# EFFICIENT EQUALITY TEST TECHNIQUE USING IDENTITY-BASED ENCRYPTION FOR TELEMEDICINE SYSTEMS

Chenguang Wang and Huiyan Chen

Department of Cryptozoology and Technology, Beijing Electronic Science and Technology Institute, Beijing, China

***ABSTRACT***

*Telemedicine systems play an important role in early HIV screening, but data privacy in the medical system has always been a challenging issue. For data privacy in the medical system, using identity encryption-based equivalence testing schemes to protect private data and screen for early AIDS has important prospects. The aim of this paper is to address the challenge of data privacy protection in medical systems by proposing a novel identity encryption scheme. The main subject revolves around the inefficiencies and lack of support for revocability and quantum resistance in existing identity encryption-based equality test schemes. My achievement lies in developing the first equality test scheme that supports both revocable encryption and quantum resistance, offering superior memory and computational performance compared to other schemes.*

***KEYWORDS***

*Telemedicine Systems, Equality test, Identity Encryption.*

## 1. INTRODUCTION

Under the public key infrastructure system PKI, to ensure users' public key authentication, there is a trusted third-party certification center CA under this system. The CA binds each user's identity information and public key information together. However, managing digital certificates often consumes a lot of resources, so in order to reduce the complexity of digital certificate management, A. Shamir et al.[1] proposed the concept of identity encryption and gave a preliminary identity-based signature scheme. This solution uses the user's identity information as the public key and does not require a digital certificate to verify the authenticity of the public key, thereby reducing the complexity of digital certificate management under the PKI system.

In 2001, D. Boneh et al.[2] proposed an identity encryption scheme based on bilinear mapping, which proved that the selected plaintext is safe under the random fable model under the assumption of the Diffie-Hellman problem. J. Horwitz et al.[3] proposed a two-layer hierarchical identity encryption scheme, which consists of a root private key generator (PKG), a domain PKG and a user, and these three components are all associated with user identity information, and at the same time proved the selected ciphertext is safe under the random oracle model. D. Boneh et al.[4] proposed an efficient identity encryption scheme, which proved that the selected identity is safe under the random fable model. A. Sahai et al.[5] proposed a fuzzy identity encryption scheme, which realizes fuzzy encryption of identities under multiple attributes and has good fault tolerance and anti-collusion attack capabilities. J. Baek et al.[6] proposed a multi-recipient

David C. Wyld et al. (Eds): SEC, NLPML, CYBI, SIGL, CRIS, AIAPP, CoSIT- 2024
pp. 77-92, 2024. CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.140707

identity encryption scheme that uses one pairing computation to encrypt a single message for n recipients and proved that the scheme is adaptively secure. X. Boyen[7] et al. proposed an anonymous identity encryption scheme that realizes the anonymity of ciphertext and the delegation of hierarchical keys. It is the first hierarchical identity encryption scheme that achieves complete anonymity in a hierarchical structure. A. Boldyreva et al.[8] proposed a revocable identity encryption scheme, which is based on the ideas of fuzzy IBE primitives and binary tree data structures, and significantly improves the efficiency of key update. S. S. Chow et al.[9] designed the first leak-proof identity encryption scheme under the standard model, which solved the elastic leakage problem under the standard model while retaining the efficiency of the original scheme. J. Chen[10] et al. proposed a lattice-based revocable identity encryption scheme, which uses trapdoor technology and key revocation to achieve key revocation, and is proven to be selectively secure under the standard model and LWE model. J. Kim et al.[11] proposed an identity broadcast encryption system based on adaptive security, which uses dual-system encryption technology and proved to be adaptively secure under the assumption of general sub-policy. S. Park[12] et al. designed a new revocable identity encryption scheme, which combines a hierarchical identity encryption scheme and a multi-linear mapping public key broadcast encryption scheme, and uses the PKBE scheme for revocation in bilinear mapping to achieve short-term revocation keys and update keys. J. Yu et al.[13] proposed an identity encryption scheme that is resistant to intrusion. Compared with other schemes, the ciphertext of any time period is safe and achieves stronger security. J. Zhang et al.[14] proposed an efficient lattice-based identity encryption scheme, which uses programmable hash functions to shorten the key length and improve computing performance. In order to shorten the key length of the traditional identity encryption scheme, S. Yamada[15] et al. used a verifiable random function with a large input space to construct the scheme, and used a new partitioning technology to compress the parameter length. J. Wei et al.[16] proposed a revocable identity encryption scheme applied to cloud computing scenarios. This scheme achieves forward or backward security by introducing the functions of user revocation and ciphertext update, and also has advantages in performance and efficiency. X. Zhang et al.[17] proposed an agent-based identity encryption scheme for cloud storage scenarios, which uses encryption and original image sampling technology to resist keyword guessing attacks inside the cloud server under error learning conditions. C. Ge et al.[18] proposed a revocable identity-based broadcast proxy re-encryption scheme, in which the proxy can revoke a set of delegations specified by the principal from the re-encryption key. Y. Sun et al.[19] proposed a revocable identity encryption scheme for the Internet of Things. The scheme uses the SM9 encryption algorithm to protect data privacy in the Internet of Things, and proved the security of the scheme based on the Diffe-Hellman problem. J. Zhang et al.[20] proposed an identity-based broadcast proxy re-encryption scheme applied to the Internet of Vehicles scenario, which protects data privacy through completely anonymous data sharing.

## 2. RELATED WORK

In order to solve the problem of large number of users in cloud computing scenarios[21-25], different users use different public keys for encryption. For example, in a telemedicine social system, if two patients with the same symptoms want to communicate, they only need to upload their respective ciphertexts and trapdoors to the cloud server, and the cloud server will treat different patients without decrypting the ciphertexts. By performing an equivalence test on the ciphertext, patients with the same symptoms can be matched. G. Yang et al.[26] first proposed the concept of equivalence testing based on identity encryption in 2010. This scheme allows anyone to conduct equivalence testing on ciphertext encrypted by two users using different keys. Q. Tang, [27] et al. proposed an equivalence testing scheme based on identity encryption that supports fine-grained authorization. In this scheme, only authorized agents can perform equivalence testing on ciphertext. D. H. Duong et al.[28] proposed a specific construction of a lattice-based equivalence test scheme under the standard model, which has higher performance

and is proven to be resistant to security attacks. G. L. D. Nguyen et al.[29] proposed an equivalence test scheme based on identity encryption based on error learning problem. Compared with the previous scheme, this scheme uses flexible authorization to enhance the privacy protection of the scheme. Z. Yang et al.[30] proposed an efficient identity encryption-based equivalence test scheme for cloud computing scenarios. This scheme improves computing performance by embedding the hash value of the plaintext into the test trapdoor and proves that it is in quantum security. The model is one-way secure against chosen ciphertext attacks.

## 3. OUR CONTRIBUTION

There has been a lot of research on identity encryption, and there are also a lot of applications for the Internet of Things and cloud computing. There are also some related studies on special applications of equality test, but there are some shortcomings in terms of performance, so this article has made some improvements to address these shortcomings. In summary, the research contributions of this paper are as follows:

(1) This paper proposes for the first time an equivalence test scheme with revocable identity-based encryption based on lattice cryptography. Compared with other equivalence test schemes, the design of this scheme does not embed plaintext hash values as input in the encryption algorithm. In the equality testing algorithm, the hash value of the plaintext is embedded as input. Through this design, an efficient identity encryption equivalence test scheme is proposed, and the scheme in this article is applied to the telemedicine system.

