HRA-Secure Puncturable Attribute-Based Proxy Re-Encryption from Lattices for Secure Cloud Sharing

Tianqiao Zhang, Mingming Jiang, Fucai Luo, Yuyan Guo, Jinqiu Hou

***Abstract*—With the rapid advancement of cloud computing technology, outsourcing massive datasets to cloud servers has become a prominent trend, making secure and efficient data sharing mechanisms a critical requirement. Attribute-based proxy re-encryption (ABPRE) has emerged as an ideal solution due to its support for fine-grained, one-to-many access control and robust ciphertext transformation capabilities. However, existing ABPRE schemes still exhibit shortcomings in addressing forward security issues caused by long-term private key leakage, threats from quantum computer attacks, and vulnerabilities to honest re-encryption attacks (HRA). To simultaneously resolve these challenges, this paper introduces a novel cryptographic primitive termed puncturable attribute-based proxy re-encryption with switchable tags (PABPRE-ST), constructing a secure cloud data sharing scheme that supports fine-grained revocation. By integrating puncturable encryption (PE) mechanisms into the ABPRE framework, the scheme achieves fine-grained ciphertext revocation based on tags. In PABPRE-ST, data owners embed tags into ciphertexts, enabling data users to puncture specific tags and thereby revoke access to corresponding ciphertexts at a granular level. Furthermore, the scheme allows delegators to switch ciphertext tags, enhancing sharing flexibility. We formalize the security definitions for the proposed puncturable attribute-based proxy re-encryption scheme and prove its security under the learning with errors (LWE) assumption, which is widely believed to be resistant to quantum computer attacks. Security analysis demonstrates that the proposed scheme achieves HRA security in the standard model.**

***Index Terms*—Cloud data sharing, attribute-based encryption (ABE), proxy re-encryption (PRE), puncturable encryption, learning with errors (LWE).**

I. Introduction

WITH the increasing maturity and widespread adoption of cloud computing technology, outsourcing data to cloud servers for storage and processing has become a mainstream paradigm for individuals and enterprises to enhance operational efficiency and reduce management costs. Cloud platforms, with their exceptional features such as on-demand services, elastic scalability, and global accessibility, significantly facilitate efficient data access and flexible sharing, making cross-organizational and cross-regional collaboration more convenient than ever before. However, entrusting sensitive data to not fully trusted cloud service providers raises serious data privacy and security challenges [1]. Malicious activities within the cloud provider or external attacks could lead to unauthorized disclosure of user-sensitive data.

To ensure data confidentiality, a straightforward approach is for data owners to encrypt data before uploading, thereby guaranteeing that only users with appropriate access privileges can decrypt it. Although such encryption effectively protects data at rest, it restricts flexible sharing among authorized users. Consider the following scenario: a doctor on vacation wishes to securely share encrypted patient medical data stored in the cloud with a colleague. Using traditional methods, the doctor must either download the ciphertext, decrypt it using their private key, re-encrypt it with the recipient's public key, and then upload it—a process that is not only cumbersome and inefficient, contradicting the agile principle of on-demand cloud computing, but also severely disrupts their vacation—or risk sharing their private key directly with the recipient or the cloud server, significantly increasing the risk of key leakage.

To resolve the core conflict between ”data security” and ”sharing convenience,” proxy re-encryption (PRE) [2] emerged. As an elegant cryptographic primitive, PRE allows a data owner to authorize a semi-trusted proxy to transform a ciphertext encrypted under Alice's public key into one that can be decrypted by Bob's private key, without the proxy needing knowledge of the underlying plaintext or user private keys. The entire process is performed on ciphertexts, ensuring the proxy gains no access to the plaintext, and Alice never exposes her private key, thereby achieving seamless and secure sharing of encrypted data while maintaining confidentiality.

However, traditional proxy re-encryption schemes suffer from certificate management issues. Subsequently developed identity-based proxy re-encryption (IB-PRE) schemes [3] simplified public key management to some extent but only support one-to-one sharing modes, making it difficult to achieve fine-grained access control and one-to-many data sharing requirements. To overcome these limitations, Liang et al. [4] proposed integrating attribute-based encryption (ABE) [5, 6] with proxy re-encryption. Depending on how access policies are bound, ABE is mainly divided into key-policy ABE (KP-ABE) [7–9] and ciphertext-policy ABE (CP-ABE) [10, 11]. In KP-ABE, the access policy is embedded in the user's key, while the ciphertext is associated with a set of descriptive attributes; conversely, in CP-ABE, the access policy is embedded in the ciphertext by the data owner, and user keys are associated with user attributes. The fusion of PRE and ABE led to the emergence of a new cryptographic primitive: attribute-

Tianqiao Zhang is with the School of Computer Science and Technology, Huaibei Normal University, Huaibei 235000, China (e-mail: zhangxiaoqiao0199@163.com).

Mingming Jiang is with the School of Computer Science and Technology, Huaibei Normal University, Huaibei 235000, China (e-mail: jiangmm3806586@126.com).

Fucai Luo is with the School of Computer Science and Technology, Zhejiang Gongshang University, Hangzhou 310018, China (e-mail: lfucai@126.com).

Yuyan Guo is with the School of Computer Science and Technology, Huaibei Normal University, Huaibei 235000, China (e-mail: guoyuyan428@163.com).

Jinqiu Hou is with the School of Management Science and Engineering, Anhui University of Finance and Economics, Bengbu 233000, China (email: jinqiu_hou@126.com).

based proxy re-encryption (ABPRE). ABPRE schemes enable a cloud server to securely transform a ciphertext encrypted under one access policy into a ciphertext under another policy. This inherits the powerful fine-grained access control and one-to-many encryption capabilities of ABE while incorporating the flexible delegation and forwarding mechanism of PRE. Subsequently, a series of ABPRE schemes have been proposed [12–23], demonstrating significant application potential.

A. Motivation

Although several ABPRE schemes have been proposed that are applicable to secure cloud data sharing [12-23, 39], existing schemes still suffer from critical limitations and fail to meet fundamental practical requirements. The key issues are outlined as follows:

(1) **Vulnerability to Quantum Attacks.** The rapid advancement of quantum computing threatens to undermine encryption schemes based on classical hard problems, such as large integer factorization and discrete logarithms. Consequently, developing cloud data sharing schemes with post-quantum security has become an urgent priority.

(2) **Lack of Fine-Grained Message Erasure.** Compromising a user's long-term private key jeopardizes the confidentiality of all previously encrypted data. An attacker could retain ciphertext copies and decrypt them upon key leakage. This risk highlights the critical need for a "burn-after-reading" mechanism, i.e., message erasure. Existing approaches are categorized as coarse-grained [24] or fine-grained [25]. Coarse-grained erasure revokes access to all historical ciphertexts after key update, offering limited flexibility. In contrast, fine-grained erasure is better suited for practical cloud data sharing.

(3) **Susceptibility to Honest Re-encryption Attacks.** Cohen [26] pointed out that the security of most existing proxy re-encryption schemes satisfying only chosen-plaintext security may be insufficient in practice. Specifically, the CPA security model does not consider the risk that an "honest but curious" delegate might learn the delegator's private key through a single honestly generated re-encryption ciphertext. In ABPRE, if the delegate can learn the delegator's private key, the delegator would lose access control, which is unacceptable.

To address issue (1), an effective approach is to construct lattice-based ABPRE schemes. Lattice-based cryptographic primitives are widely believed to be resistant to quantum attacks. In recent years, several lattice-based attribute-based encryption schemes have emerged, but most lack ciphertext transformation capability, resulting in insufficient flexibility for data sharing. Although some lattice-based ABPRE schemes [20-22] meet both requirements (1) and (3), they fail to satisfy requirement (2).

Regarding issue (2), puncturable encryption (PE) [25] offers a feasible approach for achieving fine-grained message erasure by providing a fine-grained forward security mechanism. In this mechanism, the encryptor can embed multiple tags (e.g., sender, date, file type) into the ciphertext. The data receiver then updates their key by puncturing specific ciphertext tags and discards the old key. This ensures that the new key cannot decrypt ciphertexts containing the punctured tags, thereby achieving fine-grained message erasure. Several fine-grained erasure schemes based on PE have been proposed [27-35]; however, they also lack ciphertext transformation functionality.

Existing schemes such as PPRE [36] and PIB-PRE [37, 38] support ciphertext transformation and partially fulfill requirement (2). However, PPRE [36] suffers from complex certificate management issues, and neither scheme satisfies requirements (1) and (3). Furthermore, both lack fine-grained access control and one-to-many sharing capabilities. The PCP-ABPRE scheme [23], while post-quantum secure, deviates from conventional puncturable encryption frameworks. In its design, the data owner embeds a tag t into ciphertexts and distributes updated puncture keys to authorized users; only those holding the latest key are permitted to decrypt newly generated ciphertexts tagged with t. This mechanism more closely resembles a revocation strategy, as it merely blocks access to new ciphertexts while still allowing historical messages to be decrypted with the existing private key. Consequently, the scheme fails to achieve requirement (2). Moreover, it only provides weak CPA security, thus falling short of requirement (3), and is restricted to AND-gate access policies. Its reliance on a trusted key generation center (KGC) to produce re-encryption keys violates the non-interactivity requirement of PRE and introduces additional overhead. In addition, PPRE [36], PIB-PRE [37, 38], and PCP-ABPRE [23] do not support dynamic tag switching, which severely limits their flexibility in practical fine-grained erasure scenarios. The CCA-secure PABPRE scheme [39] attempts to achieve forward security through attribute revocation and proxy re-encryption; however, its security guarantee becomes ineffective if an adversary has previously retained copies of old ciphertexts.

In summary, existing schemes can only partially meet the requirements for secure cloud data sharing. Therefore, a natural question arises:

"*Can we construct a flexible and secure cloud data sharing scheme that supports post-quantum security, fine-grained message erasure, and resistance to honest re-encryption attacks?*"

B. Our Contributions

This paper proposes a cloud data sharing scheme with fine-grained erasure capabilities, successfully addressing the aforementioned problem. Specifically, we innovatively introduce a lattice-based puncturable attribute-based proxy re-encryption primitive with switchable tags (PABPRE-ST), which achieves secure re-encryption, fine-grained erasure, and a tag-switching mechanism; see Table I. The main contributions are as follows:

- We introduce a novel cryptographic primitive termed puncturable attribute-based proxy re-encryption with switchable tags. We formalize its security definitions and prove its security under the selective HRA security model. This model represents a significant enhancement over the traditional selective CPA security, effectively resisting honest re-encryption attacks.
- We propose a lattice-based, single-hop, and unidirectional PABPRE-ST scheme and prove its security under the LWE assumption, which is widely used in constructing cryptographic primitives resistant to quantum computer

TABLE I
COMPARISON WITH THE CURRENT STATE-OF-THE-ART ABPRE SCHEMES AND PE SCHEMES

Schemes	F1	F2	F3	F4	F5	F6
[14]	●	●	○	○	−	○
[15]	●	●	○	○	−	○
[16]	●	●	○	○	−	○
[17]	●	●	●	○	−	○
[18]	●	●	○	○	−	●
[19]	●	●	○	○	−	●
[20]	●	●	●	○	−	●
[21]	●	●	●	○	−	●
[22]	●	●	●	○	−	●
[23]	●	●	○	○	○	●
[30]	○	○	−	●	○	●
[31]	○	○	−	●	○	○
[32]	○	○	−	●	○	○
[33]	○	○	−	●	○	○
[34]	○	○	−	●	○	●
[35]	○	○	−	●	○	○
[36]	●	○	○	●	○	○
[37]	●	○	○	●	○	○
[38]	●	○	○	●	○	○
[39]	●	●	●	○	○	○
Ours	●	●	●	●	●	●

F1: Flexible sharing; F2: One-to-many sharing; F3: HRA-secure; F4: Fine-grained erasure; F5: Tag switching; F6: Quantum-secure; "●": Achieved; "○": Not achieved; "−": Not applicable.

attacks. The scheme satisfies properties such as non-interactivity, proxy transparency, key optimality, and non-transitivity. Technically, our PABPRE-ST scheme is built upon the key-switching technique used by Brakerski et al. [40] in fully homomorphic encryption scheme and the classical KP-ABE scheme proposed by Boneh et al. [7].

- Under the defined selective HRA security model and based on the standard LWE assumption, we provide a complete and rigorous security proof for the proposed PABPRE-ST scheme, ensuring its theoretical soundness.

II. RELATED WORK

In this section, we provide a systematic overview of existing Attribute-Based Proxy Re-Encryption (ABPRE) and Puncturable Encryption (PE) schemes.

A. Attribute-Based Proxy Encryption

Proxy re-encryption (PRE), as an elegant cryptographic primitive first introduced by Blaze et al. [2], enables authorized users to generate re-encryption keys for semi-trusted proxies. These proxies can then transform original ciphertexts for other users without exposing the underlying plaintext or the delegator's private key. To address certificate management problem, Green and Ateniese [3] proposed identity-based proxy re-encryption, though it was limited to one-to-one encryption and delegation patterns.

To overcome the one-to-one delegation constraint, Liang et al. [4] integrated attribute-based encryption with PRE, yielding the novel cryptographic primitive of attribute-based proxy re-encryption (ABPRE), which enables fine-grained access control and one-to-many data sharing. Subsequently, Liang et al. [12] constructed a single-hop, unidirectional ciphertext-policy ABPRE (CP-ABPRE) scheme based on LSSS, relying on the Decisional q-parallel BDHE Assumption and achieving selective CCA security in the random oracle model. To eliminate the random oracle requirement and achieve adaptive security, Liang et al. [13] combined dual system encryption with selective proof techniques, constructing a unidirectional, single-hop CP-ABPRE scheme that achieves adaptive CCA security in the standard model. Ge et al. [14] proposed a verifiable and fair CP-ABPRE scheme that allows data sharers to verify the correctness of re-encrypted ciphertexts returned by cloud servers, thereby preventing malicious behavior from cloud providers. Their scheme was proven adaptively CPA-secure under the decisional q-parallel BDHE assumption. To address user revocation challenges, Ge et al. [15] introduced a directly revocable CP-ABPRE scheme where revocation is executed by cloud servers without additional burden on delegators, proving selective CPA security under the same assumption. Duan et al. [16] incorporated multi-authority attribute-based encryption into the PRE framework, constructing a multi-authority ABPRE scheme that mitigates centralization issues inherent in single-authority ABPRE systems, with selective CPA security established under the q-parallel BDHE assumption. Li et al. [17] proposed an ABPRE scheme supporting attribute revocation and demonstrated selective CCA security. While these ABPRE schemes support ciphertext transformation and one-to-many sharing, they remain vulnerable to quantum computer attacks.

To achieve quantum resistance, Li et al. [18] proposed the first lattice-based CP-ABPRE scheme supporting positive/negative attributes with AND-gate access policies, proving CPA security under the LWE assumption. However, this scheme requires trusted authority involvement in re-key generation, lacks non-interactivity, supports only limited AND-gate policies, and is vulnerable to honest re-encryption attacks (HRA). Luo et al. [19] proposed a multi-hop KP-ABPRE scheme that supports a priori bounded Boolean circuits; however, this scheme only achieves CPA security and remains vulnerable to HRA. Subsequently, Susilo et al. [20] constructed the first KP-ABPRE scheme with provable HRA security under the LWE assumption. Further addressing the proxy single-point-of-failure issue in ABPRE systems, Zhao et al. [21] introduced a threshold KP-ABPRE scheme with selective HRA security based on LWE. Zhao et al. [22] designed a revocable KP-ABPRE scheme featuring time-switching functionality, which employs partial key randomization to resist decryption key exposure attacks, yet it remains susceptible to collusion between revoked and unauthorized users.

B. Puncturable Encryption

Puncturable encryption, a concept first introduced by Green and Miers [25], enables fine-grained message erasure through a novel approach: by embedding tags (such as sender, file type, date, etc.) within ciphertexts, users can selectively "puncture" specific tags after decryption—that is, update their private keys to render them incapable of decrypting any future ciphertexts containing the punctured tags. This design philosophy has attracted significant research attention for its ability to achieve precise message revocation.

Susilo et al. [41] proposed a generic lattice-based construction for puncturable encryption based on delegatable fully key-homomorphic encryption, proving that the scheme achieves selective CPA security under the LWE assumption. Phuong et al. [27] pioneered the integration of puncturable encryption with attribute-based encryption (ABE), constructing the first puncturable ABE scheme. Dutta et al. [28] then combined puncturable encryption with identity-based encryption, proposing the first lattice-based puncturable identity-based encryption scheme. Subsequently, Dutta et al. [29] extended this work to hierarchical identity-based encryption, developing a lattice-based puncturable HIBE scheme resistant to quantum computer attacks. Further advancing this line, Dutta et al. [30] integrated attribute-based encryption with puncturable encryption to create a lattice-based puncturable ABE scheme supporting a priori bounded polynomial-depth circuits while maintaining quantum resistance.

Cui and Yi [42] integrated puncturable encryption with revocable attribute-based encryption (ABE), constructing a puncturable and revocable ABE scheme that supports attribute revocation. They proved that the scheme achieves selective CPA security under the q-1 assumption in the random oracle model. Mei et al. [31] proposed a puncturable ABE scheme supporting unbounded attributes and tags, proving adaptive CPA security under the matrix decisional Diffie-Hellman (MDDH) assumption. To address single-point-of-failure concerns in puncturable ABE systems, Wei et al. [32] introduced a multi-authority puncturable ABE scheme, establishing selective CPA security under a modified q-decisional parallel bilinear Diffie-Hellman exponent assumption (termed q-DPBDHE3) in the random oracle model. Subsequently, Deng et al. [33] presented a multi-authority puncturable ABE scheme in the standard model, proving CPA security under the decisional q-1 assumption. Zhang et al. [34] proposed the first provably secure lattice-based puncturable CP-ABE scheme, achieving selective CPA security under the LWE assumption, and further incorporated a revocation mechanism to obtain a combined puncturable and revocable CP-ABE scheme. Wang et al. [35] introduced a puncturable registered ABE scheme that completely eliminates key authorities, demonstrating adaptive CPA security under the decisional bilinear Diffie-Hellman (DBDH) assumption. While these puncturable schemes enable fine-grained message erasure and protect historical data security, they lack ciphertext transformation capabilities.

