Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.027276

ARTICLE

Quantum-Resistant Multi-Feature Attribute-Based Proxy Re-Encryption Scheme for Cloud Services

Jinqiu Hou[1], Changgen Peng[1,*], Weijie Tan[1,2] and Hongfa Ding[3]

[1]State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang, 550025, China

[2]Key Laboratory of Advanced Manufacturing Technology, Ministry of Education, Guizhou University, Guiyang, 550025, China

[3]College of Information, Guizhou University of Finance and Economics, Guiyang, 550025, China

*Corresponding Author: Changgen Peng. Email: cgpeng@gzu.edu.cn

Received: 22 October 2022 Accepted: 28 April 2023 Published: 22 September 2023

ABSTRACT

Cloud-based services have powerful storage functions and can provide accurate computation. However, the question of how to guarantee cloud-based services access control and achieve data sharing security has always been a research highlight. Although the attribute-based proxy re-encryption (ABPRE) schemes based on number theory can solve this problem, it is still difficult to resist quantum attacks and have limited expression capabilities. To address these issues, we present a novel linear secret sharing schemes (LSSS) matrix-based ABPRE scheme with the fine-grained policy on the lattice in the research. Additionally, to detect the activities of illegal proxies, homomorphic signature (HS) technology is introduced to realize the verifiability of re-encryption. Moreover, the non-interactivity, unidirectionality, proxy transparency, multi-use, and anti-quantum attack characteristics of our system are all advantageous. Besides, it can efficiently prevent the loss of processing power brought on by repetitive authorisation and can enable precise and safe data sharing in the cloud. Furthermore, under the standard model, the proposed learning with errors (LWE)-based scheme was proven to be IND-sCPA secure.

KEYWORDS

Lattice; learning with errors; attribute-based proxy re-encryption; linear secret sharing schemes

1 Introduction

It is worth noting that sensitive data is being shared and held increasingly in the third party, such as AWS, AliCloud, iCloud, etc. In the current era of cloud computing and data protection, fine-grained access management of encrypted data is a crucial requirement. While sharing data in an open, complicated network environment, there is a chance that personal information will be compromised. For example, in telemedicine system, patients have to store the medical data to the cloud server of the hospital, so that medical service personnel can better analyze the health status of patients after downloading from the cloud. While sharing medical data brings about much convenience to patients and medical service personnel in the system, it also causes new privacy and security issues. Medical data usually contains patients' sensitive information, thus, it is extremely important for patients. In

addition, patients would only like the medical data to be obtained by authorized medical service personnel. In an ideal situation, people hope to encrypt data to a semi-trusted cloud service provider for privacy protection purposes. At the same time, the encrypted data can realize data access control and ciphertext selection calculation as well. In other words, a cryptographic mechanism is needed to make sure "who" can access the encrypted data, and that they can get "what" from the encrypted data.

Proxy re-encryption (PRE) [1] is regarded as a particular primitive in public key cryptography that can give flexible data access authorization for encrypted data in accordance with user needs. Additionally, due to the capability of PRE to securely convert ciphertext, this technology has undergone extensive research and has proven to be quite useful in the cloud context. Currently, PRE is widely employed in many areas of the cloud computing environment, including access control, distributed file systems, encrypted mail forwarding systems, spam filtering systems, etc. ABPRE provides a good solution for the above scenarios. A semi-trusted proxy in an ABPRE system having access to a re-encryption key (created by a delegator) can convert ciphertext that satisfies an access policy into another ciphertext for a delegatee that complies with a new access policy. This greatly reduces the overhead of the data encryption process, enhances non-interactive fine-grained access control, and significantly improves efficiency. Users can share or access data securely, and reliably through a semi-trusted cloud computing service provider. It should be noted that in this process, the proxy is unable to obtain anything about the plaintext. Users achieve the goal of sharing data file safely and efficiently in this way. Due to its characteristics, ABPRE is very suitable for cloud storage environment.

1.1 Related Works

In 2011, Boneh et al. [2] put forth the idea of functional encryption (FE), which broke the deadlock of the original "all or nothing" access mode. FE can not only accomplish the objective of fine-grained access control (only users who meet certain policies can decrypt), but also can select ciphertext. Compared with traditional public-key cryptography (PKC), FE has stronger expression ability. Later, the fuzzy identity-based encryption was constructed by Sahai et al. [3], which was also regarded as the original form of attribute-based encryption (ABE). Especially, ABE is a special FE. In an ABE system, the ciphertext and the secrect key correspond to the attribute set and access policy respectively. The user can decode and access the ciphertext via the secret key after the attribute set of the ciphertext fulfills the access policy.

Lattice cryptography is a kind of PKC, which is widely considered to not be threatened by quantum computing. What's more, the security of lattice cryptography is based on the difficulty of solving lattice problems in the average case. Based on this superior feature, scholars began to focus on the design of the FE schemes on lattice. Boyen [4] realized FE for access structures based on the LWE hardness assumptions in 2013. More specifically, the key-policy attribute-based encryption (KP-ABE) scheme with monotonic access structure was the first ABE scheme on lattice that supported general Boolean expressions. In 2018, Dai et al. [5] first reported their implementation of a lattice-based KP-ABE scheme, which used short secret keys. What's more, the homomorphism of the public key and the ciphertext was considered in their proposed scheme as well. In 2019, Tsabary [6] constructed a fully secure ciphertext-policy attribute-based encryption (CP-ABE) for t-CNF from LWE. Varri et al. [7] presented a CP-ABE scheme from the lattice. It is noteworthy that their scheme only allowed valid users of the access policy to conduct keyword searches on the encrypted index, but users who were not in the access policy could not obtain documents in the ciphertext. Recently, to realize attribute revocation, Zhao et al. [8] presented a revocable ABE scheme, which can expediently renew users' attributes to revoke or grant their access rights. Besides, Fu et al. [9] put forward an offline/online

CP-ABE, which had better computational performance for mobile device scenario. In the same year, Fu et al. [10] comprehensively summarized various kinds of ABE schemes from the lattice in terms of complexity assumptions, expressiveness, security, efficiency. In addition, they discussed ABE schemes on lattices which were deserving further research.

With the rapid development of cloud storage technology, the problems of data security and sharing have received extensive attention from industry and academia. PRE is an encryption method that can safely convert ciphertext. It allows that a non-completely trusted third party can directly convert the user Alice's ciphertext into other users' ciphertext without decryption, which guarantees the privacy and security of the data left with the third party.

The following desired characteristics should be met by a pretty PRE scheme:

- Proxy transparency: In the transparent PRE scheme, neither the delegator nor the delegatee knows the existence of the proxy, meanings that the ciphertext sent to the delegatee after re-encryption is indistinguishable from the ciphertext originally sent to the delegatee;
- Non-interactivity: The delegator does not require the assistance of the delegatee or any other third party for the generation of the proxy re-encryption key;
- Unidirectionality: The non-reliable proxy can only change the ciphertext of the delegator into the ciphertext of the delegatee; Conversely, it cannot change the delegatee's ciphertext;
- Multi-use: The non-reliable proxy can also repeatedly re-encrypt the ciphertext that has already been re-encrypted in the unidirectional PRE, as shown in Fig. 1.

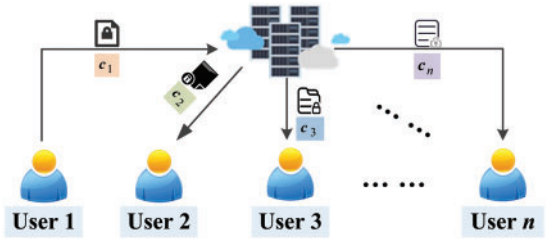

Figure 1: Schematic of multi-use PRE

The initial proposal for a lattice-based multi-bit encryption, unidirectional, and multi-use PRE scheme was made by Jiang et al. [11]. They improved the encryption efficiency. Besides, they proved that in the standard model under LWE, the scheme achieved CPA security. In 2016, PRE with multiple properties was put forward by Kim et al. [12]. What's more, the scheme was key optimality, non-interactivity, unidirectional, invisibility, collusion-resistant and non-transferability. But unfortunately, only in the random oracle model was it demonstrated to be secure. In 2021, Dutta et al. [13] presented the concrete constructions identity-based PRE, which proved both selective and adaptive security. Furthermore, the scheme they came up with satisfied the nature of unidirectionality and anti-collusion. However, fine-grained access control was not supported by the proposed scheme. Therefore, the subsequent research work was aimed at designing a new cryptographic primitive combining ABE and PRE [14,15] in order to meet the requirements.

The ABPRE scheme combines ABE with PRE. This not only ensures that the new encryption scheme has the special conversion property of PRE, but also enables accessing the encrypted data of users who satisfy the access structure. This is achieved by setting up a corresponding access structure. The owner of the data has total authority over the data, while ensuring data confidentiality. Nowadays,

ABPRE is widely used in distributed file systems, electronic medical systems, cloud storage services and other scenarios. Li et al. [16] constructed a key-policy attribute-based proxy re-encryption (KP-ABPRE) scheme based on the matrix access structure, but it was difficult to resist quantum attacks. Although lattice-based ABE and PRE schemes have been realized with the continuous development of lattice theory, the ABPRE schemes against quantum attacks have been realized only in the last few years. Therefore, there is less relevant literature about lattice-based ABPRE. Li et al. [17] designed the first ciphertext-policy ABPRE under the LWE assumption. Their scheme only supported the access structure of the AND gate, and the access policy's expressiveness was relatively weak. To be able to control the proxy's ability to re-encrypt the original ciphertext, Liang et al. [18] constructed the conditional ABPRE from lattice firstly, which met the requirements of more fine-grained data sharing. However, the schemes of [17] and [18] only satisfied the nature of single-use. This limited the practicality of the scheme. In addition, Susilo et al. [19] constructed an ABPRE that is honest and secure against re-encryption attacks, improving the security of ABPRE. To illustrate the potential of our research, we compare the existing (attribute-based) PRE schemes with our put forward scheme in Table 1.

