = \{t_1, \cdots , t_{l_2}\} \in \{-1,1\}^l \) satisfies \( h_i' = 1 + \sum_{j=1}^{l_2} h_{i,j} = 0 \) and together with \( id' = \{b_1, \cdots , b_i\} \in \{-1,1\}^l \) satisfies \( h_{id'} = 1 + \sum_{j=1}^{l_2} h_{i,j} = 0 \). If not, the challenger aborts.

Next, we show that the outputs of SampleRight in game 3 are indistinguishable from the outputs of SampleLeft in game 2. First, we note that the respond to the private key query, since \( h_{id} \) is non-zero in step 3 of PriKeyGen, the matrix \( T_{B_1} \) is a trapdoor of \( h_{id} B_1 \). Theorem 3 describes when
Reduction from LWE. Given an adversary can distinguish game 3 from game 4 with a non-negligible advantage, we will design an algorithm by using of the adversary to solve LWE problem. The challenger proceeds as follows:

**Instance.** $B$ requests samples from $O$ and gets pair $(u_i, v_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^m$ for each $i = 0, \ldots , m - 1$.

**Setup.** $B$ constructs public parameters $P$ in the following step:

1. Assemble the random matrix $A_0$ using the $i$-th column of $A_0$ be vector $u$, for $i = 1, \ldots , m$, where $u$, from the previously given LWE pair $(u_i, v_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^m$.
2. Assign the public random $n$-vector $u = u_0 \in \mathbb{Z}_q^n$.
3. Construct the public parameters $A_0, C_j, B_1, B_2$ as in game 3 using $h_i, s_i, R_i^*$.

**Query.** $B$ answers each private key and key update queries as in game 3 include the abort events.

**Challenge.** When $A$ sends a challenge message bit $b' \in \{0, 1\}$, a challenge identity and time pair $(d', t') \in \{-1, 1\}^2$, let $id' = (b', \ldots , b')$, $t' = (t', \ldots , t')$. Then $B$ constructs a challenge cipher-text corresponding $(id', t')$ as following:

1. Let $v_0, \ldots , v_m$ be entries of LWE samples.

$$v = \begin{bmatrix} v_0 \\ \vdots \\ v_m \end{bmatrix} \in \mathbb{Z}_q^n \quad \text{and} \quad y = \begin{bmatrix} y_0 \\ \vdots \\ y_m \end{bmatrix} \in \mathbb{Z}_q^m$$

2. Let $c_0^* = v_0 + b' \frac{Q}{2} \in \mathbb{Z}_q$.

3. Set $R_i^* := \sum_{j=1}^h b_j R_i^j, R_i^* := \sum_{j=1}^h t_j R_i^j, R_i^*$, where $R_i^*$, $i > 0$ is generated in setup phase.

4. Set $c_1^* = \begin{bmatrix} R_i^* + (R_i^*)^T v \\ (R_i^*)^T v \end{bmatrix} \in \mathbb{Z}_q^m$.

5. Send cipher-text $CT = (c_0^*, c_1^*)$ to adversary $A$.

Next, we demonstrate that if the LWE oracle is pseudorandom (i.e. $O = O_1$) then the distribution of $CT$ is the same as in game 3 only when the abort event not happens. First, since $h_i = 0, h_i = 0$, we have

$$F_{a'd', t'} = (A_0 B_1 + \frac{1}{2} b_j^* A_0 B_2 + \frac{1}{2} t_j^* C_j)$$

Second, from the definition of $O$, we conclude $v = A_0 s + y$, where noise vector $y \in \mathbb{Z}_q^m$ is distributed with $\mathcal{N}_q^m$. Therefore, $c_i^*$ defined in step (3) above satisfies

$$c_1^* = \begin{bmatrix} v^* \\ (R_i^*)^T v^* \end{bmatrix} = \begin{bmatrix} A_i s + y \\ (R_i^*)^T A_i s + (R_i^*)^T y \\ (R_i^*)^T y \\ A_i^* s + y \\ (A_0 R_i^*)^T s + (R_i^*)^T y \\ (A_0 R_i^*)^T s + (R_i^*)^T y \end{bmatrix}$$

And the right of the above equation is precisely the $c_1$ part of a valid challenge cipher-text in game 3. Note that $v_0 = u_0^* s + x$, just as the $c_0$ part of a valid challenge cipher-text in game 3.