(2) The encryption algorithm in our scheme uses a programmable hash function to compress the key length. In performance test experiments, our scheme has more efficient computing performance and storage performance than other schemes.

(3) The scheme in this paper is based on the difficulty of the LWE problem, which proves that the selected ciphertext is safe under the random fable model and can resist post-quantum attacks, which ensures that the solution is safe and reliable when applied to telemedicine systems.

## 4. PRELIMINARIES

**Notation.**In this study, we introduce a negligible function that is less than any polynomial fraction for all sufficiently large *n*. An event is considered to have an overwhelming probability of occurring if it happens with a probability $1 - \epsilon(n)$ greater than or equal to that of some negligible function. Additionally, we denote matrices with uppercase letters.

### 4.1. The RLWE Hardness Assumption

Difficult problems on the grid mainly include the shortest vector problem (SVP), the closest vector problem (CVP), the small integer solution problem (SIS), and the learning problem with error (LWE problem). Difficult problems are generally divided into decision-type and search-type. Decision-type difficult problems are generally oriented to distinguish variables in difficult problems from randomly selected variables, while search-type difficult problems are oriented to solve difficult problems. The hard problem adopted in this paper is the hard problem based on LWE.

Definition 1 (**Decision LWE**). Given that the polynomial matrix $A \in Z_q^{m \times n}$ satisfies $b = As + e$, where $s \in Z_q^n$ and $e$ is an error polynomial vector satisfying a discrete Gaussian distribution, then

distinguishing $(A, B = As + e)$ from uniformly randomly selected (a,b) is a problem solved by decision-type RLWE.

Definition 4 (**Search LWE**). Given the polynomial matrix $A \in Z_q^{m\times n}$ and $b \in Z_q^n$, and e is an error polynomial vector satisfying the discrete Gaussian distribution, then the search-type RLWE problem is solved by finding the vector s satisfying $b = As + e$.

## 4.1. Integer Lattice and Ideal Lattice

Definition 2 (**Integer Lattice**). Assuming that q is a prime number, given a matrix $A \in Z_q^{m\times n}$ and a vector $u \in Z_q^n$,define the integer lattice as:

$$\Lambda_q(A) = \{e \in Z^m \, s.t. \exists \, s \in Z_q^n \text{ where } A^T s = e \bmod q\}$$
$$\Lambda_q^{\perp}(A) = \{e \in Z^m \, s.t. \; A^T e = 0 \bmod q\}$$
$$\Lambda_q^{u}(A) = \{e \in Z^m \, s.t. \; A^T e = u \bmod q\}$$

## 4.2. IBE

The identity encryption system has four algorithms, namely initialization (Setup), key generation (Extract), encryption (Encrypt), and decryption (Decrypt). The main function of the initialization algorithm is to generate the public key PP and master key MK of the system. This algorithm is executed by PKG, and only PKG holds the master key. In the key generation stage, the master key is used to generate the private key corresponding to the user id. This algorithm is also executed by PKG. If the user needs to decrypt the operation, he needs to apply to PKG to obtain the private key. The encryption stage is to use the public key to encrypt the plaintext message, and the encryption algorithm is generally executed by the message sender. The decryption algorithm uses the user's private key to decrypt the ciphertext, and the decryption is generally performed by the message receiver. For identity information $id = (id_1, id_2, \ldots, id_l)$,describe the IBE system as follows:

**Setup($\lambda$)**：On input the public parameter λ, the algorithm outputs the public parameter PP, and the master key MK。

**Extract(PP,MK,id)**: On input the public parameter PP, identity $id \in \mathbb{Z}_q^n$, master key MK, and output the corresponding private key $SK_{id}$.

**Encrypt(PP, μ, id)**: On input the public parameter PP, an identity information id and a message μ, and the algorithm outputs a ciphertext CT.

**Decrypt(PP,CT, $SK_{id}$)**: On input public parameters PP, ciphertext CT, private key $SK_{id}$, and the algorithm output message μ.

**Security Game.** Provable security is to link the security of a cryptographic scheme with a specific difficult problem, and reduce the security of a cryptographic scheme to a specific mathematically difficult problem. The mathematically difficult problem used in the security proof in this paper is the RLWE problem. We use an indistinguishable stochastic countermeasure to define adaptive security. Where $M_\lambda$ and $C_\lambda$ are plaintext space and ciphertext space respectively. The specific security proof process is as follows:

**Setup**: The challenger runs Setup(λ) and sends the public parameter PP to the challenger.
**Phase 1**: The attacker sends private key queries $q_1, q_2, \ldots, q_m$,where event $q_i$ corresponds to identity $id_i$. The challenger runs the Extract algorithm to generate the private key $sk_i$, where $sk_i$ corresponds to the identity $id_i$, and sends it to the attacker.

**Challenge**: The attacker sends a plaintext $M \in M_\lambda$ and a challenge identity $c^*$, where $c^*$ did not appear in phase 1. The challenger chooses a random bit $r \in \{0,1\}$ and a random ciphertext $C \in C_\lambda$. If r=0, the challenger sets the challenge ciphertext $C^* = \mathrm{Encrypt}(PP, M, id^*)$. Otherwise, the challenger sets the challenge ciphertext $C^* = C$. The challenger sends $c^*$ to the challenger.

**Phase 2**: The attacker executes an adaptive query $q_{m+1}, q_{m+2}, \ldots, q_n$. This query event $q_i$ corresponds to the identity $id_i$, where $id_i$ is not equal to $C^*$. The challenger's response is the same as in phase 1, generating a private key corresponding to the identity and sending it to the attacker.
**Guess**: The attacker outputs a guess $r' \in \{0,1\}$, if $r = r'$, the attacker wins the game.

The game described above is a security game based on IND-ID-CPA, and we define the advantage of the attacker as:

$$\mathrm{Adv}_{A,\varepsilon}(\lambda)=\left|\Pr[r' = r] - \frac{1}{2}\right|$$

Definition 4. If the attacker's advantage $\mathrm{Adv}_{\varepsilon,A}(\lambda)$ is a negligible function for all IND-αID-CPA polynomial-time attackers A, then scheme ε is indistinguishable under selective chosen identity and chosen plaintext attacks, that is to say, scheme ε is IND-αID-CPA secure.

### 4.3. Programmable Hash Functions on Lattice

Definition 6 (Programmable hash functions). Given the security parameter κ, let the hash function $\mathcal{H}: \chi \to \mathbb{Z}_q^{n\times m}$ consist of the algorithm $(\mathcal{H}.\mathrm{Gen}, \mathcal{H}.\mathrm{Eval})$, and there is a polynomial time trapdoor generation algorithm $\mathcal{H}.\mathrm{Gen}(1^\kappa)$ that outputs the hash key K. For any input $X \in \chi$ there is a polynomial time trapdoor evaluation algorithm $\mathcal{H}.\mathrm{Eval}(K, X)$ that outputs a hash value $Z \in \mathbb{Z}_q^{n\times m}$. The specific implementation process is as follows:

(1) $\mathcal{H}.\mathrm{Gen}(1^\kappa)$: Randomly select matrix $A_0, .., A_d \leftarrow \mathbb{Z}_q^{n\times nk}$,return key $K = \{A_i\}_{i\in\{0,\ldots,d\}}$.
(2) $\mathcal{H}.\mathrm{Eval}(K, X)$: Given key $K = \{A_i\}_{i\in\{0,\ldots,d\}}$ and vector $u \in Z_q^n$ as input, let $(u_{[1]}, u_{[2]}, \ldots, u_{[d]})$ be any d-segmented representation of vector u, calculate and return $Z = A_0 + \sum_{i=1}^{d} A_i X_i$, where $X_i = \mathrm{BD}_q(H(u_{[i]})G)$.