Table 1: Qualitative comparison of (attribute-based) PRE schemes

Cryptosystem	Assumptions	Post quantum	Proxy transparency	Non-interactivity	Unidirectionality
DQW20 [20]	CDH	×	×	✓	✓
GSB22 [21]	BDHE	×	✓	✓	✓
XWZ22 [22]	DBDH	×	✓	✓	✓
HJG19 [23]	LWE	✓	✓	×	×
Our scheme	LWE	✓	✓	✓	✓

From the Table 1, we can see that the existing PRE schemes generally have the following obvious problems:

- The majority of the assumptions underlying current ABPRE scheme research come from traditional number theory. However, these traditional encryption schemes can not resist quantum attacks;
- The feasibility of the existing PRE schemes is somewhat hampered by the fact that they only meet two or three characteristics. Besides, the proxy is regarded as a semi-trusted party, but there are few restrictive measures to check the legitimacy on the malicious proxy's activities;
- At present, most ABPRE schemes expression strategies are limited, which seriously hinders the feasibility of the ABPRE schemes in practical applications.

In summary, it is crucial to create a powerful ABPRE scheme to withstand quantum attacks in cryptography. Fortunately, the lattice-based cryptosystems can effectively resist quantum attacks. Consequently, constructing lattice-based ABPRE schemes with multiple properties has important theoretical significance and broad application prospects.

1.2 Our Contributions

Designing a post-quantum secure ABPRE scheme with a variety of properties under the standard model is a very meaningful research project. Therefore, we constructed an ABPRE scheme based on key-policy with re-encryption verifiability in the research (named KPAB-VPRE):

- LSSS matrix is adopted to obtain a KP-ABPRE scheme that supports any monotonic policies. The delegator can formulate the corresponding attribute sets and encrypt the message on these attribute sets. Only when the attribute sets on the ciphertext meet the delegatee's access policy can the ciphertext be decrypted. Our KP-ABPRE scheme uses the access structure constructed by the attribute sets to control the delegatee that can realize the flexible PRE. The delegatee can be one person, one organization or multiple organizations;
- Taking the activities of corrupt proxy into consideration, the scheme is combined with homomorphic signature technology to realize the verifiability of re-encryption. In other words, during the re-encryption process, our KP-ABPRE with re-encryption verifiability (KPAB-VPRE) scheme can be verified whether the proxy performed an honest re-encryption operation. This property greatly enhances the security of the PRE;
- In general, few ABPRE schemes can satisfy three or more properties. While, we design a multi-feature KPAB-VPRE scheme with proxy transparency, non-interactivity, unidirectionality multi-use and anti-quantum attack, which can greatly enhance the practicability of the program. What's more, under the standard model, our KPAB-VPRE scheme is proven to be selectively IND-CPA secure.

1.3 Organization

In order to better obtain an understanding of this article, we introduce the relevant notations in the next Section 2, and briefly discuss some basic knowledge that we use. The definition and security model of the KPAB-VPRE scheme are described in the part of Section 3. Then, we construct a KPAB-VPRE scheme with various properties based on LWE. Besides, the security and properties of KPAB-VPRE are analyzed in Section 4. Furthermore, we also assess the effectiveness of the KPAB-VPRE in comparison to relevant literature in Subsection 4.5. Finally, Section 5 presents the conclusions.

2 Preliminaries

2.1 Notations

In this paper, we apply some initial symbols, as shown in the Table 2. $\mathbb{Z}$ and $\mathbb{R}$ represent the sets of integers and real numbers, respectively. For a matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$, we adopt $\mathbf{A}^T$ to mark the transposed matrix. What's more, we shorten algorithm names. For example, **ReEncrypt** is recorded as re-encryption algorithm.

2.2 Preliminaries

We give the following useful definitions and lemmas according to literature [4,24–27], consisting of lattice, some algorithms, decision LWE, LSSS and HS.

Table 2: Symbol description

Symbols	Definitions
$x \in \mathbb{Z}_q$	Random numbers on integer module q spaces
$\mathbf{b} \in \mathbb{Z}_q^n$	n-dimensional random vectors on integer module q space
$\mathbf{B} \in \mathbb{Z}_q^{n\times m}$	n-row m-column matrices on integer module q space
$\|\mathbf{A}\|$	l_2-norm of a matrix $\mathbf{A}$
Λ	Lattice
Ψ	Gaussian noise distribution
$\widetilde{\mathbf{B}}$	Gram-Schmidt orthogonalization result of a matrix $\mathbf{B}$
$O(\cdot)$	Asymptotic upper bound
$[l]$	Set $\{1, 2, \ldots, l\}$
$Att_i \vDash (\mathbf{M}, \rho)$	Attribute set Att_i matches the policy $(\mathbf{M}, \rho)$

2.2.1 Lattice

Definition 2.1. Given a $n \times m$ matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m] \in \mathbb{R}^{n\times m}$ with linearly independent columns $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m \in \mathbb{R}^n$. The $\mathbf{A}$ forms a lattice Λ. Besides, Λ^* is called dual lattice.

$$\Lambda(\mathbf{A}) = \left\{ \mathbf{u} \in \mathbb{R}^n, \ \ s.t. \ \ \exists \mathbf{s} \in \mathbb{Z}^m, \quad \mathbf{u} = \mathbf{A}\mathbf{s} = \sum_{i=1}^{m} s_i a_i \right\} \tag{1}$$

$$\Lambda^* = \left\{ \mathbf{h} \in \mathbb{R}^m \ \ s.t. \ \ \exists \mathbf{u} \in \Lambda, \ \ \mathbf{h}^T\mathbf{u} = \langle \mathbf{h}, \mathbf{u} \rangle \in \mathbb{Z} \right\}. \tag{2}$$

Lemma 2.1. Set the following parameters, $q \geq 3, m = \lceil 6n \log q \rceil$. Then, there exists a probability polynomial time (PPT) algorithm **TrapGen**(n, m, q, σ), which generates a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a basis $\mathbf{T}_\mathbf{A} \in \mathbb{Z}_q^{m\times m}$ for $\Lambda_q^\perp(\mathbf{A})$ meeting the following equation:

$$\|\widetilde{\mathbf{T}_\mathbf{A}}\| \leq O(\sqrt{n \log q}), \|\mathbf{T}_\mathbf{A}\| \leq O(n \log q). \tag{3}$$

Here $\widetilde{\mathbf{T}_\mathbf{A}}$ is the Schmidt orthogonalization matrix of $\mathbf{T}_\mathbf{A}$, and $\|\mathbf{T}_\mathbf{A}\|$ is the Euclidean norm of matrix $\mathbf{T}_\mathbf{A}$.

Lemma 2.2. We assume that given a prime number $q > 2$, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}, \sigma \geq \|\widetilde{\mathbf{T}_\mathbf{A}}\| \cdot \omega(\sqrt{\log m})$, where $\mathbf{T}_\mathbf{A}$ is the basis of $\Lambda_q^\perp(\mathbf{A})$. Next, we choose two vectors $\mathbf{u} \in \mathbb{Z}_q^n$ and $\mathbf{c} \in \mathbb{Z}^m$. Then,

- $Pr[\mathbf{x} \leftarrow D_{\Lambda_q^\perp(\mathbf{A},\sigma)} : \|\mathbf{x}\| > \sigma\sqrt{m}] \leq negl(n)$, where $negl(n)$ is a negligible function;
- There exists a PPT algorithm **SamplePre**$(\mathbf{A}, \mathbf{T}_\mathbf{A}, \sigma, \mathbf{u})$, which makes it output a vector $\mathbf{x} \in \Lambda_q^\mathbf{u}(\mathbf{A})$ with statistical properties similar to the discrete Gaussian distribution $D_{\Lambda_q^\mathbf{u}(\mathbf{A}),\sigma,\mathbf{c}}$;
- There is a PPT algorithm **SampleBasisLeft**$(\mathbf{A}, \mathbf{B}, \mathbf{T}_\mathbf{A}, \sigma)$, which makes it output a set of basis $\mathbf{T}_{(\mathbf{A}|\mathbf{B})}$ on the lattice $\Lambda_q^\perp(\mathbf{A}|\mathbf{B})$ that is statistically close to the distribution $D^m_{\sigma,\Lambda_q^\perp(\mathbf{A}|\mathbf{B})}$;
- Given three matrices $\mathbf{A}, \mathbf{G} \in \mathbb{Z}^{n\times m}$ and $\mathbf{S} \in \mathbb{Z}^{m\times m}$, $\mathbf{T}_\mathbf{G}$ is a short basis for the lattice $\Lambda_q^\perp(\mathbf{G})$. Then, there is a PPT algorithm **SampleBasisRight**$(\mathbf{A}, \mathbf{G}, \mathbf{S}, \mathbf{T}_\mathbf{G}, \sigma)$, which makes it output a basis $\mathbf{T}_{(\mathbf{A}|\mathbf{A}\mathbf{S}+\mathbf{G})}$ for the lattice $\Lambda_q^\perp(\mathbf{A}|\mathbf{A}\mathbf{S} + \mathbf{G})$ distributed statistically close to the distribution $D^m_{\sigma,\Lambda_q^\perp(\mathbf{A}|\mathbf{A}\mathbf{S}+\mathbf{G})}$.

Lemma 2.3. Given a positive integer n, a prime number q, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, and a vector $\mathbf{e} \in D_{\mathbb{Z}^m,\sigma}$. We set $m \geq 2n\log_2 q, \sigma \geq \omega(\sqrt{\log_2 m})$. If so, the distribution of the vector $\mathbf{u} = \mathbf{Ae} \bmod q$ is statistically close to the uniform distribution on $\mathbb{Z}_q^n$.

The above lemmas are used in the security proof of our scheme to show that the simulated system is indistinguishable from the real system.

2.2.2 Decision LWE$_{q,\Psi}$

Definition 2.2. We assume that the positive integers $n, m, q \in \mathbb{Z}$, and set Ψ_α^m be an error distribution on $\mathbb{Z}_q^m$. Let $\mathbf{A}$ and $\mathbf{e}$ be uniformly chosen at random from $\mathbb{Z}_q^{n\times m}$ and Ψ_α^m, respectively. Then, select secretly a vector $\mathbf{s} \in \mathbb{Z}_q^n$ with uniform distribution. There is a LWE oracle with two algorithms:

- O_s : Outputs the samples $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$ according to LWE distribution, where $\mathbf{b} = \mathbf{A}^T\mathbf{s} + \mathbf{e} (\bmod q)$;
- $O_\$$: Outputs the samples $(\mathbf{A}, \mathbf{b})$ from the uniformly random distribution $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$.

