

Large-Universe (Multi-Authority) ABE from LWE*

Pratish Datta¹, Yannis Rouselakis¹, Junichi Tomida¹, and Nikhil Vanjani^{2,3,**}

¹ NTT Research

{pratish.datta,ioannis.rouselakis,junichi.tomida}@ntt-research.com

² Carnegie Mellon University

³ Byzantine Research Inc.

nikhilvanjani61@gmail.com

September 11, 2026

Abstract. An attribute-based encryption (ABE) scheme is “large-universe” if its attribute universe is superpolynomial and is not enumerated during setup. In the multi-authority setting, we further require that each authority can independently manage a superpolynomial set of attributes and dynamically issue an arbitrary polynomial number of secret keys per user. Although large-universe (multi-authority) ABE from pairings is well studied, *explicit* lattice-based constructions have remained elusive. In the centralized setting, a standard workaround is to instantiate lattice-based ABE for general circuits and encode each attribute as a bit string; however, unless one adopts non-standard lattice assumptions, this approach typically yields prohibitively large ciphertexts. In the multi-authority setting, even though lattice-based ABE for general circuits is known, this bit-encoding approach applied to those schemes does not yield a genuine large-universe construction. We close this gap by presenting the *first* lattice-based large-universe (multi-authority) ABE schemes under the Learning With Errors (LWE) assumption, achieving ciphertext and key sizes that are comparable to those in the pairing-based setting. Concretely, we construct:

- a large-universe key-policy ABE scheme with ciphertext size $O(t)$;
- a large-universe ciphertext-policy ABE scheme with ciphertext size $O(|f|)$; and
- a large-universe multi-authority ABE scheme,

where t is the number of attributes, $|f|$ is the policy size, and the $O(\cdot)$ notation suppresses $\tilde{O}(\lambda)$ factors. All schemes support policies in disjunctive normal form (DNF) and are proved secure in the random oracle model. We further develop more efficient variants of our key-policy and ciphertext-policy ABE schemes over ideal lattices under the Ring-LWE assumption, aiming for practical performance on the order of seconds to minutes. Experimental results from our implementations confirm practical runtimes and memory consumption, providing concrete evidence that large-universe lattice-based ABE is feasible for efficient real-world deployment.

* This article is the full version of the article under the same name appearing at ACM CCS 2026.

** Part of this work was conducted while the author was an intern at NTT Research.

Table of Contents

1	Introduction	3
1.1	Our Contributions	4
1.2	Technical Overview	6
2	Preliminaries	10
2.1	Attribute-based Encryption	10
2.2	Multi-Authority Attribute-Based Encryption	10
2.3	Lattices	11
3	Large-Universe ABE for Subset Policies	12
3.1	KP-ABE	13
3.2	CP-ABE	16
4	Large-Universe MA-ABE for Subset Policies	21
5	Ring Instantiations	26
5.1	Preliminaries on Ideal Lattices	26
5.2	Ring Generalized Related-Trapdoor LWE	26
5.3	Our KP-ABE from Ring-LWE	27
5.4	Our CP-ABE from Ring-LWE	30
6	Implementation	36
6.1	System Parameters	36
6.2	Environment and Benchmark Configuration	37
6.3	Conclusion	37
7	Open Science	38
	Acknowledgements	38
	References	38
A	Our CP-ABE and MA-ABE with Shorter Secret Keys	43
A.1	CP-ABE	43
A.2	MA-ABE	47
B	Ring-RTLWE and Ring-LHL with Auxiliary Information	52
B.1	Preliminaries	52
B.2	Ring-LHL with Auxiliary Linear Information	54
B.3	Ring Preimage Sampling	57
B.4	Ring Gadget Matrix and Ring Trapdoors	57
B.5	Ring Generalized Related-Trapdoor LWE Assumption	61
B.6	Reduction of Ring-RTLWE to Ring-LWE	62

1 Introduction

Attribute-based encryption. Attribute-based encryption (ABE) generalizes public-key encryption to enable fine-grained access control over encrypted data. An ABE scheme is parameterized by a predicate $P : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$: an encryptor generates a ciphertext for any $x \in \mathcal{X}$, while a central authority holding the master secret key derives secret keys for any $y \in \mathcal{Y}$. Decryption succeeds on (x, y) if and only if $P(x, y) = 1$.

Typically, one of x and y is an attribute set $X \subseteq \mathcal{U}$ (e.g., `Title:Professor, Department:Engineering`) over an attribute universe $\mathcal{U}$, and the other is a Boolean policy f over $\mathcal{U}$ (e.g., `Title:Professor AND Department:Economics`). Formally, each input wire i of f is labeled by an attribute $u_i \in \mathcal{U}$. To evaluate $f(X)$ we set input i to 1 iff $u_i \in X$, and define $P(X, f) = 1$ (equivalently, $P(f, X) = 1$) iff $f(X) = 1$. If the policy appears in the key (i.e., $y = f$) we obtain key-policy (KP)-ABE. If it appears in the ciphertext (i.e., $x = f$) we obtain ciphertext-policy (CP)-ABE.

Since its inception in [SW05, GPSW06], the design of robust and practical ABE schemes has been an active and enduring research direction, yielding a long line of works that navigate fundamental trade-offs among expressiveness, efficiency, security notions, and underlying assumptions [BSW07, OSW07, Wat09, LOS⁺10, LW10, OT10, AFV11, LW11b, Wat11, LW12, OT12, Wat12, Boy13, GGH⁺13, GVV13, Att14, BGG⁺14, Wee14, CGW15, Att16, BV16, ABGW17, GKW17, CGKW18, Att19, AMY19, GWW19, KW19, Tsa19, KW20, AY20a, AY20b, BV20, GW20, LL20b, LL20a, GW20, AT20, Wee21a, Wee21b, Wee22, RW22, AYY22, LLL22, CW23, JLL23, HLL23, ARYY23, CW24, Wee24, DHM⁺24, AKY24, HLL24, WW24, CW25a, Wee25]. In parallel, ABE has matured into a deployable access-control primitive, with adoption in practical security systems, notably for cloud environments and the Internet of Things (IoT). A prominent real-world deployment is Cloudflare’s Portunus, an ABE-based access-control system used at global scale to terminate TLS connections [LVV⁺23]. Reflecting this growing relevance, standardization bodies such as NIST and ETSI have initiated efforts toward formal ABE standardization.

Multi-Authority Attribute-Based Encryption. To eliminate the centralized trust inherent in standard ABE—a major obstacle to scalable real-world deployment—Chase [Cha07] introduced multi-authority ABE (MA-ABE), in which multiple independent authorities manage disjoint sets of attributes and issue user keys for the attributes they control, without requiring mutual coordination. Concretely, an attribute can be modeled as a pair (α, u) , where $\alpha \in \mathcal{AU}$ is an authority identifier (for an identifier space $\mathcal{AU}$) and $u \in \mathcal{U}$ is an attribute from an attribute universe $\mathcal{U}$. A user holding an attribute set $\{(\alpha_i, u_i)\}_i$ obtains secret keys from the corresponding authorities $\{\alpha_i\}_i$ and can decrypt any ciphertext whose access policy is satisfied by $\{\alpha_i, u_i\}_i$. The *decentralized* variant of MA-ABE, first introduced by Lewko and Waters [LW11a] and subsequently studied extensively [OT13, OT20, AG21, DKW21, WWW22, Ven23, DKW23a, DKW23b, CCG⁺23, dIPVA23, GGL24, RVV24, DTV25, CLW25], further increases flexibility: any party may become an authority by publishing a public key and issuing user keys for the attributes it controls, with no global coordination beyond an initial trusted global parameter generation. Authorities operate independently, need not even be aware of one another, and may join at any time; in particular, there is no a priori bound on the number of authorities over the system’s lifetime.

Large-Universe (MA-)ABE. A central design dimension in ABE is the size of the attribute universe $\mathcal{U}$. The literature typically distinguishes two regimes [GPSW06]. In a “small-universe” scheme, $|\mathcal{U}|$ is polynomially bounded, the public parameters typically scale with $|\mathcal{U}|$, and the universe $\mathcal{U}$ is fixed and explicitly enumerated at setup. In contrast, a “large-universe” scheme supports a super-polynomial attribute universe while keeping the public parameters independent of $|\mathcal{U}|$; equivalently, essentially any string can serve as an attribute and be introduced dynamically over time, without requiring global re-setup. In the setting of large-universe (decentralized) MA-ABE, we further require that each authority can (independently) manage a super-polynomial number of attributes assigned over time, with a public key whose size remains independent of the number of attributes it governs, and that it can dynamically issue an arbitrary polynomial number of secret keys to each user as application demand evolves.⁴ [RW15]

This flexibility is crucial in practice, especially for highly dynamic and large-scale infrastructures. Cloudflare’s Portunus, an ABE-based access-control system deployed across more than 400 global data centers, identifies flexible and dynamic attributes as an explicit requirement because customer needs

⁴ Indeed, since decentralized MA-ABE admits an arbitrary number of attribute authorities, one could trivially obtain a super-polynomial universe by introducing a super-polynomial number of authorities. The true challenge lies in ensuring that each individual authority can itself handle a super-polynomial domain of attributes without inflating its key sizes.

evolve over time, for example when new compliance standards arise, and accordingly uses a scheme in which any string can serve as an attribute [LVV⁺23]. Further, a large universe allows new locations, roles, resources, and compliance labels to be introduced without enumerating them at setup or updating global parameters. This avoids brittle mappings to pre-allocated indices and supports long-lived deployments [LVV⁺23].

The State-of-the-Art from LWE. Boolean circuits are a natural abstraction for ABE access policies in practice because real-world authorization is inherently compositional “if-then” logic over attributes (AND/OR/NOT, with nesting and exceptions). They capture this structure directly, while remaining expressive enough to model both simple role-based checks and intricate compliance rules. In the pairing setting, large-universe ABE for expressive Boolean circuits is well developed (e.g., [GPSW06, OSW07, OT10, Att14, AC17, Att19, TKN20, AT20, RW22]) and MA-ABE is likewise well studied (e.g., [RW15, RVV24]).⁵ In contrast, to the best of our knowledge, there are no works that explicitly consider large-universe ABE for boolean access policies from lattices.

In principle, in the KP-ABE setting, the large-universe property can be obtained from existing lattice-based KP-ABE schemes for general Boolean/arithmetic circuits under the LWE assumption (e.g., [GVW13, BGG⁺14, BV16, CW23, JLL23, CW24]), by applying standard arithmetization techniques [Tha22] where needed and encoding each attribute as a bit string (viewed as field elements 0 and 1). Concretely, for an attribute universe $\mathcal{U} = \{0, 1\}^\ell \setminus \{0^\ell\}$, evaluating a circuit f with t inputs over $\mathcal{U}$ can be reduced to evaluating a derived circuit f' over $t\ell$ input bits representing an attribute set of size at most t . However, this reduction inflates ciphertext size to $O(t\ell)$ (since ciphertext size in these constructions scales as $O(v)$ for v -bit inputs), whereas pairing-based large-universe ABE achieves essentially $O(t)$ -size ciphertexts.⁶ Even for unbounded schemes [CW24, BV16], simply enumerating an exponentially large universe is not a viable route to large-universe ABE: after arithmetization, decryption must account for attributes referenced by the access policy but absent from the ciphertext’s attribute set, which requires providing ciphertext components corresponding to the “zero” value for those attributes.

For CP-ABE, the analogous route via the LWE-based NC¹ CP-ABE of [DKW21] incurs a blow-up of the input size of the converted f' to $O(2^{\bar{\delta}})$ in the depth δ of f' , yielding $O(\lambda^2 2^{\bar{\delta}})$ -size ciphertexts⁷, where $\bar{\delta}$ is an upperbound of δ . Alternative LWE-based routes via inner-product encryption (IPE) [KSW08, AFV11, Xag13] only yield large-universe ABE for conjunctions and with ciphertexts still scaling as $O(\bar{t}, \text{poly}(\ell), |f|)$ where $\bar{t}$ and $|f|$ are the upper bound on t and $|f|$, respectively.

The state-of-the-art is even worse for lattice-based large-universe MA-ABE. Existing lattice-based MA-ABE constructions cover DNF under LWE [DKW21] and evasive LWE [WWW22] while general circuits under evasive and tensor LWE [CLW24], but all of these are small-universe constructions. Moreover, the aforementioned bit-string encoding route to large universes is inapplicable here: DNF policies do not support the membership testing, and the circuit-based MA-ABE of [CLW24] is defined in a way that fundamentally conflicts with the large-universe MA-ABE setting. Specifically, the correctness and security definitions in [CLW24] restrict any authority to providing at most a fixed number of keys per user. While Garg, Goyal, and Lu give MA-ABE for circuits from public-key encryption, their scheme tolerates only bounded collusion [GGL24]. Consequently, large-universe MA-ABE schemes from lattices remain completely unknown.

In view of the state of the art, the following fundamental questions naturally arise:

Open Problem 1. *Can we construct a large-universe KP-ABE (resp. CP-ABE) scheme for some class of boolean access-policies with $O(t)$ -size (resp. $O(|f|)$ -size) ciphertexts from plain LWE?*

Open Problem 2. *Can we construct a large-universe MA-ABE scheme for some class of boolean access-policies from lattices (ideally from plain LWE) with $O(|f|)$ -size ciphertexts?*

1.1 Our Contributions

We answer the questions above in the affirmative. In particular, we construct large-universe KP-ABE, CP-ABE, and MA-ABE schemes for subset policies secure under the LWE assumption in the random

⁵ While Okamoto–Takashima MA-ABE [OT20] supports an exponentially large per-authority universe, it restricts each authority to issuing only a bounded number of keys to any fixed user, which severely constrains scalability and curtails the flexibility typically afforded by the large-universe MA-ABE setting.

⁶ Here and in what follows, we omit $\text{poly}(d) \cdot \tilde{O}(\lambda)$ factors in ciphertext sizes where d is the maximum depth of f , λ is a security parameter, $\tilde{O}(\cdot)$ hides polylogarithmic factors. Roughly speaking, the size of the term $\mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top$ is $\text{poly}(d) \cdot \tilde{O}(\lambda)$ in most lattice-based ABE schemes.

⁷ The factor λ^2 comes from the fact that they used an exponential modulus in their scheme.

oracle model. Note that large-universe (MA-)ABE for subset policies can be lifted to large-universe (MA-)ABE for DNF [WWW22]. DNF policies suffice for most real-life use cases. Cloudflare’s Portunus also demonstrates the practical relevance of DNF: its example policy `country: US or region: EU` is a disjunction of two singleton conjunctions, and the deployment reports that most policies are combinations of geographic properties [LVV⁺23].

As in prior lattice-based constructions, our KP-ABE and CP-ABE schemes achieve selective security [GPSW06], while our MA-ABE scheme satisfies static security [RW15].

As an additional benefit, all of our schemes rely on LWE with a polynomial modulus-to-noise ratio and are “unbounded” in the sense that there is no a priori bound on the cardinality of attribute sets. By contrast, the maximum policy size must be fixed at setup; however, assuming LWE with a superpolynomial modulus-to-noise ratio, this bound can be removed for our KP-ABE scheme (removing it for our CP-ABE and MA-ABE schemes remains unclear, even under this stronger assumption). For CP-ABE and MA-ABE, we also present variants with shorter secret keys; in exchange, their ciphertext sizes become $O(|f|)$ rather than $O(|f|)$.

We compare our schemes in Table 1 with prior large-universe ABE schemes from LWE obtained via transformations of lattice-based ABE for general circuits, as discussed above. In this upgrade, there is no bound on the number of conjunctions in the DNF; the only policy bound is the maximum size of each conjunction. Moreover, if the secret-key size (resp. ciphertext size) of KP-ABE (resp. CP or MA-ABE) for subset policies is $O(|f|)$, then the corresponding size in the converted DNF scheme remains $O(|f|)$, and the sizes of the other elements are unchanged. Note that since all MA-ABE schemes based on LWE require the random oracle model [DKW21, WWW22], its use is not an additional cost for the proposed MA-ABE. Another appealing feature of our constructions is that they avoid the expensive homomorphic evaluation mechanisms used in most prior lattice-based schemes [GVW13, BGG⁺14, BV16, Wee22, CW23, JLL23, HLL23, CW24, Wee24, HLL24, Wee25]. This is an important step toward practical deployment.

Beyond plain LWE, the bit-string transformation above can also be applied to circuit ABE from newer lattice assumptions. Evasive-LWE-based instantiations include the KP-ABE of [HLL23] and the CP-ABE schemes of [Wee22, HLL24]; however, we omit them from our quantitative comparison because the security of the relevant evasive-LWE formulations remains unsettled in light of recent attacks and counterexamples [BÜW24, DJM⁺25, AMYY25, HJL25, HHY25]. ABE for uniform computation models, including automata, RAMs, and Turing machines [AMR25, AMY19, Wee21a, JLL23, AKY24, DHM⁺24, CW25b], is likewise outside our scope, since it is unclear how to formulate a comparable large-universe notion in these models. This also excludes the decomposed-LWE KP-ABE for RAM programs of [AMR25]. The closest directly comparable baselines are therefore the succinct-LWE constructions of [Wee24, Wee25]. Let $v := \bar{\ell}\ell$ denote the bit-input length after the transformation. The KP-ABE of [Wee24] achieves $|\text{mpk}| = O(v^{2\alpha} + v^{1-\alpha})$, $|\text{ct}| = O(v^{1-\alpha})$, and $|\text{sk}| = O(1)$ for $1/3 \leq \alpha \leq 1$, so its $O(1)$ -ciphertext endpoint has $|\text{mpk}| = O(v^2)$. [Wee25] gives both KP-ABE and CP-ABE with $|\text{mpk}| = |\text{ct}| = |\text{sk}| = O(1)$, where these bounds suppress $\text{poly}(d, \lambda)$ factors. Despite these communication gains, encryption must still process the v -bit representation or the corresponding membership circuitry, incurring at least an ℓ -factor overhead relative to our direct per-attribute constructions. Moreover, the stated general-circuit parameterizations invoke succinct LWE with a subexponential modulus-to-noise ratio and choose a superpolynomial, depth-dependent modulus.

Nevertheless, a major bottleneck remains: in these schemes, only one bit can be encrypted at a time, leading to substantial communication overhead. Motivated by deployability, we therefore turn to ideal-lattice techniques. Over the last decade, lattice-based cryptography has advanced significantly, driven both by substantial efficiency gains from algebraic structure [LPR10] and by the versatility of ideal-lattice problems, notably ring and module LWE [LPR10, LPR13, LS15]. Indeed, prominent post-quantum encryption [ABD⁺21] and signature schemes [SBKL⁺20] selected by NIST for standardization are based on ideal-lattice assumptions. Harnessing this efficiency, we instantiate variants of our KP-ABE and CP-ABE schemes over ideal lattices under the Ring-LWE assumption, yielding the *first* ABE schemes with plausible post-quantum security guarantees that are actually deployable in practical systems.

The resulting ideal-lattice translations provide a substantial concrete improvement over their integer-lattice counterparts. In both the KP-ABE and CP-ABE settings, the Ring-LWE instantiations replace ciphertexts that encrypt only a single bit with ciphertexts that encrypt an entire ring element, corresponding to n message bits. After normalizing by the encrypted payload size, the master public keys and ciphertexts in the Ring-LWE constructions are approximately n times shorter, while the secret keys are of comparable size, up to lower-order terms and under the corresponding parameter choices. These savings are especially significant for deployment, where ciphertext expansion and key size are among the main obstacles to practical lattice-based ABE.

Table 1. Comparison with prior lattice-based large-universe ABE from LWE. For sizes of $\text{mpk}, \text{ct}, \text{sk}$, t denotes the cardinality of the attribute set, ℓ denotes $\log_2 |\mathcal{U}|$ where $\mathcal{U}$ is the attribute universe, $|f|$ denotes the size of the policy f , $|f'|$ denotes the size of f' obtained from f via the naive conversion to a large-universe scheme as described in the introduction, and δ denotes the depth of f' . For each parameter p , $\bar{p}$ denotes the bound (or maximum value) of p . $O(\cdot)$ hides $\tilde{O}(\lambda^2)$ factors in $|\text{mpk}|$ and $\tilde{O}(\lambda)$ factors in $|\text{ct}|, |\text{sk}|$. $O(\cdot)^\dagger$ hides $\text{poly}(d) \cdot \tilde{O}(\lambda^2)$ factors in $|\text{mpk}|$ and $\text{poly}(d) \cdot \tilde{O}(\lambda)$ factors in $|\text{ct}|, |\text{sk}|$ where d is the maximum depth of f . SP and circ. stand for subset policies and to circuits, respectively. The secret key sizes of MA-ABE refer to the total size of secret keys tied together with the same GID.

Reference	$ \text{mpk} $	$ \text{ct} $	$ \text{sk} $	Class	Notes
KP-ABE					
AFV11 [AFV11]	$O(\bar{\ell}\ell^3 f)$	$O(\bar{\ell}\ell^3 f)$	$O(1)$	SP	from IPE
X13 [Xag13]	$O(\bar{\ell} f)$	$O(\bar{\ell} f)$	$O(1)$	SP	from IPE
GVW13 [GVW13]	$O(\bar{\ell}\ell)^\dagger$	$O(\bar{\ell}\ell)^\dagger$	$O(f')^\dagger$	circ.	
BGGHNS14 [BGG ⁺ 14]	$O(\bar{\ell}\ell)^\dagger$	$O(\bar{\ell}\ell)^\dagger$	$O(1)^\dagger$	circ.	
BV16 [BV16]	$O(1)^\dagger$	$O(\bar{\ell}\ell)^\dagger$	$\bar{\ell}\ell + O(1)^\dagger$	circ.	
CW23 [CW23]	$O(\bar{\ell}\ell)^\dagger$	$O(\bar{\ell}\ell)^\dagger$	$O(\lambda)$	circ.	
CW24 [CW24]	$O(1)^\dagger$	$O(\bar{\ell}\ell)^\dagger$	$O(\lambda\bar{\ell}\ell)^\dagger$	circ.	
This work (§3.1)	$O(1)$	$O(t)$	$O(f)$	SP	ROM
CP-ABE					
AFV11 [AFV11]	$O(\bar{\ell}\ell^3 f)$	$O(\bar{\ell}\ell^3 f)$	$O(1)$	SP	from IPE
X13 [Xag13]	$O(\bar{\ell} f)$	$O(\bar{\ell} f)$	$O(1)$	SP	from IPE
Univ. & [BGG ⁺ 14]	$O(f')^\dagger$	$O(f')^\dagger$	$O(1)^\dagger$	circ.	KP-ABE
DKW21 [DKW21]	$O(\lambda^2 2^\delta)$	$O(\lambda^2 2^\delta)$	$O(\lambda 2^\delta)$	NC ¹	
This work (§3.2)	$O(1)$	$O(f)$	$O((t + f) f)$	SP	ROM
This work (§A.1)	$O(1)$	$O(f)$	$O(t + f)$	SP	ROM
MA-ABE					
This work (§4)	$O(1)$	$O(f)$	$O(t f)$	SP	ROM
This work (§A.2)	$O(1)$	$O(f)$	$O(t)$	SP	ROM

We further implemented the Ring-LWE instantiations of our KP-ABE and CP-ABE constructions using the NTK library (“NTT Intelligent Encryption Toolkit”). This is a highly optimized proprietary C/C++ library, developed specifically for functional encryption and applicable across a broad range of use cases. As a result, the underlying library already supported several auxiliary components required by the implementation, including attribute and policy parsing, DNF satisfiability checking, secret sharing, hybrid encryption, and key serialization/storage.