## 5. SYSTEM MODEL

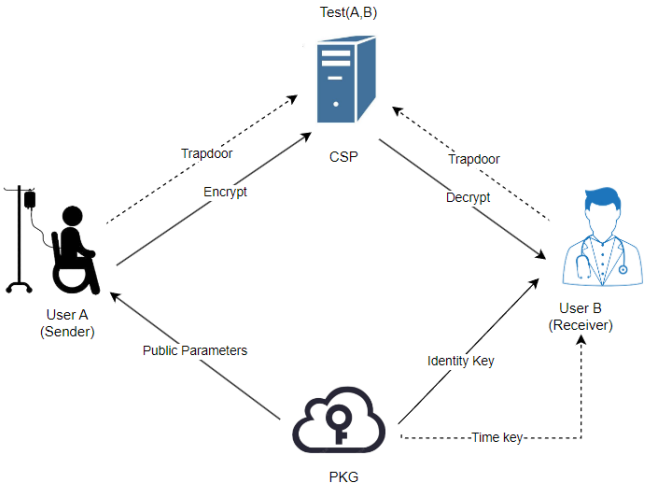


Figure 1. System model.

As shown in Figure 1, the described system model supports revocable functionality and is oriented to the field of telemedicine systems. There are three main entities in this model, namely users, key generation center(PKG), and cloud server providers(CSP). Their specific functions are as follows:

**User:** Users are divided into two types, namely senders and receivers. In a telemedicine system, the sender is generally a patient and the receiver is a doctor. The sender passes the plaintext to the cloud server, the receiver decrypts the ciphertext from the cloud server, and both the sender and the receiver send test trapdoors to the cloud server.

**Cloud server providers:** The cloud server provider receives the ciphertext from the sender and passes the ciphertext to the receiver, and the cloud server processes the equivalent test trapdoors from both the sender and the receiver and outputs the test results.

**Key generation center:** The main function of the key generation center is to generate keys for the sender and receiver, send the public key to the sender, and send the private key and update key to the receiver.

## 6. CONSTRUCTION OF OUR SCHEME

In our identity-based encryption scheme, $\mathcal{H} = (\mathcal{H}.\mathrm{gen}, \mathcal{H}.\mathrm{Eval})$ is a $(1, v, \beta)$ programmable hash function with parameters $\mathcal{H}.\mathrm{TrapGen}$ and $\mathcal{H}.\mathrm{TrapEval}$ is a pair of trapdoor generation and trapdoor evaluation algorithms. Then, we set $\mathrm{k} = \lceil \log_2 \mathrm{q} \rceil$, $\bar{m} = O(nlogq)$, $\mathrm{m} = \mathrm{m}' + \bar{\mathrm{m}}$ as system parameters.

### A. Concrete construction

**Setup($1^{\lambda}$):** Run the trapdoor function $\mathrm{TrapGen}(1^{\mathrm{n}}, 1^{\bar{\mathrm{m}}}, \mathrm{q})$, generate the matrix $\mathrm{A} \epsilon \mathrm{Z}_{\mathrm{q}}^{\mathrm{n} \times \bar{\mathrm{m}}}$ and the corresponding trapdoor $\mathrm{T_A} \in \mathrm{Z}_{\mathrm{q}}^{(\bar{\mathrm{m}}-\mathrm{nk}) \times \mathrm{nk}}$, and $\mathcal{H}.\mathrm{gen}(1^{\lambda})$ generate $\mathrm{K} = \{ \mathrm{A}, \{\mathrm{A_i}\}_{\mathrm{i} \in \{0,1\ldots,\ell'-1\}} \}$ ,

among which, $\ell' = \log_2 \ell$ randomly select $u \in Z_q^n$, and finally get PP = ( A, K, u), msk= $T_A$.

**PriKeyGen(PP,msk,id,RL,ST):** Input public parameters mpk, master key msk, an identity $id \in Z_q^n$, revocable list RL , status list ST , and output the private key of the identity id. Specific steps are as follows:

(1) Randomly select an empty leaf node v corresponding to the user id from the binary tree BT , and calculate path (v).
(2) calculate $A_{id} = \mathcal{H}.\mathrm{Eval}(K, id) \in Z_q^{n\times m}$.
(3) $\forall \theta \in \mathrm{path}(v)$, if $u_{\theta,1}$ and $u_{\theta,2}$ are not selected, the vector $u_{\theta,1} \in Z_q^n$ is randomly selected, and set $u_{\theta,2} = u - u_{\theta,1}$.
(4) $\forall \theta \epsilon \mathrm{path}(v)$, obtained $e_{\theta,1} \in Z_q^m$ by running the sampling algorithm $\mathrm{SampleD}(T_A, A_{id}, I_n, u, s)$.
(5) Output the user private key $sk_{id} = \{(\theta, e_{\theta,1})\}_{\theta\in \mathrm{path}(v)}$ and status list ST.

**KeyUpdate( PP, msk, id, RL, ST):** Input public parameters mpk, master key msk, an identity $t \in z_q^n$, revocable list RL , status list ST , and output the private key of identity t. Specific steps are as follows:

(1) Calculate $A_t = H.\mathrm{Eval}(K, t)$.
(2) $\forall \theta \in \mathrm{KUNodes}(BT, RL, t)$, if $u_{\theta,1}$ and $u_{\theta,2}$ are not selected, the vector $u_{\theta,1} \in Z_q^n$ is randomly selected, and set $u_{\theta,2} = u - u_{\theta,1}$.
(3) $\forall \theta \in \mathrm{KUNodes}(BT, RL, t)$, run the sampling algorithm $\mathrm{SampleD}(T_A, A_t, I_n, u, s)$to get $e_{\theta,2} \in Z_q^{m\times n}$.
(4) Output update key $ku_t = \{(\theta, e_{\theta,2})\}_{\theta\in \mathrm{KUNodes}(BT,RL,t)}$.

**DeckeyGen($sk_{id}$, $ku_t$):** Input the user private key $sk_{id}$ and update key $ku_t$, and output the decryption key $sk_{id,t}$. Specific steps are as follows:

(1) $\forall (i, e_{i1}) \in sk_{id}, (j, e_{j,2}) \in ku_t$, if $\exists i = j$, set $e_1 = e_{i,1}, e_2 = e_{j,2}, sk_{id,t} = (e_1, e_2)$, otherwise, $sk_{id,t} = \perp$.
(2) Output $sk_{id,t}$.

**Encrypt (PP ,id,t,m):** Input the public parameter mpk, an identity $id \in \{0,1\}^n$, plain text $m \in \{0,1\}^n$, and a time $t \in \{0,1\}^n$. Specific steps are as follows:

(1) Calculate $(K', td) \leftarrow \mathcal{H}.TrapGen(1^\lambda, A, B)$, $(R_{id}, S_{id}) = \mathcal{H}.TrapGen(td, K, id)$,$(R_t, S_t) = \mathcal{H}.TrapGen(td, K, t)$, where $B \in \mathbb{Z}_q^{n\times m'}$.

(2) Let $F_{id,t} = (A|A_{id}|A_t)$, $x \leftarrow_r D_{\mathbb{Z}^n,\alpha q}$, $y \leftarrow_r D_{\mathbb{Z}^{\overline{m}},\alpha q}$ and $s \leftarrow_r z_q^n$ are randomly selected. Set $z_1 = R_{id}^T y$, $z_2 = R_t^T y$, then $s_1$ is the first element of s.