The attacker $\mathscr{A}$ could find a solution to the decision LWE$_{q,\Psi}$ problem, if and only if its advantages $Adv(\mathscr{A}) = |Pr[\mathscr{A}^{O_s} = 1] - Pr[\mathscr{A}^{O_\$} = 1]| > negl(n)$.

2.2.3 LSSS

Definition 2.3. If the following requirements are met by a secret sharing system Π over a group of participants P, we call it LSSS over $\mathbb{Z}_P$:

- Each participant's shares compose a vector over $\mathbb{Z}_P$;
- The share generating matrix $\mathbf{M}$ for Π is a matrix of l rows and θ columns. Additionally, the function ρ is defined by the participant labeling row i as $\rho(i)$ for all the i-th row of $\mathbf{M}(i = 1, 2, \ldots, l)$. Besides, the column vector $\mathbf{v} = (s, r_2, \ldots, r_n)$ is predetermined, where $s \in \mathbb{Z}_P$ is a secret that is about to be shared, $r_2, \ldots, r_n \in \mathbb{Z}_P$ are selected at random. Moreover, $\mathbf{M} \cdot \mathbf{v}$ is the vector of l shares of s in the light of Π. Furthermore, the participant $\rho(i)$ owns the share $\lambda_i = (\mathbf{M} \cdot \mathbf{v})_i$.

In addition, a LSSS has linear reconstruction's characteristics.

2.2.4 Homomorphic Signature-HS

Lemma 2.4. The HS scheme typically makes up the following four algorithms:

- **HS.KeyGen**(λ, d, n): The algorithm returns a pair of keys $(hssk, hsvk)$ by inputting the security parameter λ, the depth d of a circuit, and the message length n.
- **HS.Sign** $(hssk, m)$: The algorithm enters $hssk$ and a message m, produces the original signature σ.
- **HS.SignEval**$(\delta, (m_i, \sigma_i))$: The algorithm inputs an evaluation circuit $\delta : \{0,1\}^N \to \{0,1\}^N$ with maximum depth d, and the pair (m_i, σ_i), outputs an evaluation signature $\sigma_{m'}$.
- **HS.Verify**$(hsvk, \delta, m', \sigma_{m'})$: The algorithm inputs $hsvk$, δ, an evaluation massage m', and an evaluation signature $\sigma_{m'}$. If verification is successful, returns result 1, otherwise returns 0.

Correctness: The HS scheme is correct if the following equation holds for any $\lambda, d, n, \delta, m \in \{0,1\}^n$, **HS.KeyGen** $(\lambda, d, n) \to (hssk, hsvk)$, **HS.Sign** $(hssk, m_i) \to \sigma_i$, $m' = \delta(m_i)$:

$$Pr\left[\textbf{HS.Verify}(hsvk, \delta, m', \textbf{HS.SignEval}(\delta, (m_i, \sigma_i))) = 1\right] = 1 \tag{4}$$

3 Defnition and the Security Model of KPAB-VPRE

3.1 System Model

Fig. 2 is an illustration of KPAB-VPRE. And in this encryption system, if user 1 and user 2 want to share encrypted data files, they need to execute the following series of algorithms (**Setup**, **KeyGen**, **Encrypt**, **ReKeyGen**, **ReEncrypt**, **ReEncVer**, **Decrypt**). Especially, an additional verification algorithm **ReEncVer**, executed by the verification server, is used to determine whether the ciphertext is an honest transformation. First, User 2 sends a sharing request to User 1, and uploads the access policy to the key generation center. Next, the key generation center issues public and private keys to User 1 and User 2, respectively. After receiving the request, user 1 encrypts the data to be shared with its own private key and calculates the re-encryption key. Then sends the original ciphertext and re-encrypted key to the proxy. The proxy re-encrypts the ciphertext with the re-encryption key and sends the re-encryption ciphertext to the verification server. The verification server sends the re-encrypted ciphertext to User 2 after judgment. Finally, User 2 decrypts the shared data of User 1 by its private key.

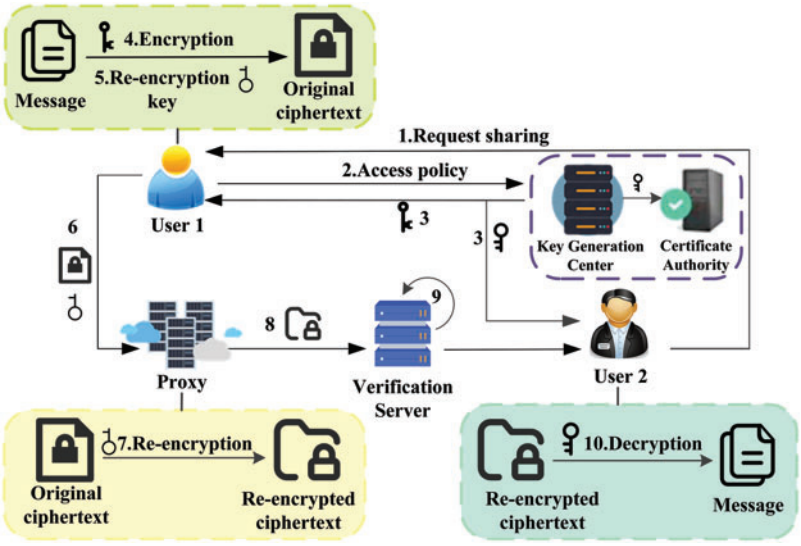

Figure 2: The system model of our proposed KPAB-VPRE

Definition 3.1. The following seven algorithms make up a KPAB-VPRE scheme:

- **Setup**$(1^{\lambda}, \mathscr{U})$: Inputs a security parameter 1^{λ} and an attribute domain $\mathscr{U}$, then the algorithm returns the public parameters pp and the master key msk.
- **KeyGen**(pp, msk, P): Inputs the parameters pp generated by **Setup** and an access policy P on an attribute domain $\mathscr{U}$ to return the secret key sk.
- **Encrypt**(pp, m, Att): Enters pp, message m and an attribute set Att on the attribute domain $\mathscr{U}$ to return the original ciphertext C_1.
- **ReKeyGen**(pp, sk_1, Att'): Enters pp, sk_1 and a new attribute set Att' to return the re-encryption key $rk_{1\to 2}$.
- **ReEncrypt**$(pp, rk_{1\to 2}, C_1)$: Enters pp, $rk_{1\to 2}$ and the original ciphertext C_1, if $Att' \models P$, outputs the re-encrypted ciphertext C_2; otherwise, outputs the symbol $\perp$.
- **ReEncVer**$(pp, hsvk, C_1, C_2)$: Enters pp, $hsvk$, the ciphertext C_1 and C_2, if C_2 is correctly converted from C_1, outputs the result 1, which means the verification is passed; otherwise, outputs 0.

- **Decrypt**(pp, sk, C): Enters pp, sk and $C_i (i = 1, 2)$, then returns the message m. Note that
  - (i) For the original ciphertext C_1, enters the user 1's secret key sk_1. Finally, decryption successfully returns m; otherwise, the error mark $\perp$ is output.
  - (ii) For the converted ciphertext C_2, inputs another user 2's secret key sk_2. After that, the decryption successfully outputs the underlying message m; otherwise, the error mark $\perp$ is output.

Correctness: For parameters $\lambda, m, Att \in \mathscr{U}$, if the attribute sets satisfy the access policy, then a correct KPAB-VPRE scheme should meet the following Eqs. (5)–(7) requirements:

$$\textbf{Decrypt}(pp, sk_1, \textbf{Encrypt}(pp, m, Att)) = m \tag{5}$$

$$\textbf{Decrypt}(pp, sk_2, \textbf{ReEncrypt}(pp, rk_{1\to 2}, C_1)) = m. \tag{6}$$

$$Pr\,[\textbf{ReEncVer}(pp, hsvk, C_1, C_2) = 1] = 1. \tag{7}$$

3.2 Security Model

In the part, we mainly describe the security model of KPAB-VPRE, which is based on the indistinguishability under chosen-plaintext attack in the selective security model (IND-sCPA). And we illustrate the model through the interactive games between the adversary $\mathscr{A}$ and the challenger $\mathscr{C}$. First, we input the security parameter λ, an attribute domain $\mathscr{U} = \{Att_1, Att_2, \ldots, Att_l\}$, where $Att_i = \{att_{i_1}, att_{i_2}, \ldots, att_{i_t}\}$, $(t \le l)$. Then, the game is comprised of $\mathscr{A}$'s operations and the following oracles $(\mathscr{O}_{sk}, \mathscr{O}_{rk}, \mathscr{O}_{rv})$, which could be called more than once in any order. Besides, the following is the interaction process between them:

- **Target.** The adversary $\mathscr{A}$ initially chooses a set of target attributes Att^* to challenge.
- **Instance.** The adversary $\mathscr{A}$ receives the system settings as a consequence of the challenger $\mathscr{C}$'s execution of the **Setup** algorithm.
- **Phase 1.** The adversary $\mathscr{A}$ is allowed to perform the following inquiries multiple times:
  - (i) Secret Key Query $\mathscr{O}_{sk}$: The attacker $\mathscr{A}$ sends the access policy P to a challenger $\mathscr{C}$ for secret key queries $\mathscr{O}_{sk}$. However, the target attribute Att^* on P cannot be queried. Finally, the challenger $\mathscr{C}$ calls the secret key generation algorithm **KeyGen**(pp, msk, P) to return the generated users' secret key sk to $\mathscr{A}$.
  - (ii) Re-Encryption Key Query $\mathscr{O}_{rk}$: The adversary $\mathscr{A}$ initiates $\mathscr{O}_{rk}$ to the challenger $\mathscr{C}$. When the challenger $\mathscr{C}$ receives the query, first runs the private key generation algorithm **KeyGen**(pp, msk, P) to generate sk, and next carries out the **ReKeyGen**(pp, sk, Att') algorithm to sends $rk_{1\to 2}$ to $\mathscr{A}$.
  - (iii) Re-Encryption Verification Query $\mathscr{O}_{rv}$: After receiving the converted ciphertext C_2, $\mathscr{C}$ needs to access the oracle $\mathscr{O}_{rv}$ and implement an algorithm **ReEncVer**$(pp, hsvk, C_1, C_2)$ to check whether the C_2 is legal.
- **Challenge.** First, $\mathscr{A}$ chooses two plaintext messages m_0 and m_1 of equal length. Next, sends them to $\mathscr{C}$. After $\mathscr{C}$ obtains two messages, a value $b \in \{0, 1\}$ is randomly selected. Besides, $\mathscr{C}$ enforces the algorithm **Encrypt**(pp, m, Att), and finally exports the resulting ciphertext to $\mathscr{A}$.
- **Phase 2:** $\mathscr{A}$ proceeds with **Phase 1** queries during this process.