To address this limitation, Phuong et al. [36] combined puncturable encryption with proxy re-encryption, constructing the first puncturable proxy re-encryption (PPRE) scheme, though it suffered from certificate management issues. Xiong et al. [37] then proposed a puncturable identity-based proxy re-encryption (PIB-PRE) scheme that eliminated certificate management, proving adaptive security under the DBDH assumption. Subsequently, Liu et al. [38] introduced a novel PIB-PRE scheme supporting fine-grained sharing mechanisms and unlimited tags. While these schemes provide both fine-grained message erasure and ciphertext forwarding capabilities, they rely on classical hardness assumptions, making them vulnerable to quantum attacks and unable to support one-to-many sharing.

Li and Shi [39] proposed a CCA-secure puncturable ABPRE scheme based on the DBDH problem, where a trusted authority revokes user attributes and proxy servers re-encrypt ciphertexts to achieve forward security—ensuring that users' new keys cannot decrypt previous data. However, if attackers preserve ciphertext copies, the confidentiality of historical data remains compromised. Affum et al. [23] presented a lattice-based puncturable CP-ABPRE (PCP-ABPRE) scheme where data owners broadcast updated puncture keys to users; only non-revoked users receive these keys and can decrypt ciphertexts containing specific tags. From an implementation perspective, their approach diverges from traditional puncturable encryption—they utilize tags primarily for revoking user decryption privileges, making it more akin to a revocable encryption scheme. Consequently, all users' private keys retain the ability to decrypt previously generated ciphertexts, failing to protect historical data confidentiality.

III. Puncturable Attribute-Based Proxy Re-Encryption with Switchable Tags

In this section, we will present the definitions of the system model, syntax, and security model for PABPRE-ST.

A. The System Model

As shown in Fig. 1, the PABPRE-ST system model comprises the following five core entities:

- **System Administrator:** This entity is considered fully trusted and typically operates as the key generation center (KGC). Its primary responsibilities include system initialization, generating the system's public parameters and master private key, processing user registration requests, and distributing initial keys to users based on predefined policy functions.
- **Data Owner:** A data owner first registers with the system administrator to obtain their initial key. They can then encrypt a message by selecting an attribute set and a tag set according to their requirements, generating an original ciphertext which is subsequently uploaded to the cloud server.
- **Cloud Server:** This entity is typically assumed to be ”semi-trusted.” It is responsible for storing and managing the original ciphertexts uploaded by data owners. Upon receiving a re-encryption key from an original recipient, the cloud server performs the re-encryption operation, transforming the original ciphertext into a re-encrypted ciphertext. Furthermore, it responds to user download requests by providing the corresponding original or re-encrypted ciphertexts.
- **Original Recipient:** This refers to a user who satisfies the access policy of the original ciphertext. Such a user can download and decrypt the original ciphertext. They can also autonomously define an access policy and a tag set to generate a re-encryption key, which is sent to the cloud server to share the data with other users. After decryption, they can further revoke the decryption capability for that specific ciphertext in a fine-grained manner by puncturing particular ciphertext tags.

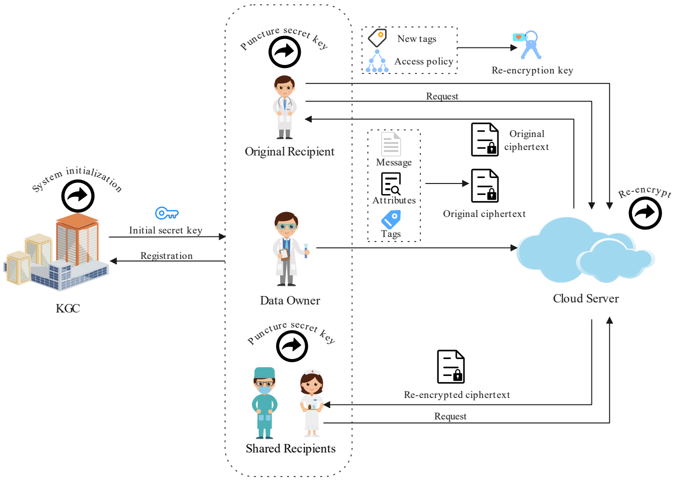

Fig. 1. Syestem model.

- **Shared Recipients:** This refers to users who do not satisfy the access policy of the original ciphertext but are granted data access through authorization from an original recipient. They can download the re-encrypted ciphertext from the cloud server and decrypt it using their own private keys. Similarly, after decryption, they can also revoke their own decryption capability for the ciphertext by puncturing its tags.

B. Syntax

Let $\mathcal{F} = \{f : \mathbb{Z}_q^\ell \to \mathbb{Z}_q\}$ be a family of functions. The KP-PABPRE scheme has an attribute space $\mathcal{X} = \mathbb{Z}_q^\ell$ and a tag space $\mathcal{T} = \mathbb{Z}_q^d$. A single-hop, unidirectional PABPRE-ST scheme is defined by the following seven algorithms (**Setup**, **KeyGen**, **ReKeyGen**, **Enc**, **Puncture**, **ReEnc**, **Dec**):

- **Setup**$(1^\lambda, \ell, d) \to (PP, msk)$: On input the security parameter λ, the attribute length ℓ, and the maximum number of tags per ciphertext d, this algorithm outputs the public parameters PP and the master secret key msk.
- **KeyGen**$(PP, msk, f) \to sk_{f,\emptyset}$: On input the public parameters PP, the master secret key msk, and a policy $f \in \mathcal{F}$, this algorithm outputs the initial secret key $sk_{f,\emptyset}$ for policy f.
- **ReKeyGen**$(PP, sk_{f,\mathcal{P}_{f,\eta}}, f, g, t_1, t_2) \to rk_{(f,t_1)\to(g,t_2)}$: On input the public parameters PP, the latest punctured secret key $sk_{f,\mathcal{P}_{f,\eta}}$ for policy f (where $\mathcal{P}_{f,\eta}$ is the set of punctured tags), and two policy-tag pairs (f, t_1) and (g, t_2) (where $t_1 = (t_{1,1}, ..., t_{1,d}) \in \mathcal{T}$, $t_2 = (t_{2,1}, ..., t_{2,d}) \in \mathcal{T}$), this algorithm outputs a unidirectional re-encryption key $rk_{(f,t_1)\to(g,t_2)}$.
- **Enc**$(PP, x, t_1, \mu) \to ct$: On input the public parameters PP, an attribute vector $x \in \mathcal{X}$, a tag vector $t_1 \in \mathcal{T}$, and a plaintext message $\mu \in \mathcal{M}$, this algorithm outputs a ciphertext ct associated with the attribute vector x and tag vector t_1.
- **Puncture**$(PP, sk_{f,\mathcal{P}_{f,\eta-1}}, \hat{t}_\eta) \to sk_{f,\mathcal{P}_{f,\eta}}$: On input the public parameters PP, a punctured secret key $sk_{f,\mathcal{P}_{f,\eta-1}}$, and a tag $\hat{t}_\eta$ to be punctured, this algorithm outputs an updated punctured secret key $sk_{f,\mathcal{P}_{f,\eta}}$.
- **ReEnc**$(PP, rk_{(f,t_1)\to(g,t_2)}, ct) \to ct'$: When entering the public parameters PP, an original ciphertext ct (encrypted under the attribute vector x and the tag vector $t_1 = (t_{1,1}, ..., t_{1,d})$), and a re-encryption key $rk_{(f,t_1)\to(g,t_2)}$, this algorithm outputs a re-encrypted ciphertext ct' (under the attribute vector y and the tag vector t_2) if $f(x) = 0 \wedge \mathcal{P}_{f,\eta} \cap \{t_{1,1}, ..., t_{1,d}\} = \emptyset$. Otherwise, it outputs the error symbol $\perp$ (indicating that ct is invalid).
- **Dec**$(PP, sk_{f,\mathcal{P}_{f,\eta}}, ct) \to \mu$: On input the public parameters PP, a ciphertext ct associated with attribute vector x and tag vector $t = (t_1, ..., t_d)$, and the latest punctured secret key $sk_{f,\mathcal{P}_{f,\eta}}$ for policy f, this algorithm outputs the plaintext μ if $f(x) = 0 \wedge \mathcal{P}_{f,\eta} \cap \{t_1, ..., t_d\} = \emptyset$. Otherwise, it outputs the error symbol $\perp$.

Correctness. A single-hop, unidirectional PABPRE-ST scheme (**Setup**, **KeyGen**, **Enc**, **Puncture**, **ReEnc**, **Dec**) is correct if for all $\mu \in \mathcal{M}$, the following conditions hold:

1) For all initial secret keys $sk_{f,\emptyset}$ output by **KeyGen** for policy f, and for all $\mu \in \mathcal{M}$, if $f(x) = 0$, then:
$$\mathbf{Dec}(PP, sk_{f,\emptyset}, \mathbf{Enc}(PP, x, t_1, \mu)) = \mu.$$
2) For all punctured secret keys $sk_{f,\mathcal{P}_{f,\eta}}$ output by **Puncture** for policy f corresponding to a sequence of punctured tags $\mathcal{P}_{f,\eta} = \{\hat{t}_1, ..., \hat{t}_\eta\}$, and for all $\mu \in \mathcal{M}$, if $f(x) = 0 \wedge \mathcal{P}_{f,\eta} \cap \{t_1, ..., t_d\} = \emptyset$, then:
$$\mathbf{Dec}(PP, sk_{f,\mathcal{P}_{f,\eta}}, \mathbf{Enc}(PP, x, t_1, \mu)) = \mu.$$
3) For any re-encryption key $rk_{(f,t_1)\to(g,t_2)}$ output by **ReKeyGen**$(PP, sk_{f,\mathcal{P}_{f,\eta}}, f, g, t_1, t_2)$, and for any ciphertext $ct = \mathbf{Enc}(PP, x, t_1, \mu)$, if $f(x) = 0 \wedge \mathcal{P}_{f,\eta} \cap \{t_{1,1}, ..., t_{1,d}\} = \emptyset \wedge g(y) = 0$, then:
$$\mathbf{Dec}(PP, sk_{g,\emptyset}, \mathbf{ReEnc}(PP, rk_{(f,t_1)\to(g,t_2)}, ct)) = \mu.$$
4) For any re-encryption key $rk_{(f,t_1)\to(g,t_2)}$ output by **ReKeyGen**$(PP, sk_{f,\mathcal{P}_{f,\eta}}, f, g, t_1, t_2)$, and for any ciphertext $ct = \mathbf{Enc}(PP, x, t_1, \mu)$, if $f(x) = 0 \wedge \mathcal{P}_{f,\eta} \cap \{t_{1,1}, ..., t_{1,d}\} = \emptyset \wedge g(y) = 0 \wedge \mathcal{P}_{g,\eta} \cap \{t_{2,1}, ..., t_{2,d}\} = \emptyset$, then:
$$\mathbf{Dec}(PP, sk_{g,\mathcal{P}_{g,\eta}}, \mathbf{ReEnc}(PP, rk_{(f,t_1)\to(g,t_2)}, ct)) = \mu.$$

C. Security Model

We define a security model for single-hop unidirectional PABPRE-ST, denoted IND-sPun-HRA, in the selective setting. This model captures security against honest re-encryption attacks for selective attributes and selective tags.

Let $\mathcal{A}$ be a probabilistic polynomial time (PPT) adversary, and let Π = (**Setup**, **KeyGen**, **ReKeyGen**, **Enc**, **Puncture**, **ReEnc**, **Dec**) be a PABPRE-ST scheme with message space $\mathcal{M}$, attribute space $\mathcal{X}$, tag space $\mathcal{T}$, and ciphertext space $\mathcal{C}$. Let $\mathcal{F} = \{f : \mathbb{Z}_q^\ell \to \mathbb{Z}_q\}$ be a family of functions.

The security game is defined via the experiment $\mathbf{Exp}_{\Pi,\mathcal{A}}^{\text{IND-sPun-HRA}}(\lambda)$ between $\mathcal{A}$ and a challenger, as detailed below:

1) **Initialization:** The adversary $\mathcal{A}$ declares the challenge attribute vector x^* and the challenge tag set $t^* = \{t_1^*, ..., t_d^*\}$.

2) **Setup:** The challenger runs **Setup**$(1^\lambda, \ell, d)$ to obtain (PP, msk) and gives the public parameters PP to $\mathcal{A}$. The challenger also initializes the following state:
    - Maintains a table of tuples $\{f, sk_{f,\mathcal{P}_{f,\eta}}, P_f, C_f\}$ representing the key state for each policy f, where P_f and C_f are initially empty sets.
    - Initializes a counter numCt to 0.
    - Initializes a key-value store $\mathcal{H}$ as empty.
    - Initializes a set Derive as empty.

3) **Phase 1:** The adversary $\mathcal{A}$ can adaptively and polynomially many times query the following oracles:
    - $\mathcal{O}_{\mathsf{Puncture}}(f, \hat{t})$: On input a policy $f \in \mathcal{F}$ and a tag $\hat{t}$:
        - If a tuple $\{f, sk_{f,\mathcal{P}_{f,\eta}}, P_f, C_f\}$ exists, the challenger runs $sk_{f,\mathcal{P}_{f,\eta+1}} \leftarrow$ **Puncture**$(PP, sk_{f,\mathcal{P}_{f,\eta}}, \hat{t})$, adds $\hat{t}$ to P_f, and updates the tuple to $\{f, sk_{f,\mathcal{P}_{f,\eta+1}}, P_f, C_f\}$.
        - Otherwise, the challenger runs $sk_{f,\emptyset} \leftarrow$ **KeyGen**(PP, msk, f), then runs $sk_{f,\mathcal{P}_{f,1}} \leftarrow$ **Puncture**$(PP, sk_{f,\emptyset}, \hat{t})$, adds $\hat{t}$ to P_f, and creates a new tuple $\{f, sk_{f,\mathcal{P}_{f,1}}, P_f, C_f\}$.
    - $\mathcal{O}_{\mathsf{Corrupt}}(f)$: On the first query for a policy $f \in \mathcal{F}$:
        - **Case 1:** $f(x^*) \neq 0$
            a) If a tuple $\{f, sk_{f,\mathcal{P}_{f,\eta}}, P_f, C_f\}$ exists, the challenger gives $sk_{f,\mathcal{P}_{f,\eta}}$ to $\mathcal{A}$ and sets $C_f \leftarrow P_f$.
            b) Otherwise, the challenger runs $sk_{f,\emptyset} \leftarrow$ **KeyGen**(PP, msk, f), gives $sk_{f,\emptyset}$ to $\mathcal{A}$, creates a new tuple $\{f, sk_{f,\emptyset}, P_f, C_f\}$, and sets $C_f \leftarrow P_f$ $(= \emptyset)$.
        - **Case 2:** $f(x^*) = 0$
            a) If a tuple $\{f, sk_{f,\mathcal{P}_{f,\eta}}, P_f, C_f\}$ exists, the challenger checks if $P_f \cap \{t_1^*, ..., t_d^*\} = \emptyset$. If true, it outputs $\perp$; otherwise, it gives $sk_{f,\mathcal{P}_{f,\eta}}$ to $\mathcal{A}$ and sets $C_f \leftarrow P_f$.
            b) If no such tuple exists, the challenger outputs $\perp$ (indicating no tags have been punctured for this key).

        For all subsequent Corrupt queries for policy f, the challenger outputs $\perp$.
    - $\mathcal{O}_{\mathsf{ReKeyGen}}(f, g, t_1, t_2)$: On input two policy-tag pairs (f, t_1) and (g, t_2):
        a) If no tuple $\{f, sk_{f,\mathcal{P}_{f,\eta}}, P_f, C_f\}$ exists, generate $sk_{f,\emptyset} \leftarrow$ **KeyGen**(PP, msk, f) and initialize the tuple $\{f, sk_{f,\emptyset}, P_f, C_f\}$ with $P_f = \emptyset$, $C_f = \emptyset$.
        b) If $f(x^*) = 0 \wedge P_f \cap \{t_1^*, ..., t_d^*\} = \emptyset$, then:
            - If $g(x^*) \neq 0$ or $(g(x^*) = 0 \wedge t_2 \neq t^*)$, output $\perp$.
            - Otherwise (i.e., $g(x^*) = 0 \wedge t_2 = t^*$), output $rk_{(f,t_1)\rightarrow(g,t_2)} \leftarrow$ **ReKeyGen**$(PP, sk_{f,\mathcal{P}_{f,\eta}}, f, g, t_1, t_2)$.
        c) If $f(x^*) \neq 0$ or $(f(x^*) = 0 \wedge P_f \cap \{t_1^*, ..., t_d^*\} \neq \emptyset)$, output $rk_{(f,t_1)\rightarrow(g,t_2)} \leftarrow$ **ReKeyGen**$(PP, sk_{f,\mathcal{P}_{f,\eta}}, f, g, t_1, t_2)$.
    - $\mathcal{O}_{\mathsf{Enc}}(x, t, \mu)$: On input an attribute set x, a tag set t, and a message μ, output the ciphertext $ct \leftarrow$ **Enc**(PP, x, t, μ). Increment numCt and store ct in $\mathcal{H}$ with key (x, t, numCt).
    - $\mathcal{O}_{\mathsf{ReEnc}}(f, g, t_2, (x, t, k))$: On input two policies f, g and a tag vector t_2, and a handle (x, t, k) (where $t = \{t_1, ..., t_d\}$, $k \leq \mathsf{numCt}$):
        - If no value exists in $\mathcal{H}$ for key (x, t, k), output $\perp$. Otherwise, let ct be the corresponding value.
        - If no tuple $\{f, sk_{f,\mathcal{P}_{f,\eta}}, P_f, C_f\}$ exists, generate $sk_{f,\emptyset} \leftarrow$ **KeyGen**(PP, msk, f) and initialize the tuple $\{f, sk_{f,\emptyset}, P_f, C_f\}$ with $\mathcal{P}_f = \emptyset$, $\mathcal{C}_f = \emptyset$.
        - If $f(x) \neq 0$ or $(f(x) = 0 \wedge P_f \cap \{t_1, ..., t_d\} \neq \emptyset)$, output $\perp$ (indicating ct is invalid for this query). Otherwise, output the re-encrypted ciphertext $ct' \leftarrow$ **ReEnc**$(PP, rk_{(f,t_1)\rightarrow(g,t_2)}, ct)$.

