A Lattice Attribute-Based Encryption Scheme for Cloud Storage Based on R-LWE Problem

Sa Lu¹, Rui Jiang¹*, Bei Pei² and Songyang Wu²

¹School of Cyber Science and Engineering, Southeast University, Nanjing, China
²The Third Research Institute of Ministry of Public Security, Shanghai, China

*Corresponding author email: R.Jiang@seu.edu.cn

Abstract. Attribute-Based Encryption (ABE) is generally applied on Cloud Storage to ensure the security and efficient access control of uploaded data. However, with the development of quantum computing, Diffie-Hellman Problem and Discrete Logarithm Problem, which all ABE scheme based on, have been solved. Therefore, anti-quantum ABE schemes become a research hotspot. However, current anti-quantum ABE schemes cannot make attribute revocation. In order to solve these problems, we propose a lattice attribute-based encryption scheme for cloud storage based on R-LWE problem (L-ABE). Firstly, we construct public/secret key pairs based on the hardness of Ring Learning with Error problem (R-LWE) to achieve quantum attack resistance effectively. Secondly, our scheme can achieve attribute revocation to ensure fine-grained access control. Finally, we prove that our scheme can resist chosen plaintext attack.

Keywords: Attribute-Based Encryption; Anti-quantum; Fine-grained access control; Chosen plaintext attack resistance.

1. Introduction

Attribute-based Encryption (ABE) is regarded as one of the most suitable methods for data access control in cloud storage systems. It is firstly proposed by Sahai and Waters [1] in 2005, based on Identity-Based Encryption (IBE) [2]. Ciphertext-policy attribute-based encryption (CP-ABE) is one kind of ABE schemes. In an CP-ABE scheme, one user possesses an attributes set which generated according to the user’s identity. Data owners encrypt their data under an access police constructed according to a set of attributes. All users receive secret keys from the Key Generate Center (KGC) in the light of their attributes. Users are able to recover a message correctly when the attributes of their keys satisfy the requirement of attribute list on the ciphertexts.

The attribute revocation is a necessary demand for cloud data to realize fine-grained access control. For example, users’ attributes may be changed dynamically in cloud storage systems. A user may be entitled some new attributes or revoked some current attributes and his permission of data access should be changed accordingly. To solve this problem, some attribute revocation methods have been proposed. Unfortunately, most revocable ABE scheme [3,4,5] cannot ensure security facing with quantum computer. The reason is that most ABE schemes are based on the bilinear maps and Discrete Logarithm Problem (DLP). Hence, with the development of post-quantum, ABE would get restricted. Shor’s algorithm [6] can solve the Discrete Logarithm problem on quantum computers in polynomial time, which means attackers can directly figure out the secret keys with public keys in an ABE scheme. Therefore, the bilinear pairing algorithm would be unable to ensure the security of encryption facing with large-scale quantum computers.
To solve the quantum attack problems, Lattice encryption becomes the most effective theory. The LWE problem is considered as hardness on the worst case even under quantum situation. Therefore, the LWE problem is widely used to ensure the data security in cloud computing to achieve quantum attack resistance. Regev proposed the LWE problem and a scheme based on the LWE problem in paper \[7\]. However, the scheme had big storage load. The schemes may be not efficient enough. Gentry et al \[8\] built and formulate the trapdoor functions based on the standard LWE problem. Then these functions are widely used in lattice encryption schemes. In 2009, Steinfeld et al. \[9\] defined the Ring Learning With Error problem (R-LWE). In 2010, Peikert et al. \[10\] perfected the definition of it. Compared to the LWE schemes. the schemes based on R-LWE can decrease the storage load and achieve high efficiency. Then anti-quantum ABE schemes make great progress. In recent years, a few researches have been made to merge CP-ABE and lattice encryption into one to achieve fine-grained access control and quantum attack resistance for cloud storage. In scheme \[11\], the author applied a fast decryption operation to increase the efficiency of performance, but their scheme could not resist collusion attack and could not make attribute revocation. Scheme \[12\] proposed a scheme which could achieve high efficiency by adding virtual attributes and be secure against chosen plaintext attack. Wang \[13\] proposed a lattice CP-ABE scheme, but the scheme could not realize fine-grained access control based on attribute revocation. The scheme \[14\] spent too much computation cost on encryption and decryption and could not deal with attribute revocation.