- **Guess.** $\mathscr{A}$ gives out his guess $b' \in \{0,1\}$ to $\mathscr{C}$. What's more, when $b' = b$, means that $\mathscr{A}$ wins the interactive game. In addition, the $\mathscr{A}$'s advantage to victory is

$$\begin{aligned}\varepsilon &= Adv_{IND-sCPA}^{\mathscr{O}_{sk},\mathscr{O}_{rk},\mathscr{O}_{rv}}(1^{\lambda},\mathscr{U}) \\ &= \left|Pr[b'=b]-\frac{1}{2}\right|.\end{aligned} \tag{8}$$

For any PPT $\mathscr{A}$, if $\mathscr{A}$'s advantage $Adv_{IND-sCPA}^{\mathscr{O}_{sk},\mathscr{O}_{rk},\mathscr{O}_{rv}}(1^{\lambda},\mathscr{U})$ against the KPAB-VPRE scheme is negligible, then the KPAB-VPRE scheme is IND-sCPA secure.

4 Scheme of our KPAB-VPRE

4.1 Construction

- **Setup**$(\lambda, n, q, m, \mathscr{U})$: This algorithm takes security parameters λ, positive integer $n > \Omega(\lambda)$, prime number $q > 2$, lattice dimension $m > 5n\log q$ and an attribute domain $\mathscr{U} = \{Att_1, Att_2, \ldots, Att_l\}$, where $Att_i = \{att_1, att_2, \ldots, att_t\}$, $(t \le l)$ as input. Then,
  - (i) For each attribute $att_i \in \mathscr{U}$, performs the algorithm **TrapGen**(n, m, q, σ) to produce two matrices, one is the uniform and random matrix $\mathbf{A}_{att_i} \in \mathbb{Z}_q^{n\times m}$, $i \in [l]$. The other is a small norm matrix $\mathbf{T}_{\mathbf{A}_{att_i}} \in \mathbb{Z}_q^{m\times m}$, which is a trapdoor basis for $\Lambda_q^{\perp}(\mathbf{A}_{att_i})$, where $\|\widetilde{\mathbf{T}_{\mathbf{A}_{att_i}}}\| \le m\cdot\omega\sqrt{\log m}$;
  - (ii) Chooses a uniform random variable $s \in \mathbb{Z}_q$;
  - (iii) Finally, returns the public parameters $pp = \{\mathbf{A}_{att_i}, s\}_{att_i\in\mathscr{U}}$ and the master key $msk = \mathbf{T}_{\mathbf{A}_{att_i}}$.
- **KeyGen**(pp, msk, P): pp, msk and the access policy P are inputs into this algorithm. Next, the algorithm extracts the private key as follows:
  - (i) Converts the user's access policy P into a shared access policy $(\mathbf{M}, \rho)$ by the linear secret sharing theory. Here, $\mathbf{M}$ is a $n \times n$ share-generating matrix. Additionally, $\rho(i) : [n] \to \mathscr{U}$ is a function which maps the i-th row of the matrix $\mathbf{M}$ to the attribute domain $\mathscr{U}$;
  - (ii) Lets $\mathbf{H} = \mathbf{MG}$, $\mathbf{G} \in \mathbb{Z}^{n\times m}$ is the gadget matrix. Generates an extended trapdoor $\mathbf{T}_{(\mathbf{A}_{att_i}|\mathbf{H})} \in \mathbb{Z}_q^{2m\times 2m}$ by running the **SampleBasisLeft**$(\mathbf{A}_{att_i}, \mathbf{H}, \mathbf{T}_{\mathbf{A}_{att_i}}, \sigma)$ algorithm, and then sets $\mathbf{F}_1 = (\mathbf{A}_{att_i}|\mathbf{H}) \in \mathbb{Z}_q^{n\times 2m}$;
  - (iii) Finally, outputs $sk = \mathbf{T}_{(\mathbf{A}_{att_i}|\mathbf{H})} = \mathbf{T}_{\mathbf{F}_1}$.
- **Encrypt**(pp, m, Att_i): As input, the algorithm accepts pp, $m \in \{0,1\}$ and an attribute set $Att_i = \{att_{i_1}, att_{i_2}, \ldots, att_{i_t}\}$, $(t \le l)$, then does:
  - (i) Chooses uniformly at random a matrix $\mathbf{S} \in \{\pm 1\}^{m\times m}$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$ and two noises $\mathbf{e}_1 \in \Psi_\alpha^{2m}$, $e_2 \in \Psi_\alpha$;
  - (ii) Sets
$$\mathbf{c}_{1,0} = \mathbf{F}_1^T\mathbf{u} + \mathbf{e}_1 \in \mathbb{Z}_q^{2m}; \tag{9}$$
  - (iii) Computes
$$c_{1,1} = [s, 0, \ldots, 0]\mathbf{u} + e_2 + m\lfloor q/2\rfloor \in \mathbb{Z}_q, i \in [n]; \tag{10}$$
  - (iv) Outputs the ciphertext $C_1 = (\mathbf{c}_{1,0}, c_{1,1})$.

- **ReKeyGen**(pp, sk, Att_j): This algorithm enters pp, sk and a new attribute set $Att_j = \{att_{j_1}, att_{j_2}, \ldots, att_{j_t}\}$, $(t \leq l)$ to return the re-encryption key $rk_{1\to 2}$ in the following way:
  (i) For each attribute $att_j \in \mathscr{U}$, chooses a uniformly random matrix $\mathbf{A}_{att_j} \in \mathbb{Z}_q^{n\times m}$, $j \in [l]$. Then sets
  $$\mathbf{F}_2 = (\mathbf{A}_{att_j}|\mathbf{H}) \in \mathbb{Z}_q^{n\times 2m}; \tag{11}$$
  (ii) Creates a low-norm matrix $\mathbf{R}_{1\to 2} \in \mathbb{Z}_q^{2m\times 2m}$ with the **SamplePre**$(\mathbf{F}_1, \mathbf{T}_{\mathbf{F}_1}, \mathbf{F}_2, \sigma)$, with the goal of ensuring that
  $$\mathbf{F}_1 \cdot \mathbf{R}_{1\to 2} = \mathbf{F}_2; \tag{12}$$
  Therefore, we set $rk_{1\to 2} = \mathbf{R}_{1\to 2}$ as re-encryption key.
  (iii) Runs the algorithm **HS.KeyGen**(λ, d, n) to generate a key pair $(hssk, hsvk)$, and parses each row of $\mathbf{R}_{1\to 2}$ to $\mathbf{z}_x \in \mathbb{Z}_q^{2m}$, $(1 \leq x \leq 2m)$, using the algorithm **HS.Sign** $(hssk, \mathbf{z}_x)$ to each $\mathbf{z}_x$ to get the sign σ_x, $(1 \leq x \leq 2m)$, where $hsvk$ is used to verify the signature;
  (iv) Finally, delivers $rk_{1\to 2}$ and the associated signature σ_x over a secure channel to the proxy server.
- **ReEncrypt**$(pp, rk_{1\to 2}, C_1)$: This algorithm accepts as inputs pp, $rk_{1\to 2}$ and C_1. The proxy then produces the re-encrypted ciphertext in the manner described below:
  (i) Finds the vector $\mathbf{g} = (g_1, g_2, \ldots, g_n)$, $\forall i \in [n] : (\mathbf{g}_i = 0) \vee (i \in [Att_i])$ so that
  $$\mathbf{g} \cdot \mathbf{M} = (1, 0, \ldots, 0); \tag{13}$$
  (ii) Sets two vectors
  $$\mathbf{c}_{2,0} = rk_{1\to 2}^T \cdot \mathbf{c}_{1,0} (\text{mod } q); \tag{14}$$
  $$c_{2,1} = c_{1,1}; \tag{15}$$
  (iii) Computes a signature $\sigma_{1\to 2} \leftarrow$ **HS.SignEval**(δ_{C_1}, σ_x) homomorphically where the evaluation circuit $\delta_{C_1}(rk_{1\to 2})$ is defined by the original ciphertext as follows:
  $$\delta_{C_1}(rk_{1\to 2}) = \mathbf{c}_{i,0}; \tag{16}$$
  (iv) Finally, outputs the re-encryption ciphertext $C_2 = (\mathbf{c}_{2,0}, c_{2,1})$ and the signature $\sigma_{1\to 2}$.
- **ReEncVer**$(pp, hsvk, C_1, C_2)$: $pp, hsvk, C_1$ with $\sigma_{*\to 2}$ (if C_1 is generated by encrypting a plaintext m, $\sigma_{*\to 2}$ is empty) and the transformed ciphertext C_2 with $\sigma_{1\to 2}$ are all inputted into this algorithm. Then, executes the algorithm **HS.Verify**$(hsvk, \delta_{C_1}, C_2, \sigma_{1\to 2})$. If it outputs 1, it means that the re-encryption ciphertext is legal, otherwise, outputs 0 indicates that the re-encryption ciphertext is illegal.
- **Decrypt**(sk, C): The user inputs the secret key sk and a ciphertext C. If the attribute sets satisfy the access policy, then follow the steps below to decrypt.
  (i) Construct two new vectors
  $$\mathbf{v} = (s, v_2, v_3, \ldots, v_m)^T \in \mathbb{Z}_q^m; \tag{17}$$
  $$\mathbf{v}' = (s, v_2', v_3', \ldots, v_k')^T \in \mathbb{Z}_q^k; \tag{18}$$

where $v_2, v_3, \ldots, v_m, v'_2, v'_3, \ldots, v'_k \in \mathbb{Z}_q$ are chosen at random, and computes the matrix multiplication $\mathbf{M}_i \cdot \mathbf{v}$, $\mathbf{M}'_j \cdot \mathbf{v}'$, denotes the result by

$$\lambda_1^T = \mathbf{M}_i \cdot \mathbf{v} \in \mathbb{Z}_q \tag{19}$$

$$\lambda_2^T = \mathbf{M}'_j \cdot \mathbf{v}' \in \mathbb{Z}_q \tag{20}$$

where $\{\mathbf{M}_i\}_{i\in[n]}$ is the i-th row of $\mathbf{M} \in \mathbb{Z}_q^{n\times m}$ and $\{\mathbf{M}'_j\}_{j\in[n]}$ is the j-th row of $\mathbf{M}' \in \mathbb{Z}_q^{n\times m}$;