Our implementation is driven by efficiency, practical deployability, and constant-time execution with respect to secret-dependent data. To this end, we leverage the Number Theoretic Transform (NTT) for efficient arithmetic, optimize the Gaussian sampling and preimage generation procedures, and systematically explore parameter configurations to balance performance, correctness, and security.

1.2 Technical Overview

In this section, we explain the high-level ideas behind our ABE constructions. For simplicity, we present the schemes over integer lattices first, and close the section by describing how they are transported to ideal lattices and which ingredients have to be reworked along the way. In what follows, we refer to ABE schemes that satisfy the properties described in Open Problem 1 as a large-universe scheme. First, we consider the reason that while we can easily convert a small universe scheme into a large-universe scheme using a random oracle in pairing-based schemes [BSW07, RW22], it seems difficult to apply the random oracle methodology to lattice-based schemes. Informally, in pairing-based small universe schemes, a master public key contains a group element g^{w_i} for each attribute $u_i \in \mathcal{U}$, where w_i is a random exponent. The possession of attribute u_i is expressed by the existence of the term g^{sw_i} in a ciphertext or secret key, where s is a random element chosen by encryption or key generation algorithm. Then, we can expand $\mathcal{U}$ to the set of any strings by generating g^{w_i} with a hash function modeled as a random oracle on the fly. The point is that non-possession of attribute u_i can be expressed by the absence of the term g^{sw_i} , and the number of non-possessed attributes does not matter.

On the other hand, most lattice-based schemes require a ciphertext or secret key to have some element for both possessed and non-possessed attributes, e.g., [GVW13, BGG⁺14, BV16, DKW21]. For instance,

a ciphertext has the term $\mathbf{s}^\top(\mathbf{B} + \mathbf{G})$ for positive attributes while it has $\mathbf{s}^\top\mathbf{B}$ for negative attributes in the scheme by [BGG⁺14]. The problem is that if the universe size is superpolynomial, at least one of the numbers of positive and negative attributes becomes also superpolynomial, and this makes encryption or key generation inefficient.

Our KP-ABE. Given the above observation, we can hope to apply the random oracle methodology to a lattice-based KP-ABE scheme where a ciphertext does not contain elements for negative attributes. Fortunately, we have such a KP-ABE scheme by Boyen [Boy13] for NC¹, the original scheme of which turned out to be insecure unless policies are restricted to DNF, and fixed later [ABN⁺22, DKW21]. Note that when we require the property that a ciphertext does not contain elements for negative attributes, we cannot use the NC¹ computation in his scheme and need to stick to DNF. His scheme can be described as follows: for attribute universe $\mathcal{U} = [L]$, a message $\mu \in \{0, 1\}$ with attribute set $X = \{x_1, \dots, x_t\} \subseteq [L]$, and DNF policy f over $\mathcal{U}$, we compute

$$\begin{aligned} \text{mpk} &= (\mathbf{A}_1, \dots, \mathbf{A}_L, \mathbf{B}, \mathbf{p}), \quad \text{msk} = \mathbf{T}_{\mathbf{A}_1}, \dots, \mathbf{T}_{\mathbf{A}_L}, \quad \text{sk} = \mathbf{K}_f \\ \text{ct} &= \mathbf{s}^\top [\underbrace{\mathbf{A}_{x_1} | \dots | \mathbf{A}_{x_t}}_{\text{wavy lines}} | \mathbf{B} | \mathbf{p}] + (\mathbf{0}, \mu \lfloor q/2 \rfloor) \end{aligned}$$

where $\mathbf{A}_i \in \mathbb{Z}_q^{n \times m}$ is statistically close to uniform, $\mathbf{T}_{\mathbf{A}_i}$ is a trapdoor of $\mathbf{A}_i$, $\mathbf{B} \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{s}, \mathbf{p} \leftarrow \mathbb{Z}_q^n$, and the expressions with wavy lines hide noise terms. $\mathbf{K}_f$ is a trapdoor of

$$\mathbf{M} = \begin{bmatrix} \mathbf{A}_1 & f_{1,0}\mathbf{B} & f_{1,1}\mathbf{Z}_1 & \dots & f_{1,\theta}\mathbf{Z}_1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_L & f_{L,0}\mathbf{B} & f_{L,1}\mathbf{Z}_1 & \dots & f_{L,\theta}\mathbf{Z}_1 \end{bmatrix}$$

where $\mathbf{Z} \leftarrow \mathbb{Z}_q^{n \times m}$, and $\mathbf{F} = (f_{i,j})_{i \in [L], j \in [0, \theta]}$ is a share generating matrix of policy f , i.e., if and only if f accepts X , the rows of $\mathbf{F}$ corresponding to the elements in X span $(1, 0, \dots, 0)$. We omit how decryption works since it is not important here.

First, we use the fact that KP-ABE for DNF can be easily converted from KP-ABE for subset policies. The subset policies for KP-ABE refer to the predicate $P : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ where $\mathcal{X} = \mathcal{Y} = [L]$ and $P(X, Y) = 1 \Leftrightarrow X \supseteq Y$ (i.e., $Y = \{y_1, \dots, y_u\}$ represents the conjunction $\bigwedge_{i \in [u]} y_i \in X$). By restricting policies to subset policies, we can simplify his scheme as follows:

$$\begin{aligned} \text{mpk} &= (\mathbf{A}_1, \dots, \mathbf{A}_L, \mathbf{p}), \quad \text{msk} = \mathbf{T}_{\mathbf{A}_1}, \dots, \mathbf{T}_{\mathbf{A}_L}, \quad \text{sk} = \{\mathbf{k}_y\}_{y \in Y} \\ \text{ct} &= \mathbf{s}^\top [\underbrace{\mathbf{A}_{x_1} | \dots | \mathbf{A}_{x_t}}_{\text{wavy lines}} | \mathbf{p}] + (\mathbf{0}, \mu \lfloor q/2 \rfloor) \end{aligned}$$

where $\mathbf{p}_y \leftarrow \mathbb{Z}_q^n$ conditioned on $\sum_{y \in Y} \mathbf{p}_y = \mathbf{p}$ for $y \in Y$, and $\mathbf{k}_y$ is sampled from a small discrete Gaussian $D_{\mathbb{Z}}^m$ conditioned on $\mathbf{A}_y \mathbf{k}_y = \mathbf{p}_y$ using $\mathbf{T}_{\mathbf{A}_y}$ (we denote this operation by $\mathbf{k}_y \leftarrow \mathbf{A}_y^{-1}(\mathbf{p}_y)$ and omit the Gaussian parameter in this overview). Correctness follows from the fact that $\sum_{y \in Y} \mathbf{s}^\top \mathbf{A}_y \mathbf{k}_y = \mathbf{s}^\top \mathbf{p}$ since $\mathbf{k}_y$ is small.

The security proof can be also simplified as follows, which is inspired by the security proof of the CP-ABE scheme by Waters, Wee, and Wu [WWW22]. In the proof, we use the following fact that

$$(\mathbf{A}, \mathbf{A}^{-1}(\mathbf{p}), \mathbf{p}) \approx_s (\mathbf{A}, \mathbf{k}, \mathbf{A}\mathbf{k}) \quad (1)$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{p} \leftarrow \mathbb{Z}_q^n$, $\mathbf{k} \leftarrow D_{\mathbb{Z}}^m$. We consider a hybrid where the challenger samples $\mathbf{A}_{x_1^*}, \dots, \mathbf{A}_{x_t^*} \leftarrow \mathbb{Z}_q^{n \times m}$ while it generates the rest of $\mathbf{A}_i$ with its trapdoor, where $X = (x_1^*, \dots, x_t^*)$ is the challenge attribute. For each key generation for Y , the challenger samples $\mathbf{k}_y \leftarrow D_{\mathbb{Z}}^m$ and computes $\mathbf{p}_y = \mathbf{A}_y \mathbf{k}_y$ for $y \in X \cap Y$ and uniformly samples $\mathbf{p}_y$ from $\mathbb{Z}_q^n$ for $y \in Y \setminus X$ conditioned on $\sum_{y \in Y} \mathbf{p}_y = \mathbf{p}$. Finally, it samples $\mathbf{k}_y \leftarrow \mathbf{A}_y^{-1}(\mathbf{p}_y)$ for $y \in Y \setminus X$ with $\mathbf{T}_{\mathbf{A}_y}$.

The point is that for all key queries, we have $X \cap Y \neq Y$, which follows from the game condition that the adversary cannot ask for secret keys for Y such that $X \supseteq Y$. Hence, in the original game each $\mathbf{p}_y$ for $y \in X \cap Y$ is independently and uniformly distributed in $\mathbb{Z}_q^n$ (i.e., the marginal distribution of $\{\mathbf{p}_y\}_{y \in X \cap Y}$ is uniformly random), and Eq. (1) implies the statistical indistinguishability between the original game and the hybrid. Then, we can apply LWE to $\mathbf{s}^\top [\underbrace{\mathbf{A}_{x_1^*} | \dots | \mathbf{A}_{x_t^*}}_{\text{wavy lines}} | \mathbf{p}]$ in ct to completely remove the information of the challenge bit.

Following the pairing-based ABE, what we want to do here is to generate $\mathbf{A}_i$ with a hash function modeled as a random oracle to make $\mathcal{U}$ be of superpolynomial size. However, it immediately turns out

that this idea does not work since the authority cannot generate a trapdoor for matrices $\mathbf{A}_i$. Our idea to solve this problem is to modify the scheme as follows:

$$\begin{aligned} \text{mpk} &= (\mathbf{A}, \mathbf{D}_1, \dots, \mathbf{D}_L, \mathbf{p}), \quad \text{msk} = \mathbf{T}_{\mathbf{A}}, \quad \text{sk} = \{\mathbf{k}_y\}_{y \in Y} \\ \text{ct} &= \underbrace{\mathbf{s}^\top [\mathbf{A}|\mathbf{D}_{x_1}| \dots |\mathbf{D}_{x_t}|\mathbf{p}]} + (\mathbf{0}, \mu \lfloor q/2 \rfloor) \end{aligned}$$

where $\mathbf{D}_i \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{T}_{\mathbf{A}}$ is a trapdoor of $\mathbf{A}$, $\mathbf{s}, \mathbf{p} \leftarrow \mathbb{Z}_q^n$, $\mathbf{p}_y \leftarrow \mathbb{Z}_q^n$ conditioned on $\sum_{y \in Y} \mathbf{p}_y = \mathbf{p}$ for $y \in Y$, and $\mathbf{k}_y \leftarrow [\mathbf{A}|\mathbf{D}_y]^{-1}(\mathbf{p}_y)$. Using the trapdoor delegation [CHKP10], we can sample $\mathbf{k}_y$ with the trapdoor of $\mathbf{A}$. The modified scheme does not need trapdoors for $\mathbf{D}_i$, and thus we can generate it by a hash function as $\mathbf{D}_i = H(i)$.

The security proof in the random oracle model is similar to the previous scheme. In the hybrid, we sample $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and program the random oracle as $H(i) = \mathbf{D}_i$, where $\mathbf{D}_i \leftarrow \mathbb{Z}_q^{n \times m}$ for $i \in X$, and $\mathbf{D}_i$ is generated with a trapdoor for $i \notin X$. A secret key $\{\mathbf{k}_y\}_{y \in Y}$ for Y is generated as follows:

1. sample $\mathbf{k}_y \leftarrow D_{\mathbb{Z}}^m$ and computes $\mathbf{p}_y = \mathbf{A}_y \mathbf{k}_y$ for $y \in X \cap Y$;
2. sample $\mathbf{p}_y$ for $y \in Y \setminus X$ uniformly from $\mathbb{Z}_q^n$ conditioned on $\sum_{y \in Y} \mathbf{p}_y = \mathbf{p}$; and
3. sample $\mathbf{k}_y \leftarrow [\mathbf{A}|\mathbf{D}_y]^{-1}(\mathbf{p}_y)$ for $y \in Y \setminus X$ relying on the trapdoors for $\mathbf{D}_y$.

The indistinguishability between the original game and the hybrids follows similarly to the previous scheme. In particular, we use the modified version of Eq. (1) for the change in item (1):

$$(\mathbf{A}, \mathbf{D}, [\mathbf{A}|\mathbf{D}]^{-1}(\mathbf{p}), \mathbf{p}) \approx_s (\mathbf{A}, \mathbf{D}, \mathbf{k}, [\mathbf{A}|\mathbf{D}]\mathbf{k})$$

where $\mathbf{A}, \mathbf{D} \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{p} \leftarrow \mathbb{Z}_q^n$, $\mathbf{k} \leftarrow D_{\mathbb{Z}}^{2m}$, and the fact that the distribution $\mathbf{k}_y \leftarrow [\mathbf{A}|\mathbf{D}_y]^{-1}(\mathbf{p}_y)$ relying on the trapdoor of $\mathbf{A}$ and that relying on the trapdoor of $\mathbf{D}_y$ are statistically close for the change in item (3). Then, we can apply LWE to $\underbrace{\mathbf{s}^\top [\mathbf{A}|\mathbf{D}_{x_1}^*| \dots |\mathbf{D}_{x_t}^*|\mathbf{p}]}_{\text{LWE}}$ in the challenge ciphertext.

Our CP-ABE and MA-ABE. Similar to the case of KP-ABE, our starting point is CP and MA-ABE schemes where secret keys do not contain elements for negative attributes. Waters, Wee, and Wu presented CP-ABE (informally) and MA-ABE schemes for subset policies that satisfy the property [WWW22]. Recall that the predicate P for CP-ABE for subset policies is defined as $P(X, Y) = 1 \Leftrightarrow X \subseteq Y$. Their CP-ABE scheme is described as follows: for encryption of a message $\mu \in \{0, 1\}$ with $X = \{x_1, \dots, x_t\}$ and key generation for $Y = \{y_1, \dots, y_u\}$, we compute

$$\begin{aligned} \text{mpk} &= \{\mathbf{A}_i, \mathbf{B}_i, \mathbf{p}_i\}_{i \in [L]}, \text{msk} = (\mathbf{T}_{\mathbf{A}_1}, \dots, \mathbf{T}_{\mathbf{A}_L}), \text{sk} = (\mathbf{r}, \{\mathbf{k}_y\}_{y \in Y}) \\ \text{ct} &= \underbrace{\mathbf{s}^\top \left[\mathbf{A}_{x_1} | \dots | \mathbf{A}_{x_t} | \sum_{x \in X} \mathbf{B}_x | \sum_{x \in X} \mathbf{p}_x \right]} + (\mathbf{0}, \mu \lfloor q/2 \rfloor) \end{aligned} \quad (2)$$

where $\mathbf{A}_i \in \mathbb{Z}_q^{n \times m}$ is statistically close to uniform, $\mathbf{T}_{\mathbf{A}_i}$ is a trapdoor of $\mathbf{A}_i$, $\mathbf{B}_i \leftarrow \mathbb{Z}_q^{n \times m(2L-1)}$, $\mathbf{s}, \mathbf{p}_i \leftarrow \mathbb{Z}_q^n$, $\mathbf{r} \leftarrow D_{\mathbb{Z}}^{m(2L-1)}$ and $\mathbf{k}_y \leftarrow \mathbf{A}_y^{-1}(\mathbf{B}_y \mathbf{r} + \mathbf{p}_y)$. If $X \subseteq Y$, decryption works since

$$\underbrace{\sum_{x \in X} \mathbf{s}^\top \mathbf{A}_x \mathbf{k}_x}_{\text{LWE}} - \underbrace{\mathbf{s}^\top \sum_{x \in X} \mathbf{B}_x \mathbf{r}}_{\text{LWE}} = \underbrace{\mathbf{s}^\top \sum_{x \in X} \mathbf{p}_x}_{\text{LWE}}$$

It is obvious that we cannot generate $\mathbf{A}_i, \mathbf{B}_i, \mathbf{p}_i$ by a hash function because the trapdoor for $\mathbf{A}_i$ cannot be generated. We can circumvent the issue in the same manner as our KP-ABE scheme. The modified scheme is described as follows:

$$\begin{aligned} \text{mpk} &= (\mathbf{A}, \{\mathbf{D}_i, \mathbf{B}_i, \mathbf{p}_i\}_{i \in [L]}), \quad \text{msk} = \mathbf{T}_{\mathbf{A}}, \quad \text{sk} = (\mathbf{r}, \{\mathbf{k}_y\}_{y \in Y}) \\ \text{ct} &= \underbrace{\mathbf{s}^\top \left[\mathbf{A}|\mathbf{D}_{x_1}| \dots |\mathbf{D}_{x_t}| \sum_{x \in X} \mathbf{B}_x | \sum_{x \in X} \mathbf{p}_x \right]} + (\mathbf{0}, \mu \lfloor q/2 \rfloor) \end{aligned}$$

where $\mathbf{k}_y \leftarrow [\mathbf{A}|\mathbf{D}_y]^{-1}(\mathbf{B}_y \mathbf{r} + \mathbf{p}_y)$. Then, we can generate $(\mathbf{D}_i, \mathbf{B}_i, \mathbf{p}_i)$ by a hash function.

Another issue is that the width of $\mathbf{B}_i$ and the length of $\mathbf{r}$ depend on the universe size L . When we make the universe size be superpolynomial, L seems to also become superpolynomial. In fact, this is not the case, and it is sufficient if the width of $\mathbf{B}_i$ and the length of $\mathbf{r}$ exceeds the cardinality of X . Hence,

the security proof goes well if we bound the cardinality of X at most L . This is basically the variant of our CP-ABE scheme described in [Section A.1](#).

However, the ciphertext size of the above scheme is $O(|f|)$ since the width of $\mathbf{B}_i$ is the bound L of the size of f , while our goal is to achieve $O(|f|)$ -size ciphertexts. To resolve this, we introduce an additional modification to the above scheme. Specifically, we use a series of hash functions $H_1, \dots, H_L$ where $H_\ell : \{0, 1\}^* \rightarrow \mathbb{Z}_q^{n \times (m + m(2^\ell - 1) + 1)}$ and compute

$$\begin{aligned} \text{mpk} &= \mathbf{A}, \quad \text{msk} = \mathbf{T}_\mathbf{A}, \quad \text{sk} = (\{\mathbf{r}_\ell, \{\mathbf{k}_{y,\ell}\}_{y \in Y}\}_{\ell \in [L]}) \\ \text{ct} &= \mathbf{s}^\top \left[\underbrace{\mathbf{A}|\mathbf{D}_{x_1,t}| \cdots |\mathbf{D}_{x_t,t}| \sum_{x \in X} \mathbf{B}_{x,t} \sum_{x \in X} \mathbf{p}_{x,t}}_{\text{wavy line}} \right] + (\mathbf{0}, \mu \lfloor q/2 \rfloor) \end{aligned} \quad (3)$$

where $[\mathbf{D}_{x,\ell}|\mathbf{B}_{x,\ell}|\mathbf{p}_{x,\ell}] = H_\ell(x)$, $\mathbf{r}_\ell \leftarrow D_{\mathbb{Z}}^{m(2^\ell - 1)}$, $\mathbf{k}_{y,\ell} \leftarrow [\mathbf{A}|\mathbf{D}_{y,\ell}]^{-1}(\mathbf{B}_{y,\ell}\mathbf{r}_\ell + \mathbf{p}_{y,\ell})$. The decryption works similarly to the above schemes by using $(\mathbf{r}_t, \{\mathbf{k}_{y,t}\}_{y \in Y})$. Observe that now the size of the ciphertext is $O(t) = O(|f|)$.

At a high level, the security proof of the above scheme is the combination of the one for our KP-ABE and the one for the CP-ABE scheme by [\[WWW22\]](#). We consider a hybrid where we sample $\mathbf{A} \leftarrow \mathbb{Z}_p^{n \times m}$ and program the random oracle as $H_\ell(i) = [\mathbf{D}_{i,\ell}|\mathbf{B}_{i,\ell}|\mathbf{p}_{i,\ell}]$ for $\ell \in [L]$, where $\mathbf{D}_{i,\ell}$ is generated with a trapdoor if $\ell \neq t$ or $i \notin X$ (t is the length of the challenge policy X), and $\mathbf{D}_{i,t}$ for $i \in X$, $\mathbf{B}_{i,\ell}$, $\mathbf{p}_{i,\ell}$ are sampled uniformly. A secret key $(\{\mathbf{r}_\ell, \{\mathbf{k}_{y,\ell}\}_{y \in Y}\}_{\ell \in [L]})$ for Y is generated as follows. For $\ell \in [L] \setminus \{t\}$, sample $\mathbf{r}_\ell \leftarrow D_{\mathbb{Z}}^{m(2^\ell - 1)}$ and $\mathbf{k}_{y,\ell} \leftarrow [\mathbf{A}|\mathbf{D}_{y,\ell}]^{-1}(\mathbf{B}_{y,\ell}\mathbf{r}_\ell + \mathbf{p}_{y,\ell})$ for $y \in Y$ relying on the trapdoors for $\mathbf{D}_{y,\ell}$. For $\ell = t$,

1. sample $\mathbf{k}_{y,t} \leftarrow D_{\mathbb{Z}}^m$ for $y \in X \cap Y$;
2. (inefficiently) sample $\mathbf{r}_t \leftarrow D_{\mathbb{Z}}^{m(2^t - 1)}$ conditioned on

$$[\mathbf{A}|\mathbf{D}_{y,t}]\mathbf{k}_{y,t} = \mathbf{B}_{y,t}\mathbf{r}_t + \mathbf{p}_{y,t} \quad \text{for } y \in X \cap Y$$

3. sample $\mathbf{k}_y \leftarrow [\mathbf{A}|\mathbf{D}_{y,t}]^{-1}(\mathbf{B}_{y,t}\mathbf{r}_t + \mathbf{p}_{y,t})$ for $y \in Y \setminus X$ relying on the trapdoors for $\mathbf{D}_{y,t}$.

Showing the statistical indistinguishability between the original game and the hybrid is not a trivial task, but this part mostly follows [\[WWW22\]](#) using the fact that the distribution $\mathbf{k}_{y,\ell} \leftarrow [\mathbf{A}|\mathbf{D}_{y,\ell}]^{-1}(\mathbf{B}_{y,\ell}\mathbf{r}_\ell + \mathbf{p}_{y,\ell})$ relying on the trapdoor of $\mathbf{A}$ and that relying on the trapdoor of $\mathbf{D}_{y,\ell}$ are statistically close.