Therefore, most of the existing lattice ABE schemes cannot actually realize fine-grained access control based on attribute revocation.

To solve the above problems, we proposed a lattice ABE scheme for cloud storage based on R-LWE problem (L-ABE). The contribution of our L-ABE scheme is three-folds:

- Our L-ABE scheme can resist the quantum computing attack. We propose a lattice encryption scheme combining R-LWE problem with ciphertext-policy attribute-based encryption algorithm. We first construct a ring of integer polynomials modulo and generate public/secret key pairs using trapdoor function. Then we merge the CP-ABE structure. Finally, we propose the whole L-ABE scheme.
- Our L-ABE scheme can deal with attribute revocation to actually realize fine-grained access control. We utilize the CP-ABE algorithm to assign attributes to the users and embed the access policy into ciphertext, which is different from previous lattice encryption schemes that cannot deal with attribute revocation. The users can decrypt ciphertexts successfully only when their attributes satisfy the access police. Finally, we construct secure attribute revocation to the users by replacing the revoked attribute values with the renewed components and updating legal users’ attributes.
- Our scheme can resist chosen plaintext attack in the selective security model under the LWE assumptions.

2. Preliminaries

Definition 1 (Lattice) \[7\]: Let \( B = \{b_1, b_2, \ldots, b_n\} \) be a linear independent vector group, where \( b_i \in \mathbb{R}, 1 \leq i \leq n \). The \( n \)-dimensional lattice \( \Lambda \) is generated by \( B \) :
\[
\Lambda = L(B) = \{ y \in \mathbb{R}^n \mid y = \sum_{i=1}^{n} x_i b_i, x_i \in \mathbb{Z} \}.
\]

Definition 2 (Discrete Gaussian Distribution) \[7\]: For any \( c > 0 \), the Gaussian function centered at \( t \in \mathbb{R} \) with parameter \( c \) is \( \rho_{c,t}(x) = \exp(-\pi \|x-t\|^2 / c^2), x \in \mathbb{R} \).

Let \( \rho_{c,t}(\Lambda) = \sum_{x \in \Lambda} \rho_{c,t}(x) \). Define the discrete gaussian distribution as: \( D_{\Lambda,c,t}(x) = \frac{\rho_{c,t}(x)}{\rho_{c,t}(\Lambda)}, x \in \Lambda \).

Definition 3 (R-LWE)\[9\]: For any real \( \alpha > 0 \), Define \( \Phi_\alpha \) as the distribution obtained by sampling from a normal variable with mean 0 and standard deviation \( \sqrt{2\pi} \),
\[ \Phi_\alpha (y) = \sum_{k=-\infty}^{\infty} \frac{1}{\alpha} \exp\left(-\pi \left(\frac{y-k}{\alpha}\right)^2\right), \quad y \in [0,1). \] For any probability distribution \( \Phi_\alpha \), generate an discrete distribution \( \varphi \) over \( Z_p \) according to the random variable \( p \cdot X_\varphi \), in which \( X_\varphi \) has distribution \( \Phi_\alpha \).

Let \( v, u_i \in R_p, \chi_i \in \varphi, 1 \leq i \leq m \), \( \alpha_i, \chi_i \) are chosen independently. Giving a list of equations \( c_i = u_i^T \cdot v + \chi_i, 1 \leq i \leq m \), the R-LWE problem donates recovering secret value \( s \) from these equations.

For \( U = [u_1, u_2, \ldots, u_m] \in R^m_p \), \( \in R^m_p C_0 = (C_{0,1}, C_{0,2}, \ldots, C_{0,m}) \), \( \chi = (\chi_1, \chi_2, \ldots, \chi_m) \in R^m_p \), denote the R-LWE problem as \( C_0 = U^T v + \chi \).

**Definition 4 (Trapdor Function)**[8]: \( \text{TrapGen}(n, m, p) \rightarrow (A, T) \): Firstly, the function randomly generates a matrix \( a \in R^m_p \) with \( n \) rows and \( m \) columns. Then the function outputs \( T \in \Lambda^\perp(A) = \{ Ae = 0 \pmod{p} \mid e \in Z^m_p \} \), which is constituted by the integer vectors orthogonal to \( A \).