(ii) Lets $\boldsymbol{\mu}_1 = \left(\lambda_1^T, 0, \ldots, 0\right)^T \in \mathbb{Z}_q^n$, $\boldsymbol{\mu}_2 = \left(\lambda_2^T, 0, \ldots, 0\right)^T \in \mathbb{Z}_q^n$, $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ respectively correspond to attribute $\rho(i)$ and $\rho(j)$ of the access policy $(\mathbf{M}, \rho)$ and $(\mathbf{M}', \rho)$;

(iii) Given a Gaussian parameter σ, and then runs the **SamplePre** algorithm to sample a vector $\mathbf{k}_t \in \mathbb{Z}_q^{2m}$, $t \in \{1, 2\}$ such that $\mathbf{F}_1 \cdot \mathbf{k}_1 = \boldsymbol{\mu}_1 \mod q$, $\mathbf{F}_2 \cdot \mathbf{k}_2 = \boldsymbol{\mu}_2 \mod q$;

(iv) Calculates

$$m' = \mathbf{c}_{t,1} - \sum_{i\in\mathscr{S}} g_i \mathbf{k}_t^T \mathbf{c}_{t,0}, t \in \{1, 2\},$$

$$\mathbf{g} = (g_1, g_2, \ldots, g_n), \quad \forall i \in [n] : (\mathbf{g}_i = 0) \vee (i \in [Att_i]); \tag{21}$$

(v) Outputs the result:

$$m = \begin{cases} 0, & |m'| \leq \left\lfloor \frac{q}{4} \right\rfloor \\ 1, & |m'| \geq \left\lfloor \frac{q}{4} \right\rfloor \end{cases}. \tag{22}$$

4.2 Correctness and Parameter

4.2.1 Proof of Correctness

(i) The accuracy of unconverted ciphertext decoding. When $t = 1$, if $Att_i \vDash (\mathbf{M}, \rho)$, the original ciphertext C_1 is decrypted as follows:

$$\begin{aligned} m' &= c_{1,1} - \sum_{i\in S} g_i \mathbf{k}_1^T \mathbf{c}_{1,0} \,(\text{mod } q) \\ &= c_{1,1} - \sum_{i\in S} g_i \mathbf{k}_1^T (\mathbf{F}_1^T \mathbf{u} + \mathbf{e}_1) \,(\text{mod } q) \\ &= [s, 0, \ldots, 0]\mathbf{u} + m\lfloor q/2 \rfloor - \sum_{i\in S} g_i \boldsymbol{\mu}_1^T \mathbf{u} + noise_1 \,(\text{mod } q) \\ &= [s, 0, \ldots, 0]\mathbf{u} + m\lfloor q/2 \rfloor - [1, 0, \ldots, 0][s, v_2, v_3, \ldots, v_k]\mathbf{u} \\ &\quad + noise_1 \,(\text{mod } q) \\ &= m\lfloor q/2 \rfloor + noise_1 \,(\text{mod } q) \end{aligned} \tag{23}$$

(ii) Decryption correctness of converted ciphertext. When $t = 2$, for the re-encryption ciphertext C_2, if $Att_j \vDash (\mathbf{M}, \rho)$, we need to compute

$$\begin{aligned} m' &= c_{2,1} - \sum_{i\in S} g_i \mathbf{k}_2^T \mathbf{c}_{2,0} \,(\text{mod } q) \\ &= c_{2,1} - \sum_{i\in S} g_i \mathbf{k}_2^T (rk_{1\to 2}^T \cdot \mathbf{c}_{1,0}) \end{aligned}$$

$$
\begin{aligned}
&= c_{2,1} - \sum_{i \in S} g_i \mathbf{k}_2^T (\mathbf{F}_2^T \mathbf{u} + \mathbf{R}_{1\to 2}^T \mathbf{e}_1) \\
&= [s, 0, \ldots, 0]\,\mathbf{u} + m\,\lfloor q/2 \rfloor - \sum_{i \in S} g_i \boldsymbol{\mu}_2^T \mathbf{u} + noise_2 \ (\mathrm{mod}\ q) \\
&= [s, 0, \ldots, 0]\,\mathbf{u} + m\,\lfloor q/2 \rfloor - \sum_{i \in S} g_i \left[\mathbf{M}_j' \cdot \mathbf{v}', 0, \ldots, 0\right] \mathbf{u} \\
&\quad + noise_2 \ (\mathrm{mod}\ q) \\
&= m\,\lfloor q/2 \rfloor + noise_2 \ (\mathrm{mod}\ q)
\end{aligned}
\tag{24}
$$

(iii) Correctness of re-encryption ciphertext verification. The effectiveness of the re-encryption ciphertext verification depends on the output of the verification algorithm **HS.Verify**. Besides, in the **ReEncrypt**$(pp, rk_{1\to 2}, C_1)$, the the evaluation circuit is defined by $\delta_{C_1}(rk_{1\to 2}) = \mathbf{c}_{j,0}$ with the original ciphertext $\mathbf{c}_{i,0}$ and $rk_{1\to 2}$. From the correctness definition of HS, it can be seen that if the signature $\sigma_{1\to 2}$ is the result of an honest calculation by the proxy, then the algorithm **HS.Verify**$(hsvk, \delta_{C_1}, C_2, \sigma_{1\to 2})$ will be accepted with overwhelming probability. In summary, the validity of the converted ciphertext has been verified.

Remark 4.1. If the parameter setting is reasonable and the noise item in Eqs. (23) and (24) is small enough, the plaintext message can be recovered correctly after decryption.

4.2.2 Parameter Settings

(i) According to the assumption of the LWE problem, for the Gaussian noise distribution $\mathbf{e}_i \leftarrow \Psi_\alpha^m$ and the normalized variance $\alpha \geq 2\sqrt{m}/q$, the length of the vector satisfies $O(\alpha q\sqrt{m}) \leq 2m$ in Regev's proof [26];

(ii) According to the algorithm **TrapGen**(n, m, q, σ), for the safety parameter $n = \Omega(\lambda)$, we require that the prime modulus $q > 2$, the Gaussian parameter $\sigma \geq m \cdot \omega(\log n)$ and the dimension of the lattice base $m \geq 5n \log q$. If the constraints of each parameter can be satisfied, the length of the lattice base generated by the trapdoor generation algorithm is at most $m \cdot \omega(\sqrt{\log m})$;

(iii) According to the algorithm **SamplePre**$(\mathbf{F}_1, \mathbf{T}_{\mathbf{F}_1}, \mathbf{F}_2, \sigma)$, we set the discrete Gaussian distribution parameter $\sigma \geq \|\widetilde{\mathbf{T}}\| \cdot \omega(\sqrt{\log m})$, the parameter $q \geq 2$ and $m \geq 2n \log q$. If the above constraints are met, $rk_{1\to 2} = \mathbf{R}_{1\to 2}$ is created with overwhelming probability advantage. Besides, the length of the $rk_{1\to 2}$ satisfies $\|\mathbf{R}_{1\to 2}\| \leq 2\sigma m$;

(iv) In order to make the error term $|noise| < 5/q$, α must satisfy the conditions

$$\alpha \leq \frac{1}{5}\left[\left(\omega\left(\sqrt{\log m}\right) + 1\right)(1 + l)\left(m^{1.5}\omega\left(\sqrt{\log m}\right)\right)\right]^{-1};$$

Since $q\alpha \geq \sqrt{m}/2$, q must meet

$$q \geq \frac{5}{2}\sqrt{m}\left[\left(\omega\left(\sqrt{\log m}\right) + 1\right)\left(1 + l\left(m^{1.5}\omega\left(\sqrt{\log m}\right)\right)\right)\right].$$

Thus, we decided on the following scheme parameters:

$$m = 5n^{1+\delta},\ \delta > \lceil \log q \rceil,\ q = \frac{5}{2}\sqrt{m}\left[\left(\omega\left(\sqrt{\log m}\right) + 1\right)\left(1 + l\left(m^{1.5}\omega\left(\sqrt{\log m}\right)\right)\right)\right],$$

$$\alpha = \frac{1}{5}\left[\left(\omega\left(\sqrt{\log m}\right) + 1\right)(1 + l)\left(m^{1.5}\omega\left(\sqrt{\log m}\right)\right)\right]^{-1},\ \sigma = m \cdot \omega(\log n).$$

where l is the count of attributes on the access policy. If the parameters are set reasonably, it can be correctly recovered after the above decryption.

4.3 Security Analysis

Theorem 4.1. Assume that σ, m, n, α and q are the same as in the previous sentence. In the standard model, our KP-ABPRE scheme is IND-sCPA secure under the assumption that the decision $\mathrm{LWE}_{q,\Psi}$ problem is difficult.

Proof. We utilize the assumption that there is a PPT adversary $\mathscr{A}$ with a non-negligible advantage to attack our system under the selective security model via "game-hopping" in order to demonstrate the theorem. The basic idea is to construct three games. The first game is an actual IND-sCPA attack, however the last game is impossible for the attacker to win; Then, based on some difficult assumptions on lattice, it is proved that the first game and the last game are equivalent. We have to demonstrate that these three games are indistinguishable from each other for PPT adversary $\mathscr{A}$ in the following ways.