4) **Challenge:** $\mathcal{A}$ submits two messages $\mu_0, \mu_1 \in \mathcal{M}$ to the challenger. The challenger randomly selects a bit $\beta \leftarrow \{0, 1\}$, and computes the challenge ciphertext $ct_\beta^* \leftarrow$ **Enc**$(PP, x^*, t^*, \mu_\beta)$. It increments numCt, adds numCt to the set Derive, and stores ct_β^* in $\mathcal{H}$ with key $(x^*, t^*, \mathsf{numCt})$.

5) **Phase 2:** After receiving the challenge ciphertext, $\mathcal{A}$ can continue querying the oracles $\mathcal{O}_{\mathsf{Puncture}}, \mathcal{O}_{\mathsf{Corrupt}}, \mathcal{O}_{\mathsf{ReKeyGen}}, \mathcal{O}_{\mathsf{Enc}}$, and $\mathcal{O}_{\mathsf{ReEnc}}$ with the same rules as in Phase 1, except the following restriction for $\mathcal{O}_{\mathsf{ReEnc}}$: If $k \in$ Derive and $(g(x^*) \neq 0$ or $(g(x^*) = 0 \wedge t_2 \neq t^*))$, then output $\perp$.

6) **Guess:** The adversary $\mathcal{A}$ outputs a guess β'. If $\beta' = \beta$, then $\mathcal{A}$ wins the game.

The advantage of the adversary $\mathcal{A}$ in the experiment $\mathsf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{IND\text{-}sPun\text{-}HRA}}(\lambda)$ is defined as $|\Pr[\beta' = \beta] - \frac{1}{2}|$.

Definition 1. A PABPRE-ST scheme is IND-sPun-HRA secure if every PPT adversary $\mathcal{A}$ achieves at most a negligible advantage in the experiment $\mathsf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{IND\text{-}sPun\text{-}HRA}}(\lambda)$.

IV. Preliminaries

In this section, we present the notation and background knowledge on lattices.

A. Notation

Let $\mathbb{R}$ denote the set of real numbers, $\mathbb{Z}$ the set of integers, and $\mathbb{Z}_q$ the ring of integers modulo q. We represent $\mathbb{Z}_q$ as integers in $(-q/2, q/2]$. We use lowercase boldface for column vectors (e.g., $\mathbf{a}$), and uppercase boldface for matrices (e.g., $\mathbf{A}$). The notation $(*, *)$ denotes the column concatenation of matrices or vectors, while $(*|*)$ denotes their row concatenation. A function $\mathsf{negl}(n)$ denotes a negligible function in n. Furthermore, we say a probability is overwhelming if it is $1 - \mathsf{negl}(n)$. The statistical distance between two distributions X and Y over a countable domain D is defined to be $\frac{1}{2}\sum_{d \in D} |X(d) - Y(d)|$. We say that two distributions are statistically close if their statistical distance is $\mathsf{negl}(n)$.

B. Background on Lattices

Definition 2. (q-ary Lattices) For integers $n, m, q \geq 2$,$\mathbf{u} \in \mathbb{Z}_q^n$, and a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, we define:

$$\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{A}\mathbf{y} = \mathbf{0} \pmod q\}$$
$$\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{A}\mathbf{y} = \mathbf{u} \pmod q\}$$

Note that $\Lambda_q^{\mathbf{u}}(\mathbf{A})$ is a shift (coset) of $\Lambda_q^{\perp}(\mathbf{A})$.

Lemma 1 [40]. Let q be a prime modulus, and n, m be positive integers. Define $k = \lceil \log q \rceil$. For any column vectors $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_q^n$, define the following algorithms:

1) $\mathsf{BD}(\mathbf{x}) = (\mathbf{u}_0, \mathbf{u}_1, ..., \mathbf{u}_{k-1}) \in \mathbb{Z}_q^{nk}$, where $\mathbf{x} = \sum_{j=0}^{k-1} 2^j \cdot \mathbf{u}_j$ and each $\mathbf{u}_j \in \{0,1\}^n$.
2) $\mathsf{P}_2(\mathbf{y}) = (\mathbf{y}, 2\mathbf{y}, ..., 2^{k-1}\mathbf{y}) \in \mathbb{Z}_q^{nk}$.

Then the following identities hold:

1) Vector Identity: $\mathsf{BD}(\mathbf{x})^{\top} \cdot \mathsf{P}_2(\mathbf{y}) = \mathbf{x}^{\top}\mathbf{y}$.
2) Matrix Identity: For any matrix $\mathbf{A} = (\mathbf{y}_1 \mid \mathbf{y}_2 \mid \cdots \mid \mathbf{y}_m) \in \mathbb{Z}_q^{n\times m}$, define
$$\mathsf{P}_2(\mathbf{A}) = (\mathsf{P}_2(\mathbf{y}_1) \mid \mathsf{P}_2(\mathbf{y}_2) \mid ... \mid \mathsf{P}_2(\mathbf{y}_m)) \in \mathbb{Z}_q^{nk\times m}.$$
Then, $\mathsf{BD}(\mathbf{x})^{\top} \cdot \mathsf{P}_2(\mathbf{A}) = \mathbf{x}^{\top}\mathbf{A}$.

Proof: By definition of BD and P_2, we have:

$$\begin{aligned}\mathsf{BD}(\mathbf{x})^{\top} \cdot \mathsf{P}_2(\mathbf{y}) &= \sum_{j=0}^{k-1} \mathbf{u}_j^{\top} \cdot \left(2^j \mathbf{y}\right) \\ &= \sum_{j=0}^{k-1} 2^j \left(\mathbf{u}_j^{\top}\mathbf{y}\right) = \left(\sum_{j=0}^{k-1} 2^j \mathbf{u}_j\right)^{\top} \mathbf{y} = \mathbf{x}^{\top}\mathbf{y}.\end{aligned}$$

Let $\mathbf{A} = (\mathbf{y}_1 \mid \mathbf{y}_2 \mid ... \mid \mathbf{y}_m)$. Then:

$$\begin{aligned}\mathsf{BD}(\mathbf{x})^{\top} \cdot \mathsf{P}_2(\mathbf{A}) &= \mathsf{BD}(\mathbf{x})^{\top} \cdot (\mathsf{P}_2(\mathbf{y}_1) \mid \mathsf{P}_2(\mathbf{y}_2) \mid ... \mid \mathsf{P}_2(\mathbf{y}_m)) \\ &= \big(\mathsf{BD}(\mathbf{x})^{\top} \cdot \mathsf{P}_2(\mathbf{y}_1) \mid \mathsf{BD}(\mathbf{x})^{\top} \cdot \mathsf{P}_2(\mathbf{y}_2) \mid ... \\ &\quad \mid \mathsf{BD}(\mathbf{x})^{\top} \cdot \mathsf{P}_2(\mathbf{y}_m)\big) \\ &= (\mathbf{x}^{\top}\mathbf{y}_1 \mid \mathbf{x}^{\top}\mathbf{y}_2 \mid ... \mid \mathbf{x}^{\top}\mathbf{y}_m) = \mathbf{x}^{\top}\mathbf{A}.\end{aligned}$$

Lemma 2. (Matrix Norms [7]) For a vector $\mathbf{u}$ we let $\|\mathbf{u}\|$ denote its ℓ_2 norm. For a matrix $\mathbf{R} \in \mathbb{Z}^{k\times m}$, let $\tilde{\mathbf{R}}$ be the result of applying Gram-Schmidt (GS) orthogonalization to the columns of $\mathbf{R}$. We define three matrix norms:

- $\|\mathbf{R}\|$ denotes the ℓ_2 length of the longest column of $\mathbf{R}$.
- $\|\mathbf{R}\|_{\mathsf{GS}} = \|\tilde{\mathbf{R}}\|$, where $\tilde{\mathbf{R}}$ is the GS orthogonalization of $\mathbf{R}$.
- $\|\mathbf{R}\|_2$ is the operator norm of $\mathbf{R}$ defined as $\|\mathbf{R}\|_2 = \sup_{\|\mathbf{x}\|=1} \|\mathbf{R}\mathbf{x}\|$.

Note that $\|\mathbf{R}\|_{\mathsf{GS}} \le \|\mathbf{R}\| \le \|\mathbf{R}\|_2 \le \sqrt{k}\|\mathbf{R}\|$ and that $\|\mathbf{R} \cdot \mathbf{S}\|_2 \le \|\mathbf{R}\|_2 \cdot \|\mathbf{S}\|_2$.

Lemma 3. (Discrete Gaussians [43]) Let $\Lambda \subseteq \mathbb{Z}^m$ be a lattice. For a vector $\mathbf{c} \in \mathbb{R}^m$ and a parameter $\sigma \in \mathbb{R}$, define:

$$\rho_{\mathbf{c},\sigma}(\mathbf{x}) = \exp\left(-\pi \frac{\|\mathbf{x} - \mathbf{c}\|^2}{\sigma^2}\right)$$
$$\rho_{\mathbf{c},\sigma}(\Lambda) = \sum_{\mathbf{x}\in\Lambda} \rho_{\mathbf{c},\sigma}(\mathbf{x}).$$

The discrete Gaussian distribution over Λ with center $\mathbf{c}$ and parameter σ is

$$\mathcal{D}_{\mathbf{c},\sigma}(\Lambda)(\mathbf{y}) = \frac{\rho_{\mathbf{c},\sigma}(\mathbf{y})}{\rho_{\mathbf{c},\sigma}(\Lambda)}, \quad \forall \mathbf{y} \in \Lambda.$$

Lemma 4 [7]. For integers $n, m, k, q, \sigma > 0$, matrices $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{U} \in \mathbb{Z}_q^{n\times k}$, if $\mathbf{R} \in \mathbb{Z}^{m\times k}$ is sampled from $\mathcal{D}_{\sigma}(\Lambda_q^{\mathbf{U}}(\mathbf{A}))$ and $\mathbf{S}$ is sampled uniformly in $\{\pm 1\}^{m\times m}$ then

$$\|\mathbf{R}^{\top}\|_2 \le \sigma\sqrt{mk}, \quad \|\mathbf{R}\|_2 \le \sigma\sqrt{mk}, \quad \|\mathbf{S}\|_2 \le 20\sqrt{m}$$

with overwhelming probability in m.

Lemma 5. (Learning with Errors (LWE) [44]) Fix integers n, m, a prime integer q and a noise distribution χ over $\mathbb{Z}$. The (n, m, q, χ)-LWE problem is to distinguish the following two distributions:

$$(\mathbf{A}, \mathbf{A}^{\top}\mathbf{s} + \mathbf{e}) \quad \text{and} \quad (\mathbf{A}, \mathbf{u})$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \chi^m$, $\mathbf{u} \leftarrow \mathbb{Z}_q^m$ are independently sampled. Throughout the paper we always set $m = \Theta(n \log q)$ and simply refer to the $\mathsf{DLWE}_{n,m,q,\chi}$ problem.

We say that a noise distribution χ is B-bounded if its support is in $[-B, B]$. For any fixed $d > 0$ and sufficiently large q, Regev [44] (through a quantum reduction) and Peikert [45] (through a classical reduction) show that taking χ as a certain q/n^d-bounded distribution, the (n, q, χ)-LWE problem is as hard as approximating the worst-case GapSVP to $n^{O(d)}$ factors, which is believed to be intractable. More generally, let $\chi_{\max} < q$ be the bound on the noise distribution. The difficulty of the LWE problem is measured by the ratio $q/\chi_{\max}$. This ratio is always bigger than 1 and the smaller it is the harder the problem. The problem appears to remain hard even when $q/\chi_{\max} < 2^{n^{\epsilon}}$ for some fixed $\epsilon \in (0, 1/2)$.

Lemma 6. Let $n, m, q > 0$ be integers with q prime. There are polynomial time algorithms with the properties below:

- $\mathsf{TrapGen}(1^n, 1^m, q) \to (\mathbf{A}, \mathbf{T_A})$ ([43],[46]): a randomized algorithm that, when $m = \Theta(n \log q)$, outputs a full-rank matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and basis $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$ for $\Lambda_q^{\perp}(\mathbf{A})$ such that $\mathbf{A}$ is $\mathsf{negl}(n)$-close to uniform and $\|\mathbf{T_A}\|_{\mathsf{GS}} = O(\sqrt{n \log q})$, with all but negligible probability in n.
- $\mathsf{ExtendRight}(\mathbf{A}, \mathbf{T_A}, \mathbf{B}) \to \mathbf{T}_{(\mathbf{A}|\mathbf{B})}$ ([47]): a deterministic algorithm that given full-rank matrices $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_q^{n\times m}$ and a basis $\mathbf{T_A}$ of $\Lambda_q^{\perp}(\mathbf{A})$ outputs a basis $\mathbf{T}_{(\mathbf{A}|\mathbf{B})}$ of $\Lambda_q^{\perp}(\mathbf{A} \mid \mathbf{B})$ such that $\|\mathbf{T_A}\|_{\mathsf{GS}} = \|\mathbf{T}_{(\mathbf{A}|\mathbf{B})}\|_{\mathsf{GS}}$.
- $\mathsf{ExtendLeft}(\mathbf{A}, \mathbf{G}, \mathbf{T_G}, \mathbf{S}) \to \mathbf{T_H}$ where $\mathbf{H} = (\mathbf{A} \mid \mathbf{G} + \mathbf{A}\mathbf{S})$ ([48]): a deterministic algorithm that given full-rank matrices $\mathbf{A}, \mathbf{G} \in \mathbb{Z}_q^{n\times m}$ and a basis $\mathbf{T_G}$ of $\Lambda_q^{\perp}(\mathbf{G})$ outputs a basis $\mathbf{T_H}$ of $\Lambda_q^{\perp}(\mathbf{H})$ such that $\|\mathbf{T_H}\|_{\mathsf{GS}} \le \|\mathbf{T_G}\|_{\mathsf{GS}} \cdot (1 + \|\mathbf{S}\|_2)$.

Lemma 7. (Gadget Matrix [46]) For positive integers $n, q \in \mathbb{N}$, $k = \lceil \log q \rceil$, let $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}^{\top} \in \mathbb{Z}_q^{n\times nk}$ be the gadget matrix where $\mathbf{I}_n$ is the identity matrix of dimension n, $\mathbf{g}^{\top} = (1, 2, ..., 2^{\lceil \log q \rceil - 1}) \in \mathbb{Z}_q^{1\times k}$. The lattice $\Lambda^{\perp}(\mathbf{G})$ has a known basis $\mathbf{T_G} \in \mathbb{Z}^{nk\times nk}$ with $\|\mathbf{T_G}\|_{\mathsf{GS}} \le \sqrt{5}$. We write $\mathbf{G}^{-1} : \mathbb{Z}_q^n \to \{0,1\}^{nk}$ to denote the operator that expands each component of the input into its binary decomposition (i.e., $\mathbf{G}\cdot\mathbf{G}^{-1}(\mathbf{x}) = \mathbf{x}$ for all $\mathbf{x}\in\mathbb{Z}_q^n$). We extend $\mathbf{G}^{-1}(\cdot)$ to operate on matrices in a column-wise manner. For $m > nk$, we overload $\mathbf{G}$ to denote the padded gadget matrix $\mathbf{G} = (\mathbf{I}_n \otimes \mathbf{g}^\top | \mathbf{0}_{n\times(m-nk)})$. We define $\mathbf{G}^{-1}$ analogously (i.e., by padding the output with row-wise zeroes).

Lemma 8. Let $\mathbf{A}\in\mathbb{Z}_q^{n\times m}$ and $\mathbf{T_A}\in\mathbb{Z}^{m\times m}$ be a basis for $\Lambda_q^{\perp}(\mathbf{A})$. Let $\mathbf{U}\in\mathbb{Z}_q^{n\times k}$. There are polynomial time algorithms that output $\mathbf{X}\in\mathbb{Z}^{m\times k}$ satisfying $\mathbf{AX}=\mathbf{U}$ with the properties below:

- $\mathsf{SampleD}(\mathbf{A},\mathbf{T_A},\mathbf{U},\sigma)\to\mathbf{X}$ ([43]): a randomized algorithm that, when $\sigma = \|\mathbf{T_A}\|_{\mathsf{GS}}\cdot\omega(\sqrt{\log m})$, outputs a random sample $\mathbf{X}$ from a distribution that is statistically close to $\mathcal{D}_\sigma(\Lambda_q^{\mathbf{U}}(\mathbf{A}))$.
- $\mathsf{RandBasis}(\mathbf{A},\mathbf{T_A},\sigma)\to\mathbf{T}'_{\mathbf{A}}$ ([47]): a randomized algorithm that, when $\sigma = \|\mathbf{T_A}\|_{\mathsf{GS}}\cdot\omega(\sqrt{\log m})$, outputs a basis $\mathbf{T}'_{\mathbf{A}}$ of $\Lambda_q^{\perp}(\mathbf{A})$ sampled from a distribution that is statistically close to $(\mathcal{D}_\sigma(\Lambda_q^{\perp}(\mathbf{A})))^m$. Note that $\|\mathbf{T}'_{\mathbf{A}}\|_{\mathsf{GS}} < \sigma\sqrt{m}$ with all but negligible probability.

Lemma 9 [48]. Let $m > (n+1)\log_2 q + \omega(\log n)$ and $q>2$ is prime. Let $\mathbf{R}$ be an $m\times k$ matrix chosen uniformly in $\{+1,-1\}^{m\times k} \mod q$ where $k = k(n)$ is polynomial in n. Let $\mathbf{A}$ and $\mathbf{U}$ be matrices chosen uniformly in $\mathbb{Z}_q^{n\times m}$ and $\mathbb{Z}_q^{n\times k}$ respectively. Then, for all vectors $\mathbf{e}$ in $\mathbb{Z}_q^m$, the distribution $(\mathbf{A},\mathbf{AR},\mathbf{R}^\top\mathbf{e})$ is statistically close to the distribution $(\mathbf{A},\mathbf{U},\mathbf{R}^\top\mathbf{e})$.