Finally, we randomize the challenge ciphertext with the generalized related trapdoor LWE (RT-LWE) assumption by [\[WWW22\]](#), which is implied by the standard LWE assumption. Roughly speaking, the RT-LWE assumption asserts that

$$\mathbf{s}^\top \left[\underbrace{\mathbf{A}|\mathbf{D}_{x_1,t}| \cdots |\mathbf{D}_{x_t,t}| \sum_{x \in X} \mathbf{B}_{x,t} \sum_{x \in X} \mathbf{p}_{x,t}}_{\text{wavy line}} \right]$$

looks random even if given uniformly distributed matrices $(\mathbf{A}, \{\mathbf{D}_{x,t}, \mathbf{B}_{x,t}, \mathbf{p}_{x,t}\}_{x \in X})$ and an oracle that, on input $Y' \subsetneq X$ and $\{\mathbf{t}_y\}_{y \in Y'} \in (\mathbb{Z}_q^n)^{Y'}$, returns an element $\mathbf{r}$ sampled according to $D_{\mathbb{Z}}^{m(2^t - 1)}$ conditioned on $\mathbf{B}_{y,t}\mathbf{r} = \mathbf{t}_y$ for all $y \in Y'$. By using the oracle with $Y' = X \cap Y$ and $\mathbf{t}_y = [\mathbf{A}|\mathbf{D}_{y,t}]\mathbf{k}_{y,t} - \mathbf{p}_{y,t}$, we can simulate the inefficient part of the hybrid, i.e., $\mathbf{r}_t$ in item (2).

The above techniques for our CP-ABE can also be applied to the MA-ABE scheme by [\[WWW22\]](#), and then we obtain the first large-universe ABE schemes from LWE. However, the security proofs need more careful analysis than the case of small-universe ABE. Recall that in small-universe MA-ABE, it is sufficient to consider the case where each authority controls one attribute [\[LW11a\]](#). Therefore, each $(\mathbf{A}_i, \mathbf{B}_i, \mathbf{p}_i)$ in [Eq. \(2\)](#) is translated into the master public key for authority i in the MA-ABE scheme by [\[WWW22\]](#). Hence, the adversary's view in the security game is basically the same in the cases of CP-ABE and MA-ABE except for corruptions. On the other hand, in our large-universe MA-ABE, $\mathbf{A}$ in [Eq. \(3\)](#) is generated by each authority. Thus, from the adversary's viewpoint, it appears that multiple instances of our CP-ABE scheme are executed, and we need to handle each instance in a hybrid manner.

Our Ring Instantiations. For deployment we instantiate our KP-ABE and CP-ABE over the $2n$ -th cyclotomic ring $R_q = \mathbb{Z}_q[x]/(x^n + 1)$: an LWE secret $\mathbf{s} \in \mathbb{Z}_q^n$ becomes a single ring element, $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ collapses to $\mathbf{a} \in R_q^{1 \times m}$, and a ciphertext carries n message bits rather than one. Our three lattice ingredients all have ring analogues—the gadget matrix $\mathbf{I}_n \otimes \mathbf{g}^\top$ becomes the gadget *vector* $\mathbf{g}$ ([Definition B.1](#)),

a trapdoor for $\mathbf{a}$ is a short $\mathbf{R}$ with $\mathbf{aR} = hg$ (Theorem B.2), and preimage sampling and delegation port directly (Lemma B.5 and Lemma B.6)—so the constructions of Section 3.1 and Section 3.2 transfer essentially verbatim, with the trapdoor instantiated statistically for the reasons given in Section 5.3. What does require new work is the CP-ABE proof, which needs the ring analogue of generalized related-trapdoor LWE (Assumption 5.1), whose reduction rests on a leftover hash lemma surviving the leak $\mathbf{e}^\top \mathbf{R}$ (Lemma B.4). Over R_q the standard bound charges the full support q^n rather than q , forcing $\hat{m} \gtrsim 2 + 2n/L$ and $\chi_R = 2^{O(n/L)}$ (Remark B.11); we conjecture that only the same-ideal component matters (Assumption B.2), under which $\hat{m} = 3$ suffices and q stays polynomial.

2 Preliminaries

Notations. For $m \in \mathbb{N}$, $[m]$ denotes a set $\{1, \dots, m\}$. For finite sets X and Y , 2^X denotes the power set of X , and Y^X denotes all maps from X to Y . Any function $\phi : X \rightarrow 2^Y$ can be represented by a set $S = \{(x, y) \in X \times Y \mid y \in \phi(x)\}$. Throughout the paper, we often identify ϕ with S . For a matrix $\mathbf{M}$ over $\mathbb{Z}$, $\|\mathbf{M}\|$ denotes the largest absolute value over all the entries. For a set (resp. a distribution) X , $x \leftarrow X$ means x is sampled uniformly from (resp. according to) X . For families of distributions $X = \{X_\lambda\}_{\lambda \in \mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda \in \mathbb{N}}$, $X \approx_c Y$ and $X \approx_s Y$ denote computational indistinguishability and statistical indistinguishability, respectively.

2.1 Attribute-based Encryption

Definition 2.1 (Attribute-Based Encryption). Let $P : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ be a predicate family where $\mathcal{X}$ and $\mathcal{Y}$ denote ciphertext attribute and key attribute spaces. An attribute-based encryption (ABE) scheme for a predicate family P and a message space $\mathcal{M}$ consists of four algorithms:

Setup(1^λ): It takes a security parameter 1^λ and outputs a master public key mpk and a master secret key msk .

Enc(mpk, x, μ): It takes mpk , an attribute $x \in \mathcal{X}$ and a message $\mu \in \mathcal{M}$, and outputs a ciphertext ct .

KeyGen($\text{mpk}, \text{msk}, y$): It takes mpk , msk , and an attribute $y \in \mathcal{Y}$, and outputs a secret key sk .

Dec($\text{mpk}, \text{ct}, \text{sk}, x, y$): It takes mpk , ct , sk , x , y , and outputs a message μ' or a symbol $\perp$.

Correctness. An ABE scheme is *correct* if it satisfies the following condition. For all $\lambda \in \mathbb{N}$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$ such that $P(x, y) = 1$, and $\mu \in \mathcal{M}$, there exist a negligible function negl , and we have

$$\Pr \left[\begin{array}{l} (\text{mpk}, \text{msk}) \leftarrow \text{Setup}(1^\lambda) \\ \mu = \text{Dec}(\text{mpk}, \text{ct}, \text{sk}) : \text{ct} \leftarrow \text{Enc}(\text{mpk}, x, \mu) \\ \text{sk} \leftarrow \text{KeyGen}(\text{mpk}, \text{msk}, y) \end{array} \right] \geq 1 - \text{negl}(\lambda).$$

Security. An ABE scheme is *selectively secure* if it satisfies the following condition. That is, the advantage of $\mathcal{A}$ defined as follows is negligible in λ for all stateful PPT adversary $\mathcal{A}$:

$$\Pr \left[\begin{array}{l} x^* \leftarrow \mathcal{A}(1^\lambda) \\ (\text{mpk}, \text{msk}) \leftarrow \text{Setup}(1^\lambda) \\ (\mu_0, \mu_1) \leftarrow \mathcal{A}^{\text{KeyGen}(\text{mpk}, \text{msk}, \cdot)}(\text{mpk}) \\ \beta \leftarrow \{0, 1\}, \text{ct}^* \leftarrow \text{Enc}(\text{mpk}, x^*, \mu_\beta) \\ \beta' \leftarrow \mathcal{A}^{\text{KeyGen}(\text{mpk}, \text{msk}, \cdot)}(\text{ct}^*) \end{array} \right] - \frac{1}{2}$$

where all y on which $\mathcal{A}$ queries **KeyGen** must satisfy $P(x^*, y) = 0$. Furthermore, we consider the security in the random oracle model and prohibit $\mathcal{A}$ from querying the random oracle before outputting x^* .

2.2 Multi-Authority Attribute-Based Encryption

We slightly modify the definition of large-universe multi-authority ABE by [RW15], that is, while they consider a map from an attribute-universe $\mathcal{U}$ to an authority universe $\mathcal{AU}$ in [RW15], we capture each attribute as an element from $\mathcal{AU} \times \mathcal{U}$ (in the following definition $\mathcal{Y}$ corresponds to $\mathcal{U}$). Note that in the large-universe setting, both definitions are essentially the same. The definition of static security completely follows that in [RW15].

Definition 2.2 (Multi-Authority Attribute-Based Encryption). Let $P : \mathcal{X} \times (2^{\mathcal{Y}})^{\mathcal{AU}} \rightarrow \{0, 1\}$ be a predicate family where $\mathcal{X}, \mathcal{Y}, \mathcal{AU}$ denote ciphertext attribute space, key attribute space, and authority identifier space, respectively. We denote $\mathbb{Y} = (2^{\mathcal{Y}})^{\mathcal{AU}}$ in what follows. Observe that a map $\sigma \in \mathbb{Y}$ can be represented by a set $Y = \{(\alpha, y) \in \mathcal{AU} \times \mathcal{Y} \mid y \in \sigma(\alpha)\}$. We will use both representations for convenience and identify σ with Y . For $\sigma = Y \in \mathbb{Y}$, we denote $\{\alpha \in \mathcal{AU} \mid \sigma(\alpha) \neq \emptyset\}$ by A_Y . For P , we require that

1. All $x \in \mathcal{X}$ specify relevant authorities $A_x \subseteq \mathcal{AU}$, i.e., there is a function $f : \mathcal{X} \rightarrow 2^{\mathcal{AU}}$, and we denote $f(x)$ by A_x .
2. For any $x \in \mathcal{X}$, if $P(x, Y) = 1$ for some $Y \in \mathbb{Y}$, then $P(x, Y') = 1$ for all $Y' \in \mathbb{Y}$ such that $Y \subset Y'$. In other words, $P(x, \cdot)$ is a monotone access structure.
3. $P(x, Y) = P(x, Y \cup (\alpha, y))$ for all $x \in \mathcal{X}, Y \in \mathbb{Y}, \alpha \in \mathcal{AU} \setminus A_x, y \in \mathcal{Y}$. In other words, $P(x, \sigma)$ depends only on $\{\sigma(\alpha)\}_{\alpha \in A_x}$.

Let $\mathcal{GU}$ be a global identifier (GID) space. A global identifier is used as a linchpin to tie secret keys together. A multi-authority attribute-based encryption (MA-ABE) scheme for a predicate family P and a message space $\mathcal{M}$ consists of the following algorithms:

GSetup(1^λ): It takes a security parameter 1^λ and outputs a global parameter gp . The other algorithms implicitly take gp .

ASetup(α): It takes an authority identifier α , and outputs a pair $(\text{mpk}_\alpha, \text{msk}_\alpha)$ of a master public key and a master secret key.

Enc($x, \{\text{mpk}_\alpha\}_{\alpha \in A_x}, \mu$): It takes attribute $x \in \mathcal{X}$, set $\{\text{mpk}_\alpha\}_{\alpha \in A_x}$ of master public keys, and a message $\mu \in \mathcal{M}$, and outputs a ciphertext ct .

KeyGen($\text{msk}, \text{gid}, y$): It takes msk , a global identifier $\text{gid} \in \mathcal{GU}$, and an attribute $y \in \mathcal{Y}$, and outputs a secret key sk .

Dec($\text{ct}, \{\text{sk}_{\alpha, y}\}_{(\alpha, y) \in Y}, x, Y$): It takes ct and a set $\{\text{sk}_{\alpha, y}\}_{(\alpha, y) \in Y}$ of secret keys, $x \in \mathcal{X}, Y \in \mathbb{Y}$, and outputs a message μ' or a symbol $\perp$.

Correctness. An MA-ABE scheme is *correct* if it satisfies the following condition. For all $\lambda \in \mathbb{N}$, $\text{gid} \in \mathcal{GU}$, $x \in \mathcal{X}$, $Y \in \mathbb{Y}$ such that $P(x, Y) = 1$, and $\mu \in \mathcal{M}$, the following probability is negligibly close to 1:

$$\Pr \left[\begin{array}{l} \text{gp} \leftarrow \text{GSetup}(1^\lambda) \\ (\text{mpk}_\alpha, \text{msk}_\alpha) \leftarrow \text{ASetup}(\alpha) \text{ for } \alpha \in A_x \cup A_Y \\ \mu = \mu' : \text{ct} \leftarrow \text{Enc}(x, \{\text{mpk}_\alpha\}_{\alpha \in A_x}, \mu) \\ \text{sk}_{\alpha, y, \text{gid}} \leftarrow \text{KeyGen}(\text{msk}_\alpha, \text{gid}, y) \text{ for } (\alpha, y) \in Y \\ \mu' = \text{Dec}(\text{ct}, \{\text{sk}_{\alpha, y, \text{gid}}\}_{(\alpha, y) \in Y}, x, Y) \end{array} \right]$$

Security. An MA-ABE scheme is *statically secure* if it satisfies the following condition. That is, the advantage of $\mathcal{A}$ defined as follows is negligible in λ for all stateful *admissible* PPT adversary $\mathcal{A}$:

$$\Pr \left[\begin{array}{l} \text{gp} \leftarrow \text{GSetup}(1^\lambda) \\ (x^*, \mu_0, \mu_1, \mathcal{C}, \{\text{mpk}_\alpha\}_{\alpha \in \mathcal{C}}, \mathcal{N}) \leftarrow \mathcal{A}(1^\lambda, \text{gp}) \\ \{\text{gid}, Y_{\text{gid}}\}_{\text{gid} \in \mathcal{Q}} \\ (\text{mpk}_\alpha, \text{msk}_\alpha) \leftarrow \text{ASetup}(\alpha) \text{ for } \alpha \in \mathcal{N} \\ \beta \leftarrow \{0, 1\}, \text{ct}^* \leftarrow \text{Enc}(x^*, \{\text{mpk}_\alpha\}_{\alpha \in A_{x^*}}, \mu_\beta) \\ \text{sk}_{\alpha, y, \text{gid}} \leftarrow \text{KeyGen}(\text{msk}_\alpha, \text{gid}, y) \text{ for } (\alpha, y) \in Y_{\text{gid}}, \text{gid} \in \mathcal{Q} \\ \beta' \leftarrow \mathcal{A}(\{\text{msk}_\alpha\}_{\alpha \in \mathcal{N}}, \text{ct}^*, \{\text{sk}_{\alpha, y, \text{gid}}\}_{(\alpha, y) \in Y_{\text{gid}}, \text{gid} \in \mathcal{Q}}) \end{array} \right] - \frac{1}{2}$$

where $x^* \in \mathcal{X}$ is the challenge attribute, $\mu_0, \mu_1 \in \mathcal{M}$ are the challenge messages, $\mathcal{C} \subseteq \mathcal{AU}$ is a set of corrupted authorities, $\mathcal{N} \subseteq \mathcal{AU}$ is a set of non-corrupted authorities, $\mathcal{Q} \subseteq \mathcal{GU}$ is a set of GIDs, $Y_{\text{gid}} \in \mathbb{Y}$ is a set of pairs of authority and key attribute associated with gid . $\mathcal{A}$ is admissible if in the challenge, it holds that $\mathcal{C} \cap \mathcal{N} = \emptyset$, $A_{x^*} \subseteq \mathcal{C} \cup \mathcal{N}$, $A_{Y_{\text{gid}}} \subseteq \mathcal{N}$ for all $\text{gid} \in \mathcal{Q}$, $P(x^*, Y) = 0$ for all $Y \in \mathbb{Y}$ such that $A_Y = \mathcal{C}$, and $P(x^*, Y \cup Y_{\text{gid}}) = 0$ for all $Y \in \mathbb{Y}$ such that $A_Y = \mathcal{C}$ and $\text{gid} \in \mathcal{Q}$. Furthermore, we consider the security in the random oracle model and prohibit $\mathcal{A}$ from querying the random oracle before outputting the challenge instance.

2.3 Lattices

Discrete Gaussians. We denote by $D_{\mathbb{Z}, \chi}$ the centered discrete Gaussian distribution over $\mathbb{Z}$ with parameter $\chi \in \mathbb{R}^+$ (see e.g., [Reg05] for discrete Gaussians). For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a vector $\mathbf{u} \in \mathbb{Z}_q^n$, $\mathbf{A}_\chi^{-1}(\mathbf{u})$ denotes the probability distribution of $\mathbf{x} \in \mathbb{Z}^m$ which follows $D_{\mathbb{Z}, \chi}^m$ conditioned on $\mathbf{A}\mathbf{x} = \mathbf{u} \bmod q$.

Lemma 2.1 (Lattice Trapdoors [Ajt99, AP09, CHKP10]). Let $n, m, q \in \mathbb{N}$ such that $m = O(n \log q)$. There exist the two PPT algorithms:

- $\text{TrapGen}(1^n, m, q)$: It outputs $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a τ -trapdoor $\mathbf{T}_\mathbf{A}$.
- $\text{SampleLeft}([\mathbf{A}|\mathbf{B}], \mathbf{T}_\mathbf{A}, \mathbf{u}, \chi)$: It takes $[\mathbf{A}|\mathbf{B}] \in \mathbb{Z}_q^{n \times 2m}$, a trapdoor $\mathbf{T}_\mathbf{A}$, $\mathbf{u} \in \mathbb{Z}_q^n$, and a parameter χ as input and outputs $\mathbf{k} \in \mathbb{Z}_q^{2m}$.

The above algorithms satisfy the following properties:

- $\mathbf{A}$ and $\mathbf{A}'$ are statistically indistinguishable where $(\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \text{TrapGen}(1^n, m, q)$ and $\mathbf{A}' \leftarrow \mathbb{Z}_q^{n \times m}$.
- $\text{TrapGen}(1^n, m, q)$ outputs a τ -trapdoor such that with overwhelming probability $\tau = O(\sqrt{n \log q})$. The parameter τ refers to the quality of the trapdoor $\mathbf{T}_\mathbf{A}$.
- Let $\mathbf{T}_\mathbf{A}$ be a τ -trapdoor for $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. If $\chi = \tau \cdot \omega(\sqrt{\log m})$, then for all $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{u} \in \mathbb{Z}_q^n$, $\mathbf{k}$ and $\mathbf{k}'$ are statistically indistinguishable where

$$\mathbf{k} \leftarrow \text{SampleLeft}([\mathbf{A}|\mathbf{B}], \mathbf{T}_\mathbf{A}, \mathbf{u}, \chi) \quad \text{and} \quad \mathbf{k}' \leftarrow [\mathbf{A}|\mathbf{B}]_\chi^{-1}(\mathbf{u})$$

We sometimes abuse the notation and denote by $\mathbf{k} \leftarrow \text{SampleLeft}([\mathbf{B}|\mathbf{A}], \mathbf{T}_\mathbf{A}, \mathbf{u}, \chi)$ the following operation: compute $\begin{bmatrix} \mathbf{k}_1 \\ \mathbf{k}_2 \end{bmatrix} \leftarrow \text{SampleLeft}([\mathbf{A}|\mathbf{B}], \mathbf{T}_\mathbf{A}, \mathbf{u}, \chi)$ and output $\mathbf{k} = \begin{bmatrix} \mathbf{k}_2 \\ \mathbf{k}_1 \end{bmatrix}$.

Then, the distribution $\text{SampleLeft}([\mathbf{B}|\mathbf{A}], \mathbf{T}_\mathbf{A}, \mathbf{u}, \chi)$ is statistically close to $[\mathbf{B}|\mathbf{A}]_\chi^{-1}(\mathbf{u})$ [CHK09, Theorem 3.4].

Lemma 2.2 (Tail Bound [GPV08]). Let $\chi > 1$ and n be polynomial in λ . Then, we have

$$\Pr \left[\|\mathbf{v}\| > \sqrt{\lambda} \chi : \mathbf{v} \leftarrow D_{\mathbb{Z}, \chi}^n \right] \leq \text{negl}(\lambda)$$

Lemma 2.3 (Preimage Sampling [GPV08]). Let $n, m, q \in \mathbb{N}$ such that $m \geq 2n \log_2 q$ and $\chi = \omega(\sqrt{\log m})$. Then, $(\mathbf{A}, \mathbf{x}, \mathbf{A}\mathbf{x})$ and $(\mathbf{A}, \mathbf{A}_\chi^{-1}(\mathbf{y}), \mathbf{y})$ are statistically indistinguishable, where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{x} \leftarrow D_{\mathbb{Z}, \chi}^m$, $\mathbf{y} \leftarrow \mathbb{Z}_q^n$.

Definition 2.3 (Learning with Errors [Reg05]). Let n, m, q, χ be parameters depending on the security parameter λ . The $\text{LWE}_{n, m, q, \chi}$ assumption states that $(\mathbf{A}, \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top)$ and $(\mathbf{A}, \mathbf{u}^\top)$ are computationally indistinguishable where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow D_{\mathbb{Z}, \chi}^m$, $\mathbf{u} \leftarrow \mathbb{Z}_q^m$.

Definition 2.4 (Generalized Related-Trapdoor LWE [WWW22]). Let $n, m, \hat{m}, \chi, q$ be lattice parameters depending on security parameter λ where q is a prime, and L be a length parameter. We define the distributions as follows: $b \leftarrow \{0, 1\}$, $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{B} \leftarrow \mathbb{Z}_q^{nL \times \hat{m}(2L-1)}$, $\mathbf{R} \leftarrow \{-1, 1\}^{\hat{m}L \times \hat{m}(L-1)}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow D_{\mathbb{Z}, \chi}^m$, $\mathbf{v} \leftarrow \mathbb{Z}_q^m$, $\hat{\mathbf{e}}_0 \leftarrow D_{\mathbb{Z}, \chi}^{\hat{m}L}$, $\hat{\mathbf{e}} = \hat{\mathbf{e}}_0 [\mathbf{I}_{\hat{m}L} | \mathbf{R}]$, $\hat{\mathbf{v}} \leftarrow \mathbb{Z}_q^{\hat{m}(2L-1)}$. We say the $\text{RTLWE}_{n, m, \hat{m}, q, \chi, L}$ assumption holds is the following probability is negligibly indistinguishable to $1/2$ in λ for all stateful admissible PPT adversaries $\mathcal{A}$:

$$\Pr \left[\begin{array}{l} \mathbf{u} \in \{0, 1\}^L \leftarrow \mathcal{A}(1^\lambda) \\ b = b' : P_0 = (\mathbf{A}, \mathbf{B}, \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top, \mathbf{s}^\top (\mathbf{u}^\top \otimes \mathbf{I}_n) \mathbf{B} + \hat{\mathbf{e}}^\top), P_1 = (\mathbf{A}, \mathbf{B}, \mathbf{v}^\top, \hat{\mathbf{v}}^\top) \\ b' \leftarrow \mathcal{A}^{\text{tdO}(\cdot, \cdot)}(P_b) \end{array} \right]$$

where $\text{tdO}(\mathbf{M}, \mathbf{t})$ returns $\mathbf{y} \leftarrow ((\mathbf{M} \otimes \mathbf{I}_n) \mathbf{B})_\chi^{-1}(\mathbf{t})$, and $\mathcal{A}$ is admissible if, for all $\mathbf{M}$ queried to tdO , it holds that $\mathbf{M} \in \mathbb{Z}_q^{k \times L}$ where $k < L$, and $\begin{bmatrix} \mathbf{u}^\top \\ \mathbf{M} \end{bmatrix}$ is full row rank.