\( \text{SampleD}(Aa, c, l) \): Take a \( m \)-dimensional lattice \( \Lambda \), a parameter \( c > 0 \) and a center \( l \) as input, where \( c \in Z_p, l \in R_p \), the algorithm tries to output the closest vector in the lattice from center \( l \). Define that \( m \) is the number of iterations. Then the function sets \( v_m = 0 \) and \( l_m = l \). Output \( l_i = \langle l_i, b_i \rangle / \langle b_i, b_i \rangle \), \( c_i = \| b_i \| \) and choose a \( z_i \) from \( D_{Z^m, c, l} \), in which \( b_i \) is the \( i \)th basis of \( \Lambda \).

Finally define \( l_{-1} = l_i - z_i b_i \in Z^m_p \) and \( v_{-1} = v_i + z_i b_i \in Z^m_p \), the function outputs \( v_0 \).

\( \text{SamplePre}(A, T, c, u) \rightarrow e \): For \( \{ Ae = u \pmod{p} \mid e \in Z^m_p \} \), find an \( l \) that \( Al = u \pmod{p} \) according to the map \( e + A^\perp \rightarrow Ae \pmod{p} \). Then the conditional distribution of \( e \) is exactly \( D_{\Lambda^\perp, c, l} \).

Therefore, we can sample a \( v \) from \( \text{SampleD}(\Lambda^\perp, c, l) \) and output a \( e = l + v \). Since \( e \) is small enough, it is hard enough to successfully figure out \( e \).

**3. L-ABE Scheme**

**3.1. System Model**

As is shown in figure 1, we propose a cloud storage system which includes five entities as follows:
Figure 1. System Model.

**CA:** The CA is a global trusted authority in the system. Firstly, the CA sets up the system. Then every user needs to send their identity information to the CA for registration. The CA doesn’t participate in any attribute assignment or any generation of secret keys which are related to attributes.

**KGC:** The KGC is an authority that is in charge of assigning, revoking, and renewing users’ attributes. Once receiving the certificate from a user, the KGC assigns a set of corresponding attributes to the user and generates the secret key $s$. The KGC is also responsible for updating secret keys when attribute revocation happens.

**CS:** The CS is responsible for storing ciphertext uploaded from users and offers data access service. When an attribute revocation happens, the CS can update the ciphertext.

**User:** After registering on the CA, each user can query the ciphertexts from the CS. They can also encrypt their own data and upload the ciphertext to the CS.

### 3.1.1. Threat Model

In our L-AWE scheme, the threat model is defined in terms of the trusted CA and KGC, the curious but honest CS, legal users, revoked users and online intruders.

The CA and the KGC are trusted, which means that the CA and the KGC will always execute the requirements of all entities in the scheme correctly and are not curious about the content of the messages. The CS is curious but honest, which means that it will never deny service queried by any authority user, and will correctly execute the instructions all the time. However, it is probably curious about the content of the ciphertext or other receiving data.

The legal users are users who have the access authority to decrypt ciphertexts.

The revoked users are users whose attributes are revoked and attempt to decrypt the ciphertext they have no access to.

The online intruders are the network attackers who try to decrypt the ciphertexts with no access authority and secret keys.

In this paper, the online intruders may collude with the legal users and the revoked users. They may exchange messages with each other to launch the collusion attacks.

### 3.2. Construction

#### 3.2.1. UserRegister

Users ought to send their identity information to the CA to make register. If the user’s identity is legal, the CA assigns a unique identity $uid \in R_p$ to this user and generate the certificate $cert(uid)$. Then the CA sends the certificate $cert(uid)$ to the corresponding user.
3.2.2. KGCSetup. The KGC selects a random integer \( \eta \), a security parameter \( n \) as a power of 2, a large prime modulus \( p = 1 \mod (2n) \) and a smaller positive integer \( q \). Let \( y(x) = x^p + 1 \). Then, the KGC selects a set of attributes \( W = \{ \text{attr}_1, \text{attr}_2, \ldots, \text{attr}_W \} \), where \( |W| \) represents the number of attributes in the system. Then the KGC sets a Hash function \( H_1 \), which maps \( x \in R \) to \( H(x) \in R \), a Hash function \( H_2 \), which maps \( x \in R^2 \) to \( H_2(x) \in R^2 \), and a positive integer \( m > 5n \log q \), where \( m \) should be an integer multiple of \( |W| \), namely \( m = \lambda |W| \).