Lemma 10. (Homomorphic Encoding [7]) For integers n and $q=q(n)$ let $m=\Theta(n\log q)$. Let $\mathbf{G}\in\mathbb{Z}_q^{n\times m}$ be the fixed matrix from Lemma 2.4 (part 4). For $x\in\mathbb{Z}_q$, $\mathbf{B}\in\mathbb{Z}_q^{n\times m}$, s $\in\mathbb{Z}_q^n$, and $\delta>0$ define the set

$$E_{\mathbf{s},\delta}(x,\mathbf{B}) = \{(x\mathbf{G}+\mathbf{B})^\top\mathbf{s}+\mathbf{e},\ \text{where } \|\mathbf{e}\|<\delta\}.$$

There are three efficient deterministic algorithms $\mathsf{Eval}_{\mathsf{pk}}$, $\mathsf{Eval}_{\mathsf{ct}}$, $\mathsf{Eval}_{\mathsf{sim}}$ that satisfy the following properties with respect to some family of functions $\mathcal{F}=\{f:\mathbb{Z}_q^\ell\to\mathbb{Z}_q\}$ and a function $\alpha_{\mathcal{F}}:\mathbb{Z}\to\mathbb{Z}$.

- $\mathsf{Eval}_{\mathsf{pk}}(f\in\mathcal{F},\overline{\mathbf{B}}\in(\mathbb{Z}_q^{n\times m})^\ell)\to\mathbf{B}_f\in\mathbb{Z}_q^{n\times m}$.
- $\mathsf{Eval}_{\mathsf{ct}}(f\in\mathcal{F},((x_i,\mathbf{B}_i,\mathbf{c}_i))_{i=1}^\ell)\to\mathbf{c}_f\in\mathbb{Z}_q^m$. Here $x_i\in\mathbb{Z}_q,\mathbf{B}_i\in\mathbb{Z}_q^{n\times m}$ and $\mathbf{c}_i\in E_{\mathbf{s},\delta}(x_i,\mathbf{B}_i)$ for some $\mathbf{s}\in\mathbb{Z}_q^n$ and $\delta>0$. Note that the same $\mathbf{s}$ is used for all $\mathbf{c}_i$. The output $\mathbf{c}_f$ must satisfy $\mathbf{c}_f\in E_{\mathbf{s},\Delta}(f(\mathbf{x}),\mathbf{B}_f)$ where $\mathbf{B}_f=\mathsf{Eval}_{\mathsf{pk}}(f,(\mathbf{B}_1,...,\mathbf{B}_\ell))$ and $\mathbf{x}=(x_1,...,x_\ell)$. We further require that $\Delta<\delta\cdot\alpha_{\mathcal{F}}(n)$ for some function $\alpha_{\mathcal{F}}(n)$ that measures the increase in the noise magnitude in $\mathbf{c}_f$ compared to the input ciphertexts.
- $\mathsf{Eval}_{\mathsf{sim}}(f\in\mathcal{F},((x_i^*,\mathbf{S}_i))_{i=1}^\ell,\mathbf{A})\to\mathbf{S}_f\in\mathbb{Z}_q^{m\times m}$. Here $x_i^*\in\mathbb{Z}_q$ and $\mathbf{S}_i\in\mathbb{Z}_q^{m\times m}$. With $\mathbf{x}^*=(x_1^*,...,x_\ell^*)$, the output $\mathbf{S}_f$ satisfies $\mathbf{AS}_f-f(\mathbf{x}^*)\mathbf{G}=\mathbf{B}_f$ where $\mathbf{B}_f=\mathsf{Eval}_{\mathsf{pk}}(f,(\mathbf{AS}_1-x_1^*\mathbf{G},...,\mathbf{AS}_\ell-x_\ell^*\mathbf{G}))$. For all $f\in\mathcal{F}$, if $\mathbf{S}_1,...,\mathbf{S}_\ell$ are random matrices in $\{\pm1\}^{m\times m}$ then $\|\mathbf{S}_f\|_2<\alpha_{\mathcal{F}}(n)$ with all but negligible probability.

V. Puncturable Attribute-Based Proxy Re-Encryption with Switchable Tags from LWE

In this section, we present the concrete construction of the PABPRE-ST scheme and prove its correctness and security.

A. Construction

In this section, we present our construction of PABPPRE-ST. We set the parameters as the following:

- $\mathbf{G}\in\mathbb{Z}_q^{n\times m}$ is a gadget matrix for integer n, large enough prime power $q=\mathrm{poly}(n)$, and $m=\Theta(n\log q)$. Let $k=\lceil\log q\rceil$.
- Let d be the maximum number of tags per ciphertext.
- Consider the message space is $\mathcal{M}=\{0,1\}^m$, the attribute space is $\mathcal{X}=\mathbb{Z}_q^\ell$, and the tag space is $\mathcal{T}=\mathbb{Z}_q^d$.
- Let χ be a $\chi_{\max}$-bounded distribution for which $\mathsf{DLWE}_{n,m,q,\chi}$ is hard.
- For the correctness and security requirements of the scheme, the Gaussian parameter is set to $\sigma_0=\omega(\alpha_{\mathcal{F}}\cdot\sqrt{\log m})$.
- Let $\mathcal{F}=\{f:\mathbb{Z}_q^\ell\to\mathbb{Z}_q\}$ be the family of functions.
- Let $\mathcal{G}=\{g_{\hat{t}}:\mathbb{Z}_q^d\to\mathbb{Z}_q,\forall\hat{t}\in\mathbb{Z}_q\}$ be the family of functions, where $g_{\hat{t}}(t)\neq 0\bmod q$ if $\hat{t}\in\{t_1,\cdots,t_d\}$, $t=(t_1,\cdots,t_d)$, otherwise $g_{\hat{t}}(t)=0\bmod q$.

The proposed PABPRE-ST consists of the following algorithms:

Setup$(1^\lambda,\ell,d)$**:** On input a security parameter λ, ℓ and d, do as follows:

1) Generate $(\mathbf{A}_0,\mathbf{T}_{\mathbf{A}_0})\leftarrow\mathsf{TrapGen}(1^n,1^m,q)$, where $\mathbf{A}_0\in\mathbb{Z}_q^{n\times m}$, and $\mathbf{T}_{\mathbf{A}_0}\subseteq\Lambda_q^{\perp}(\mathbf{A}_0)$.
2) Choose $\ell+d+1$ uniformly random matrices $\mathbf{A}_1,\cdots,\mathbf{A}_\ell,\mathbf{B}_1,\cdots,\mathbf{B}_d,\mathbf{U}\in\mathbb{Z}_q^{n\times m}$.
3) Output the public parameter $PP=\{\mathbf{A}_0,\mathbf{A}_1,\cdots,\mathbf{A}_\ell,\mathbf{B}_1,\cdots,\mathbf{B}_d,\mathbf{U}\}$ and the master secret key $msk=\{\mathbf{T}_{\mathbf{A}_0}\}$.

KeyGen$(PP,msk,f\in\mathcal{F})$**:** On input the public parameter PP, master secret key msk, and a policy $f\in\mathcal{F}$, do as follows:

1) Evaluate $\mathbf{A}_f\leftarrow\mathsf{Eval}_{\mathsf{pk}}^{\mathcal{F}}((\mathbf{A}_i)_{i=1}^\ell,f)$.
2) Compute $\mathbf{T}_f^{ER}\leftarrow\mathsf{ExtendRight}(\mathbf{A}_0,\mathbf{A}_f,\mathbf{T}_{\mathbf{A}_0})$.
3) Compute $\mathbf{T}_f\leftarrow\mathsf{RandBasis}((\mathbf{A}_0\mid\mathbf{A}_f),\mathbf{T}_f^{ER},\sigma_0)$.
4) Output the secret key $sk_{f,\emptyset}=\mathbf{T}_f\in\mathbb{Z}_q^{2m\times m}$ for the policy f.

Enc$(PP,x\in\mathbb{Z}_q^\ell,t_1\in\mathbb{Z}_q^d,\mu\in\{0,1\}^m)$**:** On input the public parameter PP, the set of attributes $x=(x_1,\cdots,x_\ell)\in\mathbb{Z}_q^\ell$, the set of tags $t_1=(t_{1,1},\cdots,t_{1,d})\in\mathbb{Z}_q^d$, and message $\mu\in\{0,1\}^m$, do as follows:

1) Choose a uniformly random vector $\mathbf{s}\leftarrow\mathbb{Z}_q^n$.
2) Choose ℓ uniformly random matrices $\mathbf{S}_i\leftarrow\{+1,-1\}^{m\times m}$ for $i\in\{1,\cdots,\ell\}$.
3) Choose d uniformly random matrices $\mathbf{R}_j\leftarrow\{+1,-1\}^{m\times m}$ for $j\in\{1,\cdots,d\}$.
4) Choose error vectors $\mathbf{e}_0,\mathbf{e}_{out}\in\chi^m$.
5) Set

$$\begin{aligned}\mathbf{H}_{x,t_1}=(\mathbf{A}_0\mid x_1\mathbf{G}+\mathbf{A}_1\mid\cdots\mid x_\ell\mathbf{G}+\mathbf{A}_\ell\\ \mid t_{1,1}\mathbf{G}+\mathbf{B}_1\mid\cdots\mid t_{1,d}\mathbf{G}+\mathbf{B}_d)\\ \in\mathbb{Z}_q^{n\times(\ell+d+1)m}.\end{aligned}$$

6) Set $\mathbf{e}=(\mathbf{I}_m\mid\mathbf{S}_1\mid\cdots\mid\mathbf{S}_\ell\mid\mathbf{R}_1\mid\cdots\mid\mathbf{R}_d)^\top\cdot\mathbf{e}_0=(\mathbf{e}_{in},\mathbf{e}_1,\cdots,\mathbf{e}_\ell,\bar{\mathbf{e}}_1,\cdots,\bar{\mathbf{e}}_d)\in\mathbb{Z}_q^{(\ell+d+1)m}$.
7) Compute $c=\mathbf{H}_{x,t_1}^\top\mathbf{s}+\mathbf{e}\in\mathbb{Z}_q^{(\ell+d+1)m}$ and $c_{out}=\mathbf{U}^\top\mathbf{s}+\mathbf{e}_{out}+\lfloor q/2\rfloor\cdot\mu\in\mathbb{Z}_q^m$. Here,

$c = (c_{in}, c_1, \cdots, c_\ell, \bar{c}_1, \cdots, \bar{c}_d) \in \mathbb{Z}_q^{(\ell+d+1)m}$, where $c_{in} = \mathbf{A}_0^\top \mathbf{s} + \mathbf{e}_{in}$, $c_i = (x_i \mathbf{G} + \mathbf{A}_i)^\top \mathbf{s} + \mathbf{e}_i$ for all $i \in \{1, \cdots, \ell\}$, and $\bar{c}_j = (t_{1,j}\mathbf{G} + \mathbf{B}_j)^\top \mathbf{s} + \bar{\mathbf{e}}_j$ for all $j \in \{1, \cdots, d\}$.

8) Output the ciphertext $ct = (c_{in}, c_1, \cdots, c_\ell, \bar{c}_1, ..., \bar{c}_d, c_{out}) \in \mathbb{Z}_q^{(\ell+d+2)m}$ with the tag vector t_1 and the attribute vector x.

Puncture$(PP, sk_{f,\mathcal{P}_{f,\eta-1}}, \hat{t}_\eta)$**:** On input the public parameters PP, a punctured secret key $sk_{f,\mathcal{P}_{\eta-1}}$, and a tag $\hat{t}_\eta \in \mathbb{Z}_q$, do as follows:

1) Evaluate $\mathbf{B}_{g_{\hat{t}_\eta}} \leftarrow \mathsf{Eval}_{\mathsf{pk}}^{\mathcal{G}}((\mathbf{B}_i)_{i=1}^d, g_{\hat{t}_\eta})$.
2) Compute $\mathbf{T}_{f,\mathcal{P}_{f,\eta}}^{ER} \leftarrow \mathsf{ExtendRight}((\mathbf{A}_0 \mid \mathbf{A}_f \mid \mathbf{B}_{g_{\hat{t}_1}} \mid \cdots \mid \mathbf{B}_{g_{\hat{t}_{\eta-1}}}), \mathbf{B}_{g_{\hat{t}_\eta}}, \mathbf{T}_{f,\mathcal{P}_{f,\eta-1}})$.
3) Compute $\mathbf{T}_{f,\mathcal{P}_{f,\eta}} \leftarrow \mathsf{RandBasis}((\mathbf{A}_0 \mid \mathbf{A}_f \mid \mathbf{B}_{g_{\hat{t}_1}} \mid \cdots \mid \mathbf{B}_{g_{\hat{t}_\eta}}), \mathbf{T}_{f,\mathcal{P}_{f,\eta}}^{ER}, \sigma_\eta)$, where $\sigma_\eta = (\sqrt{m \log m})^\eta$.
   Here, $\mathcal{P}_{f,\eta-1} = \{\hat{t}_1, \hat{t}_2, \cdots, \hat{t}_{\eta-1}\}$ and $\mathcal{P}_{f,\eta} = \{\hat{t}_1, \hat{t}_2, \cdots, \hat{t}_{\eta-1}, \hat{t}_\eta\}$.
4) Output the new punctured secret key $sk_{f,\mathcal{P}_{f,\eta}} = \mathbf{T}_{f,\mathcal{P}_{f,\eta}} \in \mathbb{Z}_q^{(\eta+2)m \times (\eta+2)m}$ for the policy f.

Dec$(PP, sk_{f,\mathcal{P}_{f,\eta}}, (ct, x, t))$**:** On input the public parameter PP, the secret key $sk_{f,\mathcal{P}_{f,\eta}}$, and a ciphertext ct, do as follows:

1) If $f(x) \neq 0$, output $\perp$.
2) For $t = (t_1, t_2, \cdots, t_d)$, if there exist some $j \in \{1, \cdots, \eta\}$ such that $g_{\hat{t}_j}(t) \neq 0$, output $\perp$.
3) Otherwise, do as follows:
   - Sample $\mathbf{R}_{f,\eta} \leftarrow \mathsf{SampleD}((\mathbf{A}_0 \mid \mathbf{A}_f \mid \mathbf{B}_{g_{\hat{t}_1}} \mid \cdots \mid \mathbf{B}_{g_{\hat{t}_\eta}}), \mathbf{T}_{f,\mathcal{P}_{f,\eta}}, \mathbf{U}, \sigma_\eta)$.
   - Parse ct as $(c_{in}, c_1, \cdots, c_\ell, \bar{c}_1, ..., \bar{c}_d, c_{out})$, evaluate $c_f \leftarrow \mathsf{Eval}_{\mathsf{ct}}^{\mathcal{F}}((x_i, \mathbf{A}_i, c_i)_{i=1}^\ell, f)$, and evaluate $c_{g_{\hat{t}_j}} \leftarrow \mathsf{Eval}_{\mathsf{ct}}^{\mathcal{G}}((t_j, \mathbf{B}_j, \bar{c}_j)_{j=1}^d, g_{\hat{t}_j})$ for all $j \in \{1, \cdots, \eta\}$. Let $c'_f = \{c_{in}, c_f, c_{g_{\hat{t}_1}}, ..., c_{g_{\hat{t}_\eta}}\} \in \mathbb{Z}_q^{(2+\eta)m}$.
   - Compute $\mu = (\mu_1, \cdots, \mu_m) = c_{out} - \mathbf{R}_{f,\eta}^\top \mathbf{c}'_f \pmod q$. For each i, if $|\mu_i| < q/4$, take $\mu_i = 0$, otherwise take $\mu_i = 1$.
   - Output $\mu = (\mu_1, \cdots, \mu_m)$.

ReKeyGen$(PP, sk_{f,\mathcal{P}_{f,\eta}}, f, g, t_1, t_2)$**:** On input the public parameter PP, two policy-tag pairs $(f, t_1), (g, t_2)$, and the punctured secret key $sk_{f,\mathcal{P}_{f,\eta}}$ for the policy f (where $\mathcal{P}_\eta$ is the set of punctured tags), do as follows:

1) Sample $\mathbf{R}_{f,\eta} \leftarrow \mathsf{SampleD}((\mathbf{A}_0 \mid \mathbf{A}_f \mid \mathbf{B}_{g_{\hat{t}_1}} \mid \cdots \mid \mathbf{B}_{g_{\hat{t}_\eta}}), \mathbf{T}_{f,\mathcal{P}_{f,\eta}}, \mathbf{U}, \sigma_\eta)$.
2) Select an attribute vector $y = (y_1, \cdots, y_\ell)$ such that $g(y) = 0$. Let $t_2 = (t_{2,1}, ..., t_{2,d})$.
3) Construct $\mathbf{H}_{y,t_2} = (\mathbf{A}_0 \mid y_1\mathbf{G} + \mathbf{A}_1 \mid \cdots \mid y_\ell \mathbf{G} + \mathbf{A}_\ell \mid t_{2,1}\mathbf{G} + \mathbf{B}_1 \mid \cdots \mid t_{2,d}\mathbf{G} + \mathbf{B}_d) \in \mathbb{Z}_q^{n \times (\ell+d+1)m}$.
4) Choose a uniformly random matrix $\overline{\mathbf{R}}_1 \leftarrow \mathbb{Z}_q^{(2+\eta)mk \times n}$.
5) Choose $\overline{\mathbf{R}}_2 \leftarrow \chi^{(2+\eta)mk \times (\ell+d+1)m}$ and $\overline{\mathbf{R}}_3 \leftarrow \chi^{(2+\eta)mk \times m}$.
6) Construct the unidirectional re-encryption key

$$\begin{aligned} & rk_{(f,t_1)\to(g,t_2)} \\ & = \begin{bmatrix} \overline{\mathbf{R}}_1 \mathbf{H}_{y,t_2} + \overline{\mathbf{R}}_2 & \overline{\mathbf{R}}_1 \mathbf{U} + \overline{\mathbf{R}}_3 - \mathsf{P}_2(\mathbf{R}_{f,\eta}) \\ \mathbf{0}_{m \times (\ell+d+1)m} & \mathbf{I}_{m \times m} \end{bmatrix} \\ & \in \mathbb{Z}_q^{((2+\eta)mk+m) \times (\ell+d+2)m}. \end{aligned}$$

7) Output $rk_{(f,t_1)\to(g,t_2)}$ along with the attribute vector y and the tag vector t_2.