(3) Calculate $c_0 = u^T s + x + m\left\lfloor \frac{q}{2} \right\rfloor$, $c_1 = F_{id,t}^T s + \begin{bmatrix} y \\ z_1 \\ z_2 \end{bmatrix}$.

(4) Output the ciphertext CT=$(c_0, c_1, s_1)$.

**Decrypt( PP,$sk_{id,t}$, CT ):** Input the public parameter PP , a decryption key $sk_{id,t}$, and a ciphertext CT. Specific steps are as follows :

(1) Set $c_1 = \begin{bmatrix} c_{1,0} \\ c_{1,1} \\ c_{1,2} \end{bmatrix} \in \mathbb{Z}_q^{3m}$.

(2) Calculate $w = c_0 - e_1^T \begin{bmatrix} c_{1,0} \\ c_{1,1} \end{bmatrix} - e_2^T \begin{bmatrix} c_{1,0} \\ c_{1,2} \end{bmatrix}$,if $\left| w - \left\lfloor \frac{q}{2} \right\rfloor \right| < \frac{q}{4}$, then output m =1, otherwise m =0.

**Td(PP,$msk$, $id$, $H'(m)$):** Input public parameter PP, private key $sk'_{id,t}$, ciphertext CT, hash function $H'(m)$.

(1) Let $H'(m) = (H'(m)_1, H'(m)_2, \cdots, H'(m)_t) \in \{0,1\}^t$, $A_{id} = \mathcal{H}.Eval(K, id) \in \mathbb{Z}_q^{n \times m}$.

(2) Run $\text{RandBasis}(\text{ExtentBasis}(A, A_{id}))$ to generate a trapdoor $T_{id}$ of the lattice $\Lambda_q^{\perp}(A_{id})$. And run $\text{RandBasis}(\text{ExtentBasis}(A, A_t))$ to generate a trapdoor $T_t$ of the lattice $\Lambda_q^{\perp}(A_t)$.

(3) Calculate $v_i = H'(m) \cdot \left\lfloor \frac{q}{2} \right\rfloor \cdot s_1^{-1} \bmod q$, where, $i = 1,2,\ldots,t$, $s_1$ is the first value of s.

(4) Set $\tilde{v}_1 \in \mathbb{Z}_q^n, \tilde{v}_2 \in \mathbb{Z}_q^n$, and satisfy the following relationship,

$\tilde{v}_1 = [v_1 \quad \cdots \quad v_j \quad \cdots \quad 0]$

$\tilde{v}_2 = [0 \quad \cdots \quad v_j \quad \cdots \quad v_t]$

$\tilde{v}_1 + \tilde{v}_2 = [v_1 \quad \cdots \quad v_j \quad \cdots \quad v_t]$

Among them, j is a randomly selected value and satisfies $0 < j < t$.

(5) Run the sampling algorithm $SampleD(T_{id}, A_t, I_n, \tilde{v}_1, s)$ to get $E_1 \in \mathbb{Z}_q^n$, at the same time, run the sampling algorithm $SampleD(T_t, A_t, I_n, \tilde{v}_2, s)$ to get $E_2 \in \mathbb{Z}_q^n$.

(6) Output $E_1$ as a test trapdoor $td_{id}$ for identity id and output $E_2$ as a test trapdoor $td_t$ for time t.

**Test($td_{id}, td_t, CT, td'_{id}, td'_t, CT'$):** Input the ciphertext CT and CT from two different users $CT'$, the test trapdoors of their related identity are $td_{id}$ and $td'_{id}$ respectively, and the test trapdoors of their related time are $td_t$ and $td'_t$ respectively.

(1) Set $c_1 = \begin{bmatrix} c_{1,0} \\ c_{1,1} \\ c_{1,2} \end{bmatrix} \in \mathbb{Z}_q^{3m}$.

(2) Calculate $tw = E_1^T \begin{pmatrix} c_{1,0} \\ c_{1,1} \end{pmatrix} + E_2^T \begin{pmatrix} c_{1,0} \\ c_{1,2} \end{pmatrix}$ and $tw' = E_1'^T \begin{pmatrix} c'_{1,0} \\ c'_{1,1} \end{pmatrix} + E_2'^T \begin{pmatrix} c'_{1,0} \\ c'_{1,2} \end{pmatrix}$.

(3) If $|tw_i - \left\lfloor \frac{q}{2} \right\rfloor| \leq \frac{q}{4}$, set $tm_i = 1$, else set to $tm_i = 0$. And the vector $tm'$are generated in a similar way.

(4) If $tm = tm'$it outputs 1, otherwise it outputs 0.

**Rev(id,t,RL,ST):** Given the user identity $id \in I$, time $t \in T$, revocation list RL and status SL, the key management center PKG performs the following operations:

(1) Add the leaf node corresponding to the user ID and time t to RL.

(2) Output the revocation list RL.

## B. Correctness proof

According to the decryption algorithm in the scheme, we can know,

$$w = c_0 - e_1^T \begin{bmatrix} c_{1,0} \\ c_{1,1} \end{bmatrix} - e_2^T \begin{bmatrix} c_{1,0} \\ c_{1,2} \end{bmatrix}$$

$$= u^T s + x + m \left\lfloor \frac{q}{2} \right\rfloor - ((A|A_{id})e_1 + (A|A_t)e_2)^T - e_1^T \begin{pmatrix} y \\ z_1 \end{pmatrix} - e_2^T \begin{pmatrix} y \\ z_2 \end{pmatrix}$$

$=m\left\lfloor\frac{q}{2}\right\rfloor+(x-e_1^T\begin{pmatrix}y\\z_1\end{pmatrix}-e_2^T\begin{pmatrix}y\\z_2\end{pmatrix})$,

where $x-e_1^T\begin{pmatrix}y\\z_1\end{pmatrix}-e_2^T\begin{pmatrix}y\\z_2\end{pmatrix}$ is error term. As long as the boundary of the error term is smaller than $\frac{q}{5}$, this scheme can always decrypt the plaintext correctly.

## 7. SECURITY PROOF

To simplify the proof, we divide attackers into two types:

Type I attacker: A Type I attacker chooses to be challenged on the target identity id *, but has been revoked before t* or at t* .

Type II attacker: A Type II attacker does not challenge id* at any time .

In this paper, a random bit is selected to guess the type of attacker that will be faced. In the following game sequence, all other games except Game2 are indistinguishable. The specific proof is as follows:

**Game0:** The challenger Ctruly simulates the IND-αRID-CPA security game. The specific process is as follows:

**Setup:** Given the security parameters $\lambda$, first run $(A, T_A) \leftarrow \mathrm{TrapGen}(1^n, 1^{\bar{m}}, q)$, generate the matrix $A\epsilon Z_q^{n\times\bar{m}}$ and the corresponding trapdoor matrix $T_A \in Z_q^{(\bar{m}-nk)\times nk}$, and then randomly select the matrix $U \in Z_q^{n\times t}$ for calculation $K \leftarrow \mathcal{H}.\mathrm{gen}(1^{\lambda})$. Finally, send the public key $\mathrm{mpk} = (A, K, U)$ to the attacker A, keeping the master private key $T_A$ private.