Lemma 2.4 ([WWW22]). Suppose that $q > 2$ is a prime, $\chi = 2\hat{m}^2 L^2 \cdot \omega(\sqrt{\log n})$, and $\hat{m} = O(n \log q)$. Then, the $\text{LWE}_{n, m + \hat{m}L, q, \chi}$ assumption implies the $\text{RTLWE}_{n, m, \hat{m}, q, \chi, L}$ assumption.

3 Large-Universe ABE for Subset Policies

Before diving into our constructions, we prove the following lemma that is useful to prove the security of our constructions.

Lemma 3.1. Let $n, m, q \in \mathbb{N}$ such that $m \geq 2n \log_2 q$, $\ell_1, \ell_2 \in \mathbb{N}$ is polynomial in n , and $\chi = \omega(\sqrt{\log m})$. Then, $(\mathbf{A}, \{\mathbf{B}_i, \mathbf{X}_i, \mathbf{Y}_i\}_{i \in [\ell_1]})$ and $(\mathbf{A}, \{\mathbf{B}_i, \hat{\mathbf{X}}_i, \hat{\mathbf{Y}}_i\}_{i \in [\ell_1]})$ are statistically indistinguishable in n , where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{B}_i \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{X}_i \leftarrow D_{\mathbb{Z}, \chi}^{2m \times \ell_2}$, $\mathbf{Y}_i = [\mathbf{A}|\mathbf{B}_i] \mathbf{X}_i$, $\hat{\mathbf{Y}}_i \leftarrow \mathbb{Z}_q^{n \times \ell_2}$, and $\hat{\mathbf{X}}_i \leftarrow [\mathbf{A}|\mathbf{B}_i]_\chi^{-1}(\hat{\mathbf{Y}}_i)$.

Proof. By a simple hybrid argument, Lemma 2.3 implies that $(\hat{\mathbf{A}}, \mathbf{X}, \hat{\mathbf{A}}\mathbf{X})$ and $(\hat{\mathbf{A}}, \hat{\mathbf{A}}_\chi^{-1}(\mathbf{Y}), \mathbf{Y})$ are statistically indistinguishable, where $\hat{\mathbf{A}} \leftarrow \mathbb{Z}_q^{n \times \hat{m}}$, $\mathbf{X} \leftarrow D_{\mathbb{Z}, \chi}^{\hat{m} \times \ell_2}$, $\mathbf{Y} \leftarrow \mathbb{Z}_q^{n \times \ell_2}$ and $\hat{m} \geq 4n \log_2 q$. By setting $[\mathbf{A}|\mathbf{B}_i] = \hat{\mathbf{A}}$, $\mathbf{X}_i = \mathbf{X}$, $\hat{\mathbf{Y}}_i = \mathbf{Y}$ and applying the indistinguishability ℓ_1 times, we can get the indistinguishability that the lemma asserts. $\square$

3.1 KP-ABE

First, we define large-universe KP-ABE for subset policies.

Definition 3.1 (Large-Universal KP-ABE for Subset Policies). Let $\mathcal{U}_\lambda$ be an attribute universe such that $|\mathcal{U}_\lambda| = \lambda^{\omega(1)}$, i.e., the size is superpolynomial. Large-Universal KP-ABE for subset policies (or conjunctions) is ABE as per [Definition 2.1](#) for the following predicate $P : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$:

$$\mathcal{X} = \mathcal{Y} = 2^{\mathcal{U}_\lambda}, \quad P(X, Y) = 1 \Leftrightarrow X \supseteq Y$$

Remark 3.1 (KP-ABE for DNFs). Large-universe KP-ABE for DNFs is defined as ABE for $P : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ where

$$\mathcal{X} = 2^{\mathcal{U}_\lambda}, \quad \mathcal{Y} = \bigcup_{i \in \mathbb{N}} (2^{\mathcal{U}_\lambda})^i \quad P(X, (Y_1, \dots, Y_s)) = 1 \Leftrightarrow \bigvee_{i \in [s]} X \supseteq Y_i$$

The conversion from large-universe KP-ABE for subset policies to that for DNFs is quite simple, that is, just defining the secret key $\mathbf{sk}$ for $(Y_1, \dots, Y_s)$ as a set $\{\mathbf{sk}_i\}_{i \in [s]}$ of secret keys of underlying scheme for Y_i suffices. The collusion resistance of the converted scheme follows from the fact that any combination of unauthorized keys cannot decrypt the ciphertext due to the collusion resistance of the underlying scheme.

In our construction, we have a restriction such that $|Y| \leq L$ where L is a parameter that is polynomial in λ . This limitation can be removed if we allow LWE with superpolynomial modulus-to-noise ratio as described in [Remark 3.2](#). Furthermore, our scheme is unbounded in a sense that number of attributes that can be associated with a ciphertext is not a priori bounded.

Construction. Let $n, m, q \in \mathbb{N}$ and $\chi \in \mathbb{R}^+$ be lattice parameters that depend on the security parameter λ , where $q > 2$ is a prime. Let the message space $\mathcal{M} = \{0, 1\}$. Let $L \in \mathbb{N}$ be the maximum size of the set used in KeyGen. Let $H : \{0, 1\}^* \rightarrow \mathbb{Z}_q^{n \times m}$ be a hash function.

Setup(1^λ): It outputs mpk and msk as follows:

$$(\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \text{TrapGen}(1^n, m, q), \quad \mathbf{p} \leftarrow \mathbb{Z}_q^n \\ \text{mpk} = (\mathbf{A}, \mathbf{p}), \quad \text{msk} = \mathbf{T}_\mathbf{A}$$

Enc(mpk, X, μ): It parses $X = \{x_1, \dots, x_t\}$ and outputs ct as follows.

$$\mathbf{s} \leftarrow \mathbb{Z}_q^n, \quad \mathbf{e}_1 \leftarrow D_{\mathbb{Z}, \chi}^{(t+1)m}, \quad e_2 \leftarrow D_{\mathbb{Z}, \chi}, \quad \mathbf{D} = [H(x_1) \parallel \dots \parallel H(x_t)] \\ \text{ct} = (\mathbf{c}_1^\top, c_2) = (\mathbf{s}^\top [\mathbf{A} \mathbf{D}] + \mathbf{e}_1^\top, \mathbf{s}^\top \mathbf{p} + e_2 + \mu \lfloor q/2 \rfloor)$$

KeyGen($\text{mpk}, \text{msk}, Y$): It parses $Y = \{y_1, \dots, y_u\}$ where it must hold that $u \leq L$. It computes, for $i \in [u]$,

$$\mathbf{p}_i \leftarrow \mathbb{Z}_q^n \quad \text{if } i < u, \quad \mathbf{p}_u = \mathbf{p} - \sum_{j \in [u-1]} \mathbf{p}_j \\ \mathbf{k}_i \leftarrow \text{SampleLeft}([\mathbf{A} H(y_i)], \mathbf{T}_\mathbf{A}, \mathbf{p}_i, \chi)$$

Then, it outputs $\mathbf{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u)$.

Dec($\text{mpk}, \text{ct}, \mathbf{sk}, X, Y$): It parses $\text{ct} = (\mathbf{c}_1^\top, c_2)$ where $\mathbf{c}_1^\top = [\mathbf{c}_{1,0}^\top \parallel \dots \parallel \mathbf{c}_{1,t}^\top] \in (\mathbb{Z}_q^m)^{t+1}$, $\mathbf{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u)$, $X = \{x_1, \dots, x_t\}$, and $Y = \{y_1, \dots, y_u\}$. If $X \supseteq Y$, it computes $\rho : [u] \rightarrow [t]$ such that $y_i = x_{\rho(i)}$ and outputs

$$\left\lfloor \frac{2}{q} \cdot \left(c_2 - \sum_{i \in [u]} [\mathbf{c}_{1,0}^\top \parallel \mathbf{c}_{1,\rho(i)}^\top] \mathbf{k}_i \right) \right\rfloor$$

Otherwise, output $\perp$.

Parameter Selection. From [Theorem 3.1](#) and [Theorem 3.2](#), we can choose parameters as follows:

$$n = \lambda, \quad q = O(n^3 L^2), \quad m = O(n \log q), \quad \chi = O(\log m)$$

Observe that we can choose q, χ as polynomials in λ . Hence, our scheme is secure under the LWE with a polynomial modulus-to-noise ratio.

Correctness. For correctness, we have the following theorem.

Theorem 3.1. Suppose $q \geq 5(\sqrt{\lambda}\chi + 2mL\lambda\chi^2)$, $m = O(n \log q)$, and $\chi = \omega(\sqrt{\log m})$. Then, our scheme is correct.

Proof. In decryption, suppose $\text{ct} = (\mathbf{c}_1^\top, c_2)$ where $\mathbf{c}_1^\top = [\mathbf{c}_{1,0}^\top | \dots | \mathbf{c}_{1,t}^\top] \in (\mathbb{Z}_q^m)^{t+1}$, $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u)$ are a ciphertext and a secret key for X and Y , respectively, such that $X \supseteq Y$. In decryption, we have

$$\begin{aligned} c_2 - \sum_{i \in [u]} [\mathbf{c}_{1,0}^\top | \mathbf{c}_{1,\rho(i)}^\top] \mathbf{k}_i \\ &= \mathbf{s}^\top \mathbf{p} + e_2 + \mu \lfloor q/2 \rfloor - \sum_{i \in [u]} (\mathbf{s}^\top [\mathbf{A} | \mathbf{D}_i] + [\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,\rho(i)}^\top]) \mathbf{k}_i \\ &= \mathbf{s}^\top \mathbf{p} + e_2 + \mu \lfloor q/2 \rfloor - \mathbf{s}^\top \mathbf{p} - \sum_{i \in [u]} [\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,\rho(i)}^\top] \mathbf{k}_i \\ &= \underbrace{\mu \lfloor q/2 \rfloor + e_2 - \sum_{i \in [u]} [\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,\rho(i)}^\top] \mathbf{k}_i}_{\tilde{e}} \end{aligned}$$

where $\mathbf{D}_i = H(y_i)$, $\mathbf{e}_1^\top = [\mathbf{e}_{1,0}^\top | \dots | \mathbf{e}_{1,t}^\top]$, and in the second equality we use the fact that $[\mathbf{A} | \mathbf{D}_i] \mathbf{k}_i = \mathbf{p}_i$ and $\sum_{i \in [u]} \mathbf{p}_i = \mathbf{p}$. From the parameter setting and Lemma 2.2, we have $|e_2| \leq \sqrt{\lambda}\chi$ and $\|[\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,\rho(i)}^\top]\| \leq \sqrt{\lambda}\chi$ with overwhelming probability. Thanks to the property of lattice trapdoors, the distribution of $\mathbf{k}_i$ is statistically close to $[\mathbf{A} | \mathbf{D}_i]_\chi^{-1} \mathbf{p}_i$. Since the marginal distribution of $\mathbf{p}_i$ is uniform in $\mathbb{Z}_q^n$, the marginal distribution of $\mathbf{k}_i$ is statistically close to $D_{\mathbb{Z}, \chi}^{2m}$ from Lemma 2.3. Thus, we have $\|\mathbf{k}_i\| \leq \sqrt{\lambda}\chi$ with overwhelming probability again from Lemma 2.2. In summary, we have, with overwhelming probability,

$$|\tilde{e}| = \left| e_2 - \sum_{i \in [u]} [\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,i}^\top] \mathbf{k}_i \right| \leq \sqrt{\lambda}\chi + 2mu\lambda\chi^2 \leq \sqrt{\lambda}\chi + 2mL\lambda\chi^2$$

Thus, $|\tilde{e}| \leq q/5$ and correctness holds. $\square$

Security Analysis. For security, we have the following theorem.

Theorem 3.2. Suppose $m = O(n \log q)$, $\chi = \omega(\sqrt{\log n})$, the hash function H is modeled as a random oracle, t is the size of the attribute X for the challenge query, and $\text{LWE}_{n, m(t+1)+1, q, \chi}$ holds. Then, our scheme satisfies selective security as per Definition 2.1.

Proof. We prove the theorem by a series of hybrids described as follows.

- H_0 : This is the original game. In this game, $\mathcal{A}$ first outputs the challenge attribute $X = (x_1, \dots, x_t)$. Then, the challenger computes

$$\begin{aligned} (\mathbf{A}, \mathbf{T}_\mathbf{A}) &\leftarrow \text{TrapGen}(1^n, m, q), \mathbf{p} \leftarrow \mathbb{Z}_q^n \\ \text{mpk} &= (\mathbf{A}, \mathbf{p}), \quad \text{msk} = \mathbf{T}_\mathbf{A} \end{aligned}$$

and gives mpk to $\mathcal{A}$. When $\mathcal{A}$ asks for a secret key for $Y = (y_1, \dots, y_u)$, the challenger computes

$$\begin{aligned} \mathbf{p}_i &\leftarrow \mathbb{Z}_q^n \quad \text{if } i < u, \quad \mathbf{p}_u = \mathbf{p} - \sum_{j \in [u-1]} \mathbf{p}_j \\ \mathbf{k}_i &\leftarrow \text{SampleLeft}([\mathbf{A} | H(y_i)], \mathbf{T}_\mathbf{A}, \mathbf{p}_i, \chi) \end{aligned}$$

and gives $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u)$ to $\mathcal{A}$. For the challenge, $\mathcal{A}$ outputs μ_0, μ_1 . Then, the challenger chooses a challenge bit $b \leftarrow \{0, 1\}$ and gives ct to $\mathcal{A}$ where

$$\begin{aligned} \mathbf{s} &\leftarrow \mathbb{Z}_q^n, \mathbf{e}_1 \leftarrow D_{\mathbb{Z}, \chi}^{(t+1)m}, e_2 \leftarrow D_{\mathbb{Z}, \chi}, \mathbf{D} = [H(x_1) | \dots | H(x_t)] \\ \text{ct} &= (\mathbf{c}_1^\top, c_2) = (\mathbf{s}^\top [\mathbf{A} | \mathbf{D}] + \mathbf{e}_1^\top, \mathbf{s}^\top \mathbf{p} + e_2 + \mu_b \lfloor q/2 \rfloor) \end{aligned}$$

The post-challenge key queries are the same as the pre-challenge queries. Finally, $\mathcal{A}$ outputs b' . During the game, the challenger simulates the random oracle as follows. It maintains a dictionary $\mathcal{D}$ that is initially empty. When H is invoked on input x , the challenger checks if $\mathcal{D}(x) = \perp$. If this is the case, the challenger samples $\mathbf{D} \leftarrow \mathbb{Z}_q^{n \times m}$, updates $\mathcal{D}(x) = \mathbf{D}$, and returns $\mathbf{D}$. Otherwise, it returns $\mathcal{D}(x)$.

- H_1 : This is the same as H_0 except that the challenger simulates the random oracle as follows. Right after $\mathcal{A}$ outputs the challenge attribute $X = (x_1, \dots, x_t)$, the challenger computes $\mathbf{D}_i \leftarrow \mathbb{Z}_q^{n \times m}$ and updates $\mathcal{D}(x_i) = \mathbf{D}_i$ for all $i \in [t]$. During the game, when H is invoked on input x , the challenger checks if $\mathcal{D}(x) = \perp$. If this is the case, the challenger samples $(\mathbf{D}, \mathbf{T}_\mathbf{D}) \leftarrow \text{TrapGen}(1^n, m, q)$, updates $\mathcal{D}(x) = \mathbf{D}$, and returns $\mathbf{D}$. Otherwise, it returns $\mathcal{D}(x)$. Since the property of lattice trapdoors (Lemma 2.1), $\mathbf{D}$ is statistically close to uniform, and thus H_0 and H_1 are statistically indistinguishable given $m = O(n \log q)$.
- H_2 : This is the same as H_1 except the followings. In this hybrid, the challenger samples $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ instead of using TrapGen for mpk . For all secret key queries on $Y = (y_1, \dots, y_u)$, let $Y_X = Y \cap X$ and $Y_{\bar{X}} = Y \setminus X$. The challenger computes

$$\begin{aligned} \mathbf{p}_i &\leftarrow \mathbb{Z}_q^n \text{ for } i : y_i \in Y_X, \\ \mathbf{p}_i &\leftarrow \mathbb{Z}_q^n \text{ conditioned on } \sum_{i \in [u]} \mathbf{p}_i = \mathbf{p} \text{ for } i : y_i \in Y_{\bar{X}}, \\ \mathbf{k}_i &\leftarrow \begin{cases} [\mathbf{A}|H(y_i)]_\chi^{-1}(\mathbf{p}_i) & (i : y_i \in Y_X) \\ \text{SampleLeft}([\mathbf{A}|H(y_i)], \mathbf{T}_{H(y_i)}, \mathbf{p}_i, \chi) & (i : y_i \in Y_{\bar{X}}) \end{cases} \end{aligned}$$

and gives $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u)$ to $\mathcal{A}$. Note that the sampling of $\mathbf{k}_i$ for $i : y_i \in Y_X$ is inefficient. It is easy to see that $H_1 \approx_s H_2$ since $\mathbf{A}$ is statistically close to uniform in H_1 , $\mathbf{p}_i$ for $i \in [u]$ are uniformly distributed conditioned on $\sum_{i \in [u]} \mathbf{p}_i = \mathbf{p}$ in both hybrids, and, for all $i \in [u]$,

$$\begin{aligned} &\text{SampleLeft}([\mathbf{A}|H(y_i)], \mathbf{T}_{H(y_i)}, \mathbf{p}_i, \chi) \\ &\approx_s \text{SampleLeft}([\mathbf{A}|H(y_i)], \mathbf{T}_\mathbf{A}, \mathbf{p}_i, \chi) \\ &\approx_s [\mathbf{A}|H(y_i)]_\chi^{-1}(\mathbf{p}_i) \end{aligned}$$

which follows from Lemma 2.1 if $m = O(n \log q)$.

- H_3 : This game is the same as H_2 except the followings. For all secret key queries on $Y = (y_1, \dots, y_u)$, let $Y_X = Y \cap X$, $Y_{\bar{X}} = Y \setminus X$. The challenger computes

$$\begin{aligned} \mathbf{k}_i &\leftarrow D_{\mathbb{Z}, \chi}^{2m}, \mathbf{p}_i = [\mathbf{A}|H(y_{i_1})]\mathbf{k}_i \text{ for } i : y_i \in Y_X \\ \mathbf{p}_i &\leftarrow \mathbb{Z}_q^n \text{ for } i : y_i \in Y_{\bar{X}} \text{ conditioned on } \sum_{i \in [u]} \mathbf{p}_i = \mathbf{p} \\ \mathbf{k}_i &\leftarrow \text{SampleLeft}([\mathbf{A}|H(y_i)], \mathbf{T}_{H(y_i)}, \mathbf{p}_i, \chi) \text{ for } i : y_i \in Y_{\bar{X}} \end{aligned} \tag{4}$$

and gives $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u)$ to $\mathcal{A}$. We prove $H_2 \approx_s H_3$ in Lemma 3.2.

- H_4 : This game is the same as H_3 except the followings. The challenger generates the challenge ciphertext as

$$\text{ct} = (\mathbf{c}_1^\top, c_2) \leftarrow \mathbb{Z}_q^{(t+1)m} \times \mathbb{Z}_q$$

We prove $H_3 \approx_c H_4$ in Lemma 3.3.

It is obvious that $\mathcal{A}$ does not obtain the information on b , and thus its advantage is 0 in the last hybrid. This concludes the proof. $\square$

Lemma 3.2. $H_2 \approx_s H_3$ if $m = O(n \log q)$ and $\chi = \omega(\sqrt{\log m})$.

Proof. Let Q be the number of secret key queries. Due to the game condition, we have $Y_X \subsetneq Y$ for all Y in secret key queries. Hence, $\mathbf{p}_i^{(j)}$ for all $i : y_i^{(j)} \in Y_X^{(j)}$, $j \in [Q]$ are uniformly and independently distributed in $\mathbb{Z}_q^n$. Recall that $\mathbf{A}$ is uniformly distributed in $\mathbb{Z}_q^{n \times m}$. Thanks to Lemma 3.1 for $\ell_1 = |X|$ and $\ell_2 = Q$, the joint distribution of $(\mathbf{A}, \mathbf{k}_i^{(j)}, \mathbf{p}_i^{(j)})$ for $i : y_i^{(j)} \in Y_X^{(j)}$, $j \in [Q]$ in H_2 is statistically close to the distribution of $(\mathbf{A}, \mathbf{k}_i^{(j)}, \mathbf{p}_i^{(j)})$ for $i : y_i^{(j)} \in Y_X^{(j)}$, $j \in [Q]$ where we first sample $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{k}_i^{(j)} \leftarrow D_{\mathbb{Z}, \chi}^{2m}$ and set $\mathbf{p}_i^{(j)} = [\mathbf{A}|H(y_i^{(j)})]\mathbf{k}_i^{(j)}$. Hence, we have $H_2 \approx_s H_3$. $\square$

Lemma 3.3. $H_3 \approx_c H_4$ under the $\text{LWE}_{n, m(t+1)+1, q, \chi}$ assumption where t is the size of the attribute X for the challenge query.

Proof. We prove that if there is an adversary $\mathcal{A}$ that distinguishes H_3 and H_4 with non-negligible probability, we can construct a reduction $\mathcal{B}$ that breaks the $\text{LWE}_{n, m(t+1)+1, q, \chi}$ assumption. We can construct $\mathcal{B}$ as follows.