ReEnc$(PP, rk_{(f,t_1)\to(g,t_2)}, (ct, x, t_1))$**:** On input the public parameter, the re-encryption key $rk_{(f,t_1)\to(g,t_2)}$, and ciphertext ct, do as follows:

1) If $f(x) \neq 0$, output $\perp$.
2) For $t_1 = (t_{1,1}, t_{1,2}, \cdots, t_{1,d})$, if there exist some $j \in \{1, \cdots, \eta\}$ such that $g_{\hat{t}_j}(t_1) \neq 0$, output $\perp$.
3) Otherwise, do as follows:
   - Parse ct as $(c_{in}, c_1, \cdots, c_\ell, \bar{c}_1, ..., \bar{c}_d, c_{out})$, evaluate $c_f \leftarrow \mathsf{Eval}_{\mathsf{ct}}^{\mathcal{F}}((x_i, \mathbf{A}_i, c_i)_{i=1}^\ell, f)$, and evaluate $\mathbf{c}_{g_{\hat{t}_j}} \leftarrow \mathsf{Eval}_{\mathsf{ct}}^{\mathcal{G}}((t_{1,j}, \mathbf{B}_j, \bar{c}_j)_{j=1}^d, g_{\hat{t}_j})$ for all $j \in \{1, \cdots, \eta\}$.
   - Compute the re-encrypted ciphertext $ct_{(f,t_1)\to(g,t_2)} = (c'_{in}, c'_1, \cdots, c'_\ell, \bar{c}'_1, ..., \bar{c}'_d, c'_{out})$ as follows:

$$\begin{aligned} ct_{(f,t_1)\to(g,t_2)}^\top = & (\mathsf{BD}(c_{in}, c_f, c_{g_{\hat{t}_1}}, ..., c_{g_{\hat{t}_\eta}})^\top \mid c_{out}^\top) \\ & \cdot rk_{(f,t_1)\to(g,t_2)} \in \mathbb{Z}_q^{1 \times (\ell+d+2)m}. \end{aligned}$$

   - Output the re-encrypted ciphertext $ct_{(f,t_1)\to(g,t_2)}$ with the tag vector t_2 and the attribute vector y.

B. Correctness

Correctness is divided into two cases: one pertains to the original ciphertext, and the other to the re-encrypted ciphertext.

For original ciphertext: When $f(x) = 0 \wedge g_{\hat{t}_1}(t_1) = 0 \wedge ... \wedge g_{\hat{t}_\eta}(t_1) = 0$ we know by the requirement on Eval_{ct} that the resulting ciphertexts $c_f \in E_{s,\Delta_1}(0, \mathbf{A}_f)$, $\bar{c}_{g_{\hat{t}_j}} \in E_{s,\Delta_2}(0, \mathbf{B}_{g_{\hat{t}_j}})$ for $\forall j \in [\eta]$. Consequently,

$$\begin{aligned} c'_f &= (c_{in}, c_f, \bar{c}_{g_{\hat{t}_1}}, ..., \bar{c}_{g_{\hat{t}_\eta}}) \\ &= (\mathbf{A}_0 \mid \mathbf{A}_f \mid \mathbf{B}_{g_{\hat{t}_1}} \mid \cdots \mid \mathbf{B}_{g_{\hat{t}_\eta}})^\top \mathbf{s} + \mathbf{e}' \end{aligned}$$

where $\mathbf{e}' = (\mathbf{e}_0, \mathbf{e}_f, \mathbf{e}_{g_{\hat{t}_1}}, \cdots, \mathbf{e}_{g_{\hat{t}_\eta}})$ and $\| \mathbf{e}' \| < \Delta_1 + \eta\Delta_2 + \chi_{\max} < (\alpha_{\mathcal{F}} + \eta\alpha_{\mathcal{G}} + 1)\chi_{\max} < (\eta\alpha + 2)\chi_{\max}$. Where $\alpha = \max\{\alpha_{\mathcal{F}}, \alpha_{\mathcal{G}}\}$.

We know that $(\mathbf{A}_0 \mid \mathbf{A}_f \mid \mathbf{B}_{g_{\hat{t}_1}} \mid \cdots \mid \mathbf{B}_{\hat{g}_{t_\eta}}) \cdot \mathbf{R}_{f,\eta} = \mathbf{U} \pmod q$ and $\| \mathbf{R}_{f,\eta}^\top \|_2 < (\eta + 2)m\sigma_\eta$ with overwhelming probability by Lemma 4. Therefore

$$\begin{aligned} \mathbf{c}_{out} - \mathbf{R}_{f,\eta}^\top \mathbf{c}'_f &= \left(\mathbf{U}^\top \mathbf{s} + \left\lfloor \frac{q}{2} \right\rfloor \cdot \mu + \mathbf{e}_{out}\right) \\ &\quad - \left(\mathbf{U}^\top \mathbf{s} + \mathbf{R}_{f,\eta}^\top \mathbf{e}'\right) \pmod q \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot \mu + \mathbf{e}_{out} - \mathbf{R}_{f,\eta}^\top \mathbf{e}' \pmod q. \end{aligned}$$

Finally,

$$\begin{aligned} \| \mathbf{e}_{out} - \mathbf{R}_{f,\eta}^\top \mathbf{e}' \| &\leq \chi_{\max} + (\eta+2)m\sigma_\eta \cdot (\eta\alpha + 2)\chi_{\max} \\ &\leq (\eta+3)^2\alpha^2 \cdot \chi_{\max} \cdot m^{\eta/2+1} \end{aligned}$$

with overwhelming probability. By choosing appropriate parameters such that $(\eta + 3)^2\alpha^2 \cdot \chi_{\max} \cdot m^{\eta/2+1} < q/4$, the message $\mu \in \{0, 1\}^m$ can be recovered.

For re-encrypted ciphertext:

$$\begin{aligned}
&ct^\top_{(f,t_1)\to(g,t_2)}\\
&= (\mathsf{BD}(c_{in}, c_f, \bar{c}_{g_{\hat{t}_1}}, ..., \bar{c}_{g_{\hat{t}_\eta}})^\top \mid c^\top_{out}) \cdot rk_{(f,t_1)\to(g,t_2)}\\
&= (\mathsf{BD}(c_{in}, c_f, \bar{c}_{g_{\hat{t}_1}}, ..., \bar{c}_{g_{\hat{t}_\eta}})^\top \mid c^\top_{out}) \cdot\\
&\quad \begin{bmatrix} \overline{\mathbf{R}}_1\mathbf{H}_{y,t_2} + \overline{\mathbf{R}}_2 & \overline{\mathbf{R}}_1\mathbf{U} + \overline{\mathbf{R}}_3 - \mathsf{P}_2(\mathbf{R}_{f,\eta}) \\ \mathbf{0}_{m\times(\ell+d+1)m} & \mathbf{I}_{m\times m} \end{bmatrix}
\end{aligned}$$

Then,

$$\begin{aligned}
\mathbf{H}_1^\top &= \mathsf{BD}(c_{in}, c_f, \bar{c}_{g_{\hat{t}_1}}, ..., \bar{c}_{g_{\hat{t}_\eta}})^\top \overline{\mathbf{R}}_1\mathbf{H}_{y,t_2}\\
&\quad + \mathsf{BD}(c_{in}, c_f, \bar{c}_{g_{\hat{t}_1}}, ..., \bar{c}_{g_{\hat{t}_\eta}})^\top \overline{\mathbf{R}}_2.
\end{aligned}$$

$$\begin{aligned}
\mathbf{H}_2^\top &= \mathsf{BD}(c_{in}, c_f, \bar{c}_{g_{\hat{t}_1}}, ..., \bar{c}_{g_{\hat{t}_\eta}})^\top \overline{\mathbf{R}}_1\mathbf{U}\\
&\quad + \mathsf{BD}(c_{in}, c_f, \bar{c}_{g_{\hat{t}_1}}, ..., \bar{c}_{g_{\hat{t}_\eta}})^\top \overline{\mathbf{R}}_3\\
&\quad - \mathsf{BD}(c_{in}, c_f, \bar{c}_{g_{\hat{t}_1}}, ..., \bar{c}_{g_{\hat{t}_\eta}})^\top \mathsf{P}_2(\mathbf{R}_{f,\eta}) + c^\top_{out}
\end{aligned}$$

Where

$$\begin{aligned}
&\mathsf{BD}(c_{in}, c_f, \bar{c}_{g_{\hat{t}_1}}, ..., \bar{c}_{g_{\hat{t}_\eta}})^\top \mathsf{P}_2(\mathbf{R}_{f,\eta})\\
&= (c_{in}, c_f, \bar{c}_{g_{\hat{t}_1}}, ..., \bar{c}_{g_{\hat{t}_\eta}})^\top \mathbf{R}_{f,\eta}\\
&= \mathbf{s}^\top(\mathbf{A}_0 \mid \mathbf{A}_f \mid \mathbf{B}_{g_{\hat{t}_1}} \mid ... \mid \mathbf{B}_{g_{\hat{t}_\eta}}) \cdot \mathbf{R}_{f,\eta}\\
&\quad + (\mathbf{e}_0^\top \mid \mathbf{e}_f^\top \mid \mathbf{e}^\top_{g_{\hat{t}_1}} \mid ... \mid \mathbf{e}^\top_{g_{\hat{t}_\eta}})\mathbf{R}_{f,\eta}\\
&= \mathbf{s}^\top\mathbf{U} + (\mathbf{e}_0^\top \mid \mathbf{e}_f^\top \mid \mathbf{e}^\top_{g_{\hat{t}_1}} \mid ... \mid \mathbf{e}^\top_{g_{\hat{t}_\eta}}) \cdot \mathbf{R}_{f,\eta}.
\end{aligned}$$

Then,

$$\begin{aligned}
\mathbf{H}_2^\top &= \mathsf{BD}(c_{in}, c_f, \bar{c}_{g_{\hat{t}_1}}, ..., \bar{c}_{g_{\hat{t}_\eta}})^\top \overline{\mathbf{R}}_1\mathbf{U}\\
&\quad + \mathsf{BD}(c_{in}, c_f, \bar{c}_{g_{\hat{t}_1}}, ..., \bar{c}_{g_{\hat{t}_\eta}})^\top \overline{\mathbf{R}}_3 + \left\lfloor \frac{q}{2} \right\rfloor \mu^\top\\
&\quad + \mathbf{e}^\top_{out} - (\mathbf{e}_0^\top \mid \mathbf{e}_f^\top \mid \mathbf{e}^\top_{g_{\hat{t}_1}} \mid ... \mid \mathbf{e}^\top_{g_{\hat{t}_\eta}}) \cdot \mathbf{R}_{f,\eta}
\end{aligned}$$

Finally, $ct_{(f,t_1)\to(g,t_2)} = (\mathbf{H}^\top_{y,t_2}\bar{\mathbf{s}} + \bar{\mathbf{e}}_1, \mathbf{U}^\top\bar{\mathbf{s}} + \left\lfloor \frac{q}{2} \right\rfloor \mu + \bar{\mathbf{e}}_2)$, where

$$\begin{aligned}
\bar{\mathbf{s}} &= \overline{\mathbf{R}}_1^\top \mathsf{BD}\left(c_{in}, c_f, \bar{c}_{g_{\hat{t}_1}}, ..., \bar{c}_{g_{\hat{t}_\eta}}\right),\\
\bar{\mathbf{e}}_1 &= \overline{\mathbf{R}}_2^\top \mathsf{BD}\left(c_{in}, c_f, \bar{c}_{g_{\hat{t}_1}}, ..., \bar{c}_{g_{\hat{t}_\eta}}\right),\\
\bar{\mathbf{e}}_2 &= \overline{\mathbf{R}}_3^\top \mathsf{BD}\left(c_{in}, c_f, \bar{c}_{g_{\hat{t}_1}}, ..., \bar{c}_{g_{\hat{t}_\eta}}\right) + \mathbf{e}_{out}\\
&\quad - \mathbf{R}^\top_{f,\eta}(\mathbf{e}_{in}, \mathbf{e}_f, \mathbf{e}_{g_{\hat{t}_1}}, ..., \mathbf{e}_{g_{\hat{t}_\eta}}).
\end{aligned}$$

Then for $g(y) = 0 \wedge g_{\hat{t}_1}(t_2) = 0 \wedge ... \wedge g_{\hat{t}_\eta}(t_2) = 0$, following the same decryption procedure as for the original ciphertext will successfully recover the original message.

C. Security Proof

In this section, we prove that the proposed PABPRE-ST scheme is selectively HRA secure in the standard model.

Theorem 1: If the $\mathsf{DLWE}_{n,m,q,\chi}$ problem is hard, then the above PABPRE-ST scheme is selectively HRA secure in the standard model.

Proof: We prove the theorem using a sequence of games, where the first game is the real security game defined in the security model. In the final game, the adversary's advantage is zero because the challenge ciphertext is random. The hardness of $\mathsf{DLWE}_{n,m,q,\chi}$ is used to show computational indistinguishability between the last two games. All games are described as follows:

- **Game0.** This is the real selective HRA security game defined in the security model.

- **Game1.** In Game1, we change the way the public matrices $\mathbf{A}_i$ for all $i \in [\ell]$ and $\mathbf{B}_j$ for all $j \in [d]$ are generated. The challenger samples random matrices $\mathbf{S}_i^* \in \{-1, +1\}^{m\times m}$ and computes

$$\mathbf{A}_i = \mathbf{A}_0\mathbf{S}_i^* - x_i^*\mathbf{G} \quad \text{for all } i \in [\ell].$$

Then it samples random matrices $\mathbf{R}_1^*, \dots, \mathbf{R}_d^* \in \{-1, +1\}^{m\times m}$ and computes

$$\mathbf{B}_j = \mathbf{A}_0\mathbf{R}_j^* - t_j^*\mathbf{G} \quad \text{for all } j \in [d].$$

The rest remains the same as in Game0.

In Game0, the public matrices $\mathbf{A}_i$ for all $i \in [\ell]$ and $\mathbf{B}_j$ for all $j \in [d]$ are uniform in $\mathbb{Z}_q^{n\times m}$. In Game1, for all $i \in [\ell]$, we have $\mathbf{A}_i = \mathbf{A}_0\mathbf{S}_i^* - x_i^*\mathbf{G}$ where $\mathbf{S}_i^* \in \{-1, +1\}^{m\times m}$, and for all $j \in [d]$, we have $\mathbf{B}_j = \mathbf{A}_0\mathbf{R}_j^* - t_j^*\mathbf{G}$ where $\mathbf{R}_j^* \in \{-1, +1\}^{m\times m}$. We observe that $\mathbf{S}_i^*$ and $\mathbf{R}_j^*$ only appear in the construction of $\mathbf{A}_i$ and $\mathbf{B}_j$ and in the challenge ciphertext, specifically in terms like $(\mathbf{S}_i^*)^\top\mathbf{e}_0$ and $(\mathbf{R}_j^*)^\top\mathbf{e}_0$ for $i \in [\ell], j \in [d]$, and $\mathbf{e}_0 \in \mathbb{Z}_q^m$. By the Lemma 9, for all $i \in [\ell]$, the distribution $(\mathbf{A}_0, \mathbf{A}_0\mathbf{S}_i^*, (\mathbf{S}_i^*)^\top\mathbf{e}_0)$ is statistically close to the distribution $(\mathbf{A}_0, \mathbf{U}_i, (\mathbf{S}_i^*)^\top\mathbf{e}_0)$, where $\mathbf{U}_i$ is a uniform matrix in $\mathbb{Z}_q^{n\times m}$. Similarly, for all $j \in [d]$, the distribution $(\mathbf{A}_0, \mathbf{A}_0\mathbf{R}_j^*, (\mathbf{R}_j^*)^\top\mathbf{e}_0)$ is statistically close to $(\mathbf{A}_0, \mathbf{U}_j^{'}, (\mathbf{R}_j^*)^\top\mathbf{e}_0)$, where $\mathbf{U}_j^{'}$ is a uniform matrix in $\mathbb{Z}_q^{n\times m}$. Therefore, the public matrices $\{\mathbf{A}_1, \dots, \mathbf{A}_\ell, \mathbf{B}_1, \dots, \mathbf{B}_d\}$ in Game0 and Game1 are statistically indistinguishable. Since in Game1 we only changed the generation of matrices $\{\mathbf{A}_1, \dots, \mathbf{A}_\ell, \mathbf{B}_1, \dots, \mathbf{B}_d\}$, we conclude that Game0 and Game1 are statistically indistinguishable.

- **Game2.** In Game2, we change the generation of $\mathbf{A}_0$ and $\mathbf{U}$. In this game, we sample two uniform random matrices $\mathbf{A}_0 \leftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{R}^* \leftarrow \mathcal{D}_\sigma(\mathbb{Z}^{(\ell+d+1)m\times m})$ (here, let $\sigma = \omega(\sqrt{mlogm}) > \eta_\varepsilon(\mathbb{Z})$), and compute

$$\begin{aligned}
\mathbf{U} &= (\mathbf{A}_0 \mid x_1^*\mathbf{G} + \mathbf{A}_1 \mid \cdots \mid x_\ell^*\mathbf{G} + \mathbf{A}_\ell \mid t_1^*\mathbf{G} + \mathbf{B}_1 \mid \cdots \mid\\
&\quad t_d^*\mathbf{G} + \mathbf{B}_d)\mathbf{R}^*\\
&= (\mathbf{A}_0 \mid \mathbf{A}_0\mathbf{S}_1^* \mid \cdots \mid \mathbf{A}_0\mathbf{S}_\ell^* \mid \mathbf{A}_0\mathbf{R}_1^* \mid \cdots \mid \mathbf{A}_0\mathbf{R}_d^*)\mathbf{R}^*.
\end{aligned}$$

In Game1, the matrix $\mathbf{A}_0$ is generated via $\mathsf{TrapGen}(1^n, 1^m, q)$, and $\mathbf{U}$ is a uniform random matrix in $\mathbb{Z}_q^{n\times m}$. By Lemma 6, the matrix $\mathbf{A}_0$ in Game1 and Game2 are statistically close. Furthermore, by leftover hash lemma, the matrix $\mathbf{U}$ in Game1 and Game2 are statistically indistinguishable.