**Phase 1:** Attacker A sends bounded queries to the challenger.

1. User key generation query: Input public parameters PP , master key msk , an identity $id \in Z_q^n$, status list ST , and output the private key of identity id . Specific steps are as follows:

(1) The challenger first determines whether the identity id is in the state list ST. If the id is in the state list ST, the tuple is retrieved directly from $\mathrm{ST}(id, u_{\theta,1}, u_{\theta,2})$. Otherwise, randomly select an empty leaf node v corresponding to the user id from the binary tree BT , and calculate path(v). $\forall\theta\epsilon\mathrm{path}(v)$, first randomly select $u_{\theta,1} \in Z_q^{n\times n}$,and $u_{\theta,2} = u - u_{\theta,1}$, then store the tuples $(id, u_{\theta,1}, u_{\theta,2})$ in the state list ST and θ on the node .

(2) Calculate $A_{id} = \mathcal{H}.\mathrm{Eval}(K, id)$ firstly. Then, run the sampling algorithm $\mathrm{SampleD}(T_A, A_{id}, I_n, u_{\theta,1}, s)$to obtain $e_{\theta,1}$and output the user's private key $sk_{id} = \{(\theta, e_{\theta 1})\}_{\theta\epsilon\mathrm{path}(v)}$ and send it to the attacker, where $e_{\theta,1}$the distribution statistics are close to $D_{\Lambda_q^u(T_A),\delta}$.

2. Key update query: Input public parameters PP , master key msk , an identity $t \in z_q^n$, revocable list RL , status list ST , and output the private key of time t . Specific steps are as follows:

(1) Define R to be a set of revocable users at time t. For any user that satisfies the requirement $t' \le t$, if it exists $(t', id')$ ∈ RL, it is added $id'$ to the revocable list RL. For all $id \notin$ Rusers, determine whether the identity id is in the status list ST. If the id is in the status list ST, retrieve the tuple directly from ST $(id, u_{\theta,1}, u_{\theta,2})$. Otherwise,

$\forall\theta \in \mathrm{KUNotes(BT, RL, t)}$,if $u_{\theta,1}$, $u_{\theta,2}$ is not defined, randomly select $u_{\theta,1} \in Z_q^n$,let $u_{\theta,2} = u - u_{\theta,1}$.

(2) The challenger runs first $A_t = \mathcal{H}.\mathrm{Eval}(K, t)$and runs $\mathrm{SampleD}(T_A, A_t, I_n, u_{\theta,2}, s)$to get $e_{\theta,2}$, The distribution statistics of $e_{\theta,2}$ are close to $D_{\Lambda_q^u(F_t),\delta}$. Output the update key $ku_t = \{(\theta, e_{\theta,2})\}_{\theta\in \mathrm{KUNodes(BT,RL,t)}}$and send it to the attacker.

**Revocation query:** Input the revocation user identity idand time t, and the challenger returns the updated revocation list.

**Challenge:** Challenger C selects the user identity $id^* \in I$ ,time $t^* \in T$ and $M_1$two ciphertext sums of the same length $M_0$. The challenger selects them randomly $b \leftarrow \{0,1\}$and $M_b$performs encryption operations on them. First calculate

$A_{id} = (A, H_K(id)) \in Z_q^{n\times m}, A_t = (A, H_K(t)) \in Z_q^{n\times m}$, then randomly select $x \leftarrow_r D_{Z^n,\alpha q}$, $y \leftarrow_r D_{Z^{\bar{m}},\alpha q}$, $z_1 = R_{id}^T y$, $z_2 = R_t^T y$, $s \leftarrow_r Z_q^n$calculate the ciphertext

$c_0^* = u^T s + x + M\left\lfloor\frac{q}{2}\right\rfloor, c_1^* = F_{id^*,t}s + \begin{pmatrix} y \\ z_1 \\ z_2 \end{pmatrix}$, and send the challenge ciphertext $C^* = (c_0^*, c_1^*, s_1)$ to the attacker.

**Phase 2:** The attacker A can adaptively $id \neq id^*$perform more user private key queries for any identity just like Phase 1.

**Guess:** The attacker eventually returns one bit $b' \in \{0,1\}$. If $b = b'$ , the challenger outputs 1, otherwise, it outputs 0.
The event $E_i$ indicates that challenger C outputs 1 in $\mathrm{game}_i$, and $\left|\Pr(E_0) - \frac{1}{2}\right| = \epsilon$.

**Game 1:** This game is the same as game 0 except that the Setup and Challenge phases of the game have been changed.

**setup:** The challenger runs $(A, T_A) \leftarrow \mathrm{TrapGen}(1^n, 1^{\bar{m}}, q)$ generates the matrix $A\epsilon Z_q^{n\times\bar{m}}$ and the corresponding trapdoor matrix $T_A \in Z_q^{(\bar{m}-nk)\times nk}$, and then generates it $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}(A, G)$. Finally, output mpk= $(A, K', U)$, and set the master private key R and the trapdoor td private.

**Challenge:** The challenger uses the generated ciphertext $(K', td)$ during the setup phase. According to the properties of the trapdoor key that are statistically close to the function $\mathcal{H}$, it can be obtained $|\Pr(X_1) - \Pr(X_0)| \leq \mathrm{negl}(\lambda)$.

**Game 2** : In Game 2, the challenger is basically the same as Game 1, except that the following steps are added at the end of the game. First at the end of the game, the challenger is defined:

$$\tau\left(\hat{td}, \hat{R}, I^*\right) = \begin{cases} 0, \text{if } \hat{S}_{id^*} = 0, \hat{S}_{id_i} \text{ is invertible}, \ i \in \{1, \ldots, Q\} \\ 1, \text{Otherwise} \end{cases}$$

,Where $(\hat{R}_{id^*}, \hat{S}_{id^*}) = \mathcal{H}.\mathrm{TrapEval}(\hat{td}, \hat{R}, id^*), (\hat{R}_{id_i}, \hat{S}_{id_i}) = \mathcal{H}.\mathrm{TrapEval}(\hat{td}, \hat{R}, id_i)$. The challenger then proceeds as follows:

(1) **About check:** Generated $\left(\mathrm{td}, \mathrm{K}^{'}\right)$during the Setup phase . If $\tau\left(\mathrm{td}, \mathrm{K}^{'}, \mathrm{I}^{*}\right)$=1, the challenger ends the game and outputs uniform random bits.

(2) **Artificial abort:** Let p represent the probability $\mathrm{p}=\Pr\left(\tau\left(\widehat{\mathrm{td}}, \widehat{\mathrm{K}}, \mathrm{I}^{*}\right)=0\right)$ of randomly choosing $\left(\widehat{\mathrm{td}}, \widehat{\mathrm{K}}\right)$. The challenger samples $O\left(\epsilon^{-2} \log\left(\epsilon^{-1}\right) \delta^{-1} \log\left(\delta^{-1}\right)\right)$ times with probability p by running $\left(\widehat{\mathrm{K}}, \widehat{\mathrm{td}}\right) \leftarrow \mathcal{H}.\mathrm{TrapGen}(\mathrm{a}, \mathrm{g})$ and $\tau\left(\widehat{\mathrm{td}}, \widehat{\mathrm{K}}, \mathrm{I}^{*}\right)$ to obtain an estimate $\mathrm{p}^{'}$ of p. If $\mathrm{p}^{'} \geq \delta$,the challenger terminates the game with probability $\frac{\mathrm{p}^{'}-\delta}{\mathrm{p}^{'}}$ , otherwise output a uniformly random bit.