Then the KGC applies the \( \text{TrapGen}(n, m, p) \) to output matrix \( \{\cdot\cdot\cdot\}^{m}_j \) and generates \( U^\perp = \Lambda \). For each attribute \( j \), \( \text{TrapGen}(n, m, p) \) generates a matrix \( A_j \) and its corresponding \( A_j^\perp = \Lambda \). Then let the \( j \)th column vector \( u_j \in U \) represents attribute \( j \) and randomly select an integer \( \omega \geq \log m \). Run the trapdoor function \( \text{SamplePre}(A_j, T_j, s_j, u_j) \) to output \( e_j \in Z^p \), where \( A_j e_j = u_j (\mod p) \). Finally, for each attribute \( j \), the KGC randomly selects an version key \( a_j \in Z_p \) and computes the public attribute key \( PK_j = \eta a_j A_j \).

3.2.3. Secret Key Generation. The Secret Key Generation can be divided into 2 steps:

Step 1: The KGC randomly selects a reversible vectors \( f_q^{-1} \in R^2, f_p^{-1} \in R^2 \) and a vector \( h \in R \). Then compute the public key \( PK_0 = h \cdot f_p^{-1} (\mod p) \). After one user \( i \) sending certificate \( \text{cert}(uid) \) to the KGC, the KGC sends a random integer \( ipt \in R \) and a set of attribute \( iW \) to \( i \) if the certificate is legal.

Step 2: After receiving the access policy \( N \) sending by the user \( i \), the KGC generates a matrix \( U_N = \{u_1, u_2, \ldots, u_W\} \). If \( u_j \in W_i \), then \( u_j' = u_j \). Otherwise \( u_j' = u_j \). Then the KGC can compute \( T_{U_N} = \Lambda(U_N) \) according to \( T_U \). Then select a random integer \( s \geq \|T_{U_N}\| \cdot \omega \log m \) and run the trapdoor function \( \text{SamplePre}(U_N, T_{U_N}, s, f) \) to output some \( r = (r_1, r_2, \ldots, r_W) \in Z^W \), where \( U_N \cdot r = f (\mod p) \). Finally, the KGC generates the secret key \( SK_{i,j} = \{SK_{i,1} = \eta^{-1} t_j e_j H_1(\text{uid}), SK_{i,2} = t_j^i r_j a_j^{-1} H_2(\text{uid})\}, j \in W_i \} \) and sends it to the user.

3.2.4. Data Encryption. The user who owns some data (called data owner) divided the message \( M \) into two sections: \( M = M_1 \parallel M_2 \). Then randomly selects a vector \( g \in R \) and also divided it into \( g_1 \parallel g_2 \). Compute \( M_1 = (g_1 \parallel g_2) \oplus H_1(M_2) \), \( M_2 = (g_1 \parallel g_2) \oplus H_1(M_2) \), \( G = H_2(M_1 \parallel M_2 \parallel g) \) and generate the ciphertext \( CT = M_1 \parallel M_2 + q \cdot G \cdot PK_0 (\mod p) \).

3.2.5. Data Decryption. If the attributes of user \( i \) satisfy the access policy \( N \), \( i \) can successfully compute \( a = CT \cdot \sum j \in W_i (PK_j \cdot SK_{j,1}) \cdot SK_{j,2} (\mod p) = (m + q \cdot g \cdot PK_0) \cdot f (\mod p) = (m \cdot f + q \cdot g \cdot h) (\mod p) \).