- $\mathcal{A}$ declares the challenge attribute $X = (x_1, \dots, x_t)$ at the beginning of the game. $\mathcal{B}$ obtains a LWE instance $(\hat{\mathbf{A}}, \mathbf{v}^\top)$ where $\mathbf{v}^\top = \mathbf{s}^\top \hat{\mathbf{A}} + \mathbf{e}^\top$ or $\mathbf{v}^\top \leftarrow \mathbb{Z}_q^{(t+1)m+1}$. $\mathcal{B}$ parses $\hat{\mathbf{A}} = [\mathbf{A}|\mathbf{D}_1|\dots|\mathbf{D}_t|\mathbf{p}]$ and gives $\text{mpk} = (\mathbf{A}, \mathbf{p})$ to $\mathcal{A}$.
- Random oracle queries: $\mathcal{B}$ first sets $\mathcal{D}(x_i) = \mathbf{D}_i$ for all $i \in [t]$, where $\mathcal{D}$ is the dictionary for the random oracle simulation. During the game, when H is invoked on input x , the challenger checks if $\mathcal{D}(x) = \perp$. If this is the case, the challenger samples $(\mathbf{D}, \mathbf{T}_\mathbf{D}) \leftarrow \text{TrapGen}(1^n, m, q)$, updates $\mathcal{D}(x) = \mathbf{D}$, and returns $\mathbf{D}$. Otherwise, it just returns $\mathcal{D}(x)$.
- Secret key queries: when $\mathcal{A}$ queries $\mathcal{B}$ on $Y = (y_1, \dots, y_u)$, $\mathcal{B}$ defines $Y_X = Y \cap X$, $Y_{\bar{X}} = Y \setminus X$, and computes $\mathbf{k}_i$ for $i \in [u]$ as in Eq. (4). Then, returns $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u)$ to $\mathcal{A}$.
- Challenge: after $\mathcal{A}$ outputs the challenge messages (μ_0, μ_1) , $\mathcal{B}$ chooses $b \leftarrow \{0, 1\}$ and gives the challenge ciphertext $\text{ct} = \mathbf{v}^\top + (0^{(t+1)m}, \mu_b \lfloor q/2 \rfloor)$.
- Output: if $\mathcal{A}$ outputs b' at the end of the game, $\mathcal{B}$ outputs 1 if and only if $b = b'$.

Observe that $\mathcal{A}$'s view corresponds to H_3 if $\mathbf{v}^\top = \mathbf{s}^\top \hat{\mathbf{A}} + \mathbf{e}^\top$ and H_4 otherwise. This concludes the proof. $\square$

Remark 3.2 (On the bound L). We can see that the bound L in the above scheme is used only to decide the modulus q as per Theorem 3.1. Hence, if we set $q = \lambda^{\omega(1)}$ keeping other parameters the same and use LWE with a sub-exponential modulus-to-ratio, we can remove the bound L from the above construction because L is polynomial.

3.2 CP-ABE

We define large-universe CP-ABE for subset policies as follows.

Definition 3.2 (Large-Universe CP-ABE for Subset Policies). Let $\mathcal{U}_\lambda$ be an attribute universe such that $|\mathcal{U}_\lambda| = \lambda^{\omega(1)}$, i.e., the size is superpolynomial. Large-Universe CP-ABE for Subset Policies is ABE as per Definition 2.1 for the following predicate $\text{P} : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$:

$$\mathcal{X} = \mathcal{Y} = 2^{\mathcal{U}_\lambda}, \quad \text{P}(X, Y) = 1 \Leftrightarrow X \subseteq Y$$

Remark 3.3 (CP-ABE for DNFs). Large-universe CP-ABE for DNFs is defined as ABE for $\text{P} : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ where

$$\mathcal{X} = \bigcup_{i \in \mathbb{N}} (2^{\mathcal{U}_\lambda})^i, \mathcal{Y} = 2^{\mathcal{U}_\lambda}, \text{P}((X_1, \dots, X_s), Y) = 1 \Leftrightarrow \bigvee_{i \in [s]} X_i \subseteq Y$$

The conversion from large-universe CP-ABE for subset policies to that for DNFs is quite simple, that is, just defining the ciphertext ct for $(X_1, \dots, X_s)$ as a set $\{\text{ct}_i\}_{i \in [s]}$ of ciphertexts of underlying scheme for X_i suffices. The security of the converted scheme follows by a simple hybrid argument.

In our construction, we have a restriction such that $|X| \leq L$ where L is a parameter that is polynomial in λ . Notably, our scheme is unbounded in a sense that number of attributes that can be associated with a secret key is not a priori bounded.

Construction. Let $n, m, q \in \mathbb{N}$ and $\chi \in \mathbb{R}^+$ be lattice parameters that depend on the security parameter λ , where $q > 2$ is a prime. Let the message space $\mathcal{M} = \{0, 1\}$. Let L be the maximum size of the set used in Enc . Let $H_\ell : \{0, 1\}^* \rightarrow \mathbb{Z}_q^{n \times (2m\ell+1)}$ for $\ell \in [L]$ be hash functions.

$\text{Setup}(1^\lambda)$: It outputs $(\text{mpk}, \text{msk}) = (\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \text{TrapGen}(1^n, m, q)$.

$\text{Enc}(\text{mpk}, X, \mu)$: It parses $X = \{x_1, \dots, x_t\}$, where it must hold that $t < L$, and outputs ct as follows.

$$\begin{aligned} \mathbf{s} &\leftarrow \mathbb{Z}_q^n, \mathbf{e}_1 \leftarrow D_{\mathbb{Z}, \chi}^{(t+1)m}, \mathbf{e}_2 \leftarrow D_{\mathbb{Z}, \chi}^{mt}, \mathbf{e}_3 \leftarrow D_{\mathbb{Z}, \chi}, \\ \mathbf{R} &\leftarrow \{-1, 1\}^{mt \times m(t-1)}, \\ [\underbrace{\mathbf{D}_i}_m \mid \underbrace{\mathbf{B}_i}_{m(2t-1)} \mid \underbrace{\mathbf{p}_i}_1] &= H_t(x_i), \mathbf{D} = [\mathbf{D}_1 \mid \dots \mid \mathbf{D}_t], \\ \text{ct} &= (\mathbf{c}_1^\top, \mathbf{c}_2^\top, c_3), \\ &= \left(\mathbf{s}^\top [\mathbf{A}|\mathbf{D}] + \mathbf{e}_1^\top, \mathbf{s}^\top \sum_{i \in [t]} \mathbf{B}_i + \mathbf{e}_2^\top [\mathbf{I}_{mt}|\mathbf{R}], \mathbf{s}^\top \sum_{i \in [t]} \mathbf{p}_i + c_3 + \mu \lfloor q/2 \rfloor \right) \end{aligned}$$

KeyGen(mpk, msk, Y): It parses $Y = \{y_1, \dots, y_u\}$. It computes $\mathbf{r}_\ell \leftarrow D_{\mathbb{Z}, \chi}^{m(2\ell-1)}$ for $\ell \in [L]$ and, for $(i, \ell) \in [u] \times [L]$,

$$\underbrace{[\widehat{\mathbf{D}}_{i,\ell}]_m}_{m} \mid \underbrace{[\widehat{\mathbf{B}}_{i,\ell}]_{m(2\ell-1)}}_{m(2\ell-1)} \mid \underbrace{[\widehat{\mathbf{p}}_{i,\ell}]_1}_1 = H_\ell(y_i),$$

$$\mathbf{k}_{i,\ell} \leftarrow \text{SampleLeft}([\mathbf{A}|\widehat{\mathbf{D}}_{i,\ell}], \mathbf{T}_\mathbf{A}, \widehat{\mathbf{B}}_{i,\ell}\mathbf{r}_\ell + \widehat{\mathbf{p}}_{i,\ell}, \chi)$$

Then, it outputs $\mathbf{sk} = \{\mathbf{k}_{1,\ell}, \dots, \mathbf{k}_{u,\ell}, \mathbf{r}_\ell\}_{\ell \in [L]}$.

Dec(mpk, ct, sk, X, Y): It parses $\mathbf{ct} = (\mathbf{c}_1^\top, \mathbf{c}_2^\top, c_3)$ where $\mathbf{c}_1^\top = [\mathbf{c}_{1,0}^\top \cdots \mathbf{c}_{1,t}^\top] \in (\mathbb{Z}_q^m)^{t+1}$, $\mathbf{sk} = \{\mathbf{k}_{1,\ell}, \dots, \mathbf{k}_{u,\ell}, \mathbf{r}_\ell\}_{\ell \in [L]}$, $X = \{x_1, \dots, x_t\}$, and $Y = \{y_1, \dots, y_u\}$.

$$\left\lfloor \frac{2}{q} \cdot \left(c_3 - \left(\sum_{i \in [t]} [\mathbf{c}_{1,0}^\top | \mathbf{c}_{1,i}^\top] \mathbf{k}_{\rho(i),t} - \mathbf{c}_2^\top \mathbf{r}_t \right) \right) \right\rfloor$$

Otherwise, output $\perp$.

Parameter Selection. From [Theorem 3.3](#) and [Theorem 3.4](#), we can choose parameters as follows:

$$n = \lambda, \quad q = O(n^8 L^7), \quad m = O(n \log q), \quad \chi = O(m^2 L^2 \log n)$$

Observe that we can choose q, χ as polynomials in λ . Hence, our scheme is secure under the LWE with a polynomial modulus-to-noise ratio.

Correctness. For correctness, we have the following theorem.

Theorem 3.3. Suppose $q \geq 5(\sqrt{\lambda}\chi + mL(2 + 2mL - m)\lambda\chi^2)$, $m = O(n \log q)$, and $\chi = \omega(\sqrt{\log m})$. Then, our scheme is correct.

Proof. In decryption, suppose $\mathbf{ct} = (\mathbf{c}_1^\top, \mathbf{c}_2^\top, c_3)$ where $\mathbf{c}_1^\top = [\mathbf{c}_{1,0}^\top \cdots \mathbf{c}_{1,t}^\top]$, $\mathbf{sk} = \{\mathbf{k}_{1,\ell}, \dots, \mathbf{k}_{u,\ell}, \mathbf{r}_\ell\}_{\ell \in [L]}$ are a ciphertext and a secret key for X and Y , respectively, such that $X \subseteq Y$. In decryption, we have

$$\begin{aligned} & c_3 - \left(\sum_{i \in [t]} [\mathbf{c}_{1,0}^\top | \mathbf{c}_{1,i}^\top] \mathbf{k}_{\rho(i),t} - \mathbf{c}_2^\top \mathbf{r}_t \right) \\ &= \mathbf{s}^\top \sum_{i \in [t]} \mathbf{p}_i + e_3 + \mu \lfloor q/2 \rfloor - \sum_{i \in [t]} (\mathbf{s}^\top [\mathbf{A}|\mathbf{D}_i] + [\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,i}^\top]) \mathbf{k}_{\rho(i),t} + \\ & \quad \left(\mathbf{s}^\top \sum_{i \in [t]} \mathbf{B}_i + \mathbf{e}_2^\top [\mathbf{I}_{mt}|\mathbf{R}] \right) \mathbf{r}_t \\ &= \mathbf{s}^\top \sum_{i \in [t]} \mathbf{p}_i + e_3 + \mu \lfloor q/2 \rfloor - \sum_{i \in [t]} (\mathbf{s}^\top (\mathbf{B}_i \mathbf{r}_t + \mathbf{p}_i) + [\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,i}^\top] \mathbf{k}_{\rho(i),t}) + \\ & \quad \left(\mathbf{s}^\top \sum_{i \in [t]} \mathbf{B}_i + \mathbf{e}_2^\top [\mathbf{I}_{mt}|\mathbf{R}] \right) \mathbf{r}_t \\ &= \underbrace{\mu \lfloor q/2 \rfloor + e_3 - \sum_{i \in [t]} [\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,i}^\top] \mathbf{k}_{\rho(i),t} + \mathbf{e}_2^\top [\mathbf{I}_{mt}|\mathbf{R}] \mathbf{r}_t}_e \end{aligned}$$

where $[\mathbf{D}_i|\mathbf{B}_i|\mathbf{p}_i] = H_t(x_i)$, $\mathbf{e}_1^\top = [\mathbf{e}_{1,0}^\top \cdots \mathbf{e}_{1,t}^\top]$. From the parameter setting and [Lemma 2.2](#), we have $|e_3| \leq \sqrt{\lambda}\chi$, $\|[\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,i}^\top]\| \leq \sqrt{\lambda}\chi$, $\|\mathbf{e}_2\| \leq \sqrt{\lambda}\chi$, and $\|\mathbf{r}_t\| \leq \sqrt{\lambda}\chi$ with overwhelming probability. Thanks to the property of lattice trapdoors, the distribution of $\mathbf{k}_{\rho(i),t}$ is statistically close to $[\mathbf{A}|\mathbf{D}_i]_\chi^{-1}(\mathbf{B}_i \mathbf{r}_t + \mathbf{p}_i)$. Since the marginal distribution of $\mathbf{p}_i$ is uniform in $\mathbb{Z}_q^n$, the marginal distribution of $\mathbf{k}_{\rho(i),t}$ is statistically close to $D_{\mathbb{Z}, \chi}^{2m}$ from [Lemma 2.3](#). Thus, we have $\|\mathbf{k}_{\rho(i),t}\| \leq \sqrt{\lambda}\chi$ with overwhelming probability again from

Lemma 2.2. In summary, we have, with overwhelming probability,

$$\begin{aligned} |\tilde{e}| &= \left| e_3 - \sum_{i \in [t]} [\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,i}^\top] \mathbf{k}_{\rho(i),t} + \mathbf{e}_2^\top [\mathbf{I}_{mt} | \mathbf{R}] \mathbf{r}_t \right| \\ &\leq \sqrt{\lambda} \chi + 2mt\lambda\chi^2 + m^2t(2t-1)\lambda\chi^2 \\ &\leq \sqrt{\lambda} \chi + mL(2 + 2mL - m)\lambda\chi^2 \end{aligned}$$

Thus, $|\tilde{e}| \leq q/5$ with overwhelming probability and correctness holds. $\square$

Security Analysis. For security, we have the following theorem.

Theorem 3.4. Suppose $m = O(n \log q)$, $\chi = 2m^2L^2 \cdot \omega(\sqrt{\log n})$, the hash function H is modeled as a random oracle, t is the size of the attribute X for the challenge query, and $\text{LWE}_{n,m(2t+1)+1,q,\chi}$ holds. Then, our scheme satisfies selective security as per [Definition 2.1](#).

Proof. We prove the theorem by a series of hybrids described as follows.

- H_0 : This is the original game. In this game, $\mathcal{A}$ first outputs the challenge attribute $X = (x_1, \dots, x_t)$. Then, the challenger computes

$$(\text{mpk}, \text{msk}) = (\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \text{TrapGen}(1^n, m, q)$$

and gives mpk to $\mathcal{A}$. When $\mathcal{A}$ asks for a secret key for $Y = (y_1, \dots, y_u)$, the challenger computes $\mathbf{r}_\ell \leftarrow D_{\mathbb{Z}, \chi}^{m(2\ell-1)}$ for $\ell \in [L]$ and, for $(i, \ell) \in [u] \times [L]$,

$$\begin{aligned} [\underbrace{\widehat{\mathbf{D}}_{i,\ell}}_m \mid \underbrace{\widehat{\mathbf{B}}_{i,\ell}}_{m(2\ell-1)} \mid \underbrace{\widehat{\mathbf{p}}_{i,\ell}}_1] &= H_\ell(y_i), \\ \mathbf{k}_{i,\ell} &\leftarrow \text{SampleLeft}([\mathbf{A} | \widehat{\mathbf{D}}_{i,\ell}], \mathbf{T}_\mathbf{A}, \widehat{\mathbf{B}}_{i,\ell} \mathbf{r}_\ell + \widehat{\mathbf{p}}_{i,\ell}, \chi) \end{aligned}$$

and gives $\text{sk} = \{\mathbf{k}_{1,\ell}, \dots, \mathbf{k}_{u,\ell}, \mathbf{r}_\ell\}_{\ell \in [L]}$ to $\mathcal{A}$. For the challenge, $\mathcal{A}$ outputs μ_0, μ_1 . Then, the challenger chooses a challenge bit $\beta \leftarrow \{0, 1\}$ and gives ct to $\mathcal{A}$ where

$$\begin{aligned} \mathbf{s} &\leftarrow \mathbb{Z}_q^n, \mathbf{e}_1 \leftarrow D_{\mathbb{Z}, \chi}^{(t+1)m}, \mathbf{e}_2 \leftarrow D_{\mathbb{Z}, \chi}^m, e_3 \leftarrow D_{\mathbb{Z}, \chi}, \\ \mathbf{R} &\leftarrow \{-1, 1\}^{mt \times m(t-1)} \\ [\underbrace{\mathbf{D}_i}_m \mid \underbrace{\mathbf{B}_i}_{m(2t-1)} \mid \underbrace{\mathbf{p}_i}_1] &= H_t(x_i), \mathbf{D} = [\mathbf{D}_1 \mid \dots \mid \mathbf{D}_t] \\ \text{ct} &= (\mathbf{c}_1^\top, \mathbf{c}_2^\top, c_3) \\ &= \left(\mathbf{s}^\top [\mathbf{A} | \mathbf{D}] + \mathbf{e}_1^\top, \mathbf{s}^\top \sum_{i \in [t]} \mathbf{B}_i + \mathbf{e}_2^\top [\mathbf{I}_{mt} | \mathbf{R}], \mathbf{s}^\top \sum_{i \in [t]} \mathbf{p}_i + e_3 + \mu_\beta \lfloor q/2 \rfloor \right) \end{aligned}$$

The post-challenge key queries are the same as the pre-challenge queries. Finally, $\mathcal{A}$ outputs β' . During the game, the challenger simulates the random oracle as follows. It maintains a dictionary $\mathcal{D}_\ell$ for $\ell \in [L]$ that are initially empty. When H_ℓ is invoked on input x , the challenger checks if $\mathcal{D}_\ell(x) = \perp$. If this is the case, the challenger samples $[\mathbf{D} | \mathbf{B} | \mathbf{p}] \leftarrow \mathbb{Z}_q^{n \times (2m\ell+1)}$, updates $\mathcal{D}_\ell(x) = [\mathbf{D} | \mathbf{B} | \mathbf{p}]$, and returns $[\mathbf{D} | \mathbf{B} | \mathbf{p}]$. Otherwise, it returns $\mathcal{D}_\ell(x)$.

- H_1 : This is the same as H_0 except that the challenger simulates the random oracle as follows. Right after $\mathcal{A}$ outputs the challenge attribute $X = (x_1, \dots, x_t)$, the challenger computes $[\mathbf{D}_i | \mathbf{B}_i | \mathbf{p}_i] \leftarrow \mathbb{Z}_q^{n \times (2mt+1)}$ and updates $\mathcal{D}_t(x_i) = [\mathbf{D}_i | \mathbf{B}_i | \mathbf{p}_i]$ for all $i \in [t]$. During the game, when H_ℓ is invoked on input x , the challenger checks if $\mathcal{D}_\ell(x) = \perp$. If this is the case, the challenger samples $(\mathbf{D}, \mathbf{T}_\mathbf{D}) \leftarrow \text{TrapGen}(1^n, m, q)$, $[\mathbf{B} | \mathbf{p}] \leftarrow \mathbb{Z}_q^{n \times ((2\ell-1)m+1)}$, updates $\mathcal{D}_\ell(x) = [\mathbf{D} | \mathbf{B} | \mathbf{p}]$, and returns $\mathcal{D}_\ell(x)$. Otherwise, it returns $\mathcal{D}_\ell(x)$. Since the property of lattice trapdoors ([Lemma 2.1](#)), $\mathbf{D}$ is statistically close to uniform, and thus H_0 and H_1 are statistically indistinguishable given $m = O(n \log q)$.

- H_2 : This is the same as H_1 except the followings. In this hybrid, the challenger samples $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ instead of using `TrapGen` for `mpk`. For all secret key queries on $Y = (y_1, \dots, y_u)$, let $Y_X = Y \cap X$ and $Y_{\bar{X}} = Y \setminus X$. The challenger computes for $(i, \ell) \in [u] \times [L]$,

$$\begin{aligned} \mathbf{r}_\ell &\leftarrow D_{\mathbb{Z}, \chi}^{m(2t-1)}, \quad [\underbrace{\widehat{\mathbf{D}}_{i,\ell}}_m \mid \underbrace{\widehat{\mathbf{B}}_{i,\ell}}_{m(2t-1)} \mid \underbrace{\widehat{\mathbf{p}}_{i,\ell}}_1] = H_\ell(y_i) \\ \mathbf{k}_{i,\ell} &\leftarrow \begin{cases} [\mathbf{A} \mid \widehat{\mathbf{D}}_{i,\ell}]_\chi^{-1} (\widehat{\mathbf{B}}_{i,\ell} \mathbf{r}_\ell + \widehat{\mathbf{p}}_{i,\ell}) & (i : y_i \in Y_X \wedge \ell = t) \\ \text{SampleLeft}([\mathbf{A} \mid \widehat{\mathbf{D}}_{i,\ell}], \mathbf{T}_{\widehat{\mathbf{D}}_{i,\ell}}, \widehat{\mathbf{B}}_{i,\ell} \mathbf{r}_\ell + \widehat{\mathbf{p}}_{i,\ell}, \chi) & (\text{otherwise}) \end{cases} \end{aligned}$$

and gives $\text{sk} = \{\mathbf{k}_{1,\ell}, \dots, \mathbf{k}_{u,\ell}, \mathbf{r}_\ell\}_{\ell \in [L]}$ to $\mathcal{A}$. Note that the sampling of $\mathbf{k}_{i,t}$ for $i : y_i \in Y_X$ is inefficient. It is easy to see that $H_1 \approx_s H_2$ since $\mathbf{A}$ is statistically close to uniform in H_1 and, for all $(i, \ell) \in [u] \times [L]$,

$$\begin{aligned} &\text{SampleLeft}([\mathbf{A} \mid \widehat{\mathbf{D}}_{i,\ell}], \mathbf{T}_{\widehat{\mathbf{D}}_{i,\ell}}, \widehat{\mathbf{B}}_{i,\ell} \mathbf{r}_\ell + \widehat{\mathbf{p}}_{i,\ell}, \chi) \\ &\approx_s \text{SampleLeft}([\mathbf{A} \mid \widehat{\mathbf{D}}_{i,\ell}], \mathbf{T}_{\mathbf{A}}, \widehat{\mathbf{B}}_{i,\ell} \mathbf{r}_\ell + \widehat{\mathbf{p}}_{i,\ell}, \chi) \\ &\approx_s [\mathbf{A} \mid \widehat{\mathbf{D}}_{i,\ell}]_\chi^{-1} (\widehat{\mathbf{B}}_{i,\ell} \mathbf{r}_\ell + \widehat{\mathbf{p}}_{i,\ell}) \end{aligned}$$

which follows from [Lemma 2.1](#) if $m = O(n \log q)$ (precisely, the first indistinguishability applies to all the cases except for $i : y_i \in Y_X \wedge \ell = t$).