Because $\mathbf{A}_0$ is a randomly sampled matrix, the challenger does not have a trapdoor for $\Lambda_q^\perp(\mathbf{A}_0)$, but it can still answer all queries as follows:

Initialization: $\mathcal{A}$ sends a challenge attribute vector $x^* \in \mathcal{X}$ and a challenge tag vector $t^* = (t_1^*, \dots, t_d^*) \in \mathcal{T}$ to the challenger.

Setup: The challenger runs $(PP, msk) \leftarrow$ **Setup**$(1^\lambda, \ell, d)$ and gives PP to $\mathcal{A}$. The challenger also maintains a table of tuples $\{f, sk_{f,P_{f,\eta}}, P_f, C_f\}$ representing the key state for each policy f, where P_f and C_f are initially empty sets; initializes a counter numCt to 0; initializes a key-value store $\mathcal{H}$ as empty; and initializes a set Derive as empty.

Phase 1: $\mathcal{A}$ can adaptively query the following oracles polynomially many times in any order:

• $\mathcal{O}_{\mathsf{Puncture}}(f, \hat{t})$: When $\mathcal{A}$ submits any policy $f \in \mathcal{F}$ and a tag $\hat{t}$, there are two cases:

1. **Case 1:** $f(x^*) \neq 0$

(1) If a tuple $\{f, sk_{f,P_{f,\eta}}, P_f, C_f\}$ exists, the challenger directly runs $sk_{f,P_{f,\eta+1}} \leftarrow$ **Puncture**$(PP, sk_{f,P_{f,\eta}}, \hat{t})$, adds $\hat{t}$ to P_f, and updates the tuple to $\{f, sk_{f,P_{f,\eta+1}}, P_f, C_f\}$.

(2) Otherwise, the challenger runs

$$\mathbf{S}_f^* \leftarrow \mathsf{Eval}_{\mathsf{sim}}^{\mathcal{F}}(f, \{(\mathbf{S}_i^*, x_i^*)\}_{i\in[\ell]}, \mathbf{A}_0)$$

such that $\mathbf{A}_0\mathbf{S}_f^* - f(x^*)\mathbf{G} = \mathbf{A}_f$, where $\|\mathbf{S}_f^*\| \leq \alpha_{\mathcal{F}}(n)$. Then it runs

$$\mathbf{T}_f^{ER} \leftarrow \mathsf{ExtendRight}(\mathbf{A}_0, \mathbf{S}_f^*, f(x^*)\mathbf{G}, \mathbf{T}_{\mathbf{G}}),$$

where $\|\mathbf{T}_f^{ER}\|_{\mathsf{GS}} \leq \|\mathbf{T}_{\mathbf{G}}\|_{\mathsf{GS}} \cdot \|\mathbf{S}_f^*\|_2 \leq \sqrt{5}\cdot\alpha_{\mathcal{F}}(n)$. Then it randomizes the trapdoor

$$\mathbf{T}_f \leftarrow \mathsf{RandBasis}((\mathbf{A}_0 \mid \mathbf{A}_f), \mathbf{T}_f^{ER}, \sigma_0),$$

where $\sigma_0 \geq \|\mathbf{T}_f^{ER}\|_{\mathsf{GS}} \cdot \omega(\sqrt{\log m}) \geq \omega(\alpha_{\mathcal{F}}(n)\sqrt{\log m})$. Let the initial key for policy f be $sk_{f,\emptyset} = \mathbf{T}_f$. Then it executes $sk_{f,P_{f,1}} \leftarrow$ **Puncture**$(PP, sk_{f,\emptyset}, \hat{t})$, adds $\hat{t}$ to P_f, and creates a new tuple $\{f, sk_{f,P_{f,1}}, P_f, C_f\}$ where $P_f = \{\hat{t}\}$, $C_f = \emptyset$.

2. **Case 2:** $f(x^*) = 0$

(1) If a tuple $\{f, -, P_f, C_f\}$ exists, the challenger adds $\hat{t}$ to P_f and updates the tuple.

(2) Otherwise, the challenger adds $\hat{t}$ to P_f, creates a new tuple $\{f, -, P_f, C_f\}$ where $P_f = \{\hat{t}\}$, $C_f = \emptyset$.

• $\mathcal{O}_{\mathsf{Corrupt}}(f)$: When $\mathcal{A}$ first queries policy f for corruption, the challenger considers two cases:

1. **Case 1:** $f(x^*) \neq 0$

(1) If a tuple $\{f, sk_{f,P_{f,\eta}}, P_f, C_f\}$ exists, the challenger gives $sk_{f,P_{f,\eta}}$ to $\mathcal{A}$ and sets $C_f \leftarrow P_f$.

(2) Otherwise, the challenger computes the initial key $sk_{f,\emptyset}$ for policy f as in the $\mathcal{O}_{\mathsf{Puncture}}$ phase, sends it to $\mathcal{A}$, creates a new tuple $\{f, sk_{f,\emptyset}, P_f, C_f\}$, and sets $C_f \leftarrow P_f$ (which is $\emptyset$).

(3) For all subsequent queries, the challenger returns $\perp$.

2. **Case 2:** $f(x^*) = 0$

(1) If a tuple $\{f, -, P_f, C_f\}$ exists, the challenger checks if $P_f \cap \{t_1^*, \ldots, t_d^*\} = \emptyset$. If yes, return $\perp$. Otherwise, there exists $\hat{t}_j \in P_f$ such that $g_{\hat{t}_j}(t^*) \neq 0 \mod q$. Assume $P_f = \{\hat{t}_1, \ldots, \hat{t}_k\}$. Without loss of generality, assume $g_{\hat{t}_k}(t^*) \neq 0 \mod q$. Compute

$$\mathbf{A}_f \leftarrow \mathsf{Eval}_{\mathsf{pk}}^{\mathcal{F}}((\mathbf{A}_i)_{i=1}^{\ell}, f).$$

For all $j \in [d]$, compute

$$\mathbf{B}_{g_{\hat{t}_j}} \leftarrow \mathsf{Eval}_{\mathsf{pk}}^{\mathcal{G}}((\mathbf{B}_i)_{i=1}^{d}, g_{\hat{t}_j}).$$

Then run

$$\mathbf{R}_k^* \leftarrow \mathsf{Eval}_{\mathsf{sim}}^{\mathcal{G}}(g_{\hat{t}_k}, \{(\mathbf{R}_j^*, t_j^*)\}_{j\in[d]}, \mathbf{A}_0)$$

such that $\mathbf{A}_0\mathbf{R}_k^* - g_{\hat{t}_k}(t^*)\mathbf{G} = \mathbf{B}_{g_{\hat{t}_k}}$, where $\|\mathbf{R}_k^*\| \leq \alpha_{\mathcal{G}}(n)$. Then run

$$\mathbf{T}_{g_{\hat{t}_k}}^{ER} \leftarrow \mathsf{ExtendRight}(\mathbf{A}_0, \mathbf{R}_k^*, g_{\hat{t}_k}(t^*)\mathbf{G}, \mathbf{T}_{\mathbf{G}}),$$

where $\|\mathbf{T}_{g_{\hat{t}_k}}^{ER}\|_{\mathsf{GS}} \leq \|\mathbf{T}_{\mathbf{G}}\|_{\mathsf{GS}} \cdot \|\mathbf{R}_k^*\|_2 \leq \sqrt{5}\cdot\alpha_{\mathcal{G}}(n)$. Then run

$$\mathbf{T}_{f,\mathcal{P}_{f,k}}^{EL} \leftarrow \mathsf{ExtendLeft}((\mathbf{A}_0 \mid \mathbf{B}_{g_{\hat{t}_k}}), (\mathbf{A}_f \mid \mathbf{B}_{g_{\hat{t}_1}} \mid \\ \cdots \mid \mathbf{B}_{g_{\hat{t}_{k-1}}}), \mathbf{T}_{g_{\hat{t}_k}}^{ER}),$$

obtaining a trapdoor $\mathbf{T}_{(\mathbf{A}_0|\mathbf{B}_{g_{\hat{t}_k}}|\mathbf{A}_f|\mathbf{B}_{g_{\hat{t}_1}}|\cdots|\mathbf{B}_{g_{\hat{t}_{k-1}}})}$ for $(\mathbf{A}_0 \mid \mathbf{B}_{g_{\hat{t}_k}} \mid \mathbf{A}_f \mid \mathbf{B}_{g_{\hat{t}_1}} \mid \cdots \mid \mathbf{B}_{g_{\hat{t}_{k-1}}})$. Then swap rows of $\mathbf{T}_{(\mathbf{A}_0|\mathbf{B}_{g_{\hat{t}_k}}|\mathbf{A}_f|\mathbf{B}_{g_{\hat{t}_1}}|\cdots|\mathbf{B}_{g_{\hat{t}_{k-1}}})}$ to get a trapdoor $\mathbf{T}_{(\mathbf{A}_0|\mathbf{A}_f|\mathbf{B}_{g_{\hat{t}_1}}|\cdots|\mathbf{B}_{g_{\hat{t}_k}})}$ for $(\mathbf{A}_0 \mid \mathbf{A}_f \mid \mathbf{B}_{g_{\hat{t}_1}} \mid \cdots \mid \mathbf{B}_{g_{\hat{t}_k}})$. Then randomize the trapdoor

$$\mathbf{T}_{(\mathbf{A}_0|\mathbf{A}_f|\mathbf{B}_{g_{\hat{t}_1}}|\cdots|\mathbf{B}_{g_{\hat{t}_k}})} \leftarrow \mathsf{RandBasis} \\ ((\mathbf{A}_0 \mid \mathbf{A}_f \mid \mathbf{B}_{g_{\hat{t}_1}} \mid \cdots \mid \mathbf{B}_{g_{\hat{t}_k}}), \quad \mathbf{T}_{(\mathbf{A}_0|\mathbf{A}_f|\mathbf{B}_{g_{\hat{t}_1}}|\cdots|\mathbf{B}_{g_{\hat{t}_k}})}, \sigma_k).$$

Finally, set the latest punctured key $sk_{f,\mathcal{P}_{f,k}} = \mathbf{T}_{(\mathbf{A}_0|\mathbf{A}_f|\mathbf{B}_{g_{\hat{t}_1}}|\cdots|\mathbf{B}_{g_{\hat{t}_k}})}$ and send it to $\mathcal{A}$. Then set $C_f \leftarrow P_f$.

(2) Otherwise, the challenger returns $\perp$.

(3) For all subsequent queries, the challenger returns $\perp$.

• $\mathcal{O}_{\mathsf{ReKeyGen}}(f, g, t_1, t_2)$: Input two policy-tag pairs (f, t_1) and (g, t_2). The challenger does the following:

1. If $f(x^*) \neq 0$ and no tuple $\{f, sk_{f,P_{f,\eta}}, P_f, C_f\}$ exists, then the challenger generates a punctured key $sk_{f,\mathcal{P}_\emptyset}$ as in $\mathcal{O}_{\mathsf{Puncture}}$. Set $P_f = \emptyset$, and create a new tuple $\{f, sk_{f,\mathcal{P}_\emptyset}, P_f, C_f\}$. If $f(x^*) = 0$ and no tuple $\{f, -, P_f, C_f\}$ exists, then set $P_f = \emptyset$, and create a new tuple $\{f, -, P_f, C_f\}$. Let $P_f = \{\hat{t}_1, \ldots, \hat{t}_\eta\}$.

2. If $f(x^*) \neq 0$ or $(f(x^*) = 0$ and $P_f \cap \{t_1^*, \ldots, t_d^*\} \neq \emptyset)$, the challenger generates the latest punctured key $sk_{f,\mathcal{P}_{f,\eta}}$ as in $\mathcal{O}_{\mathsf{Corrupt}}$. Then, using $sk_{f,\mathcal{P}_{f,\eta}}$, the challenger runs $rk_{(f,t_1)\to(g,t_2)} \leftarrow$ **ReKeyGen**$(PP, sk_{f,\mathcal{P}_{f,\eta}}, f, g, t_1, t_2)$ to get the re-encryption key $rk_{(f,t_1)\to(g,t_2)}$ and sends it to $\mathcal{A}$.

3. If $f(x^*) = 0$ and $P_f \cap \{t_1^*, \ldots, t_d^*\} = \emptyset$ and $g(x^*) = 0$ and $t_2 = t^*$, the challenger can no longer generate the key $sk_{f,\mathcal{P}_{f,\eta}}$. However, the challenger can simulate a re-encryption key that is computationally indistinguishable. Specifically, the challenger samples $\mathbf{X}_1 \leftarrow \mathbb{Z}_q^{(2+\eta)mk\times(\ell+d+1)m}$, $\mathbf{X}_2 \leftarrow \mathbb{Z}_q^{(2+\eta)mk\times m}$, and sets

$$rk_{(f,t_1)\to(g,t_2)} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 \\ \mathbf{0}_{m\times(\ell+d+1)m} & \mathbf{I}_{m\times m} \end{bmatrix},$$

then returns the re-encryption key $rk_{(f,t_1)\to(g,t_2)}$ to $\mathcal{A}$.

• $\mathcal{O}_{\mathsf{Enc}}(x, t, \mu)$: When $\mathcal{A}$ submits any $(x, t, \mu) \in \mathcal{X}\times\mathcal{T}\times\mathcal{M}$, the challenger runs $ct \leftarrow$ **Enc**$(PP, x \in \mathcal{X}, t \in \mathcal{T}, \mu)$ and returns ct to $\mathcal{A}$. Then, the challenger increments numCt and stores ct in $\mathcal{H}$ with key (x, t, numCt).

• $\mathcal{O}_{\mathsf{ReEnc}}(f, g, t_2, (x, t, k))$: When $\mathcal{A}$ inputs two policies f, g, a tag vector t_2, and a handle (x, t, k) (where $t = \{t_1, \ldots, t_d\}$, $k \leq \mathsf{numCt}$), the challenger does the following:

1. If no value exists in $\mathcal{H}$ for key (x, t, k), output $\perp$. Otherwise, let ct be the corresponding value.

2. If $f(x^*) \neq 0$ and no tuple $\{f, sk_{f,P_{f,\eta}}, P_f, C_f\}$ exists, then the challenger generates a punctured key $sk_{f,\mathcal{P}_\emptyset}$ as in $\mathcal{O}_{\mathsf{Puncture}}$. Set $P_f = \emptyset$, and create a new tuple $\{f, sk_{f,\mathcal{P}_\emptyset}, P_f, C_f\}$. If $f(x^*) = 0$ and no tuple $\{f, -, P_f, C_f\}$ exists, then set $P_f = \emptyset$, $C_f = \emptyset$, and create a new tuple $\{f, -, P_f, C_f\}$. Let $P_f = \{\hat{t}_1, \ldots, \hat{t}_\eta\}$.

3. If $f(x) \neq 0$ or ($f(x) = 0$ and $P_f \cap \{t_1, \ldots, t_d\} \neq \emptyset$), the challenger outputs $\perp$ (indicating ct is invalid). Otherwise, the challenger parses ct as $ct = (c_{in}, c_1, \cdots, c_\ell, \bar{c}_1, \ldots, \bar{c}_d, c_{out})$, and considers the following 6 cases (summarized in Table II):

(1) If $x = x^*$ and $t = t^*$ (where $f(x) = 0$ and $P_f \cap \{t_1^*, \ldots, t_d^*\} = \emptyset$), the challenger does:

(a) Choose an attribute set $y = (y_1, \ldots, y_\ell)$ such that $g(y) = 0$.