Then limit the modulus of \( a \) into \( [-\frac{P}{2}, \frac{P}{2}] \) and generate \( M_1 \parallel M_2 = a \cdot f_q^{-1} (\mod q) \).
Then user can get $H_i(M_r)$ and compute $(M_r \| g_r) = M_2 \oplus H_i(M_r)$. Thus the right section $M_r$ has been decrypted. After that, the user can obtain the left section $M_l$ by computing $(M_l \| g_r) = M_1 \oplus H_i(M_r \| g_r)$. Therefore, the message $M = (M_l \| M_r)$ has been successfully decrypted.

In addition, the user can also get $g_r = (g_l \| g_r)$, $G = H_i(M_l \| M_r \| g_r)$. Thus the integrity of $CT$ can be verified by recalculating the $CT = M_l \| M_r + q \cdot G \cdot PK_0 (mod\ p)$.

3.2.6. Attribute Revocation. Let $W'$ be a set of attributes needed to be revoked. For each attribute $j \in W'$, the KGC generates a new version key $a_j' \in Z_p$ and updates the public attribute keys $PK_j' = \eta \cdot a_j \cdot A_j$. Then the KGC generates the key update keys $KUK_j = t_i^{-1} \cdot r_j \cdot ((a_j)^{-1} - a_j^{-1})$ and sends them to users who possess the corresponding attributes.

Once receiving the $KUK_j$ from KGC, the user $i$ updates the secret key section $SK_{j,2} = SK_{j,2} + KUK_j$, $j \in |W| \cap |W'|$.

3.3. Security Analysis

**Theorem 1:** The L-ABE scheme can resist chosen plaintext attack (CPA).

**Proof:** We describe a game [15] to make the proof. Supposing that there exists an adversary who can break our game in polynomial time with advantage $\epsilon$. Firstly, an emulator randomly selects two big prime numbers $p, q$ where $p \neq q$, positive integer $n, m$ and two reversible vector $f_0, f_1 \in R$. Then the emulator generates a Hash function $H_1(x) \in R$, a Hash function $H_2(x) \in R$, and a Hash function $H_3(x) \in R$. Then the emulator tosses coin and gets a random $\mu$. If $\mu = 0$, the emulator chooses $f_0$. Otherwise it chooses $f_1$.

**Initialization:** The challenger runs $\text{TrapGen}(n, m, p)$ to output a matrix $U = [u_1, u_2, \cdots, u_m] \in R_p^m$ and corresponding $T = \Lambda(U)$. For each attribute $j$, the challenger runs the $\text{TrapGen}(n, m, p)$ to output a matrix $T_j = \Lambda(A_j)$ and selects a random integer $s_j \geq \| T_j \| \cdot \omega \cdot \log m$. Then it runs $\text{SamplePre}(A_j, T_j, s_j, u_j)$ to generate $e_j \in Z_p^m$, where $A_je_j = u_j \pmod p$. For each attribute $j$, KGC randomly selects a $\eta \in Z_p$ and a version key $a_j \in Z_p$ and generates a public attribute key $PK_j = \eta \cdot a_j \cdot A_j$. Then the challenger creates a new access policy $L$.

**Step 1:** The adversary selects a group of $(uid, W_{uid})$ and sends it to the challenger to query the secret keys, where $uid$ has been selected during initialization and all $W_{uid}$ cannot satisfy the access policy $L$. Then the challenger runs the secret key generation algorithm to generate two vector $f_{\mu p}^{-1} \in R_p$ and $f_{\mu q}^{-1} \in R_q$. Then select a vector $h \in R$ and publish the public key $PK_{\mu 0} = h \cdot f_{\mu p}^{-1} (mod\ p)$. The challenger randomly selects $t_i \in Z_p$ and output $v_i = dt_i$. Select a random $r_j \in Z_p$. Finally, the challenger outputs the secret key $SK = (SK_{j,1} = \eta^{-1} \cdot v_i e_j H(uid)^{-1}, SK_{j,2} = v_i^{-1} r_j a_j^{-1} H(uid), j \in W)$.