Let $\widetilde{p_i}$ represent the probability that the challenger satisfies $\tau\left(\widehat{td}, \widehat{K}, I^{*}\right)=0$ in game i during the artificial abort phase, and $p_i$represent the probability that the challenger satisfies $\tau\left(\widehat{td}, \widehat{K}, I^{*}\right)=0$ in game i during the about check phase. Then let $\Gamma_i$ represent the absolute value of the difference between $\widetilde{p_i}$ and $p_i$, that is, $\Gamma_{\mathrm{i}}=\left|\widetilde{\mathrm{p}_{\mathrm{i}}}-\mathrm{p}_{\mathrm{i}}\right|$. If $\mathcal{H}$ is a $(1, \mathrm{v}, \beta, \gamma, \delta)$ programmable hash function and satisfies $\mathrm{Q} \leq \mathrm{v}$,then $\left|\Pr\left(\mathrm{X}_2\right)-\frac{1}{2}\right| \geq \frac{1}{2} \epsilon(\delta-\Gamma_2)$.

**Game 3:** This game is basically the same as Game 2, and satisfies $\mathrm{S}_{\mathrm{id}}^{'}$=0, $S_t^{'}=0$. In addition to Phase 1, Phase 2 and other stages, the following changes have been made:

**Phase 1:** Attacker A sends bounded queries to the challenger.

(1) If rev=0, simulate an interactive game with the first type of attacker.

a. In the key generation query phase, if $\theta \in \mathrm{path}(\mathrm{v})$, run the Gaussian sampling algorithm $\mathrm{SampleGaussian}\left(\mathrm{Z}_{\mathrm{q}}^{\mathrm{m}}, \delta\right)$to generate $\mathrm{e}_{\theta,1} \in \mathrm{Z}_{\mathrm{q}}^{\mathrm{m}}$, let $\mathrm{U}_{\theta,1}=\mathrm{F}_{\mathrm{id}} \mathrm{e}_{\theta,1} \bmod \mathrm{q}$,where ,
$\mathrm{F}_{\mathrm{id}}=(\mathrm{A} | \mathrm{A}_{\mathrm{id}})$， $\mathrm{A}_{\mathrm{id}}=\mathrm{AR}_{\mathrm{id}}^{'}+\mathrm{S}_{\mathrm{id}}^{'} \mathrm{B}=\mathrm{AR}_{\mathrm{id}}^{'}$.

b. In the key update phase, if $\theta \notin \mathrm{path}(\mathrm{v})$, run the Gaussian sampling algorithm $\mathrm{SampleGaussian}\left(\mathrm{Z}_{\mathrm{q}}^{\mathrm{m}}, \delta\right)$ to generate $\mathrm{e}_{\theta,2} \in \mathrm{Z}_{\mathrm{q}}^{\mathrm{m}}$, let $\mathrm{U}_{\theta,1}=\mathrm{F}_{\mathrm{id}} \mathrm{e}_{\theta,1} \bmod \mathrm{q}$, where,
$\mathrm{F}_{\mathrm{t}}=(\mathrm{A} | \mathrm{A}_{\mathrm{t}})$， $\mathrm{A}_{\mathrm{t}}=\mathrm{AR}_{\mathrm{t}}^{'}+\mathrm{S}_{\mathrm{t}}^{'} \mathrm{B}=\mathrm{AR}_{\mathrm{id}}^{'}$.

(2) If rev=0, simulate an interactive game with the second type of attacker.

In the key update phase, run the Gaussian sampling algorithm $\mathrm{SampleGaussian}\left(\mathrm{Z}_{\mathrm{q}}^{\mathrm{m}}, \mathrm{s}\right)$ to generate $\mathrm{e}_{\theta,2} \in \mathrm{Z}_{\mathrm{q}}^{\mathrm{m}}$, let $\mathrm{u}_{\theta,2}=(\mathrm{A} | \mathrm{A}_{\mathrm{t}}) \cdot \mathrm{e}_{\theta,1}$, $\mathrm{u}_{\theta,1}=\mathrm{u}-\mathrm{u}_{\theta,1}$, where, $\mathrm{A}_{\mathrm{t}}=\mathrm{AR}_{\mathrm{t}}^{'}+\mathrm{S}_{\mathrm{t}}^{'} \mathrm{b}=\mathrm{AR}_{\mathrm{id}}^{'}$.The user $\mathrm{id}^{*}$has not been queried, and the challenger uses $\left\{\theta, \mathrm{e}^{\theta,2}\right\}$ as a reply to the update query for t.

**Phase 2**：A can adaptively perform more user private key queries for any identity B just like Phase 1.

Because both $\mathrm{F}_{\mathrm{id}}$ and $\mathrm{F}_{\mathrm{t}}$ can be regarded as random matrix $\mathrm{Z}_{\mathrm{q}}^{\mathrm{n} \times \mathrm{m}}$, they $\mathrm{u}_{\theta,1}$， $\mathrm{u}_{\theta,2}$ are statistically indistinguishable from uniform distributions. Therefore, the attacker cannot distinguish the type of impersonator and has a 1/2 probability of correctly impersonating. When the game is emulated correctly, Game2 and Game3 are indistinguishable. Right now $\Pr\left(\mathrm{X}_3\right)=1/2 \Pr\left(\mathrm{X}_2\right), \Gamma_3=1/2 \Gamma_2$.

**Game 4:** This game is basically the same as Game 3, except for the following changes to the Setup , Phase 1 , Challenge , Phase 2 and other stages:

**Setup:** The challenger randomly selects the matrix $A \in Z_q^{n\times \bar{m}}$, $U \in Z_q^{n\times n}$calculates it, and $\left(K^{'},td\right) \leftarrow \mathcal{H}.\mathrm{TrapGen}(A,G)$finally outputs it $PP = \left(A,K^{'},U\right)$, and saves the trapdoor td privately.

**Phase 1:** Attacker A sends bounded queries to the challenger.

1. User key generation query.

(1) The challenger first determines whether the identity id is in the state list ST. If the id is not in the state list ST, randomly selects an empty leaf node v corresponding to the user id from the binary tree BT , and calculates path(v) . $\forall\theta\epsilon\mathrm{path}(v)$, store the id in the state list ST and θ on the node .

(2) Calculate first $\left(R_{id}^{'},S_{id}^{'}\right) = \mathcal{H}.\mathrm{TrapEval}\left(td,K^{'},id\right)$. If $S_{id}^{'} = 0$, terminate the game and send a random bit. Otherwise, run the sampling algorithm $\mathrm{SampleD}\left(R_{id},A_{id},S_{id},u_{\theta,1},s\right)$to obtain $e_{\theta,1}$, output the user's private key $sk_{id} = \{(\theta,e_{\theta,1})\}_{\theta\in path(v)}$and send it to the attacker, where $e_{\theta,1}$the distribution statistics are close to $D_{\Lambda_q^{u_{\theta,1}}(F_{id}),\delta}$, where, $F_{id} = (A|A_{id})$.

2. Key update query.

(1) Define R to be a set of revocable users at time t. For any user that satisfies the requirement $t^{'} \le t$, if it exists $\left(t^{'},id^{'}\right) \in RL$, it is added $id^{'}$ to the revocable list RL. For all $id \notin R$users, determine whether the identity id is in the status list ST. If the id is not in the status list ST, $\forall\theta \in \mathrm{KUNotes}(BT,RL,t)$, store t to the node θ.