(b) Sample three matrices $\mathbf{E}_0 \leftarrow \mathbb{Z}_q^{2mk \times n}$, $\mathbf{E}_1 \leftarrow \chi^{2mk \times (\ell+d+1)m}$, $\mathbf{E}_2 \leftarrow \chi^{2mk \times m}$, and compute

$$\mathbf{H}_{y,t_2} = (\mathbf{A}_0 \mid y_1\mathbf{G} + \mathbf{A}_1 \mid \cdots \mid y_\ell\mathbf{G} + \mathbf{A}_\ell \mid t_{2,1}\mathbf{G} + \mathbf{B}_1 \\ \mid \cdots \mid t_{2,d}\mathbf{G} + \mathbf{B}_d) \in \mathbb{Z}_q^{n \times (\ell+d+1)m}.$$

(c) Compute a dummy re-encryption key:

$$rk_{(f,t_1)\to(g,t_2)} = \begin{bmatrix} \mathbf{E}_0\mathbf{H}_{y,t_2} + \mathbf{E}_1 & \mathbf{E}_0\mathbf{U} + \mathbf{E}_2 - \mathsf{P}_2(\mathbf{R}^*) \\ \mathbf{0}_{m \times (\ell+d+1)m} & \mathbf{I}_{m \times m} \end{bmatrix} \\ \in \mathbb{Z}_q^{(2mk+m) \times (\ell+d+2)m}.$$

(d) Compute the re-encrypted ciphertext:

$$ct_{(f,t_1)\to(g,t_2)}^\top = (\mathsf{BD}(c_{in}, c_1, \cdots, c_\ell, \bar{c}_1, \ldots, \bar{c}_d)^\top \mid c_{out}^\top) \\ \cdot rk_{(f,t_1)\to(g,t_2)} \\ \in \mathbb{Z}_q^{1 \times (\ell+d+2)m}.$$

(2) If $x \neq x^*$, $t = t^*$ or $t \neq t^*$ (here $f(x^*) = 0$ and $P_f \cap \{t_1^*, \ldots, t_d^*\} = \emptyset$), the challenger computes

$$\mathbf{H}_{x,t} = (\mathbf{A}_0 \mid x_1\mathbf{G} + \mathbf{A}_1 \mid \cdots \mid x_\ell\mathbf{G} + \mathbf{A}_\ell \mid t_1\mathbf{G} + \mathbf{B}_1 \\ \mid \cdots \mid t_d\mathbf{G} + \mathbf{B}_d) \\ = (\mathbf{A}_0 \mid \mathbf{A}_0\mathbf{S}_1^* + (x_1 - x_1^*)\mathbf{G} \\ \mid \cdots \mid \mathbf{A}_0\mathbf{S}_\ell^* + (x_\ell - x_\ell^*)\mathbf{G} \mid t_1\mathbf{G} + \mathbf{B}_1 \mid \cdots \mid t_d\mathbf{G} + \mathbf{B}_d).$$

Since $x \neq x^*$, there exists at least one index $i \in [\ell]$ such that $x_i \neq x_i^*$. Without loss of generality, assume $x_l \neq x_l^*$ for some $l \in [\ell]$. Then run

$$\mathbf{T}^{ER}_{(\mathbf{A}_0|\mathbf{A}_0\mathbf{S}_l^* + (x_l - x_l^*)\mathbf{G})} \leftarrow \mathsf{ExtendRight}(\mathbf{A}_0, \mathbf{S}_l^*, (x_l - x_l^*)\mathbf{G}, \mathbf{T}_\mathbf{G}),$$

where $\|\mathbf{T}^{ER}_{(\mathbf{A}_0|\mathbf{A}_0\mathbf{S}_l^* + (x_l - x_l^*)\mathbf{G})}\|_{\mathsf{GS}} \leq \|\mathbf{T}_\mathbf{G}\|_{\mathsf{GS}} \cdot \|\mathbf{S}_l^*\|_2 \leq 20\sqrt{5m}$. Let $\mathbf{H}_1 = (\mathbf{A}_0 \mid \mathbf{A}_0\mathbf{S}_l^* + (x_l - x_l^*)\mathbf{G} \mid \mathbf{A}_0\mathbf{S}_1^* + (x_1 - x_1^*)\mathbf{G} \mid \cdots \mid \mathbf{A}_0\mathbf{S}_{l-1}^* + (x_{l-1} - x_{l-1}^*)\mathbf{G} \mid t_1\mathbf{G} + \mathbf{B}_1 \mid \cdots \mid t_d\mathbf{G} + \mathbf{B}_d)$. Then run

$$\mathbf{T}^{EL}_{\mathbf{H}_1} \leftarrow \mathsf{ExtendLeft}((\mathbf{A}_0 \mid \mathbf{A}_0\mathbf{S}_l^* + (x_l - x_l^*)\mathbf{G}), \\ (\mathbf{A}_0\mathbf{S}_1^* + (x_1 - x_1^*)\mathbf{G} \mid \cdots \mid \mathbf{A}_0\mathbf{S}_{l-1}^* + (x_{l-1} - x_{l-1}^*)\mathbf{G} \mid \\ t_1\mathbf{G} + \mathbf{B}_1 \mid \cdots \mid t_d\mathbf{G} + \mathbf{B}_d), \quad \mathbf{T}^{ER}_{(\mathbf{A}_0|\mathbf{A}_0\mathbf{S}_l^* + (x_l - x_l^*)\mathbf{G})}),$$

obtaining a trapdoor $\mathbf{T}^{EL}_{\mathbf{H}_2}$ for $\mathbf{H}_2 = (\mathbf{A}_0 \mid \mathbf{A}_0\mathbf{S}_l^* + (x_l - x_l^*)\mathbf{G} \mid \mathbf{A}_0\mathbf{S}_1^* + (x_1 - x_1^*)\mathbf{G} \mid \cdots \mid \mathbf{A}_0\mathbf{S}_{l-1}^* + (x_{l-1} - x_{l-1}^*)\mathbf{G} \mid t_1\mathbf{G} + \mathbf{B}_1 \mid \cdots \mid t_d\mathbf{G} + \mathbf{B}_d)$. Then swap rows of $\mathbf{T}^{EL}_{\mathbf{H}_2}$ to get a trapdoor $\mathbf{T}^{EL}_{\mathbf{H}_{x,t}}$ for $\mathbf{H}_{x,t}$. Then run $\mathbf{R}_{x,t} \leftarrow \mathsf{SampleD}(\mathbf{H}_{x,t}, \mathbf{T}^{EL}_{\mathbf{H}_{x,t}}, \mathbf{U}, \sigma)$. Then, the challenger computes the re-encrypted ciphertext ct' in the same way as in case (1).

(3) If $x \neq x^*$, $t = t^*$ or $t \neq t^*$ (here $f(x^*) \neq 0$). The challenger generates the punctured key $sk_{f,\mathcal{P}_{f,\eta}}$ as in $\mathcal{O}_{\mathsf{Corrupt}}$ and $\mathcal{O}_{\mathsf{Puncture}}$, runs $rk_{(f,t_1)\to(g,t_2)} \leftarrow$ **ReKeyGen**$(PP, sk_{f,\mathcal{P}_{f,\eta}}, f, g, t_1, t_2)$, and generates the re-encrypted ciphertext $ct' \leftarrow$ **ReEnc**$(PP, rk_{(f,t_1)\to(g,t_2)}, ct)$.

(4) If $x = x^*$, $t \neq t^*$ (here $f(x) = 0$ and $P_f \cap \{t_1^*, \ldots, t_d^*\} = \emptyset$). The challenger computes

$$\mathbf{H}_{x,t} = (\mathbf{A}_0 \mid x_1\mathbf{G} + \mathbf{A}_1 \mid \cdots \mid x_\ell\mathbf{G} + \mathbf{A}_\ell \mid t_1\mathbf{G} + \mathbf{B}_1 \mid \\ \cdots \mid t_d\mathbf{G} + \mathbf{B}_d) \\ = (\mathbf{A}_0 \mid x_1\mathbf{G} + \mathbf{A}_1 \mid \cdots \mid x_\ell\mathbf{G} + \mathbf{A}_\ell \mid \mathbf{A}_0\mathbf{R}_1^* + \\ (t_1 - t_1^*)\mathbf{G} \mid \cdots \mid \mathbf{A}_0\mathbf{R}_d^* + (t_d - t_d^*)\mathbf{G}).$$

Since $t \neq t^*$, there exists at least one index $j \in [d]$ such that $t_j \neq t_j^*$. Without loss of generality, assume $t_d \neq t_d^*$. Then run

$$\mathbf{T}^{ER}_{(\mathbf{A}_0|\mathbf{A}_0\mathbf{R}_d^* + (t_d - t_d^*)\mathbf{G})} \leftarrow \mathsf{ExtendRight}(\mathbf{A}_0, \mathbf{R}_d^*, (t_d - t_d^*)\mathbf{G}, \mathbf{T}_\mathbf{G}),$$

where $\|\mathbf{T}^{ER}_{(\mathbf{A}_0|\mathbf{A}_0\mathbf{R}_d^* + (t_d - t_d^*)\mathbf{G})}\|_{\mathsf{GS}} \leq \|\mathbf{T}_\mathbf{G}\|_{\mathsf{GS}} \cdot \|\mathbf{R}_d^*\|_2 \leq 20\sqrt{5m}$. Then, the challenger computes $\mathbf{R}_{x,t}$ in a similar way as above and computes the re-encrypted ciphertext ct' in the same way.

(5) If $x = x^*$, $t \neq t^*$ (here $f(x) = 0$ and $P_f \cap \{t_1^*, \ldots, t_d^*\} \neq \emptyset$). The challenger generates the punctured key $sk_{f,\mathcal{P}_{f,i}}$ as in $\mathcal{O}_{\mathsf{Corrupt}}$, runs $rk_{(f,t_1)\to(g,t_2)} \leftarrow$ **ReKeyGen**$(PP, sk_{f,\mathcal{P}_{f,i}}, f, g, t_1, t_2)$, and generates the re-encrypted ciphertext $ct' \leftarrow$ **ReEnc**$(PP, rk_{(f,t_1)\to(g,t_2)}, ct)$.

(6) If $x \neq x^*$, $t \neq t^*$ (here $f(x^*) = 0$ and $P_f \cap \{t_1^*, \ldots, t_d^*\} \neq \emptyset$), the challenger generates the punctured key $sk_{f,\mathcal{P}_{f,\eta}}$ as in $\mathcal{O}_{\mathsf{Corrupt}}$, runs $rk_{(f,t_1)\to(g,t_2)} \leftarrow$ **ReKeyGen**$(PP, sk_{f,\mathcal{P}_{f,\eta}}, f, g, t_1, t_2)$, and generates the re-encrypted ciphertext $ct' \leftarrow$ **ReEnc**$(PP, rk_{(f,t_1)\to(g,t_2)}, ct)$.

Finally, the challenger returns the re-encrypted ciphertext ct' to $\mathcal{A}$.

Challenge: $\mathcal{A}$ submits two messages $\mu_0, \mu_1 \in \{0,1\}^m$. The challenger samples a random bit $b \leftarrow \{0,1\}$, and returns the challenge ciphertext $ct_b^* \leftarrow$ **Enc**(PP, μ_b, x^*, t^*) to $\mathcal{A}$. Then, the challenger increments numCt, adds numCt to the set Derive, and stores ct_b^* in $\mathcal{H}$ with key $(x^*, t^*, \mathsf{numCt})$. This query can only be made once.

Phase 2: $\mathcal{A}$ continues to query $\mathcal{O}_{\mathsf{Puncture}}$, $\mathcal{O}_{\mathsf{Corrupt}}$, $\mathcal{O}_{\mathsf{Enc}}$, $\mathcal{O}_{\mathsf{ReKeyGen}}$, and $\mathcal{O}_{\mathsf{ReEnc}}$. The challenger answers as in Phase 1, except the following restriction for $\mathcal{O}_{\mathsf{ReEnc}}$: if $k \in$ Derive and ($g(x^*) \neq 0$ or ($g(x^*) = 0$ and $t_2 \neq t^*$)), then output $\perp$. Otherwise, the challenger answers as in Phase 1.

We argue that the punctured keys $sk_{f,\mathcal{P}_{f,i}}$, re-encryption keys $rk_{(f,t_1)\to(g,t_2)}$, and re-encrypted ciphertexts ct' generated in Game1 are indistinguishable from those in Game2. First, by Lemma 6, the punctured keys $sk_{f,\mathcal{P}_{f,i}}$ generated in Game1 are statistically indistinguishable from those generated in Game2. Second, when $f(x^*) = 0 \wedge P_f \cap \{t_1^*, \ldots, t_d^*\} = \emptyset \wedge g(x^*) = 0 \wedge t_2 = t^*$, in Game1, the re-encryption key $rk_{(f,t_1)\to(g,t_2)}$ should be

$$rk_{(f,t_1)\to(g,t_2)} = \begin{bmatrix} \overline{\mathbf{R}}_1\mathbf{H}_{y,t_2} + \overline{\mathbf{R}}_2 & \overline{\mathbf{R}}_1\mathbf{U} + \overline{\mathbf{R}}_3 - \mathsf{P}_2(\mathbf{R}_{f,\eta}) \\ \mathbf{0}_{m \times (\ell+d+1)m} & \mathbf{I}_{m \times m} \end{bmatrix} \\ \in \mathbb{Z}_q^{((2+\eta)mk+m) \times (\ell+d+2)m}.$$

TABLE II
SUMMARY OF CHALLENGER'S RESPONSES IN $\mathcal{O}_{\mathbf{ReEnc}}$

Condition	**Condition**	**Method used by Challenger**
$x = x^* \wedge t = t^*$	$f(x^*) = 0 \wedge P_f \cap \{t_1^*, \ldots, t_d^*\} = \emptyset$	Use $\mathbf{R}^*$
$x \neq x^* \wedge t = t^*$ $x \neq x^* \wedge t \neq t^*$	$f(x^*) = 0 \wedge P_f \cap \{t_1^*, \ldots, t_d^*\} = \emptyset$	Use $(x - x^*)\mathbf{G}$ trapdoor
$x \neq x^* \wedge t = t^*$ $x \neq x^* \wedge t \neq t^*$	$f(x^*) \neq 0$	Use $f(x^*)\mathbf{G}$ trapdoor
$x = x^* \wedge t \neq t^*$	$f(x^*) = 0 \wedge P_f \cap \{t_1^*, \ldots, t_d^*\} = \emptyset$	Use $(t - t^*)\mathbf{G}$ trapdoor
$x = x^* \wedge t \neq t^*$	$f(x^*) = 0 \wedge P_f \cap \{t_1^*, \ldots, t_d^*\} \neq \emptyset$	Use $g_{\hat{t}_\eta}(t^*)\mathbf{G}$ trapdoor
$x \neq x^* \wedge t \neq t^*$	$f(x^*) = 0 \wedge P_f \cap \{t_1^*, \ldots, t_d^*\} \neq \emptyset$	Use $g_{\hat{t}_\eta}(t^*)\mathbf{G}$ trapdoor

In Game2, the re-encryption key is constructed as:

$$rk_{(f,t_1)\to(g,t_2)} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 \\ \mathbf{0}_{m\times(\ell+d+1)m} & \mathbf{I}_{m\times m} \end{bmatrix},$$

where $\mathbf{X}_1$ and $\mathbf{X}_2$ are random uniform matrices. We construct an algorithm $\mathcal{D}$ that solves the DLWE problem by interacting with an adversary that distinguishes between Game1 and Game2 re-encryption keys. Algorithm $\mathcal{D}$ first receives two LWE instances $(\mathbf{A}, \mathbf{C}_1)$ and $(\mathbf{U}, \mathbf{C}_2)$, where $\mathbf{C}_1, \mathbf{C}_2$ are either random matrices in $\mathbb{Z}_q^{(2+\eta)mk\times m}$, or $\mathbf{C}_1 = \mathbf{S}^\top\mathbf{A} + \mathbf{E}_1$ and $\mathbf{C}_2 = \mathbf{S}^\top\mathbf{U} + \mathbf{E}_2$. Here $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{U} \leftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{S} \in \mathbb{Z}_q^{n\times(2+\eta)mk}$, and $\mathbf{E}_1$, $\mathbf{E}_2$ are error vectors from $\chi^{(2+\eta)mk\times m}$. Algorithm $\mathcal{D}$ constructs $\mathbf{T}_1 = \mathbf{C}_1(\mathbf{I}_m|\mathbf{S}_1^*|\ldots|\mathbf{S}_\ell^*|\mathbf{R}_1^*|\ldots|\mathbf{R}_d^*)$, $\mathbf{T}_2 = \mathbf{C}_2 + \mathbf{P}_2(\mathbf{R}_{f,\eta})$. Then it constructs the re-encryption key $rk = \begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_2 \\ \mathbf{0}_{m\times(\ell+d+1)m} & \mathbf{I}_{m\times m} \end{bmatrix}$. If the DLWE instance is pseudorandom, then in the adversary's view, rk is the re-encryption key in Game1; if the DLWE instance is random, then by the leftover hash lemma, in the adversary's view, rk is the re-encryption key in Game2. Therefore, under the DLWE assumption, the re-encryption key computations in Game1 and Game2 are computationally indistinguishable. Finally, it is not hard to verify that all re-encrypted ciphertexts generated in Game2 are of the correct form and can be decrypted correctly. Therefore, from $\mathcal{A}$'s view, the re-encrypted ciphertexts ct' generated in Game1 are statistically indistinguishable from those generated in Game2. In conclusion, Game1 and Game2 are computationally indistinguishable.

- **Game3.** This game is the same as Game2, except that we change the way the challenge ciphertext $ct^* = (c_{in}^*, c_1^*, \ldots, c_\ell^*, \bar{c}_1^*, \ldots, \bar{c}_d^*, c_{out}^*)$ is generated. In this game, we sample a random independent vector from $\mathbb{Z}_q^{(\ell+d+2)m}$ as the challenge ciphertext. In this case, the challenge ciphertext is independent of the message μ_b, so the advantage of $\mathcal{A}$ is zero.

Next, we use the $\mathsf{DLWE}_{n,q,m,\chi}$ problem to show that Game2 and Game3 are computationally indistinguishable.

Reduction from LWE. We will construct a polynomial-time algorithm $\mathcal{B}$ that can solve the $\mathsf{DLWE}_{n,q,m,\chi}$ problem if $\mathcal{A}$ can distinguish between Game2 and Game3 with non-negligible advantage.

LWE Instance: $\mathcal{B}$ obtains an LWE instance: $(\mathbf{A}_0, c_{in}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$. According to the DLWE problem, we know that c_{in} is either random or

$$c_{in} = \mathbf{A}_0^\top\mathbf{s} + \mathbf{e}_{in}$$

for random vector $\mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{e}_{in} \leftarrow \chi^m$.

Setup. $\mathcal{B}$ sets up PP as in Game2.

Phase 1. $\mathcal{B}$ answers all queries from $\mathcal{A}$ as in Game2.

Challenge. Upon receiving $\mu_0, \mu_1 \in \{0,1\}^m$, $\mathcal{B}$ samples a random bit $b \leftarrow \{0,1\}$, and creates the challenge ciphertext $ct^* = (c_{in}^*, c_1^*, \ldots, c_\ell^*, \bar{c}_1^*, \ldots, \bar{c}_d^*, c_{out}^*)$ by setting

$$\begin{aligned} &(c_{in}^*, c_1^*, \ldots, c_\ell^*, \bar{c}_1^*, \ldots, \bar{c}_d^*) \\ &\quad= (\mathbf{I}_m \mid \mathbf{S}_1^* \mid \cdots \mid \mathbf{S}_\ell^* \mid \mathbf{R}_1^* \mid \cdots \mid \mathbf{R}_d^*)^\top c_{in} \\ &c_{out}^* = ((\mathbf{I}_m \mid \mathbf{S}_1^* \mid \cdots \mid \mathbf{S}_\ell^* \mid \mathbf{R}_1^* \mid \cdots \mid \mathbf{R}_d^*)\mathbf{R}^*)^\top c_{in} + \lfloor q/2 \rfloor \mu_b. \end{aligned}$$

Then, $\mathcal{B}$ returns ct^* to $\mathcal{A}$.