**Challenge:** The adversary selects two different messages of equal length $m_0$ and $m_1$ and sends them to the challenger. Then the challenger tosses a coin to output a random $b$. Then the challenger runs the
data encryption algorithm to get \( m_{b_1} = (m_{b_2} \| g_t) \oplus H_1(m_{b_2} \| g_t) \), \( m_{b_2} = (m_{b_2} \| g_t) \oplus H_1(m_{b_1}) \), \( G = H_2(m_{b_3} \| m_{b_3} \| g) \). Then generate the ciphertext \( CT = m_{b_3} \| m_{b_2} + qG \cdot PK_s(\mod p) \).

**Step 2:** Same to Step 1, the adversary selects a group of \( L \) and sends it to the challenger to query the secret keys and all \( W_{uid} \) cannot satisfy the access policy \( L \).

**Guess:** The adversary outputs the guess \( b' = 0 \) or \( b' = 1 \).

If the adversary gives correct guess \( b' = b \), we can assume that the adversary has the advantage \( \varepsilon \) in attack L-ABE scheme, namely \( \Pr(b' = b) = \frac{1}{2} + \varepsilon \). While the adversary gives wrong guess \( b' \neq b \), we can assume that the adversary has no advantage, namely \( \Pr(b' \neq b) = \frac{1}{2} \). Therefore, the whole advantage of the adversary in winning the game equals to \( \frac{1}{2} \Pr(b' = b) + \frac{1}{2} \Pr(b' \neq b) - \frac{1}{2} = \frac{1}{2} (1 + \varepsilon) + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} = \frac{1}{2} \), which is negligible in polynomial time. Thus, our scheme is secure against CPA.

### 3.4. Storage Cost Comparison

In this section, we compare the storage load of the lattice based CP-ABE scheme [12], lattice-based encryption scheme [15] and our L-AWE scheme in Table 1. In addition, \( |W| \) represents the number of attributes in the system, \( |S| \) represents the number of attributes of user \( i \), \( |J| \) represents the number of attributes related to the ciphertext, \( |I| \) represents the number of users in the system in our scheme, \( k \) represents the bits of messages that are encrypted once in the scheme [12, 15] and \( m \) represents a trapdoor running parameter in the scheme [12, 15].

| Scheme          | KGC                                                                 | user                                                                 | CS                        |
|-----------------|----------------------------------------------------------------------|----------------------------------------------------------------------|---------------------------|
| Scheme [12]     | \((3mn|W| + 2|W| + 7) \cdot \log p\)                                      | \((mn|S|) \cdot \log p\)                                              | \((mn|J| + n) \cdot \log p\) |
| Scheme [15]     | \((2mn + 2m^2) \cdot |I| + 5\) \cdot \log p                                     | \(2m^2 \cdot \log p\)                                               | \((mn + k) \cdot |I| \cdot \log p\) |
| Our L-ABE Scheme| \((mn + mn|S| + 4n + |I|) \cdot \log p\)                              | \(2|S| \cdot \log p\)                                               | \(2n \log p\)              |

**KGC:** The KGC needs to store the public keys, public parameters and master secret keys.
As we can see in Table 1, the storage cost of KGC in our scheme is less than that in scheme [12]. Since \(|S| < |W|\), then we can get \((mn + mn|S| + 4n + |I|) < 3mn|S| < (3mn|W| + 2|W| + 7)\). Therefore, the storage cost of KGC in our scheme is less than that in scheme [12].

The storage cost of KGC in scheme [15] is \((2mn + 2m^2) \cdot |I| + 5\) \cdot \log p while it is \((mn + mn|S| + 4n + |I|) \cdot \log p\) in ours. Since that \( mn + mn|S| + 4n + |I| < 2mn + m^2 \cdot |I| < (2mn + 2m^2) \cdot |I| + 5 \), the storage cost of KGC in our scheme is less than that in scheme [15].

**User:** Users need to store their secret keys.
As is shown in Table 1, the storage load of users in our scheme is obviously less than that in scheme [12]. In addition, since \( |S| < m \), the storage load of users in our scheme is less than that in scheme [15].
CS: The CS needs to store the ciphertexts. As is shown in Table 1, the storage load of CS in our scheme is obviously less than that in scheme [12,15]. Therefore, our L-ABE scheme can achieve high efficiency.