- H_3 : This game is the same as H_2 except the followings. For all secret key queries on $Y = (y_1, \dots, y_u)$, let $Y_X = Y \cap X$, $Y_{\bar{X}} = Y \setminus X$, and $Y_X = (y_{i_1}, \dots, y_{i_k})$. The challenger computes

$$[\underbrace{\widehat{\mathbf{D}}_{i,t}}_m \mid \underbrace{\widehat{\mathbf{B}}_{i,t}}_{m(2t-1)} \mid \underbrace{\widehat{\mathbf{p}}_{i,t}}_1] = H_t(y_i) \quad \text{for } i \in [u], \quad (5)$$

$$\begin{aligned} \mathbf{k}_{i,t} &\leftarrow D_{\mathbb{Z}, \chi}^{2m} \quad \text{for } i : y_i \in Y_X \\ \mathbf{r}_t &\leftarrow \begin{bmatrix} \widehat{\mathbf{B}}_{i_1,t} \\ \vdots \\ \widehat{\mathbf{B}}_{i_k,t} \end{bmatrix}_\chi^{-1} \left(\begin{bmatrix} [\mathbf{A} \mid \widehat{\mathbf{D}}_{i_1,t}] \mathbf{k}_{i_1,t} - \widehat{\mathbf{p}}_{i_1,t} \\ \vdots \\ [\mathbf{A} \mid \widehat{\mathbf{D}}_{i_k,t}] \mathbf{k}_{i_k,t} - \widehat{\mathbf{p}}_{i_k,t} \end{bmatrix} \right) \\ \mathbf{k}_{i,t} &\leftarrow \text{SampleLeft}([\mathbf{A} \mid \widehat{\mathbf{D}}_{i,t}], \mathbf{T}_{\widehat{\mathbf{D}}_{i,t}}, \widehat{\mathbf{B}}_{i,t} \mathbf{r}_t + \widehat{\mathbf{p}}_{i,t}, \chi) \quad \text{for } i : y_i \in Y_{\bar{X}} \end{aligned} \quad (6)$$

and gives $\text{sk} = \{\mathbf{k}_{1,\ell}, \dots, \mathbf{k}_{u,\ell}, \mathbf{r}_\ell\}_{\ell \in [L]}$ to $\mathcal{A}$ where $\{\mathbf{k}_{1,\ell}, \dots, \mathbf{k}_{u,\ell}, \mathbf{r}_\ell\}_{\ell \in [L] \setminus \{t\}}$ is generated in the same way as in H_2 . We prove $H_2 \approx_s H_3$ in [Lemma 3.4](#).

- H_4 : This game is the same as H_3 except the followings. The challenger generates the challenge ciphertext as

$$\text{ct} = (\mathbf{c}_1^\top, \mathbf{c}_2^\top, c_3) \leftarrow \mathbb{Z}_q^{(t+1)m} \times \mathbb{Z}_q^{m(2L-1)} \times \mathbb{Z}_q$$

We prove $H_3 \approx_c H_4$ in [Lemma 3.5](#).

It is obvious that $\mathcal{A}$ does not obtain the information on β , and thus its advantage is 0 in the last hybrid. This concludes the proof. $\square$

Lemma 3.4. $H_2 \approx_s H_3$ if $m = O(n \log q)$ and $\chi = \omega(\sqrt{\log m})$.

Proof. Let Q be the number of secret key queries, $Y^{(j)} = (y_1^{(j)}, \dots, y_{u^{(j)}}^{(j)})$ be the attribute for the j -th secret key query, let $Y_X^{(j)} = Y^{(j)} \cap X$, $|Y_X^{(j)}| = k^{(j)}$, and $Y_{\bar{X}}^{(j)} = (y_{i_1}^{(j)}, \dots, y_{i_k}^{(j)})$. For $y_{i_1}^{(j)}$ and $y_{i_k}^{(j)}$ in the previous sentence, we omit the index (j) for conciseness in some parts, e.g. i_k should be $i_{k^{(j)}}$ precisely. We sometimes omit (j) similarly in what follows.

Let $\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_t \end{bmatrix}$ and $\mathbf{p} = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_t \end{bmatrix}$ where $[\mathbf{D}_i \mid \mathbf{B}_i \mid \mathbf{p}_i] = H_t(x_i)$. Thanks to [Lemma 2.3](#), $\mathbf{Br}_t^{(1)}, \dots, \mathbf{Br}_t^{(Q)}$

where $\mathbf{r}_t^{(j)} \leftarrow D_{\mathbb{Z}, \chi}^{m(2L-1)}$ for $j \in [Q]$ are distributed statistically close to uniform in $(\mathbb{Z}_q^{nt})^Q$, because $t \leq L$, $m = O(n \log q)$, and $\chi = \omega(\sqrt{\log m})$. Hence, the joint distribution of $\widehat{\mathbf{B}}^{(j)} \mathbf{r}_t^{(j)} + \widehat{\mathbf{p}}^{(j)}$ for all $j \in [Q]$ are also statistically close to uniform random, where $[\widehat{\mathbf{D}}_i^{(j)} \mid \widehat{\mathbf{B}}_i^{(j)} \mid \widehat{\mathbf{p}}_i^{(j)}] = H_t(y_i^{(j)})$ for $i \in \{i_1, \dots, i_k\}$ and

$$\widehat{\mathbf{B}}^{(j)} = \begin{bmatrix} \widehat{\mathbf{B}}_{i_1}^{(j)} \\ \vdots \\ \widehat{\mathbf{B}}_{i_k}^{(j)} \end{bmatrix}, \quad \widehat{\mathbf{p}}^{(j)} = \begin{bmatrix} \widehat{\mathbf{p}}_{i_1}^{(j)} \\ \vdots \\ \widehat{\mathbf{p}}_{i_k}^{(j)} \end{bmatrix}$$

since $\widehat{\mathbf{B}}_i^{(j)} \in \{\mathbf{B}_1, \dots, \mathbf{B}_t\}$ and $\widehat{\mathbf{p}}_i^{(j)} \in \{\mathbf{p}_1, \dots, \mathbf{p}_t\}$ for all $i : y_i^{(j)} \in Y_X^{(j)}, j \in [Q]$. Recall also that $\mathbf{A}$ is uniformly distributed in $\mathbb{Z}_q^{n \times m}$. Thanks to [Lemma 3.1](#) for $\ell_1 = |X|$ and $\ell_2 = Q$, the joint distribution of $(\mathbf{A}, \mathbf{k}_{i,t}^{(j)}, \widehat{\mathbf{B}}_i^{(j)} \mathbf{r}_t^{(j)} + \widehat{\mathbf{p}}_i^{(j)})$ for $i : y_i^{(j)} \in Y_X^{(j)}$ and $j \in [Q]$ in $\mathbf{H}_2$ is statistically close to the distribution of $(\mathbf{A}, \mathbf{k}_{i,t}^{(j)}, \widehat{\mathbf{B}}_i^{(j)} \mathbf{r}_t^{(j)} + \widehat{\mathbf{p}}_i^{(j)})$ where we first sample $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{k}_{i,t}^{(j)} \leftarrow D_{\mathbb{Z}, \chi}^{2m}$ and then sample

$$\mathbf{B} \leftarrow \mathbb{Z}_q^{nt \times m(2L-1)}, \mathbf{r}_t^{(j)} \leftarrow D_{\mathbb{Z}, \chi}^{m(2L-1)}, \mathbf{p} \leftarrow \mathbb{Z}_q^{nt}$$

conditioned on $[\mathbf{A}|\widehat{\mathbf{D}}_i^{(j)}]\mathbf{k}_{i,t}^{(j)} = \widehat{\mathbf{B}}_i^{(j)} \mathbf{r}_t^{(j)} + \widehat{\mathbf{p}}_i^{(j)}$ for $i : y_i \in Y_X^{(j)}$ and $j \in [Q]$. This is conceptually the same as sampling $\mathbf{B} \leftarrow \mathbb{Z}_q^{nt \times m(2L-1)}, \mathbf{p} \leftarrow \mathbb{Z}_q^{nt}$, and then sampling $\mathbf{r}_t^{(j)}$ for each secret key query as [Eq. \(6\)](#). Recall that the difference between $\mathbf{H}_2$ and $\mathbf{H}_3$ lies in $\mathbf{k}_{i,t}^{(j)}, \mathbf{r}_t^{(j)}$ for $i : y_i^{(j)} \in Y_X^{(j)}, j \in [Q]$. Thus, we have $\mathbf{H}_2 \approx_s \mathbf{H}_3$. $\square$

Lemma 3.5. $\mathbf{H}_3 \approx_c \mathbf{H}_4$ under the $\text{LWE}_{n, m(2t+1)+1, q, \chi}$ assumption if $\chi = 2m^2 L^2 \cdot \omega(\sqrt{\log n})$ and t is the size of the attribute X for the challenge query.

Proof. We prove that if there is an adversary $\mathcal{A}$ that distinguishes $\mathbf{H}_3$ and $\mathbf{H}_4$ with non-negligible probability, we can construct a reduction $\mathcal{B}$ that breaks the $\text{RTLWE}_{n, m+tm+1, m, q, \chi, t}$ assumption. Then, [Lemma 2.4](#) immediately implies [Lemma 3.5](#). We can construct $\mathcal{B}$ as follows.

- $\mathcal{A}$ declares the challenge attribute $X = (x_1, \dots, x_t)$ at the beginning of the game. $\mathcal{B}$ outputs $\mathbf{u}^\top = (\underbrace{1, \dots, 1}_t)$ for the security game for the related trapdoor LWE assumption and obtains $(\widehat{\mathbf{A}}, \mathbf{B}, \mathbf{v}^\top, \widehat{\mathbf{v}}^\top)$ where $(\mathbf{v}^\top, \widehat{\mathbf{v}}^\top) = (\mathbf{s}^\top \widehat{\mathbf{A}} + \mathbf{e}^\top, \mathbf{s}^\top (\mathbf{u}^\top \otimes \mathbf{I}_n) \mathbf{B} + \widehat{\mathbf{e}}^\top)$ or $(\mathbf{v}^\top, \widehat{\mathbf{v}}^\top) \leftarrow \mathbb{Z}_q^{m+tm+1} \times \mathbb{Z}_q^{m(2t-1)}$. $\mathcal{B}$ parses $\widehat{\mathbf{A}} = [\mathbf{A}|\mathbf{D}_1|\dots|\mathbf{D}_t|\mathbf{p}]$, $\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_t \end{bmatrix}$, and gives $\text{mpk} = \mathbf{A}$ to $\mathcal{A}$.
- Random oracle queries: $\mathcal{B}$ samples $\mathbf{p}_1, \dots, \mathbf{p}_t \leftarrow \mathbb{Z}_q^n$ conditioned on $\sum_{i \in [t]} \mathbf{p}_i = \mathbf{p}$. $\mathcal{B}$ sets $\mathcal{D}_t(x_i) = [\mathbf{D}_i|\mathbf{B}_i|\mathbf{p}_i]$ for all $i \in [t]$, where $\mathcal{D}_t$ is the dictionary for the random oracle simulation. Note that since $\mathbf{p}$ is kept hidden from the adversary, $\mathbf{p}_1, \dots, \mathbf{p}_t$ are distributed randomly from the view point of $\mathcal{A}$. During the game, when H_t is invoked on input x , the challenger checks if $\mathcal{D}_t(x) = \perp$. If this is the case, the challenger samples $(\mathbf{D}, \mathbf{T}_\mathbf{D}) \leftarrow \text{TrapGen}(1^n, m, q)$, $[\mathbf{B}|\mathbf{p}] \leftarrow \mathbb{Z}_q^{n \times ((2t-1)m+1)}$, updates $\mathcal{D}_t(x) = [\mathbf{D}|\mathbf{B}|\mathbf{p}]$, and returns $[\mathbf{D}|\mathbf{B}|\mathbf{p}]$. Otherwise, it just returns $\mathcal{D}_t(x)$. The simulation of H_ℓ for $\ell \neq t$ is the same as in $\mathbf{H}_0$.
- Secret key queries: when $\mathcal{A}$ queries $\mathcal{B}$ on $Y = (y_1, \dots, y_u)$, $\mathcal{B}$ defines $Y_X = Y \cap X$, $Y_{\bar{X}} = Y \setminus X$, $Y_X = (y_{i_1}, \dots, y_{i_k})$, $\mathbf{m}_i^\top = (0^{i-1}, 1, 0^{t-i}) \in \mathbb{Z}_t^t$ for $i \leq u$. Then, it computes

$$\begin{aligned} [\widehat{\mathbf{D}}_i|\widehat{\mathbf{B}}_i|\widehat{\mathbf{p}}_i] &= H_t(y_i) \quad \text{for } i \in [u] \\ \mathbf{k}_{i,t} &\leftarrow D_{\mathbb{Z}, \chi}^{2m} \quad \text{for } i : y_i \in Y_X \end{aligned}$$

and makes a query $\text{tdO}(\mathbf{M}, \mathbf{t})$ and obtains $\mathbf{r}_t \in \mathbb{Z}_q^{m(2t-1)}$ as a reply, where $\phi : \{i_1, \dots, i_k\} \rightarrow [t]$ is an injective function such that $\mathbf{B}_{\phi(i)} = \widehat{\mathbf{B}}_i$,

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_{\phi(i_1)}^\top \\ \vdots \\ \mathbf{m}_{\phi(i_k)}^\top \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} [\mathbf{A}|\widehat{\mathbf{D}}_{i_1}] \mathbf{k}_{i_1} - \widehat{\mathbf{p}}_{i_1} \\ \vdots \\ [\mathbf{A}|\widehat{\mathbf{D}}_{i_k}] \mathbf{k}_{i_k} - \widehat{\mathbf{p}}_{i_k} \end{bmatrix}$$

Finally, it computes $\mathbf{r}_\ell \leftarrow D_{\mathbb{Z}, \chi}^{m(2\ell-1)}$ for $\ell \in [L] \setminus \{t\}$ and

$$\mathbf{k}_{i,\ell} \leftarrow \text{SampleLeft}([\mathbf{A}|\widehat{\mathbf{D}}_{i,\ell}], \mathbf{T}_{\widehat{\mathbf{D}}_{i,\ell}}, \widehat{\mathbf{B}}_{i,\ell} \mathbf{r}_\ell + \widehat{\mathbf{p}}_{i,\ell}, \chi) \quad \text{for } i : y_i \in Y_{\bar{X}} \vee \ell \neq t$$

where $[\widehat{\mathbf{D}}_{i,\ell}|\widehat{\mathbf{B}}_{i,\ell}|\widehat{\mathbf{p}}_{i,\ell}] = H_\ell(y_i)$. Then, returns $\text{sk} = \{\mathbf{k}_{1,\ell}, \dots, \mathbf{k}_{u,\ell}, \mathbf{r}_\ell\}_{\ell \in [L]}$ to $\mathcal{A}$.

- Challenge: after $\mathcal{A}$ outputs the challenge messages (μ_0, μ_1) , $\mathcal{B}$ chooses $\beta \leftarrow \{0, 1\}$ and gives the challenge ciphertext $\text{ct} = (\mathbf{v}^\top + (0^{(t+1)m}, \mu_\beta \lfloor q/2 \rfloor), \widehat{\mathbf{v}}^\top)$.
- Output: if $\mathcal{A}$ outputs β' at the end of the game, $\mathcal{B}$ outputs 1 if and only if $\beta = \beta'$.

Observe that $\mathcal{A}$'s view corresponds to $\mathbf{H}_3$ if $(\mathbf{v}^\top, \widehat{\mathbf{v}}^\top) = (\mathbf{s}^\top \widehat{\mathbf{A}} + \mathbf{e}^\top, \mathbf{s}^\top (\mathbf{u}^\top \otimes \mathbf{I}_n) \mathbf{B} + \widehat{\mathbf{e}}^\top)$ and $\mathbf{H}_4$ otherwise. Additionally, if $\mathcal{A}$ is admissible, then $\mathcal{B}$ is admissible. This is because $Y_X \subsetneq X$ for all Y in secret key queries, and thus it is obvious that $k < t$ and thus $\{\mathbf{u}, \mathbf{m}_{\phi(i_1)}, \dots, \mathbf{m}_{\phi(i_k)}\}$ are linearly independent. This concludes the proof. $\square$

4 Large-Universal MA-ABE for Subset Policies

We define large-universal MA-ABE for subset policies as follows.

Definition 4.1 (Large-Universal MA-ABE for Subset Policies). Let $\mathcal{U}_\lambda$ be an attribute universe such that $|\mathcal{U}_\lambda| = \lambda^{\omega(1)}$, i.e., the size is superpolynomial. Large-universal MA-ABE for subset policies is MA-ABE as per [Definition 2.2](#) for the following predicate $P : \mathcal{X} \times \mathbb{Y} \rightarrow \{0, 1\}$:

$$\mathcal{X} = (2^{\mathcal{U}_\lambda})^{\mathcal{A}\mathcal{U}}, \quad \mathcal{Y} = \mathcal{U}_\lambda, \quad P(X, Y) = 1 \Leftrightarrow X \subseteq Y$$

where we view $\phi \in \mathcal{X}$ as a set $X = \{(\alpha, x) \in \mathcal{A}\mathcal{U} \times \mathcal{X} \mid x \in \phi(\alpha)\}$ similar to [Definition 2.2](#), and $Y \in \mathbb{Y} = (2^{\mathcal{Y}})^{\mathcal{A}\mathcal{U}}$. It is not hard to see that P satisfies the three conditions in [Definition 2.2](#) by defining $A_X = f(\phi) = \{\alpha \mid \phi(\alpha) \neq \emptyset\}$ for $\phi \in \mathcal{X}$.

Remark 4.1 (Large-Universal MA-ABE for DNFs). Large-universal MA-ABE for DNFs is defined as MA-ABE for $P : \mathcal{X} \times \mathbb{Y} \rightarrow \{0, 1\}$ where

$$\begin{aligned} \mathcal{X} &= \bigcup_{i \in \mathbb{N}} ((2^{\mathcal{U}_\lambda})^{\mathcal{A}\mathcal{U}})^i, \quad \mathcal{Y} = \mathcal{U}_\lambda, \\ P((X_1, \dots, X_s), Y) &= 1 \Leftrightarrow \bigvee_{i \in [s]} X_i \subseteq Y \end{aligned}$$

The conversion from large-universal MA-ABE for subset policies to that for DNFs is similar to the case for CP-ABE as described in [Remark 3.3](#).

In our construction, we have a restriction such that $|X| \leq L$ where L is a parameter that is polynomial in λ .

Construction. Let $n, m, q \in \mathbb{N}$ and $\chi \in \mathbb{R}^+$ be lattice parameters that depend on the security parameter λ , where $q > 2$ is a prime. Let the message space $\mathcal{M} = \{0, 1\}$. Let L be the maximum size of the set used in Enc. For $\ell \in [L]$, let $H_{1,\ell} : \{0, 1\}^* \rightarrow \mathbb{Z}_q^{n \times (2m\ell+1)}$ be hash functions. For $\ell \in [L]$, let $H_{2,\ell} : \mathcal{GU} \rightarrow \mathbb{Z}_q^{m(2\ell-1)}$ be a hash function modeled as a random oracle, the outputs of which are distributed according to a discrete Gaussian distribution, i.e., for each gid , $H_{2,\ell}(\text{gid})$ is sampled from $D_{\mathbb{Z}, \chi}^{m(2\ell-1)}$. Waters, Wee, and Wu showed how to implement such a random oracle from a standard random oracle that outputs uniform distribution [\[WWW22\]](#). Hence, we directly use $H_{2,\ell}$ in our construction.

GSetup(1^λ): It outputs $\text{gp} = (n, m, q, L, \chi, \{H_{1,\ell}, H_{2,\ell}\}_{\ell \in [L]})$.

ASetup(α): It outputs mpk and msk as follows. It runs $(\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \text{TrapGen}(1^n, m, q)$ and output these as $(\text{mpk}, \text{msk}) = (\mathbf{A}, (\alpha, \mathbf{A}, \mathbf{T}_\mathbf{A}))$.

Enc($X, \{\text{mpk}_\alpha\}_{\alpha \in A_X}, \mu$): It parses $X = \{(\alpha_1, x_1), \dots, (\alpha_t, x_t)\}$ and $\text{mpk}_\alpha = \mathbf{A}_\alpha$, where it must hold that $t < L$, and outputs $\text{ct} = (\{\mathbf{c}_{1,i}^\top\}_{i \in [t]}, \mathbf{c}_2^\top, c_3)$ as follows.⁸

$$\begin{aligned} \mathbf{s}_\alpha &\leftarrow \mathbb{Z}_q^n, \quad \tilde{\mathbf{e}}_{1,\alpha} \leftarrow D_{\mathbb{Z}, \chi}^m \quad \text{for } \alpha \in A_X \\ \hat{\mathbf{e}}_{1,i} &\leftarrow D_{\mathbb{Z}, \chi}^m, \quad \underbrace{[\mathbf{D}_i]}_m \mid \underbrace{[\mathbf{B}_i]}_{m(2t-1)} \mid \underbrace{[\mathbf{p}_i]}_1 = H_{1,t}((\alpha_i, x_i)) \quad \text{for } i \in [t] \\ \mathbf{e}_{1,i}^\top &= [\tilde{\mathbf{e}}_{1,\alpha_i}^\top \mid \hat{\mathbf{e}}_{1,i}^\top], \quad \mathbf{e}_2 \leftarrow D_{\mathbb{Z}, \chi}^{mt}, \quad \mathbf{e}_3 \leftarrow D_{\mathbb{Z}, \chi}, \quad \mathbf{R} \leftarrow \{-1, 1\}^{mt \times m(t-1)} \\ \mathbf{c}_{1,i}^\top &= \mathbf{s}_{\alpha_i}^\top [\mathbf{A}_{\alpha_i} \mid \mathbf{D}_i] + \mathbf{e}_{1,i}^\top \\ \mathbf{c}_2^\top &= \sum_{i \in [t]} \mathbf{s}_{\alpha_i}^\top \mathbf{B}_i + \mathbf{e}_2^\top [\mathbf{I}_{mt} \mid \mathbf{R}], \quad c_3 = \sum_{i \in [t]} \mathbf{s}_{\alpha_i}^\top \mathbf{p}_i + e_3 + \mu \lfloor q/2 \rfloor \end{aligned}$$

KeyGen($\text{msk}, \text{gid}, y$): It parses $\text{msk} = (\alpha, \mathbf{A}, \mathbf{T}_\mathbf{A})$, computes $\mathbf{r}_\ell = H_{2,\ell}(\text{gid})$, and

$$\begin{aligned} \underbrace{[\hat{\mathbf{D}}_\ell]}_m \mid \underbrace{[\hat{\mathbf{B}}_\ell]}_{m(2\ell-1)} \mid \underbrace{[\hat{\mathbf{p}}_\ell]}_1 &= H_{1,\ell}((\alpha, y)), \\ \mathbf{k}_\ell &\leftarrow \text{SampleLeft}([\mathbf{A} \mid \hat{\mathbf{D}}_\ell], \mathbf{T}_\mathbf{A}, \hat{\mathbf{B}}_\ell \mathbf{r}_\ell + \hat{\mathbf{p}}_\ell, \chi) \end{aligned}$$

for $\ell \in [L]$. Then, it outputs $\text{sk} = (\{\mathbf{k}_\ell\}_{\ell \in [L]}, \text{gid})$.

⁸ Note that the first m elements of $\mathbf{c}_{1,i}^\top$ may be redundant if $\alpha_i = \alpha_j$ for some $i \neq j$, but we use this notation for convenience.