When $O = O_1$ we have that $v_0$ is uniform and independent in $\mathbb{Z}_q$ and $v$ is uniform in $\mathbb{Z}_q^n$. Therefore defined in above step 3) is uniform and independent in $\mathbb{Z}_q^m$. Hence, the challenge cipher-text is uniform in $\mathbb{Z}_q \times \mathbb{Z}_q^m$, just as it is in game 4.

**Guess.** The adversary guesses that it is a game 3 or game 4 challengers interacting with. The challenger decided to solve LWE problem using the answer of adversary.
Note that when $O = O_3$ the adversary’s view is just the same as in game 3. When $O = O_1$ it is the same as in game 4. Hence, the challenger’s advantage to solve LWE problem is just the same as the adversary’s advantage to distinguish game 3 and game 4.

V. PERFORMANCE ANALYSES

The efficiency of our scheme is compared with the schemes [22, 24, 25, 27, 28] in Table II.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Public key length</th>
<th>Message-cipher-text expansion factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>[22]</td>
<td>$(dm^2 + mn) log q$</td>
<td>$mn log q$</td>
</tr>
<tr>
<td>[24]</td>
<td>$(mn + (l + 1)n) log q$</td>
<td>$(m + 1) log q$</td>
</tr>
<tr>
<td>[25]</td>
<td>$(5mn + n) log q$</td>
<td>$(3m + 1) log q$</td>
</tr>
<tr>
<td>[27]</td>
<td>$((l + 1)mn) log q$</td>
<td>$(l + 1)^2 m^2$</td>
</tr>
<tr>
<td>[28]</td>
<td>$(n^2 + 1) log q$</td>
<td>$(n + 1) log q$</td>
</tr>
<tr>
<td>Our scheme</td>
<td>$((l + l_2 + 2)mn + n) log q$</td>
<td>$(3m + 1) log q$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Private key length</th>
<th>Update key length</th>
</tr>
</thead>
<tbody>
<tr>
<td>[22]</td>
<td>$(mn) log q$</td>
<td>no</td>
</tr>
<tr>
<td>[24]</td>
<td>$(mn) log q$</td>
<td>no</td>
</tr>
<tr>
<td>[25]</td>
<td>$(2m) log q$</td>
<td>$(2m) log q$</td>
</tr>
<tr>
<td>[27]</td>
<td>$((l + 1)mn) log q$</td>
<td>no</td>
</tr>
<tr>
<td>[28]</td>
<td>$(n) log q$</td>
<td>no</td>
</tr>
<tr>
<td>Our scheme</td>
<td>$(2m) log q$</td>
<td>$(2m) log q$</td>
</tr>
</tbody>
</table>

In the tables II, the meaning of each symbol is illustrated as follows: $d$ is the maximum hierarchical depth, $l$ is the length of the identity information, $l_2$ is the length of the variable time.

From the table II we can see the public key space of our scheme is relatively longer compared with the existing schemes, this is because in our scheme we considered both the identity of the user and the variable time, while in other schemes except for scheme [25] the variable time is not involved only the identity of the user is associated with the cipher-text. The cipher-text length of our scheme is in the normal range and it is equals to the length of the cipher-text in the scheme [25]. The efficiency of our scheme is much similar to the scheme in [25].

VI. CONCLUSIONS

In this paper, a revocable identity-based encryption scheme from lattice is proposed in the standard model, and it is proved to be secure against adaptive-ID attacks under LWE assumption. Revocable identity-based encryption scheme from lattice is a new topic, for revocation in our scheme we use the method of revocation list, for other revocation technique such as minimum cover set still need more research works to be done, which will be a good research direction.

REFERENCES


Yaling Zhang received her B.S. degree in computer science in 1988 from Northwest University, Xi’an, China. She received her B.S. degree in computer science in 2001, and earned her Ph.D. degree in mechanism electron engineering in 2008, both from the Xi’an University of Technology, Xi’an, China. Currently, she is a professor in Xi’an University of Technology. Her current research interests include cryptography and network security.

Shangping Wang received his B.S. degree in mathematics in 1982 from Xi’an University of Technology, Xi’an, China. He received his M.S. degree in applied mathematics in 1989 from Xi’an Jiaotong University, Xi’an, China, and earned his Ph.D. degree in cryptology in 2003 from Xi’an University, Xi’an, China. Currently, he is a professor in Xi’an University of Technology. His current research interests are cryptography and information security.

Qing Du received her B.S. degree in mathematics in 2013 from Xi’an University of Technology, Xi’an, China. She received his M.S. degree in applied mathematics in 2015 from Xi’an University of Technology, Xi’an, China. Currently she is an IT worker in Xi’an, China. Her current research interests are cryptography and information security.