Phase 2. Same as in Game2.

Guess. $\mathcal{A}$ guesses whether it is interacting with the challenger of Game2 or Game3, and then $\mathcal{B}$ uses $\mathcal{A}$'s guess as the answer to the $\mathsf{DLWE}_{n,q,m,\chi}$ problem.

Next, we consider two cases: the first is when the LWE instance is pseudorandom, and the second is when the LWE instance is random.

1. We argue that if the LWE instance is pseudorandom, then the distribution of ct^* is as in Game2. Indeed, since the matrix $\mathbf{H}_{x^*,t^*}$ from the **Enc** algorithm is

$$\begin{aligned} \mathbf{H}_{x^*,t^*} &= (\mathbf{A}_0 \mid \mathbf{A}_1 + x_1^*\mathbf{G} \mid \cdots \mid \mathbf{A}_\ell + x_\ell^*\mathbf{G} \mid \mathbf{B}_1 + t_1^*\mathbf{G} \\ &\quad \mid \cdots \mid \mathbf{B}_d + t_d^*\mathbf{G}) \\ &= (\mathbf{A}_0 \mid \mathbf{A}_0\mathbf{S}_1^* \mid \cdots \mid \mathbf{A}_0\mathbf{S}_\ell^* \mid \mathbf{A}_0\mathbf{R}_1^* \mid \cdots \mid \mathbf{A}_0\mathbf{R}_d^*), \end{aligned}$$

the constructed vector $(c_{in}^*, c_1^*, \ldots, c_\ell^*, \bar{c}_1^*, \ldots, \bar{c}_d^*)$ is expressed as

$$\begin{aligned} &(c_{in}^*, c_1^*, \ldots, c_\ell^*, \bar{c}_1^*, \ldots, \bar{c}_d^*) \\ &\quad= (\mathbf{I}_m \mid \mathbf{S}_1^* \mid \cdots \mid \mathbf{S}_\ell^* \mid \mathbf{R}_1^* \mid \cdots \mid \mathbf{R}_d^*)^\top \mathbf{c}_{in} \\ &\quad= (\mathbf{I}_m \mid \mathbf{S}_1^* \mid \cdots \mid \mathbf{S}_\ell^* \mid \mathbf{R}_1^* \mid \cdots \mid \mathbf{R}_d^*)^\top (\mathbf{A}_0^\top\mathbf{s} + \mathbf{e}_{in}) \\ &\quad= \mathbf{H}_{x^*,t^*}^\top\mathbf{s} + \mathbf{e}, \end{aligned}$$

where $\mathbf{e} = (\mathbf{I}_m \mid \mathbf{S}_1^* \mid \cdots \mid \mathbf{S}_\ell^* \mid \mathbf{R}_1^* \mid \cdots \mid \mathbf{R}_d^*)^\top\mathbf{e}_{in}$. Therefore, the distribution of $(c_{in}^*, c_1^*, \ldots, c_\ell^*, \bar{c}_1^*, \ldots, \bar{c}_d^*)$ is identical to that in Game2. Furthermore, since $\mathbf{U} = (\mathbf{A}_0 \mid \mathbf{A}_0\mathbf{S}_1^* \mid \cdots \mid \mathbf{A}_0\mathbf{S}_\ell^* \mid \mathbf{A}_0\mathbf{R}_1^* \mid \cdots \mid \mathbf{A}_0\mathbf{R}_d^*)\mathbf{R}^*$, the constructed vector c_{out}^* can be expressed as

$$\begin{aligned} c_{out}^* &= ((\mathbf{I}_m \mid \mathbf{S}_1^* \mid \cdots \mid \mathbf{S}_\ell^* \mid \mathbf{R}_1^* \mid \cdots \mid \mathbf{R}_d^*)\mathbf{R}^*)^\top c_{in} + \lfloor q/2 \rfloor \mu_b \\ &= (\mathbf{R}^*)^\top (\mathbf{A}_0 \mid \mathbf{A}_0\mathbf{S}_1^* \mid \cdots \mid \mathbf{A}_0\mathbf{S}_\ell^* \mid \mathbf{A}_0\mathbf{R}_1^* \mid \cdots \mid \mathbf{A}_0\mathbf{R}_d^*)^\top \\ &\quad \mathbf{s} + \lfloor q/2 \rfloor \mu_b + ((\mathbf{I}_m \mid \mathbf{S}_1^* \mid \cdots \mid \mathbf{S}_\ell^* \mid \mathbf{R}_1^* \mid \cdots \mid \mathbf{R}_d^*)\mathbf{R}^*)^\top \mathbf{e}_{in} \\ &= \mathbf{U}^\top\mathbf{s} + \mathbf{e}_{out} + \lfloor q/2 \rfloor \mu_b, \end{aligned}$$

where $\mathbf{e}_{out} = ((\mathbf{I}_m \mid \mathbf{S}_1^* \mid \cdots \mid \mathbf{S}_\ell^* \mid \mathbf{R}_1^* \mid \cdots \mid \mathbf{R}_d^*)\mathbf{R}^*)^\top \mathbf{e}_{in}$. Therefore, the distribution of c_{out}^* is also identical to that in Game2. In conclusion, the distribution of ct^* is the same as in Game2.

2. We argue that ct^* is distributed as in Game 3 if the LWE instance is random, i.e., c_{in} is a random vector. Let $\mathbf{F} = (\mathbf{A}_0|\mathbf{A}_0\mathbf{S}_1^*|\ldots|\mathbf{A}_0\mathbf{S}_\ell^*|\mathbf{A}_0\mathbf{R}_1^*|\ldots|\mathbf{A}_0\mathbf{R}_d^*)$.

By the standard leftover hash lemma [49], $(\mathbf{A}_0, c_{in}^\top, \mathbf{A}_0\mathbf{S}_i^*, \mathbf{A}_0\mathbf{R}_j^*, c_{in}^\top\mathbf{S}_i^*, c_{in}^\top\mathbf{R}_j^*)$ is $\mathrm{negl}(n)-$uniform and $(\mathbf{F}, c_{in}^\top(\mathbf{I}_m|\mathbf{S}_1^*|\ldots|\mathbf{S}_\ell^*|\mathbf{R}_1^*|\ldots|\mathbf{R}_d^*), \mathbf{F}\mathbf{R}^*, (c_{in}^\top(\mathbf{I}_m|\mathbf{S}_1^*|\ldots|\mathbf{S}_\ell^*|\mathbf{R}_1^*|\ldots|\mathbf{R}_d^*))\mathbf{R}^*)$ is $\mathrm{negl}(n)-$uniform. Therefore, vector $(c_{in}^*, c_1^*, \ldots, c_\ell^*, \bar{c}_1^*, \ldots, \bar{c}_d^*, c_{out}^*)$ defined above is uniform and independent in $\mathbb{Z}_q^{(\ell+d+2)m}$. So, we conclude that the distribution of ct^* is as in Game 3. As described above, if the DLWE instance is pseudorandom, then $\mathcal{A}$'s view is as in Game2; if the DLWE instance is random, then $\mathcal{A}$'s view is as in Game3. Therefore, $\mathcal{B}$'s advantage in solving the DLWE problem is equal to $\mathcal{A}$'s advantage in distinguishing between Game2 and Game3. The proof of the theorem is complete.

VI. Conclusion

In this paper, we present a puncturable KP-ABPRE scheme supporting ciphertext tag switching, designed to enable secure cloud data sharing. To the best of our knowledge, this represents the first truly functional puncturable ABPRE scheme, capable of effectively protecting historical data confidentiality while achieving fine-grained message erasure. Our construction is lattice-based, providing inherent resistance against quantum computer attacks and honest re-encryption attacks (HRA). The scheme supports a priori bounded polynomial-depth circuits, offering significantly enhanced policy expressiveness. This work leaves an important open problem: the design of an HRA-secure puncturable CP-ABPRE scheme.

References

[1] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Comput.*, vol. 16, no. 1, pp. 69–73, Jan. 2012.

[2] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Proc. Int. Conf. Theory Appl. Cryptogr. Tech.*, 1998, pp. 127–144.

[3] M. Green and G. Ateniese, "Identity-based proxy re-encryption," in *Proc. Appl. Cryptogr. Netw. Secur.*, 2007, pp. 288–306.

[4] X. Liang, Z. Cao, H. Lin, and J. Shao, "Attribute based proxy re-encryption with delegating capabilities," in *Proc. 4th ACM Symp. Inf., Comput. Commun. Secur.*, 2009, pp. 276–286.

[5] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptogr. Tech.*, 2005, pp. 457–473.

[6] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, 2006, pp. 89–98.

[7] D. Boneh, C. Gentry, S. Gorbunov, S. Maffei, M. Rosen, and G. Segev, "Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits," in *Proc. Annu. Int. Conf. Theory Appl. Cryptogr. Tech.*, 2014, pp. 533–556.

[8] Z. Brakerski and V. Vaikuntanathan, "Circuit-ABE from LWE: unbounded attributes and semi-adaptive security," in *Proc. Annu. Int. Cryptol. Conf.*, 2016, pp. 363–384.

[9] V. Cini and H. Wee, "Unbounded ABE for circuits from LWE, revisited," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2024, pp. 238–267.

[10] H. Wee, "Optimal broadcast encryption and CP-ABE from evasive lattice assumptions," in *Proc. Annu. Int. Conf. Theory Appl. Cryptogr. Tech.*, 2022, pp. 217–241.

[11] P. Datta, I. Komargodski, and B. Waters, "Decentralized multi-authority ABE for DNFs from LWE," in *Proc. Annu. Int. Conf. Theory Appl. Cryptogr. Tech.*, 2021, pp. 177–209.

[12] K. Liang, L. Fang, W. Susilo, and D. S. Wong, "A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security," in *Proc. 5th Int. Conf. Intell. Netw. Collabor. Syst.*, 2013, pp. 552–559.

[13] K. Liang, M. H. Au, J. K. Liu, W. Susilo, D. S. Wong, and G. Yang, "A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing," *Future Gener. Comput. Syst.*, vol. 52, pp. 95–108, Nov. 2015.

[14] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "A verifiable and fair attribute-based proxy re-encryption scheme for data sharing in clouds," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 5, pp. 2907–2919, Sep. 2021.

[15] C. Ge, W. Susilo, Z. Liu, J. Baek, X. Luo, and L. Fang, "Attribute-based proxy re-encryption with direct revocation mechanism for data sharing in clouds," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 2, pp. 949–960, Mar.–Apr. 2024.

[16] P. Duan, Z. Ma, H. Gao, L. Zhang, and G. Liu, "Multi-Authority Attribute-Based Encryption Scheme With Access Delegation for Cross Blockchain Data Sharing," *IEEE Trans. Inf. Forensics Security*, early access, 2024.

[17] X. Li, Y. Xie, C. Peng, H. Huang, and K. Chen, "EPREAR: An Efficient Attribute-based Proxy Re-encryption Scheme with Fast Revocation for Data Sharing in AIoT," *IEEE Trans. Mobile Comput.*, early access, 2025.

[18] J. Li, C. Ma, and K. Zhang, "A novel lattice-based CP-ABPRE scheme for cloud sharing," *Symmetry*, vol. 11, no. 10, p. 1262, Oct. 2019.

[19] F. Luo, S. Al-Kuwari, F. Wang, and K. Chen, "Attribute-based proxy re-encryption from standard lattices," *Theor. Comput. Sci.*, vol. 865, pp. 52–62, Apr. 2021.

[20] W. Susilo, P. Dutta, D. H. Duong, and P. Wang, "Lattice-based HRA-secure attribute-based proxy re-encryption in standard model," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2021, pp. 169–191.

[21] F. Zhao, J. Weng, W. Xie, S. Tang, and R. H. Deng, "HRA-secure attribute-based threshold proxy re-encryption from lattices," *Inf. Sci.*, vol. 655, p. 119900, Jan. 2024.

[22] F. Zhao, J. Weng, W. Xie, S. Tang, and R. H. Deng, "Time-based attribute-based proxy re-encryption with decryption key update," *Des., Codes, Cryptogr.*, vol. 92, no. 12, pp. 4099–4129, Dec. 2024.

[23] E. Affum, X. Zhang, X. Wang, Z. Wang, and Q. Zhang, "Lattice Puncturable Attribute Based Proxy Re-encryption Scheme and Its Application in Information Centric Network," in *Proc. Future Inf. Commun. Conf.*, 2022, pp. 765–786.

[24] R. Canetti, S. Halevi, and J. Katz, "A forward-secure public-key encryption scheme," in *Proc. Annu. Int. Conf. Theory Appl. Cryptogr. Tech.*, 2003, pp. 255–271.

[25] M. D. Green and I. Miers, "Forward secure asynchronous messaging from puncturable encryption," in *Proc. IEEE Symp. Secur. Privacy*, 2015, pp. 305–320.

[26] A. Cohen, "What about bob? The inadequacy of CPA security for proxy reencryption," in *Proc. IACR Int. Workshop Public Key Cryptogr.*, 2019, pp. 287–316.

[27] T. V. X. Phuong, R. Ning, C. Xin, and Y. Wu, "Puncturable attribute-based encryption for secure data delivery in Internet of Things," in *Proc. IEEE CONF. Comput. Commun.*, 2018, pp. 1511–1519.

[28] P. Dutta, W. Susilo, D. H. Duong, and P. Wang, "Puncturable identity-based encryption from lattices," in *Proc. Australas. Conf. Inf. Secur. Privacy*, 2021, pp. 571–589.

[29] P. Dutta, M. Jiang, D. H. Duong, H. Qian, and W. Susilo, "Hierarchical identity-based puncturable encryption from lattices with application to forward security," in *Proc. 2022 ACM Asia Conf. Comput. Commun. Secur.*, 2022, pp. 408–422.

[30] P. Dutta, W. Susilo, D. H. Duong, A. P. S. Pouriyeh, and H. Choo, "Puncturable identity-based and attribute-based encryption from lattices," *Theor. Comput. Sci.*, vol. 929, pp. 18–38, Oct. 2022.

[31] Q. Mei, M. Yang, J. Chen, X. Wang, and Y. Zhang, "Expressive data sharing and self-controlled fine-grained data deletion in cloud-assisted IoT," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 3, pp. 2625–2640, May 2022.

[32] J. Wei, X. Chen, J. Wang, X. Huang, X. Liu, and Y. Xiang, "Securing fine-grained data sharing and erasure in outsourced storage systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 2, pp. 552–566, Feb. 2022.

[33] H. Deng, H. Yin, Z. Qin, L. Ou, F. Li, and N. Ge, "Toward fine-grained and forward-secure access control in cloud-assisted IoT," *IEEE Internet Things J.*, vol. 11, no. 22, pp. 36569–36580, Nov. 2024.

[34] T. Zhang, M. Jiang, F. Luo, and Y. Guo, "A lattice-based puncturable CP-ABE scheme with forward security for cloud-assisted IoT," *IEEE Internet Things J.*, vol. 12, no. 14, pp. 26538–26554, Jul. 2025.

[35] C. Wang, Y. Ming, H. Liu, T. H. Y. Duan, and K. R. Choo, "Puncturable Registered ABE for Vehicular Social Networks: Enhancing Security and Practicality," *IEEE Trans. Dependable Secure Comput.*, early access, 2025.

[36] T. V. X. Phuong, W. Susilo, J. Kim, D. H. Duong, and G. Yang, "Puncturable proxy re-encryption supporting to group messaging service," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2019, pp. 215–233.

[37] H. Xiong, L. Wang, Z. Zhou, Y. Li, C. Ge, and W. Susilo, "Burn after reading: Adaptively secure puncturable identity-based proxy re-encryption scheme for securing group message," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 11248–11260, Jul. 2021.

[38] H. Liu, Y. Ming, C. Wang, T. H. Y. Duan, and K. R. Choo, "Flexible selective data sharing with fine-grained erasure in VANETs," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 5390–5405, 2024.

[39] Z. Li and G. Shi, "A CCA-secure puncturable attribute-based proxy re-encryption scheme," *IEEE Internet Things J.*, vol. 12, no. 22, pp. 47679–47690, Nov. 2025.

[40] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," *ACM Trans. Comput. Theory*, vol. 6, no. 3, pp. 1–36, Jul. 2014.

[41] W. Susilo, D. H. Duong, H. Q. Le, and P. Dutta, "Puncturable encryption: a generic construction from delegatable fully key-homomorphic encryption," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2020, pp. 107–127.

[42] H. Cui and X. Yi, "Secure Internet of Things in cloud computing via puncturable attribute-based encryption with user revocation," *IEEE Internet Things J.*, vol. 11, no. 2, pp. 3662–3670, Jan. 2023.

[43] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. 40th Annu. ACM Symp. Theory Comput.*, 2008, pp. 197–206.

[44] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, no. 6, pp. 1–40, Sep. 2009.

[45] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, 2009, pp. 333–342.

[46] D. Micciancio and C. Peikert, "Trapdoors for lattices: Simpler, tighter, faster, smaller," in *Proc. Annu. Int. Conf. Theory Appl. Cryptogr. Tech.*, 2012, pp. 700–718.

[47] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert, "Bonsai trees, or how to delegate a lattice basis," *J. Cryptol.*, vol. 25, no. 4, pp. 601–639, Oct. 2012.

[48] S. Agrawal, D. Boneh, and X. Boyen, "Efficient lattice (H)IBE in the standard model," in *Proc. Annu. Int. Conf. Theory Appl. Cryptogr. Tech.*, 2010, pp. 553–572.

[49] V. Shoup, *A computational introduction to number theory and algebra*. Cambridge, U.K.: Cambridge Univ. Press, 2009.