(2) The challenger runs first $\left(R_t^{'},S_t^{'}\right) = \mathcal{H}.\mathrm{TrapEval}\left(td,K^{'},t\right)$ and if $S_t^{'} = 0$, terminates the game and sends a random bit. Otherwise, running $\mathrm{SampleD}\left(R_t,A_t,S_t,u_{\theta,2},s\right)$gets $e_{\theta,2}$, $e_{\theta,2}$The distribution statistics are close to $D_{\Lambda_q^{\perp}(F_t),\delta}$. Output the update key $ku_t = \{(\theta,e_{\theta,2})\}_{\theta\in KUNodes(BT,RL,t)}$ and send it to the attacker.

Challenge: challenger computation $\left(R_{id^*}^{'},S_{id^*}^{'}\right) = \mathcal{H}.\mathrm{TrapEval}\left(td,K^{'},id^*\right)$, $\left(R_{t^*}^{'},S_{t^*}^{'}\right) = \mathcal{H}.\mathrm{TrapEval}\left(td,K^{'},t^*\right)$. If the attacker is Type-2 , $S_{t^*}^{'} \neq 0$terminate the game and send random bits. If the attacker is Type- 1, check $S_{id^*}^{'} = 0$, $S_{t^*}^{'} = 0$ whether it is satisfied, if not, the game will be terminated. otherwise, $A_{id^*} = AR_{id^*}^{'} + S_{id^*}^{'}B \in Z_q^{n\times \bar{m}}, A_{t^*} = AR_{t^*}^{'} + S_{t^*}^{'}B \in \mathbb{Z}_q^{n\times \bar{m}}$. Calculate

$$c_0^* = b_1 + M\left\lfloor \frac{q}{2} \right\rfloor, c_1^* = F_{id,t}s + \begin{pmatrix} y \\ z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} A^T s + y \\ A_{id^*}^T s + R_{id^*}^T y \\ A_{t^*}^T s + R_{t^*}^T y \end{pmatrix} = \begin{pmatrix} A^T s + y \\ \left(R_{id^*}^{'}\right)^T (A^T s + y) \\ \left(R_{t^*}^{'}\right)^T (A^T s + y) \end{pmatrix} =$$

$$\begin{pmatrix} v_2 \\ \left(R_{id^*}^{'}\right)^T v_2 \\ \left(R_{t^*}^{'}\right)^T v_2 \end{pmatrix},$$

where, $v_1 = U^T s + x_1, v_2 = A^T s + y$. Finally, the challenge ciphertext $C^* = (c_0^*, c_1^*, s_1)$is sent to the attacker.

**Phase 2:** A can adaptively $id \neq id^*$perform more user private key queries for any identity just like Phase 1.

From the attacker's perspective, when game 4 is simulated correctly, game 4 and game 3 are statistically indistinguishable. Right now,

$$Pr(X_4) = Pr(X_3), \Gamma_4 = \Gamma_3$$

**Game 5**: This game is basically the same as Game 3 , except for the following changes to the challenge stage:

**Challenge:** Randomly select vectors $b_1 \in Z_q^n, v_2 \in Z_q^m$ to calculate ciphertext $C^* = (c_0^*, c_1^*, s_1)$, where,$c_0^* = b_1 + M\left\lfloor \frac{q}{2} \right\rfloor, c_1^* = \begin{pmatrix} v_2 \\ \left(R_{id^*}^{'}\right)^T v_2 \\ \left(R_{t^*}^{'}\right)^T v_2 \end{pmatrix}$. if $(u, v_1)$, $(A, v_2)$If it is a legal RL WE tuple, the attacker thinks it is in Game 4 , otherwise it is considered to be in Game 5, which indicates that Game 4 and Game 5 are indistinguishable. so, $|Pr(X_5) - Pr(X_4)| \leq \epsilon^{'}$ , $|\Gamma_5 - \Gamma_4| \leq \epsilon^{'}$ .

**Game** 6 : This game is basically the same as Game 5 , except that the challenger has made the following changes:

**Setup:** Run first $(A, T_A) \leftarrow TrapGen(1^n, 1^{\bar{m}}, q)$to generate the ring polynomial vector $a \in R_q^m$ and the corresponding trapdoor matrix $T_A \in Z_q^{(\bar{m}-nk)\times nk}$. Then challengers are randomly generated $u \in Z_q^n$and calculated separately $K \leftarrow \mathcal{H}.gen(1^\lambda)$. $\left(K^{'}, td\right) \leftarrow \mathcal{H}.TrapGen(1^\lambda, A, G)$Finally, send the master public key mpk= $(A, K, U)$and keep $\left(T_a, K^{'}, td\right)$ it private.

**Phase 1:** Attacker A sends bounded queries to the challenger.

(1) User key generation query: challenger computation $A_{id} = \left(A|H_K(id)\right) \in Z_q^{n\times m}$, where $H_K(id) = \mathcal{H}.Eval(K, id) \in Z_q^{n\times m'}$. Then, run $SampleD(T_A, A_{id}, I_n, u_{0,1}, s)$ to get $e_{id} \in Z_q^m$ and send it to the attacker.

(2) Key update query: Compute $A_t = \left(A, H_K(t)\right) \in Z_q^{n\times m}$, where, $H_K(t) = \mathcal{H}.\mathrm{Eval}(K,t) \in Z_q^{n\times m'}$. The call is then $\mathrm{SampleD}(T_A, A_t, I_n, u_{\theta,2}, s)$ obtained $e_t \in Z_q^m$ and sent to the attacker.

**Challenge:** The challenger randomly selects $c_1^* \in Z_q^n, c_2^* \in Z_q^{3m}$ and generates ciphertext $C^* = (c_1^*, c_2^*, s_1)$.

According to the function $\mathcal{H}$ are close to the properties of the trapdoor key and are randomly distributed $\mathrm{game}_4$ in $b_1, b_2, b_3$, so from the attacker's perspective, $\mathrm{game}_6$ and $\mathrm{game}_5$ are statistics indistinguishable. That is, $|\Pr(X_6) - \Pr(X_5)| \le \mathrm{negl}(\lambda)$ and $|\Gamma_5 - \Gamma_4| \le \mathrm{negl}(\lambda)$. And because the public key and the challenge ciphertext are independent of the randomly distributed td in game 6, the challenger can use it in the guessing phase calculation $\left(K', td\right) \leftarrow \mathcal{H}.\mathrm{TrapGen}\left(1^\lambda, a, g\right)$ and the termination check phase $\left(K', td\right)$, that is, $\Pr(X_6) = 0$ and $\Gamma_6 = 0$.

## 8. PARAMETER ANALYSIS

In order to verify the effectiveness of the scheme in this article, we analyzed the parameter sizes of different schemes from the perspectives of storage overhead and computing overhead. The experimental analysis results also verified the effectiveness of our ideas.

As can be seen from Table 1, the overall cost of public parameters of our scheme is smaller than that of other schemes, and the storage overhead of private keys is also smaller than that of other schemes. Moreover, our solution is revocable and can adapt to cloud computing scenarios. Among them, $\ell$ represents the length of the identity id, m, n, t represent the parameters in different solutions respectively.

Table 1. Storage overhead analysis.

| Schemes | \|PP\| | \|SK\| | \|Ciphertext\| | Assumption | Revocable |
|---|---|---|---|---|---|
| [28] | $(\ell+3)mn$ | 4mt | 2t+4m | LWE | × |
| [29] | $(\ell+3)mn$ | $4m^2$ | $m^2+6m+2t$ | LWE | × |
| Ours | $(log\ell+2)mn$ | 2m | 3m+2 | LWE | ✔ |

From Table 2, we mainly compare the number of multiplication operations in different schemes. The results also show that the performance of our scheme is better than the other two schemes in the encryption, decryption and testing phases. Among them, $\ell$ represents the length of the identity id, m, n, t represent the parameters in different solutions respectively.