Dec($\text{ct}, \{\text{sk}_{\alpha,y}\}_{(\alpha,y) \in Y}, X, Y$): It parses $\text{ct} = (\{\mathbf{c}_{1,i}^\top\}_{i \in [t]}, \mathbf{c}_2^\top, c_3)$, $\text{sk}_{\alpha,y,\ell} = (\{\mathbf{k}_{\alpha,y,\ell}\}_{\ell \in [L]}, \text{gid}_{\alpha,y})$, $X = \{(\alpha_1, x_1), \dots, (\alpha_t, x_t)\}$, and $Y = \{(\alpha'_1, y_1), \dots, (\alpha'_t, y_t)\}$. If there exists $(\alpha, y), (\alpha', y') \in Y$ such that $\text{gid}_{\alpha,y} \neq \text{gid}_{\alpha',y'}$, it outputs $\perp$. If $X \subseteq Y$, it defines $\rho: [t] \rightarrow Y$ such that $\rho(i) = (\alpha_i, x_i)$ and outputs

$$\left\lfloor \frac{2}{q} \cdot \left(c_3 - \left(\sum_{i \in [t]} \mathbf{c}_{1,i}^\top \mathbf{k}_{\rho(i),t} - \mathbf{c}_2^\top H_{2,t}(\text{gid}) \right) \right) \right\rfloor$$

Otherwise, output $\perp$.

Parameter Selection. From [Theorem 4.1](#) and [Theorem 4.2](#), we can choose parameters as follows:

$$n = \lambda, \quad q = O(n^8 L^7), \quad m = O(n \log q), \quad \chi = O(m^2 L^2 \log n)$$

Observe that we can choose q, χ as polynomials in λ . Hence, our scheme is secure under the LWE with a polynomial modulus-to-noise ratio.

Correctness. For correctness, we have the following theorem.

Theorem 4.1. *Suppose $q \geq 5(\sqrt{\lambda}\chi + mL(2 + 2mL - m)\lambda\chi^2)$, $m = O(n \log q)$, and $\chi = \omega(\sqrt{\log m})$. Then, our scheme is correct.*

Proof. In decryption, suppose $\text{ct} = (\{\mathbf{c}_{1,i}^\top\}_{i \in [t]}, \mathbf{c}_2^\top, c_3)$ and $\{\text{sk}_{\alpha,y} = (\{\mathbf{k}_{\alpha,y,\ell}\}_{\ell \in [L]}, \text{gid})\}_{(\alpha,y) \in Y}$ are a ciphertext and a secret key for $X \in \mathcal{X}$ and $(Y, \text{gid}) \in \mathbb{Y} \times \mathcal{GU}$, respectively, such that $X \subseteq Y$. In decryption, we have

$$\begin{aligned} & c_3 - \left(\sum_{i \in [t]} \mathbf{c}_{1,i}^\top \mathbf{k}_{\rho(i),t} - \mathbf{c}_2^\top H_{2,t}(\text{gid}) \right) \\ &= \sum_{i \in [t]} \mathbf{s}_{\alpha_i}^\top \mathbf{p}_i + e_3 + \mu \lfloor q/2 \rfloor - \sum_{i \in [t]} (\mathbf{s}_{\alpha_i}^\top [\mathbf{A}_{\alpha_i} | \mathbf{D}_i] + \mathbf{e}_{1,i}^\top) \mathbf{k}_{\rho(i),t} + \left(\sum_{i \in [t]} \mathbf{s}_{\alpha_i}^\top \mathbf{B}_i + \mathbf{e}_2^\top [\mathbf{I}_{mL} | \mathbf{R}] \right) \mathbf{r}_t \\ &= \sum_{i \in [t]} \mathbf{s}_{\alpha_i}^\top \mathbf{p}_i + e_3 + \mu \lfloor q/2 \rfloor - \sum_{i \in [t]} (\mathbf{s}_{\alpha_i}^\top (\mathbf{B}_i \mathbf{r}_t + \mathbf{p}_i) + \mathbf{e}_{1,i}^\top \mathbf{k}_{\rho(i),t}) + \left(\sum_{i \in [t]} \mathbf{s}_{\alpha_i}^\top \mathbf{B}_i + \mathbf{e}_2^\top [\mathbf{I}_{mL} | \mathbf{R}] \right) \mathbf{r}_t \\ &= \underbrace{\mu \lfloor q/2 \rfloor + e_3 - \sum_{i \in [t]} \mathbf{e}_{1,i}^\top \mathbf{k}_{\rho(i),t} + \mathbf{e}_2^\top [\mathbf{I}_{mL} | \mathbf{R}] \mathbf{r}_t}_{\tilde{e}} \end{aligned}$$

where $[\mathbf{D}_i | \mathbf{B}_i | \mathbf{p}_i] = H_{1,t}((\alpha_i, x_i))$, $\mathbf{r}_t = H_{2,t}(\text{gid})$. From the parameter setting and [Lemma 2.2](#), we have $|e_3| \leq \sqrt{\lambda}\chi$, $\|\mathbf{e}_{1,i}\| \leq \sqrt{\lambda}\chi$, $\|\mathbf{e}_2\| \leq \sqrt{\lambda}\chi$, and $\|\mathbf{r}_t\| \leq \sqrt{\lambda}\chi$ with overwhelming probability. Thanks to the property of lattice trapdoors, the distribution of $\mathbf{k}_{\rho(i)}$ is statistically close to $[\mathbf{A}_{\alpha_i} | \mathbf{D}_i]_\chi^{-1} (\mathbf{B}_i \mathbf{r}_t + \mathbf{p}_i)$. Since the marginal distribution of $\mathbf{p}_i$ is uniform in $\mathbb{Z}_q^n$, the marginal distribution of $\mathbf{k}_{\rho(i),t}$ is statistically close to $D_{\mathbb{Z}, \chi}^{2m}$ from [Lemma 2.3](#). Thus, we have $\|\mathbf{k}_{\rho(i),t}\| \leq \sqrt{\lambda}\chi$ with overwhelming probability again from [Lemma 2.2](#). In summary, we have, with overwhelming probability,

$$\begin{aligned} |\tilde{e}| &= \left| e_3 - \sum_{i \in [t]} \mathbf{e}_{1,i}^\top \mathbf{k}_{\rho(i),t} + \mathbf{e}_2^\top [\mathbf{I}_{mL} | \mathbf{R}] \mathbf{r}_t \right| \leq \sqrt{\lambda}\chi + 2mt\lambda\chi^2 + m^2t(2t-1)\lambda\chi^2 \\ &\leq \sqrt{\lambda}\chi + mL(2 + 2mL - m)\lambda\chi^2 \end{aligned}$$

Thus, $|\tilde{e}| \leq q/5$ with overwhelming probability and correctness holds. $\square$

Security Analysis. For security, we have the following theorem.

Theorem 4.2. *Suppose $m = O(n \log q)$, $\chi = 2m^2 L^2 \cdot \omega(\sqrt{\log n})$, the hash function H_1, H_2 are modeled as a random oracle, t is the size of the challenge attribute X , t' is the number of non-corrupted authorities related to the challenge attribute X , and $\text{LWE}_{n, m(2t+t')+1, q, \chi}$ holds. Then, our scheme satisfies static security as per [Definition 2.2](#).*

Proof. We prove the theorem by a series of hybrids described as follows.

- H_0 : This is the original game. In this game, $\mathcal{A}$ first outputs the challenge attribute $X = \{(\alpha_1^*, x_1^*), \dots, (\alpha_t^*, x_t^*)\}$, the challenge messages μ_0, μ_1 , a set of corrupted authorities $\mathcal{C} \subseteq \mathcal{AU}$ and their master public keys $\{\text{mpk}_\alpha\}_{\alpha \in \mathcal{C}}$ where $\text{mpk}_\alpha = \mathbf{A}_\alpha$, a set of non-corrupted authorities $\mathcal{N} \subseteq \mathcal{AU}$, secret key queries $\{\text{gid}, Y_{\text{gid}}\}_{\text{gid} \in \mathcal{Q}}$ where $\mathcal{Q} \subseteq \mathcal{GU}$ and $Y_{\text{gid}} \in \mathbb{Y}$ for all $\text{gid} \in \mathcal{Q}$. Then, the challenger computes for $\alpha \in \mathcal{N}$,

$$(\mathbf{A}_\alpha, \mathbf{T}_{\mathbf{A}_\alpha}) \leftarrow \text{TrapGen}(1^n, m, q), \quad (\text{mpk}_\alpha, \text{msk}_\alpha) = (\mathbf{A}_\alpha, (\alpha, \mathbf{A}_\alpha, \mathbf{T}_{\mathbf{A}_\alpha}))$$

It computes the challenge ciphertext $\text{ct}^* = (\{\mathbf{c}_{1,i}^\top\}_{i \in [t]}, \mathbf{c}_2^\top, c_3)$ as $b \leftarrow \{0, 1\}$ and

$$\begin{aligned} \mathbf{s}_\alpha &\leftarrow \mathbb{Z}_q^n, \quad \tilde{\mathbf{e}}_{1,\alpha} \leftarrow D_{\mathbb{Z}, \chi}^m \quad \text{for } \alpha \in A_X \\ \hat{\mathbf{e}}_{1,i} &\leftarrow D_{\mathbb{Z}, \chi}^m, \quad \underbrace{[\mathbf{D}_i]}_m \mid \underbrace{[\mathbf{B}_i]}_{m(2t-1)} \mid \underbrace{[\mathbf{p}_i]}_1 = H_{1,t}((\alpha_i^*, x_i^*)) \quad \text{for } i \in [t] \\ \mathbf{e}_{1,i}^\top &= [\tilde{\mathbf{e}}_{1,\alpha_i}^\top \mid \hat{\mathbf{e}}_{1,i}^\top], \quad \mathbf{e}_2 \leftarrow D_{\mathbb{Z}, \chi}^{mt}, \quad \mathbf{e}_3 \leftarrow D_{\mathbb{Z}, \chi}, \quad \mathbf{R} \leftarrow \{-1, 1\}^{mt \times m(t-1)} \\ \mathbf{c}_{1,i}^\top &= \mathbf{s}_{\alpha_i}^\top [\mathbf{A}_{\alpha_i} \mid \mathbf{D}_i] + \mathbf{e}_{1,i}^\top \\ \mathbf{c}_2^\top &= \sum_{i \in [t]} \mathbf{s}_{\alpha_i}^\top \mathbf{B}_i + \mathbf{e}_2^\top [\mathbf{I}_{mt} \mid \mathbf{R}], \quad c_3 = \sum_{i \in [t]} \mathbf{s}_{\alpha_i}^\top \mathbf{p}_i + e_3 + \mu_b \lfloor q/2 \rfloor \end{aligned}$$

It computes secret keys $\text{sk}_{\alpha,y,\text{gid}}$ for $(\alpha, y) \in Y_{\text{gid}}, \text{gid} \in \mathcal{Q}$ as $\mathbf{r}_\ell = H_{2,\ell}(\text{gid})$ and

$$\begin{aligned} \underbrace{[\hat{\mathbf{D}}_\ell]}_m \mid \underbrace{[\hat{\mathbf{B}}_\ell]}_{m(2\ell-1)} \mid \underbrace{[\hat{\mathbf{p}}_\ell]}_1 &= H_{1,\ell}((\alpha, y)), \\ \mathbf{k}_\ell &\leftarrow \text{SampleLeft}([\mathbf{A}_\alpha \mid \hat{\mathbf{D}}_\ell], \mathbf{T}_{\mathbf{A}_\alpha}, \hat{\mathbf{B}}_\ell \mathbf{r}_\ell + \hat{\mathbf{p}}_\ell, \chi), \\ \text{sk}_{\alpha,y,\text{gid}} &= (\{\mathbf{k}_\ell\}_{\ell \in [L]}, \text{gid}) \end{aligned}$$

Then, it gives $(\{\text{mpk}_\alpha\}_{\alpha \in \mathcal{N}}, \text{ct}^*, \{\text{sk}_{\alpha,y,\text{gid}}\}_{(\alpha,y) \in Y_{\text{gid}}, \text{gid} \in \mathcal{Q}})$ to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs b' . During the game, the challenger simulates the random oracle as follows. It maintains a dictionary $\mathcal{D}_{1,\ell}, \mathcal{D}_{2,\ell}$ that are initially empty. When $H_{1,\ell}$ is invoked on input x , the challenger checks if $\mathcal{D}_{1,\ell}(x) = \perp$. If this is the case, the challenger samples $[\mathbf{D} \mid \mathbf{B} \mid \mathbf{p}] \leftarrow \mathbb{Z}_q^{n \times (2m\ell+1)}$, updates $\mathcal{D}_{1,\ell}(x) = [\mathbf{D} \mid \mathbf{B} \mid \mathbf{p}]$, and returns $[\mathbf{D} \mid \mathbf{B} \mid \mathbf{p}]$. Otherwise, it returns $\mathcal{D}_{1,\ell}(x)$. Similarly, when $H_{2,\ell}$ is invoked on input gid , the challenger checks if $\mathcal{D}_{2,\ell}(\text{gid}) = \perp$. If this is the case, the challenger samples $\mathbf{r} \leftarrow D_{\mathbb{Z}, \chi}^{m(2\ell-1)}$, updates $\mathcal{D}_{2,\ell}(\text{gid}) = \mathbf{r}$, and returns $\mathbf{r}$. Otherwise, it returns $\mathcal{D}_{2,\ell}(\text{gid})$.

- H_1 : This is the same as H_0 except that the challenger simulates the random oracle as follows. Let $X = \{(\alpha_1^*, x_1^*), \dots, (\alpha_t^*, x_t^*)\}$ be the challenge attribute that $\mathcal{A}$ outputs at the beginning of the game. The challenger computes $[\mathbf{D}_i \mid \mathbf{B}_i \mid \mathbf{p}_i] \leftarrow \mathbb{Z}_q^{n \times (2mt+1)}$ and updates $\mathcal{D}_{1,t}((\alpha_i^*, x_i^*)) = [\mathbf{D}_i \mid \mathbf{B}_i \mid \mathbf{p}_i]$ for all $i \in [t]$. During the game, when $H_{1,\ell}$ is invoked on input x for any $\ell \in [L]$, the challenger checks if $\mathcal{D}_{1,\ell}(x) = \perp$. If this is the case, the challenger samples $(\mathbf{D}, \mathbf{T}_{\mathbf{D}}) \leftarrow \text{TrapGen}(1^n, m, q)$, $[\mathbf{B} \mid \mathbf{p}] \leftarrow \mathbb{Z}_q^{n \times ((2\ell-1)m+1)}$, updates $\mathcal{D}_{1,\ell}(x) = [\mathbf{D} \mid \mathbf{B} \mid \mathbf{p}]$, and returns $\mathcal{D}_{1,\ell}(x)$. Otherwise, it returns $\mathcal{D}_{1,\ell}(x)$. Since the property of lattice trapdoors ([Lemma 2.1](#)), $\mathbf{D}$ is statistically close to uniform, and thus H_0 and H_1 are statistically indistinguishable given $m = O(n \log q)$.
- H_2 : This is the same as H_1 except the followings. In this hybrid, the challenger samples $\mathbf{A}_\alpha \leftarrow \mathbb{Z}_q^{n \times m}$ for $\alpha \in \mathcal{N}$ instead of using TrapGen for mpk_α . For secret key queries on Y_{gid} for all $\text{gid} \in \mathcal{Q}$, let $Y_{\text{gid},X} = Y_{\text{gid}} \cap X$ and $Y_{\text{gid},\bar{X}} = Y_{\text{gid}} \setminus X$. The challenger computes

$$\begin{aligned} \mathbf{r}_{\text{gid},\ell} &= H_{2,\ell}(\text{gid}), \quad \underbrace{[\hat{\mathbf{D}}_{\alpha,y,\ell}]}_m \mid \underbrace{[\hat{\mathbf{B}}_{\alpha,y,\ell}]}_{m(2\ell-1)} \mid \underbrace{[\hat{\mathbf{p}}_{\alpha,y,\ell}]}_1 = H_{1,\ell}((\alpha, y)) \\ \mathbf{k}_{\alpha,y,\text{gid},\ell} &\leftarrow \begin{cases} [\mathbf{A}_\alpha \mid \hat{\mathbf{D}}_{\alpha,y,\ell}]_\chi^{-1} (\hat{\mathbf{B}}_{\alpha,y,\ell} \mathbf{r}_{\text{gid},\ell} + \hat{\mathbf{p}}_{\alpha,y,\ell}) & ((\alpha, y) \in Y_{\text{gid},X} \wedge \ell = t) \\ \text{SampleLeft}([\mathbf{A}_\alpha \mid \hat{\mathbf{D}}_{\alpha,y,\ell}], \mathbf{T}_{\hat{\mathbf{D}}_{\alpha,y,\ell}}, \hat{\mathbf{B}}_{\alpha,y,\ell} \mathbf{r}_{\text{gid},\ell} + \hat{\mathbf{p}}_{\alpha,y,\ell}, \chi) & (\text{otherwise}) \end{cases} \end{aligned}$$

and gives $\text{sk}_{\alpha,y,\text{gid}} = (\{\mathbf{k}_{\alpha,y,\text{gid},\ell}\}_{\ell \in [L]}, \text{gid})$ for $(\alpha, y) \in Y_{\text{gid}}, \text{gid} \in \mathcal{Q}$ to $\mathcal{A}$. Note that the sampling of $\mathbf{k}_{\alpha,y,\text{gid},t}$ for $(\alpha, y) \in Y_X$ is inefficient. It is easy to see that $H_1 \approx_s H_2$ since $\mathbf{A}_\alpha$ for $\alpha \in \mathcal{N}$ is statistically close to uniform in H_1 and, for all $(\alpha, y) \in Y_{\text{gid}}$ and $\ell \in [L]$,

$$\begin{aligned} &\text{SampleLeft}([\mathbf{A}_\alpha \mid \hat{\mathbf{D}}_{\alpha,y,\ell}], \mathbf{T}_{\hat{\mathbf{D}}_{\alpha,y,\ell}}, \hat{\mathbf{B}}_{\alpha,y,\ell} \mathbf{r}_{\text{gid}} + \hat{\mathbf{p}}_{\alpha,y,\ell}, \chi) \\ &\approx_s \text{SampleLeft}([\mathbf{A}_\alpha \mid \hat{\mathbf{D}}_{\alpha,y}], \mathbf{T}_{\mathbf{A}_\alpha}, \hat{\mathbf{B}}_{\alpha,y} \mathbf{r}_{\text{gid}} + \hat{\mathbf{p}}_{\alpha,y}, \chi) \\ &\approx_s [\mathbf{A}_\alpha \mid \hat{\mathbf{D}}_{\alpha,y}]_\chi^{-1} (\hat{\mathbf{B}}_{\alpha,y} \mathbf{r}_{\text{gid}} + \hat{\mathbf{p}}_{\alpha,y}) \end{aligned}$$

which follows from [Lemma 2.1](#) if $m = O(n \log q)$ (precisely, the first indistinguishability applies to all the cases except for $(\alpha, y) \in Y_{\text{gid}, X} \wedge \ell = t$).

- H_3 : This game is the same as H_2 except the followings. For secret key queries on Y_{gid} for all $\text{gid} \in \mathcal{Q}$, let $Y_{\text{gid}, X} = Y_{\text{gid}} \cap X = ((\alpha_1, y_1), \dots, (\alpha_k, y_k))$ and $Y_{\text{gid}, \bar{X}} = Y_{\text{gid}} \setminus X$. The challenger computes for all $\text{gid} \in \mathcal{Q}$,

$$\begin{aligned} & \underbrace{[\widehat{\mathbf{D}}_{\alpha, y, t}]_m}_{m} \mid \underbrace{[\widehat{\mathbf{B}}_{\alpha, y, t}]_{m(2t-1)}}_{m(2t-1)} \mid \underbrace{[\widehat{\mathbf{p}}_{\alpha, y, t}]_1}_1 = H_{1,t}((\alpha, y)) \quad \text{for } (\alpha, y) \in Y_{\text{gid}} \\ & \mathbf{k}_{\alpha, y, \text{gid}, t} \leftarrow D_{\mathbb{Z}, \chi}^{2m} \quad \text{for } (\alpha, y) \in Y_{\text{gid}, X} \\ & \mathbf{r}_{\text{gid}, t} \leftarrow \begin{bmatrix} \widehat{\mathbf{B}}_{\alpha_1, y_1, t} \\ \vdots \\ \widehat{\mathbf{B}}_{\alpha_k, y_k, t} \end{bmatrix}_{\chi}^{-1} \left(\begin{bmatrix} [\mathbf{A}_{\alpha_1} | \widehat{\mathbf{D}}_{\alpha_1, y_1, t}] \mathbf{k}_{\alpha_1, y_1, t} - \widehat{\mathbf{p}}_{\alpha_1, y_1, t} \\ \vdots \\ [\mathbf{A}_{\alpha_k} | \widehat{\mathbf{D}}_{\alpha_k, y_k, t}] \mathbf{k}_{\alpha_k, y_k, t} - \widehat{\mathbf{p}}_{\alpha_k, y_k, t} \end{bmatrix} \right) \\ & \mathbf{k}_{\alpha, y, \text{gid}, t} \leftarrow \text{SampleLeft}([\mathbf{A}_{\alpha} | \widehat{\mathbf{D}}_{\alpha, y, t}], \mathbf{T}_{\widehat{\mathbf{D}}_{\alpha, y, t}}, \widehat{\mathbf{B}}_{\alpha, y, t} \mathbf{r}_{\text{gid}, t} + \widehat{\mathbf{p}}_{\alpha, y, t}, \chi) \quad \text{for } (\alpha, y) \in Y_{\text{gid}, \bar{X}} \end{aligned} \quad (7)$$

and sets $\text{sk}_{\alpha, y, \text{gid}} = (\{\mathbf{k}_{\alpha, y, \text{gid}, \ell}\}_{\ell \in [L]}, \text{gid})$ and $H_{2,t}(\text{gid}) = \mathbf{r}_{\text{gid}, t}$ right after $\mathcal{A}$ outputs the challenge instance, where $\{\mathbf{k}_{\alpha, y, \text{gid}, \ell}\}_{\ell \in [L] \setminus \{t\}}$ are generated in the same way as in H_2 . We prove $H_2 \approx_s H_3$ in [Lemma 4.1](#).