Game sequence:

- **Game0:** $\mathscr{C}$ and $\mathscr{A}$ are playing a typical IND-sCPA game. Besides, $\mathscr{C}$ faithfully responds to different sk and $rk_{1\to 2}$ queries in accordance with the real scheme's algorithms.
- **Game1:** This game changes the generation of $\mathbf{A}_{att_i}$ and sk.
  - (i) **TrapGen**(n, m, q, σ) algorithm produces a matrix $\mathbf{A}_{att_i}$ in **Game0**, but in **Game1**, $\mathbf{A}_{att_i}$ does not contain trapdoor, the matrix $\mathbf{A}_{att_i} \in \mathbb{Z}_q^{n\times m}$ is a uniformly distributed matrix randomly selected by the challenger $\mathscr{C}$;
  - (ii) The challenger $\mathscr{C}$ sets $\mathbf{H} = \mathbf{A}_{att_i}\mathbf{S} + \mathbf{G}$, and then runs the **SampleBasisRight**$(\mathbf{A}_{att_i}, \mathbf{G}, \mathbf{S}, \mathbf{T}_{\mathbf{G}}, \sigma)$ algorithm to get the private key $sk = \mathbf{T}_{(\mathbf{A}_{att_i}|\mathbf{H})} = \mathbf{T}_{\mathbf{F}_1}$;
  - (iii) Finally, the attacker $\mathscr{A}$ receives the sk back from the challenger $\mathscr{C}$. And the settings of other parameters are the same as **Game0**.
- **Game2:** The production of the challenge ciphertext C^* distinguishes this game from **Game1**. In **Game1**, the **Encrypt**(pp, m, Att_i) algorithm creates C^*. However, in **Game2**, the C^* is a uniformly random and independent matrix from $\mathbb{Z}_q^{2m} \times \mathbb{Z}_q$. The rest of the settings are the same as **Game1**. In this case, the advantage of the attacker $\mathscr{A}$ is zero.

If **Game1** is indistinguishable from **Game0**, and **Game2** is indistinguishable from **Game1**, our KPAB-VPRE scheme is IND-sCPA secure in the standard model.

Game transfer:

- **Game0** to **Game1:** We now prove that **Game0** and **Game1** are indistinguishable through the Lemma 4.1.
- **Lemma 4.1.** If $mn > (n+1)\log_2 q + \omega(\lg n)$, then **Game1** and **Game0** are indistinguishable, and the answer to the attacker $\mathscr{A}$'s inquiries become indistinguishable from the true scheme.

  Proof.
  - (i) The matrix $\mathbf{A}_{att_i} \in \mathbb{Z}_q^{n\times m}$ in **Game0** is produced by the **TrapGen**(n, m, q, σ) algorithm. In addition, it is random. However, in **Game1**, the matrix $\mathbf{A}_{att_i}$ is randomly selected from $\mathbb{Z}_q^{n\times m}$, so, for the attacker $\mathscr{A}$ in polynomial time, the public parameter $\mathbf{A}_{att_i}$ is taken from uniformly distributed $\mathbb{Z}_q^{n\times m}$;

(ii) For the private key, sk is generated by the **SampleBasisLeft** algorithm in **Game0**, but in **Game1**, sk is generated by the **SampleBasisRight** algorithm. By definition of **SampleBasisRight**, we have that $sk = \mathbf{T}_{(\mathbf{A}_{att_i}|\mathbf{H})}$ is distributed as required. Thus, the challenger $\mathscr{C}$'s response to the secret key question is indistinguishable from that in **Game1** for the $\mathscr{A}$.

From the Lemma 4.1, we can see that the $\mathscr{A}$'s advantage in **Game1** is equal to that in **Game0**. Therefore, **Game0** and **Game1** are indistinguishable.

- **Game1** to **Game2:** Assuming that in the selective security model, an attacker $\mathscr{A}$ can discriminate between **Game1** and **Game2** with a non-negligible advantage by chosen plaintext attack, the challenger $\mathscr{C}$ will construct an algorithm $\mathscr{B}$ to solve the decision LWE problem with a non-negligible probability. In the LWE problem, the decision algorithm has access to a sampling oracle $\mathscr{O}$, which can be either a really random sampler $\mathscr{O}_s$ or a pseudo-random sampler $\mathscr{O}_s$ with an embedded secret $\mathbf{u} \in \mathbb{Z}_q^n$. Following is the reduction's progression:

  –**Target.** The challenger $\mathscr{C}$ receives a statement from the adversary $\mathscr{A}$ announcing the target attribute set Att_i^* to attack.

  –**Instance.** The challenger $\mathscr{C}$ demands $l+1$ LWE samples from the oracle $\mathscr{O}$, which we denote as:

$$\begin{aligned}
&[(\mathbf{w}_0, v_0)] \in (\mathbb{Z}_q^n \times \mathbb{Z}_q)\\
&\left[(\mathbf{w}_1^1, v_1^1), \ldots, (\mathbf{w}_1^{2m}, v_1^{2m})\right] \in (\mathbb{Z}_q^n \times \mathbb{Z}_q)^{2m}\\
&\left[(\mathbf{w}_2^1, v_2^1), \ldots, (\mathbf{w}_2^{2m}, v_2^{2m})\right] \in (\mathbb{Z}_q^n \times \mathbb{Z}_q)^{2m}\\
&\qquad\vdots\\
&\left[(\mathbf{w}_l^1, v_l^1), \ldots, (\mathbf{w}_l^{2m}, v_l^{2m})\right] \in (\mathbb{Z}_q^n \times \mathbb{Z}_q)^{2m}
\end{aligned} \tag{25}$$

  –**Setup.** The challenger $\mathscr{C}$ prepares the public parameters as follows once it has obtained the target attribute set Att_i^*:

  (i) The central authority selects the appropriate system parameters n, m, q, σ;

  (ii) If the attribute att_i is in the target attribute set Att_i^*, then the challenger $\mathscr{C}$ constructs a matrix $\mathbf{F}_1^* = \left[\mathbf{w}_i^1 | \ldots | \mathbf{w}_i^{2m}\right]$ and a vector $[s, 0, \ldots, 0]^T = \mathbf{w}_0$, where $\mathbf{w}_i (i \in [l])$ comes from LWE sampling;

  (iii) The remaining parameters are the same as **Game1** settings.

Remark 4.2. The above settings have the correct distribution.

–**Phase 1.** The attacker $\mathscr{A}$ can ask the following questions:

Secret Key Query $\mathscr{O}_{sk}$:

(i) The attacker $\mathscr{A}$ sends the access policy P to a challenger $\mathscr{C}$ for private key queries, However, the target attribute on $(\mathbf{M}, \rho)$ cannot be queried, that is, $\rho(i) \notin Att_i^*$, where $\rho(i)$ is the attribute on the access policy $(\mathbf{M}, \rho)$. If $\rho(i) \in Att_i^*$, a random bit is output and terminated;

(ii) According to the parameters in **Game1**, the challenger $\mathscr{C}$ generates sk as previously mentioned and gives it back to the attacker $\mathscr{A}$.

Re-Encryption Key Query $\mathcal{O}_{rk}$:

(i) The challenger $\mathscr{C}$ first generates sk as in $\mathcal{O}_{sk}$. Then, the challenger $\mathscr{C}$ replies with $rk_{1\to 2}$ by running the algorithm **ReKeyGen**(pp, sk, Att_j);

(ii) $rk_{1\to 2}$ is sent by the the challenger $\mathscr{C}$ to the attacker $\mathscr{A}$. The attacker $\mathscr{A}$ can query multiple times.

Re-Encryption Verification Query $\mathcal{O}_{rv}$:

(i) After $\mathscr{A}$ gets the queries $\mathcal{O}_{rk}$ from one access policy P to another access policy P', $\mathscr{A}$ can also make some queries $\mathcal{O}_{rv}$ to $\mathscr{C}$. Next, $\mathscr{C}$ only executes the algorithm **ReEncVer** faithfully;

(ii) Finally, $\mathscr{C}$ sends the result to $\mathscr{A}$ according to the verification algorithm.

Remark 4.3. Because of the unforgeability of HS, the algorithm **ReEncVer** cannot provide any additional capabilities for $\mathscr{A}$ in the selectively IND-CPA security game. That is to say, $\mathscr{A}$ cannot offer invalid ciphertext to pass the algorithm **ReEncVer**. This is very critical. What's more, we are aware that the security features of the underlying algorithm **HS.Verify** determine if the algorithm **ReEncVer** is valid. Furthermore, $\mathscr{A}$ must be able to counterfeit the signature $\sigma^*_{1\to 2}$ and pass the algorithm **HS.Verify** in order to pass the game of re-encryption verification, as this is also a legal signature for HS. In other words, $\mathscr{A}$ has the same benefits as violating the unforgeability of HS when it comes to our KPAB-VPRE scheme's re-encryption verifiability.

–**Challenge.** In order to indicate that it is open to a challenge, $\mathscr{A}$ selects two messages $\{m_0, m_1\} \leftarrow \mathbb{Z}_q^m$ of equal-length. A message m_φ ($\varphi \leftarrow \{0,1\}$) is selected at random by $\mathscr{C}$ for encryption. At last, $\mathscr{C}$ replies with C^* built as follows applying the LWE instance:

(i) Constructing ciphertext $C^* = (\mathbf{c}^*_{0,1}, c^*_{1,1})$ based on LWE samplings. Let

$$v^* = (v_i^1, v_i^2, \ldots, v_i^{2m}) \in \mathbb{Z}_q^{1\times 2m} \tag{26}$$

$$\mathbf{c}^*_{1,0} = (v^*)^T \in \mathbb{Z}_q^{2m} \tag{27}$$

$$c^*_{1,1} = v_0 + m_\varphi \lfloor q/2 \rfloor \in \mathbb{Z}_q \tag{28}$$

where v_i is taken from the LWE instance;

(ii) The challenger $\mathscr{C}$ randomly selects a bit $\varphi \leftarrow \{0,1\}$, if $\varphi = 0$, $C^* = (\mathbf{c}^*_{1,0}, c^*_{1,1})$ that will be returned to the attacker as challenge ciphertext. If $\varphi = 1$, C^* as a challenge ciphertext will be randomly chosen from $\mathbb{Z}_q^{2m} \times \mathbb{Z}_q$ and returned to the attacker $\mathscr{A}$.