4. Conclusion
In this paper, we propose a lattice attribute-based encryption scheme for cloud storage based on R-LWE problem, which is called L-ABE scheme. Our L-ABE scheme can achieve both quantum computing attack resistance by applying R-LWE problem and achieve fine-grained access control by using CP-ABE to realize attribute revocation. Additionally, the L-ABE scheme is proven to be secure against chosen plaintext attack and can reach high efficiency. Finally, this article has not made formal proof of quantum attack resistance and collusion attack resistance, which is what we plan to do in future work.

Acknowledgments
This work is supported by the Key Project of Chinese National Programs for Fundamental Research and Development (973 program) (NO.2013CB338003), the National Natural Science Foundation of China (NO.61372103), the Key Laboratory of Computer Network Technology of Jiangsu Province and the Key Laboratory Program of Information Network Security of Ministry of Public Security (No. C19607). Rui Jiang is the corresponding author.

References
[1] Sahai and B. Waters, “Fuzzy Identity-Based Encryption”, *International Conference on Theory and Applications of Cryptographic Techniques*, Aarhus, Denmark, 2005, pp. 457-473.
[2] Shamir A, “identity-based cryptosystems and signature schemes,” *The Workshop on the Theory & Application of Cryptographic Techniques*, Santa Barbara, USA, 1984, pp. 47-53.
[3] Hur J, Noh DK, “Attribute-based access control with efficient revocation in data outsourcing systems,” *IEEE Trans. Parallel Distrib. Syst.*, 2011, 22, (7), pp. 1214-1221.
[4] W. Wang, G. Zhang and Y. Shen, “A CP-ABE Scheme Supporting Attribute Revocation and Policy Hiding in Outsourced Environment,” *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, 2018, pp. 96-99.
[5] V. Reshma, S. J. Gladwin and C. Thiruvengatesan, “Pairing-Free CP-ABE based Cryptography Combined with Steganography for Multimedia Applications,” 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2019, pp. 501-505.
[6] Shor, Peter W. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. Siam Review,” *SIAM Journal on Computing*, 1997, 26, (5), pp. 1484-1509.
[7] Regev O, “On lattices, learning with errors, random linear codes, and cryptography,” in *Proc. of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, New York, USA, 2005, pp. 84-93.
[8] Gentry C, Peikert C, Vaikuntanathan V, “Trapdoors for hard lattices and new cryptographic constructions,” in *Proc. of the 40th Annual ACM Symposium on Theory of Computing*, New York, 2008, pp.197-206.
[9] D. Stehle, R. Steinfeld, K. Tanaka, K. Xagawa, “Efficient public key encryption based on ideal lattices,” *Advances in Cryptology ASIACRYPT 2009 - Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security* (Lecture Notes in Computer Science LNCS 5912), 2009, pp. 617-635.
[10] Lyubashevsky V, Peikert C, Regev O, “On Ideal Lattices and Learning with Errors over Rings,” *Proceedings of the 29th Annual international conference on Theory and Applications of Cryptographic Techniques*, Riviera, French, 2010, pp. 1-23.
[11] K. D. Gür, Y. Polyakov, K. Rohloff, G. W. Ryan, H. Sajjadpour and E. Savas, “Practical Applications of Improved Gaussian Sampling for Trapdoor Lattices,” in *IEEE Transactions on Computers*, 2019, 68, (4), pp. 570-584.
[12] Z. C. Liu, Z. L. Jiang, X. Wang, Y. L. Wu, S. M. Yiu, “Multi-authority ciphertext policy attribute-based encryption scheme on ideal lattices”, *16th IEEE International Symposium on Parallel and Distributed Processing with Applications*, Melbourne, Australia, 2018, pp. 1003-1008.

[13] Wang Y, “Lattice ciphertext policy attribute-based encryption in the standard model,” International Journal of Network Security, 2014, 16, (6), pp. 444-451.

[14] X. Liu, J. Ma, J. Xiong, Q. Li, T. Zhang and H. Zhu, “Threshold attribute-based encryption with attribute hierarchy for lattices in the standard model.” *IET Information Security*, 2014, 8, (4), pp. 217-223.

[15] Kim K S, Jeong I R, “Collusion-resistant unidirectional proxy re-encryption scheme from lattices,” Journal of Communications & Networks, 2016, 18(1): 1-7.