Table 2. Computational cost analysis.

| Shemes | Encrypt | Decrypt | Trapdoor | Test |
|---|---|---|---|---|
| [28] | $\ell mn + 4mn + 2nt$ | 4mt | 0 | 2mt |
| [29] | $\ell mn + 6mn + 2nt$ | 6mt | 0 | 3mt |
| Ours | $(log\ell)mn + 3mn$ +n | 4m | 0 | 8m |

## 9. CONCLUSION

This paper proposes for the first time an equality test scheme with revocable identity-based encryption based on lattice cryptography. This scheme is designed to use the hash value without embedding plaintext in the encryption algorithm as input, and introduces a programmable hash function, and The hash value of the plaintext is embedded as input in the equivalence testing algorithm. Finally, in the parameter analysis, it was verified that the proposed scheme is efficient in terms of storage overhead and computing overhead, and it was proved that the selected ciphertext is safe under the random oracle model.

## REFERENCES


[1] A. Shamir, "Identity-based cryptosystems and signature schemes," in CRYPTO, pp. 47–53, 1985.

[2] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in CRYPTO, pp. 213–229, 2001.

[3] J. Horwitz and B. Lynn, "Toward hierarchical identity-based encryption," in EUROCRYPT, pp. 466–481, 2002.

[4] D. Boneh and X. Boyen, "Efficient selective-id secure identity-based encryption without random ora-cles," in EUROCRYPT, pp. 223–238, 2004.

[5] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in EUROCRYPT, pp. 457–473, 20

[6] J. Baek, R. Safavi-Naini, and W. Susilo, "Efficient multi-receiver identity-based encryption and its application to broadcast encryption," in PKC, pp. 380–397, 2005.

[7] X. Boyen and B. Waters, "Anonymous hierarchical identity-based encryption (without random ora-cles)," in CRYPTO, pp. 290–307, 2006.

[8] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in CCS, pp. 417–426, 2008.

[9] S. S. Chow, Y. Dodis, Y. Rouselakis, and B. Waters, "Practical leakage-resilient identity-based en-cryption from simple assumptions," in CCS, pp. 152–161, 2010.

[10] J. Chen, H. W. Lim, S. Ling, H. Wang, and K. Nguyen, "Revocable identity-based encryption from lattices," in ACISP, pp. 390–403, 2012.

[11] J. Kim, W. Susilo, M. H. Au, and J. Seberry, "Adaptively secure identity-based broadcast encryption with a constant-sized ciphertext," IEEE Transactions on Information Forensics and Security, vol. 10, no. 3, pp. 679–693, 2015.

[12] S. Park, K. Lee, and D. H. Lee, "New constructions of revocable identity-based encryption from multilinear maps," IEEE Transactions on Information Forensics and Security, vol. 10, no. 8, pp. 1564–1577, 2015.

[13] J. Yu, R. Hao, H. Zhao, M. Shu, and J. Fan, "Iribe: Intrusion-resilient identity-based encryption," Information Sciences, vol. 329, pp. 90–104, 2016.

[14] J. Zhang, Y. Chen, and Z. Zhang, "Programmable hash functions from lattices: short signatures and ibes with small key sizes," in CRYPTO, pp. 303–332, 2016.

[15] S. Yamada, "Asymptotically compact adaptively secure lattice ibes and verifiable random functions via generalized partitioning techniques," in CRYPTO, pp. 161–193, 2017.

[16] J. Wei, W. Liu, and X. Hu, "Secure data sharing in cloud computing using revocable-storage identity-based encryption," IEEE Transactions on Cloud Computing, vol. 6, no. 4, pp. 1136–1148, 2016.

[17] X. Zhang, Y. Tang, H. Wang, C. Xu, Y. Miao, and H. Cheng, "Lattice-based proxy-oriented identity-based encryption with keyword search for cloud storage," Information Sciences, vol. 494, pp. 193–207, 2019.

[18] C. Ge, Z. Liu, J. Xia, and L. Fang, "Revocable identity-based broadcast proxy re-encryption for data sharing in clouds," IEEE transactions on dependable and secure computing, vol. 18, no. 3, pp. 1214–1226, 2019.

[19] Y. Sun, P. Chatterjee, Y. Chen, and Y. Zhang, "Efficient identity-based encryption with revocation for data privacy in internet of things," IEEE internet of things journal, vol. 9, no. 4, pp. 2734–2743, 2021.

[20] J. Zhang, S. Su, H. Zhong, J. Cui, and D. He, "Identity-based broadcast proxy re-encryption for flexible data sharing in vanets," IEEE Transactions on Information Forensics and Security, 2023.
[21] T. Dillon, C. Wu, and E. Chang, "Cloud computing: issues and challenges," in AINA, pp. 27–33, Ieee, 2010.
[22] Y. Jadeja and K. Modi, "Cloud computing-concepts, architecture and challenges," in ICCEET, pp. 877–880, IEEE, 2012.
[23] M. Ali, S. U. Khan, and A. V. Vasilakos, "Security in cloud computing: Opportunities and challenges," Information sciences, vol. 305, pp. 357–383, 2015.
[24] B. Varghese and R. Buyya, "Next generation cloud computing: New trends and research directions," Future Generation Computer Systems, vol. 79, pp. 849–861, 2018.
[25] A. Alam, "Cloud-based e-learning: scaffolding the environment for adaptive e-learning ecosystem based on cloud computing infrastructure," in ICICC, pp. 1–9, 2022.
[26] G. Yang, C. H. Tan, Q. Huang, and D. S. Wong, "Probabilistic public key encryption with equality test," in CT-RSA, pp. 119–131, 2010.
[27] Q. Tang, "Towards public key encryption scheme supporting equality test with fine-grained authoriza-tion," in ACISP, pp. 389–406, 2011.
[28] D. H. Duong, H. Q. Le, P. S. Roy, and W. Susilo, "Lattice-based ibe with equality test in standard model," in ProvSec, pp. 19–40, 2019.
[29] G. L. D. Nguyen, W. Susilo, D. H. Duong, H. Le, and F. Guo, "Lattice-based ibe with equality test supporting flexible authorization in the standard model," in INDOCRYPT, pp. 624–643, 2020.
[30] Z. Yang, D. He, L. Qu, and Q. Ye, "An efficient identity-based encryption with equality test in cloud computing," IEEE Transactions on Cloud Computing, 2023.

## AUTHORS

**Chenguang Wang** received the B.S. degree in information security from Anhui University, China, in 2021. He is currently pursuing the master's degree with the Department of Cryptographic Science and Technology, Beijing Electronic Science and Technology Institute, China.His current research interests include lattice cryptography and identity-based encryption.

**Huiyan Chen** graduated from the Graduate School of Chinese Academy of Sciences, majoring in signal and information processing. He is currently a professor at the Beijing Electronic Science and Technology Institute. And He has been engaged in the research and teaching of cryptography for a long time, and his main research fields include network security, public key cryptosystem theory, design, analysis, implementation and application, etc., with strong theoretical basis and practical scientific research ability. I have published a number of papers with high academic level and presided over or participated in many scientific research projects in the field of information security, among which more than a dozen projects have won provincial and ministerial awards. Have a deep understanding of the design and analysis of cryptographic algorithms and cryptographic protocols, and have sufficient practical experience in the application of cryptographic tools in the field of information security engineering.