Remark 4.4. The above challenge ciphertext $C^* = (\mathbf{c}^*_{1,0}, c^*_{1,1})$ has the correct distribution. A simple analysis reveals that:

∗ If $\mathcal{O} = \mathcal{O}_s$ in the instantiation process, it is a pseudo random sampling oracle embedded in the secret $\mathbf{u} \in \mathbb{Z}_q^n$. Well,

$$v_i = \mathbf{w}_i^T \mathbf{u} + e, e \in \Psi_\alpha. \tag{29}$$

Due to the setting of public parameters, the ciphertext obtained is

$$\begin{aligned}\mathbf{c}_{1,0}^{*} &= (v^{*})^{T} = [v_i^1, v_i^2, \ldots, v_i^{2m}]^T \\ &= [\mathbf{w}_i^1, \mathbf{w}_i^2, \ldots, \mathbf{w}_i^{2m}]^T\mathbf{u} + \mathbf{e}_1 \\ &= \mathbf{F}_1^{*T}\mathbf{u} + \mathbf{e}_1\end{aligned} \tag{30}$$

where $\mathbf{e}_1 \in \Psi_{\alpha}^{2m}$.

$$\begin{aligned}c_{1,1}^{*} &= v_0 + m_{\varphi}\lfloor q/2 \rfloor \\ &= \mathbf{w}_0^T\mathbf{u} + e_2 + m_{\varphi}\lfloor q/2 \rfloor \\ &= [s, 0, \ldots, 0]\mathbf{u} + e_2 + m_{\varphi}\lfloor q/2 \rfloor\end{aligned} \tag{31}$$

where $e_2 \in \Psi_{\alpha}$.

The above $C^{*} = (\mathbf{c}_{1,0}^{*}, c_{1,1}^{*})$ obeys uniform random distribution in statistics;

$*$ $C^{*} = (\mathbf{c}_{1,0}^{*}, c_{1,1}^{*})$ is chosen at random in $\mathbb{Z}_q^{2m} \times \mathbb{Z}_q$ if $\mathscr{O} = \mathscr{O}_{\$}$.

–**Phase 2.** The simulator $\mathscr{B}$ operates in a similar manner to **Phase 1**, the attacker $\mathscr{A}$ is able to query secret keys multiple times, and the set of attributes Att_i^{*} that have done so have not yet met the access structure $(\mathbf{M}, \rho)$.

–**Guess.** After enough questioning, the attacker outputs his guess of φ as φ'. If $\varphi' = \varphi$, the challenger outputs $\mathscr{O}' = \mathscr{O}_{\$}$, otherwise $\mathscr{O}' = \mathscr{O}_{s}$.

In the attacker $\mathscr{A}$'s view, the challenger $\mathscr{C}$'s behavior is close to that of a real, adaptive security experiment. In this game, the attacker $\mathscr{A}$ has the advantage since $\varepsilon = |Pr[\varphi' = \varphi] - \frac{1}{2}|$. Therefore, the advantages of the LWE predictor are as follows:

(i) In a pseudo-random sampler, an attacker $\mathscr{A}$ has the advantageous value ε. In this case, $\mathscr{O} = \mathscr{O}_{\$}$, $Pr[\varphi' = \varphi \mid \mathscr{O} = \mathscr{O}_{\$}] = \frac{1}{2} + \varepsilon$ and the challenger's advantage is

$$Pr[\mathscr{O}' = \mathscr{O} \mid \mathscr{O} = \mathscr{O}_{\$}] = \frac{1}{2} + \varepsilon; \tag{32}$$

(ii) In a true random predictor, an attacker $\mathscr{A}$ has an advantage of 0. In this case, $\mathscr{O} = \mathscr{O}_{s}$, $Pr[\varphi' = \varphi \mid \mathscr{O} = \mathscr{O}_{s}] = \frac{1}{2} + \varepsilon$, the challenger's advantage is

$$Pr[\mathscr{O}' = \mathscr{O} \mid \mathscr{O} = \mathscr{O}_{s}] = \frac{1}{2}. \tag{33}$$

Therefore, assuming that an attacker $\mathscr{A}$ guesses the correct probability is $Pr[\varphi' = \varphi] \geq \frac{1}{2} + \varepsilon$, a challenger $\mathscr{C}$ has the advantage

$$\begin{aligned}&\frac{1}{2}(Pr[\mathscr{O}' = \mathscr{O} \mid \mathscr{O} = \mathscr{O}_{\$}] + Pr[\mathscr{O}' = \mathscr{O} \mid \mathscr{O} = \mathscr{O}_{s}]) - \frac{1}{2} \\ &= \frac{1}{2}\left(\frac{1}{2} + \varepsilon + \frac{1}{2}\right) - \frac{1}{2} = \frac{\varepsilon}{2}\end{aligned} \tag{34}$$

to solve the decision $\mathrm{LWE}_{q,\Psi}$ problem.

As a result, **Game2** and **Game1** cannot be distinguished under the assumption of the decision $\mathrm{LWE}_{q,\psi}$ problem.

In conclusion, the security of our designed KPAB-VPRE scheme is compactly reduced to the decision $\mathrm{LWE}_{q,\psi}$ difficulty assumption by the use of the three equivalent games mentioned above. Additionally, under the standard model, the KPAB-VPRE scheme is selectively IND-CPA secure. This completes the proof.

4.4 Performance Analysis

- **Non-interactivity.** Our scheme is non-interactive, because the re-encryption key $rk_{1\to 2}$ is generated by the **SamplePre**$(\mathbf{F}_1, \mathbf{T}_{\mathbf{F}_1}, \mathbf{F}_2, \sigma)$ algorithm, which is created only depends on the private key $sk_1 = \mathbf{T}_{\mathbf{F}_1}$ in the list of the delegator Alice's attributes, and does not need the private key of the delegatee Bob's access structure. Therefore, the scheme is non-interactive.
- **Proxy transparency.** Our scheme satisfies this property, because the size of the re-encrypted ciphertext $C_2 \in \mathbb{Z}_q^{2m} \times \mathbb{Z}_q$ is the same as the size of the original ciphertext $C_1 \in \mathbb{Z}_q^{2m} \times \mathbb{Z}_q$.
- **Unidirectionality.** This property requires that the ciphertext direction can only be converted from Alice to Bob, not in reverse. This is true, because the proxy can get the re-encryption key $rk_{1\to 2} \leftarrow$ **SamplePre**$(\mathbf{F}_1, \mathbf{T}_{\mathbf{F}_1}, \mathbf{F}_2, \sigma)$ from Alice to Bob. However, the re-encryption key generation method requires Alice's private key. Thus, if the proxy wants to obtain the re-encryption key $rk_{2\to 1}$ from Bob to Alice, without Bob's private key, the proxy cannot convert Bob's ciphertext into Alice's ciphertext.
- **Multi-use.** We suppose that $C_1 = (\mathbf{c}_{1,0}, c_{1,1})$ is the ciphertext of the attribute list 1 for the attribute lists $1, 2, \ldots, k$. The re-encryption process is carried out in the ranges of 1 to k and 2 to $k-1$. In order to create a low-norm matrix $\mathbf{R}_{i\to i+1} \in \mathbb{Z}_q^{2m\times 2m}$ such that $\mathbf{F}_i \cdot \mathbf{R}_{i\to i+1} = \mathbf{F}_{i+1}$ and $rk_{i\to i+1} = \mathbf{R}_{i\to i+1}$, we run the algorithm **SamplePre**$(\mathbf{F}_i, \mathbf{T}_{\mathbf{F}_i}, \mathbf{F}_{i+1}, \sigma)$, $i \in [1, k-1]$. The re-encryption procedures are shown in the Eq. (35).

$$
\begin{aligned}
\mathbf{c}_{k,0} &= rk_{k-1\to k}^T \mathbf{c}_{k-1,0} (\mathrm{mod}\ q) \\
&= rk_{k-1\to k}^T (rk_{k-2\to k-1}^T \mathbf{c}_{k-2,0}) (\mathrm{mod}\ q) \\
&= rk_{k-1\to k}^T (rk_{k-2\to k-1}^T (rk_{k-3\to k-2}^T \mathbf{c}_{k-3,0})) (\mathrm{mod}\ q) \\
&\quad \vdots \\
&= \prod_{i=1}^{k-1} rk_{i\to i+1}^T \mathbf{c}_{1,0} (\mathrm{mod}\ q) \\
&= \prod_{i=1}^{k-1} rk_{i\to i+1}^T (\mathbf{F}_1^T \mathbf{u} + \mathbf{e}_1) (\mathrm{mod}\ q) \\
&= \mathbf{F}_k^T \mathbf{u} + \left(\prod_{i=1}^{k-1} rk_{i\to i+1} \right)^T \mathbf{e}_1 (\mathrm{mod}\ q)
\end{aligned}
\tag{35}
$$

Set $c_{k,1} = c_{1,1}$. Obviously, $C_k = (\mathbf{c}_{k,0}, c_{k,1})$ in the Eq. (35) is the ciphertext of the attribute list k. Using reasonable parameter settings, the noise item is sufficiently small to be decrypted correctly.

4.5 Comparison

In this part, we compare our KPAB-VPRE with other relevant schemes [13,17,28–30] in terms of the size of the ciphertext, access policy, multi-use, security model, and re-encryption verifiability. And Table 3 displays the comparison's findings.

Table 3: Comparison with previous related work

Cryptosystem	Size of ciphertext	Access policy	Multi-use	Standard model	Re-encryption verifiability
DSD21 [13]	$1+m$	×	×	✓	×
LQZ21 [28]	$1+m$	×	✓	✓	×
SRA20 [29]	$l+m$	×	×	×	×
LMZ19 [17]	$1+2lm$	AND-gate	×	✓	×
WYZ21 [30]	$l+2m$	×	✓	✓	✓
Our scheme	$1+2lm$	Any monotonic	✓	✓	✓

Note: m: The dimension of the output lattice. l: The upper limit of all attributes.

As can be seen from Table 3, in the same type of schemes, the literature [13,28–30] did not support the expression of access policy, while the literature [17] only supports "AND" expression. While, our scheme adopts the LSSS matrix to express the access policy and supports the operations of "AND, OR, THRESHOLD". What's more, matrix operation is used to realize encryption and decryption algorithm, which has higher efficiency. In terms of the re-encryption verification, only Wu et al. [30] supported this property, but this scheme cannot support arbitrary access policies. Therefore, the proposed scheme realizes fine-grained access and sharing of encrypted data, as well as meets the multi-use and re-encryption verification, which makes the scheme more practical. Especially, our KPAB-VPRE achieves IND-sCPA security under the standard model.

5 Conclusion

We present a multi-functional LSSS matrix-based KPAB-VPRE scheme from lattice that is proven to be IND-sCPA secure under the standard model. The scheme based on the lattice is implemented by matrix operation, which can facilitate parallel algorithm design and has superior efficiency, as opposed to the classic ABPRE schemes based on bilinear mapping. This scheme is based on the construction of LWE difficult problems from lattice. From the complexity of lattice difficult problems in the worst case, we can see that under the appropriate parameter selection, there is no effective algorithm to solve these difficult problems in polynomial time, even if it is a quantum computer. Therefore, this scheme can resist quantum attacks. In addition, the data owner can encrypt messages on any attribute sets. The ciphertext cannot be actively decoded until the attribute put on it complies with the user's access policy. Furthermore, the verification of re-encryption is realized by introducing homomorphic signature technology, thereby detecting the activities of corrupt proxies, which has higher security and enforceability in practical scenarios. However, in our KPAB-VPRE scheme, the size of the ciphertext is not fixed, and it grows linearly as the number of attributes increases. Therefore, the next study will focus on creating a multi-functional ABPRE system with fixed ciphertext length in the future.

Acknowledgement: The authors are willing to express our appreciation to the reviewers for their constructive comments which significantly enhanced the presentation of the study.

Funding Statement: The project is provided funding by the Natural Science Foundation of China (Nos. 62272124, 2022YFB2701400), the Science and Technology Program of Guizhou Province (No. [2020]5017), the Research Project of Guizhou University for Talent Introduction (No. [2020]61), the Cultivation Project of Guizhou University (No. [2019]56), the Open Fund of Key Laboratory of Advanced Manufacturing Technology, Ministry of Education, GZUAMT2021KF[01] and the Postgraduate Innovation Program in Guizhou Province (No. YJSKYJJ[2021]028).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Jinqiu Hou; data collection: Weijie Tan; analysis and interpretation of results: Changgen Peng. Hongfa Ding; draft manuscript preparation: Jinqiu Hou. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All data generated or analysed during this study are included in this published article.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Blaze, M., Bleumer, G., Strauss, M. (1998). Divertible protocols and atomic proxy cryptography. In: Nyberg, K. (Ed.), *Advances in cryptology-EUROCRYPT'98*, pp. 127–144. Germany: Springer. https://doi.org/10.1007/BFb0054122
2. Boneh, D., Sahai, A., Waters, B. (2011). Functional encryption: Definitions and challenges. In: Ishai, Y. (Ed.), *Theory of cryptography*, pp. 253–273. Germany: Springer. https://doi.org/10.1007/978-3-642-19571-6_16
3. Sahai, A., Waters, B. (2005). Fuzzy identity-based encryption. In: Cramer, R. (Ed.), *Advances in cryptology-EUROCRYPT 2005*, pp. 457–473. Germany: Springer. https://doi.org/10.1007/11426639_27
4. Boyen, X. (2013). Attribute-based functional encryption on lattices. In: Sahai, A. (Ed.), *Theory of cryptography*, pp. 122–142. Germany: Springer. https://doi.org/10.1007/978-3-642-36594-2_8
5. Dai, W., Doröz, Y., Polyakov, Y., Rohloff, K., Sajjadpour, H. et al. (2018). Implementation and evaluation of a lattice-based key-policy ABE scheme. *IEEE Transactions on Information Forensics and Security, 13(5),* 1169–1184. https://doi.org/10.1109/TIFS.2017.2779427
6. Tsabary, R. (2019). Fully secure attribute-based encryption for T-CNF from LWE. In: Boldyreva, A., Micciancio, D. (Eds.), *Advances in cryptology-CRYPTO 2019*, pp. 62–85. Germany: Springer. https://doi.org/10.1007/978-3-030-26948-7_3
7. Varri, U., Pasupuleti, S. K., Kadambari, K. V. (2021). CP-ABSEL: Ciphertext-policy attribute-based searchable encryption from lattice in cloud storage. *Peer-to-Peer Networking and Applications, 14(3),* 1290–1302. https://doi.org/10.1007/s12083-020-01057-3
8. Zhao, S., Jiang, R., Bhargava, B. K. (2022). RL-ABE: A revocable lattice attribute based encryption scheme based on R-LWE problem in cloud storage. *IEEE Transactions on Services Computing, 15(2),* 1026–1035. https://doi.org/10.1109/TSC.2020.2973256
9. Fu, X., Wang, Y., You, L., Ning, J., Hu, Z. et al. (2022). Offline/Online lattice-based ciphertext policy attribute-based encryption. *Journal of Systems Architecture, 130(6),* 102684. https://doi.org/10.1016/j.sysarc.2022.102684
10. Fu, X., Ding, Y., Li, H., Ning, J., Wu, T. et al. (2022). A survey of lattice based expressive attribute based encryption. *Computer Science Review, 43(5),* 100438. https://doi.org/10.1016/j.cosrev.2021.100438

11. Jiang, M., Hu, Y., Wang, B., Wang, F. H., Lai, Q. (2015). Lattice-based multi-use unidirectional proxy re-encryption. *Security and Communication Networks, 8(18),* 3796–3803. https://doi.org/10.1002/sec.1300
12. Kim, K. S., Jeong, I. R. (2016). Collusion-resistant unidirectional proxy re-encryption scheme from lattices. *Journal of Communications and Networks, 18(1),* 1–7. https://doi.org/10.1109/JCN.2016.000003
13. Dutta, P., Susilo, W., Duong, D. H., Roy, P. S. (2021). Collusion-resistant identity-based proxy re-encryption: Lattice-based constructions in standard model. *Theoretical Computer Science, 871(1),* 16–29. https://doi.org/10.1016/j.tcs.2021.04.008
14. Zhang, Y., Deng, R. H., Xu, S., Sun, J., Li, Q. et al. (2020). Attribute-based encryption for cloud computing access control: A survey. *ACM Computing Surveys, 53(4),* 83:1–83:41. https://doi.org/10.1145/3398036
15. Wang, X., Hu, A., Fang, H. (2020). Improved collusion-resistant unidirectional proxy re-encryption scheme from lattice. *IET Information Security, 14(3),* 342–351. https://doi.org/10.1049/iet-ifs.2018.5246
16. Li, K., Zhang, Y., Ma, H. (2013). Key policy attribute-based proxy re-encryption with matrix access structure. *2013 5th International Conference on Intelligent Networking and Collaborative Systems*, Xi'an, China, IEEE. https://doi.org/10.1109/INCoS.2013.17
17. Li, J., Ma, C., Zhang, K. (2019). A novel lattice-based ciphertext-policy attribute-based proxy re-encryption for cloud sharing. In: Meng, W., Furnell, S. (Eds.), *Security and privacy in social networks and big data*, pp. 32–46. Germany: Springer. https://doi.org/10.1007/978-981-15-0758-8_3
18. Liang, X., Weng, J., Yang, A., Yao, L., Jiang, Z. et al. (2021). Attribute-based conditional proxy re-encryption in the standard model under LWE. In: Bertino, E., Shulman, H., Waidner, M. (Eds.), *Computer security-ESORICS 2021*, pp. 147–168. Germany: Springer. https://doi.org/10.1007/978-3-030-88428-4_8
19. Susilo, W., Dutta, P., Duong, D. H., Roy, P. S. (2021). Lattice-based HRA-secure attribute-based proxy re-encryption in standard model. In: Bertino, E., Shulman, H., Waidner, M. (Eds.), *Computer security-ESORICS 2021*, pp. 169–191. Germany: Springer. https://doi.org/10.1007/978-3-030-88428-4_9
20. Deng, H., Qin, Z., Wu, Q., Guan, Z., Zhou, Y. (2020). Flexible attribute-based proxy re-encryption for efficient data sharing. *Information Sciences, 511(1),* 94–113. https://doi.org/10.1016/j.ins.2019.09.052
21. Ge, C., Susilo, W., Baek, J., Liu, Z., Xia, J. et al. (2022). A verifiable and fair attribute-based proxy re-encryption scheme for data sharing in clouds. *IEEE Transactions on Dependable and Secure Computing, 19(5),* 2907–2919. https://doi.org/10.1109/TDSC.2021.3076580
22. Xiong, H., Wang, L., Zhou, Z., Zhao, Z., Huang, X. et al. (2022). Burn after reading: Adaptively secure puncturable identity-based proxy re-encryption scheme for securing group message. *IEEE Internet of Things Journal, 9(13),* 11248–11260. https://doi.org/10.1109/JIOT.2021.3126230
23. Hou, J., Jiang, M., Guo, Y., Song, W. (2019). Efficient identity-based multi-bit proxy re-encryption over lattice in the standard model. *Journal of Information Security and Applications, 47,* 329–334. https://doi.org/10.1016/j.jisa.2019.05.015
24. Agrawal, S., Boneh, D., Boyen, X. (2010). Efficient lattice (H) IBE in the standard model. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Germany, Springer. https://doi.org/10.1007/978-3-642-13190-5_28
25. Howe, J., Khalid, A., Rafferty, C., Regazzoni, F., O'Neill, M. (2018). On practical discrete gaussian samplers for lattice-based cryptography. *IEEE Transactions on Computers, 67(3),* 322–334. https://doi.org/10.1109/TC.2016.2642962
26. Regev, O. (2009). On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM, 56(6),* 34:1–34:40. https://doi.org/10.1145/1568318.1568324
27. Zhao, J., Gao, H. (2017). LSSS matrix-based attribute-based encryption on lattices. *13th International Conference on Computational Intelligence and Security*, Hong Kong, China, IEEE. https://doi.org/10.1109/CIS.2017.00062
28. Li, J., Qiao, Z., Zhang, K., Cui, C. (2021). A lattice-based homomorphic proxy re-encryption scheme with strong anti-collusion for cloud computing. *Sensors, 21(1),* 288. https://doi.org/10.3390/s21010288

29. Singh, K., Rangan, C. P., Agrawal, R., Sheshank, S. (2020). Provably secure lattice based identity based unidirectional PRE and PRE^{+} schemes. *Journal of Information Security and Applications, 54,* 102569. https://doi.org/10.1016/j.jisa.2020.102569
30. Wu, L., Yang, X., Zhang, M., Wang, X. A. (2022). IB-VPRE: Adaptively secure identity-based proxy re-encryption scheme from LWE with re-encryption verifiability. *Journal of Ambient Intelligence and Humanized Computing, 13(1),* 469–482. https://doi.org/10.1007/s12652-021-02911-9