- H_4 : This game is the same as H_3 except the followings. The challenger generates challenge ciphertext $\text{ct} = (\{\mathbf{c}_{1,i}^\top\}_{i \in [t]}, \mathbf{c}_2^\top, c_3)$ as

$$\begin{aligned} & \hat{\mathbf{c}}_{1,i} \leftarrow \mathbb{Z}_q^m \quad \text{for } \alpha \in A_X \cap \mathcal{N}, \quad \hat{\mathbf{c}}_{1,i} \leftarrow \mathbb{Z}_q^m \quad \text{for } i : \alpha_i^* \in \mathcal{N} \\ & \mathbf{c}_{1,i}^\top = [\hat{\mathbf{c}}_{1,i}^\top | \hat{\mathbf{c}}_{1,i}^\top] \quad \text{for } i : \alpha_i^* \in \mathcal{N}, \quad \mathbf{c}_{1,i}^\top = \mathbf{s}_{\alpha_i^*}^\top [\mathbf{A}_{\alpha_i^*} | \mathbf{D}_i] + \mathbf{e}_{1,i}^\top \quad \text{for } i : \alpha_i^* \in \mathcal{C} \\ & (\mathbf{c}_2^\top, c_3) \leftarrow \mathbb{Z}_q^{m(2L-1)} \times \mathbb{Z}_q \end{aligned}$$

We prove $H_3 \approx_c H_4$ in [Lemma 4.2](#).

It is obvious that $\mathcal{A}$ does not obtain the information on b , and thus its advantage is 0 in the last hybrid. This concludes the proof. $\square$

Lemma 4.1. $H_2 \approx_s H_3$ if $m = O(n \log q)$ and $\chi = \omega(\sqrt{\log m})$.

Proof. Let $Y_{\text{gid}, X} = Y_{\text{gid}} \cap X$ and $Y_{\text{gid}, \bar{X}} = Y_{\text{gid}} \setminus X$. Let $\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_t \end{bmatrix}$ and $\mathbf{p} = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_t \end{bmatrix}$ where $[\mathbf{D}_i | \mathbf{B}_i | \mathbf{p}_i] = H_{1,t}((\alpha_i^*, x_i^*))$. Thanks to [Lemma 2.3](#), $\{\mathbf{B} \mathbf{r}_{\text{gid}}\}_{\text{gid} \in \mathcal{Q}}$ where $\mathbf{r}_{\text{gid}} = H_{2,t}(\text{gid})$ is distributed statistically close to uniform in $(\mathbb{Z}_q^{nt})^{|\mathcal{Q}|}$, because $m = O(n \log q)$, and $\chi = \omega(\sqrt{\log m})$. Hence, the joint distribution of $\{\widehat{\mathbf{B}}_{\alpha, y} \mathbf{r}_{\text{gid}} + \widehat{\mathbf{p}}_{\alpha, y}\}$ for all $(\alpha, y) \in Y_{\text{gid}, X}, \text{gid} \in \mathcal{Q}$ are also statistically close to uniform random in $(\mathbb{Z}_q^n)^{\sum_{\text{gid} \in \mathcal{Q}} |Y_{\text{gid}, X}|}$, where $[\widehat{\mathbf{B}}_{\alpha, y} | \widehat{\mathbf{D}}_{\alpha, y} | \widehat{\mathbf{p}}_{\alpha, y}] = H_{1,t}((\alpha, y))$ (recall that $\widehat{\mathbf{B}}_{\alpha, y} \in \{\mathbf{B}_1, \dots, \mathbf{B}_t\}$ for all $(\alpha, y) \in Y_{\text{gid}, X}$). Recall also that $\mathbf{A}_{\alpha}$ is uniformly distributed in $\mathbb{Z}_q^{n \times m}$ for $\alpha \in \mathcal{N}$. Thanks to [Lemma 3.1](#) (by applying it for each α), the joint distribution of $(\mathbf{A}_{\alpha}, \mathbf{k}_{\alpha, y, \text{gid}}, \widehat{\mathbf{B}}_{\alpha, y} \mathbf{r}_{\text{gid}} + \widehat{\mathbf{p}}_{\alpha, y})$ for $(\alpha, y) \in Y_{\text{gid}, X}, \text{gid} \in \mathcal{Q}$ in H_2 is statistically close to the distribution of $(\mathbf{A}_{\alpha}, \mathbf{k}_{\alpha, y, \text{gid}}, \widehat{\mathbf{B}}_{\alpha, y} \mathbf{r}_{\text{gid}} + \widehat{\mathbf{p}}_{\alpha, y})$ where we first sample $\mathbf{A}_{\alpha} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{k}_{\alpha, y, \text{gid}} \leftarrow D_{\mathbb{Z}, \chi}^{2m}$ and then sample

$$\widehat{\mathbf{B}}_{\alpha, y} \leftarrow \mathbb{Z}_q^{n \times m(2t-1)}, \mathbf{r}_{\text{gid}} \leftarrow D_{\mathbb{Z}, \chi}^{m(2t-1)}, \widehat{\mathbf{p}}_{\alpha, y} \leftarrow \mathbb{Z}_q^n$$

conditioned on $[\mathbf{A}_{\alpha} | \widehat{\mathbf{D}}_{\alpha, y}] \mathbf{k}_{\alpha, y, \text{gid}} = \widehat{\mathbf{B}}_{\alpha, y} \mathbf{r}_{\text{gid}} + \widehat{\mathbf{p}}_{\alpha, y}$ for $(\alpha, y) \in Y_{\text{gid}, X}, \text{gid} \in \mathcal{Q}$. This is conceptually the same as sampling $\widehat{\mathbf{B}}_{\alpha, y} \leftarrow \mathbb{Z}_q^{n \times m(2L-1)}$, $\widehat{\mathbf{p}}_{\alpha, y} \leftarrow \mathbb{Z}_q^n$ for $(\alpha, y) \in \bigcup_{\text{gid} \in \mathcal{Q}} Y_{\text{gid}, X}$, and then sampling $\mathbf{r}_{\text{gid}}$ for each $\text{gid} \in \mathcal{Q}$ as [Eq. \(7\)](#). Recall that the difference between H_2 and H_3 lies in $\mathbf{k}_{\alpha, y, \text{gid}}, \mathbf{r}_{\text{gid}}$ for $(\alpha, y) \in Y_{\text{gid}, X}, \text{gid} \in \mathcal{Q}$. Thus, we have $H_2 \approx_s H_3$. $\square$

Lemma 4.2. $H_3 \approx_c H_4$ under the $\text{LWE}_{n, m(2t+t') + 1, q, \chi}$ assumption if $\chi = 2m^2 L^2 \cdot \omega(\sqrt{\log n})$, t is the size of the challenge attribute X , and t' is the number of non-corrupted authorities related to the challenge attribute X .

Proof. We prove that if there is an adversary $\mathcal{A}$ that distinguishes H_3 and H_4 with non-negligible probability, we can construct a reduction $\mathcal{B}$ that breaks the $\text{RTLWE}_{n,(t+t')m+1,m,q,\chi,t}$ assumption. Then, Lemma 2.4 immediately implies Lemma 4.2. We can construct $\mathcal{B}$ as follows. After $\mathcal{A}$ outputs the challenge instance, $\mathcal{B}$ generates the input $(\{\text{mpk}_\alpha\}_{\alpha \in \mathcal{N}}, \text{ct}^*, \{\text{sk}_{\alpha,y,\text{gid}}\}_{(\alpha,y) \in Y_{\text{gid}}, \text{gid} \in \mathcal{Q}})$ to $\mathcal{A}$ as follows:

- Non-corrupted master public keys: $\mathcal{A}$ declares the challenge attribute $X = \{(\alpha_1^*, x_1^*), \dots, (\alpha_t^*, x_t^*)\}$ and non-corrupted authorities $\mathcal{N}$ at the beginning of the game. Let $A_X \cap \mathcal{N} = \{\alpha_1, \dots, \alpha_{t'}\}$. $\mathcal{B}$ outputs $\mathbf{u} = (u_1, \dots, u_t)$ where $u_i = 1$ if $\alpha_i^* \in \mathcal{N}$ and $u_i = 0$ otherwise, for the security game for the related trapdoor LWE assumption. Then, it obtains $(\hat{\mathbf{A}}, \mathbf{B}, \mathbf{v}^\top, \hat{\mathbf{v}}^\top)$ where $(\mathbf{v}^\top, \hat{\mathbf{v}}^\top) = (\mathbf{s}^\top \hat{\mathbf{A}} + \mathbf{e}^\top, \mathbf{s}^\top (\mathbf{u}^\top \otimes \mathbf{I}_n) \mathbf{B} + \hat{\mathbf{e}}^\top)$ or $(\mathbf{v}^\top, \hat{\mathbf{v}}^\top) \leftarrow \mathbb{Z}_q^{tm+m+1} \times \mathbb{Z}_q^{m(2t-1)}$. $\mathcal{B}$ parses

$$\hat{\mathbf{A}} = [\mathbf{A}_{\alpha_1} | \dots | \mathbf{A}_{\alpha_{t'}} | \mathbf{D}_1 | \dots | \mathbf{D}_t | \mathbf{p}], \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_t \end{bmatrix}$$

and sets $\text{mpk}_{\alpha_i} = \mathbf{A}_{\alpha_i}$ for $i \in [t']$. For $\alpha \in \mathcal{N} \setminus A_X$, $\mathcal{B}$ chooses $\text{mpk}_\alpha = \mathbf{A}_\alpha \leftarrow \mathbb{Z}_q^{n \times m}$.

- Challenge ciphertext: $\mathcal{A}$ outputs corrupted master public keys $\text{mpk}_\alpha = \mathbf{A}_\alpha$ for $\alpha \in \mathcal{C}$. $\mathcal{B}$ parses $\mathbf{v}^\top = [\mathbf{v}_{\alpha_1}^\top | \dots | \mathbf{v}_{\alpha_{t'}}^\top | \tilde{\mathbf{v}}_1^\top | \dots | \tilde{\mathbf{v}}_t^\top | v]$. $\mathcal{B}$ samples $b \leftarrow \{0, 1\}$, $\mathbf{p}_i \leftarrow \mathbb{Z}_q^n$ conditioned on $\sum_{i: \alpha_i^* \in \mathcal{N}} \mathbf{p}_i = \mathbf{p}$ for $i: \alpha_i^* \in \mathcal{N}$ and $\mathbf{p}_i \leftarrow \mathbb{Z}_q^n$ for $i: \alpha_i^* \in \mathcal{C}$. Then, $\mathcal{B}$ computes $\text{ct} = (\{\mathbf{c}_{1,i}^\top\}_{i \in [t]}, \mathbf{c}_2^\top, \mathbf{c}_3)$ as follows.

$$\begin{aligned} \mathbf{s}_\alpha &\leftarrow \mathbb{Z}_q^n \quad \text{for } \alpha \in A_X, \quad \tilde{\mathbf{e}}_{1,\alpha} \leftarrow D_{\mathbb{Z}, \chi}^m \quad \text{for } \alpha \in A_X \cap \mathcal{C} \\ \hat{\mathbf{e}}_{1,i} &\leftarrow D_{\mathbb{Z}, \chi}^m \quad \text{for } i: \alpha_i^* \in \mathcal{C} \\ \mathbf{c}_{1,i}^\top &= \begin{cases} \mathbf{s}_{\alpha_i^*} [\mathbf{A}_{\alpha_i^*} | \mathbf{D}_i] + [\mathbf{v}_{\alpha_i^*}^\top | \tilde{\mathbf{v}}_i^\top] & (i: \alpha_i^* \in \mathcal{N}) \\ \mathbf{s}_{\alpha_i^*} [\mathbf{A}_{\alpha_i^*} | \mathbf{D}_i] + [\tilde{\mathbf{e}}_{1,i}^\top | \hat{\mathbf{e}}_i^\top] & (i: \alpha_i^* \in \mathcal{C}) \end{cases} \\ \mathbf{c}_2^\top &= \sum_{i \in [t]} \mathbf{s}_{\alpha_i^*} \mathbf{B}_i + \hat{\mathbf{v}}^\top, \quad \mathbf{c}_3 = \sum_{i \in [t]} \mathbf{s}_{\alpha_i^*} \mathbf{p}_i + v + \mu_b \lfloor q/2 \rfloor \end{aligned}$$

- Random oracle programming for $H_{1,t}$: $\mathcal{B}$ sets $\mathcal{D}_{1,t}((\alpha_i^*, x_i^*)) = [\mathbf{D}_i | \mathbf{B}_i | \mathbf{p}_i]$ for all $i \in [t]$. Note that since $\mathbf{p}$ is kept hidden from the adversary, $\mathbf{p}_1, \dots, \mathbf{p}_t$ are distributed randomly from the view point of $\mathcal{A}$. During the game, when $H_{1,t}$ is invoked on input x , the challenger checks if $\mathcal{D}_{1,t}(x) = \perp$. If this is the case, the challenger samples $(\mathbf{D}, \mathbf{T}_\mathbf{D}) \leftarrow \text{TrapGen}(1^n, m, q)$, $[\mathbf{B} | \mathbf{p}] \leftarrow \mathbb{Z}_q^{n \times ((2t-1)m+1)}$, updates $\mathcal{D}_{1,t}(x) = [\mathbf{D} | \mathbf{B} | \mathbf{p}]$, and returns $[\mathbf{D} | \mathbf{B} | \mathbf{p}]$. Otherwise, it just returns $\mathcal{D}_{1,t}(x)$. Note that the oracle simulation for $H_{1,\ell}$ s.t. $\ell \neq t$ is the same as in H_1 .
- Secret keys: For each Y_{gid} in the challenge instance, $\mathcal{B}$ simulates $\text{sk}_{\alpha,y,\text{gid}} = (\{\mathbf{k}_{\alpha,y,\text{gid},\ell}\}_{\ell \in [L]}, \text{gid})$ as follows. Let $Y_{\text{gid},X} = Y_{\text{gid}} \cap X = ((\alpha_1, y_1), \dots, (\alpha_k, y_k))$ and $Y_{\text{gid},\bar{X}} = Y_{\text{gid}} \setminus X$, $\mathbf{m}_i^\top = (0^{i-1}, 1, 0^{t-i}) \in \mathbb{Z}_q^t$ for $i \leq t$. Then, it computes for all $\text{gid} \in \mathcal{Q}$,

$$\begin{aligned} [\hat{\mathbf{D}}_{\alpha,y} | \hat{\mathbf{B}}_{\alpha,y} | \hat{\mathbf{p}}_{\alpha,y}] &= H_{1,t}(\alpha, y) \quad \text{for } (\alpha, y) \in Y_{\text{gid}} \\ \mathbf{k}_{\alpha,y,\text{gid},t} &\leftarrow D_{\mathbb{Z}, \chi}^{2m} \quad \text{for } (\alpha, y) \in Y_{\text{gid},X} \end{aligned}$$

and makes a query $\text{tdO}(\mathbf{M}, \mathbf{t})$ and obtains $\mathbf{r}_{\text{gid},t} \in \mathbb{Z}_q^{m(2t-1)}$ as a reply, where $\phi: [k] \rightarrow [t]$ is an injective function such that $\mathbf{B}_{\phi(i)} = \hat{\mathbf{B}}_{\alpha_i, y_i}$,

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_{\phi(1)}^\top \\ \vdots \\ \mathbf{m}_{\phi(k)}^\top \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} [\mathbf{A}_{\alpha_1} | \hat{\mathbf{D}}_{\alpha_1, y_1}] \mathbf{k}_{\alpha_1, y_1, \text{gid}} - \hat{\mathbf{p}}_{\alpha_1, y_1} \\ \vdots \\ [\mathbf{A}_{\alpha_k} | \hat{\mathbf{D}}_{\alpha_k, y_k}] \mathbf{k}_{\alpha_k, y_k, \text{gid}} - \hat{\mathbf{p}}_{\alpha_k, y_k} \end{bmatrix}$$

Finally, it computes

$$\mathbf{k}_{\alpha,y,\text{gid},t} \leftarrow \text{SampleLeft}([\mathbf{A}_\alpha | \hat{\mathbf{D}}_{\alpha,y}], \mathbf{T}_{\hat{\mathbf{D}}_{\alpha,y}}, \hat{\mathbf{B}}_{\alpha,y} \mathbf{r}_{\text{gid},t} + \hat{\mathbf{p}}_{\alpha,y}, \chi) \quad \text{for } (\alpha, y) \in Y_{\text{gid},\bar{X}}$$

The other elements $\{\mathbf{k}_{\alpha,y,\text{gid},\ell}\}_{\ell \in [L] \setminus \{t\}}$ are generated in the same way as in H_2 .

- Random oracle programming for H_2 : $\mathcal{B}$ sets $\mathcal{D}_{2,t}(\text{gid}) = \mathbf{r}_{\text{gid},t}$ for all $\text{gid} \in \mathcal{Q}$. The rest of the simulation of $H_{2,\ell}$ for $\ell \in [L]$ is the same as the challenger in H_3 .

At the end of the game, if $\mathcal{A}$ outputs b' at the end of the game, $\mathcal{B}$ outputs 1 if and only if $b = b'$. Observe that $\mathcal{A}$'s view corresponds to H_3 if $(\mathbf{v}^\top, \hat{\mathbf{v}}^\top) = (\mathbf{s}^\top \hat{\mathbf{A}} + \mathbf{e}^\top, \mathbf{s}^\top (\mathbf{u}^\top \otimes \mathbf{I}_n) \mathbf{B} + \hat{\mathbf{e}}^\top)$ and H_4 otherwise. Especially, $\mathcal{B}$ implicitly defines $\mathbf{s}'_\alpha = \mathbf{s} + \mathbf{s}_\alpha$ for $\alpha \in A_X \cap \mathcal{N}$. Additionally, if $\mathcal{A}$ is admissible, then $\mathcal{B}$ is admissible. This is because $Y_{\text{gid},X} \subsetneq X \cap \mathcal{N}$ for all $\text{gid} \in \mathcal{Q}$, which follows from the game condition, and thus it is obvious that $k < t'$ and thus $\{\mathbf{u}, \mathbf{m}_{\phi(i_1)}, \dots, \mathbf{m}_{\phi(i_k)}\}$ are linearly independent. This concludes the proof. $\square$

5 Ring Instantiations

We develop more efficient variants of our key-policy and ciphertext-policy ABE schemes in ideal lattices under the Ring-LWE assumption, and provide an implementation.

5.1 Preliminaries on Ideal Lattices

The Ring Setting. Let $n = n(\lambda)$ be a power of two. We work with the $2n$ -th cyclotomic ring of integers

$$\mathcal{R} = \mathbb{Z}[x]/(x^n + 1),$$

whose elements are polynomials of degree less than n with integer coefficients. For an integer modulus $q = q(\lambda)$, let

$$R_q = \mathcal{R}/q\mathcal{R} = \mathbb{Z}_q[x]/(x^n + 1).$$

Throughout this document, we assume that q is a prime that splits completely in $\mathcal{R}$, i.e., $q\mathcal{R}$ factors as

$$q\mathcal{R} = \mathfrak{q}_1 \cdot \mathfrak{q}_2 \cdots \mathfrak{q}_n$$

where each $\mathfrak{q}_i$ is a prime ideal of norm $\mathcal{N}(\mathfrak{q}_i) = q$.

We define RingTrapGen and RingSampleLeft algorithms in [Theorem B.1](#) and also present their properties. We present the ring analogs of the preimage sampling and delegation in [Lemma B.5](#) and [Lemma B.6](#), respectively.

5.2 Ring Generalized Related-Trapdoor LWE

In this section, we present the ring generalization of the related-trapdoor LWE assumption, and show that it can be based on the standard Ring-LWE assumption. This is the main technical contribution of this section, and is of independent interest.

Assumption 5.1 (Ring Generalized Related-Trapdoor LWE). *Let λ be a security parameter, and let $n, q, m, \hat{m}, L, \chi, \chi_R$ be lattice parameters, where n is a power of two, q is an NTT-friendly prime, and $\chi, \chi_R > 0$ are Gaussian widths. The $\text{RTRLWE}_{n,m,\hat{m},q,\chi,\chi_R,L}$ assumption is defined by the following game between a challenger and an adversary $\mathcal{A}$:*

1. The adversary $\mathcal{A}$ chooses a vector $\mathbf{u} \in \{0, 1\}^L$ with at least one entry equal to 1.
2. The challenger samples $\mathbf{a} \xleftarrow{\$} R_q^m$, $\mathbf{B} \xleftarrow{\$} R_q^{L \times \hat{m}(2L-1)}$, and chooses a random bit $b \in \{0, 1\}$. The challenge is constructed as follows:
 - If $b = 0$ (real): sample $s \xleftarrow{\$} R_q$, $\mathbf{R} \leftarrow D_{\mathcal{R}, \chi_R}^{\text{coeff}, \hat{m}L \times \hat{m}(L-1)}$, $\mathbf{e}_1 \leftarrow D_{\mathcal{R}, \chi}^{\text{coeff}, m}$, and $\hat{\mathbf{e}}_0 \leftarrow D_{\mathcal{R}, \chi}^{\text{coeff}, \hat{m}L}$. Set $\hat{\mathbf{e}}^\top := \hat{\mathbf{e}}_0^\top \cdot [\mathbf{I}_{\hat{m}L} \mid \mathbf{R}] \in \mathcal{R}^{\hat{m}(2L-1)}$. Give to $\mathcal{A}$ the tuple

$$(\mathbf{a}, \mathbf{B}, s\mathbf{a} + \mathbf{e}_1, s \cdot \mathbf{u}^\top \mathbf{B} + \hat{\mathbf{e}}^\top).$$
 - If $b = 1$ (random): sample $\mathbf{v} \xleftarrow{\$} R_q^m$ and $\hat{\mathbf{v}} \xleftarrow{\$} R_q^{\hat{m}(2L-1)}$. Give to $\mathcal{A}$ the tuple

$$(\mathbf{a}, \mathbf{B}, \mathbf{v}, \hat{\mathbf{v}}^\top).$$

3. $\mathcal{A}$ may adaptively issue oracle queries of the form $(\mathbf{M}, \mathbf{t})$, where $\mathbf{M} \in R_q^{k \times L}$ for some $k < L$ and $\mathbf{t} \in R_q^k$. The oracle responds as follows:
 - Define the augmented matrix $\overline{\mathbf{M}} := \begin{pmatrix} \mathbf{M} \\ \mathbf{u}^\top \end{pmatrix} \in R_q^{(k+1) \times L}$. If $\overline{\mathbf{M}}$ does not have full row rank in the ring sense (i.e., it fails to be invertible modulo some prime ideal factor $\mathfrak{q}_i$ of $q\mathcal{R}$), the oracle returns $\perp$.
 - Otherwise, the oracle returns a sample $\mathbf{y} \leftarrow (\mathbf{M}\mathbf{B})_\chi^{-1}(\mathbf{t})$ — that is, a discrete Gaussian preimage $\mathbf{y} \in \mathcal{R}^{\hat{m}(2L-1)}$ satisfying $\mathbf{M}\mathbf{B}\mathbf{y} = \mathbf{t} \bmod q$, with distribution statistically close to $D_{\mathcal{R}^{\hat{m}(2L-1)}, \chi}$ conditioned on this constraint.
4. $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$.

We say that the $\text{RTRLWE}_{n,m,\hat{m},q,\chi,\chi_R,L}$ assumption holds if for every probabilistic polynomial-time adversary $\mathcal{A}$,

$$|\Pr[b' = b] - \tfrac{1}{2}| \leq \text{negl}(\lambda).$$

Theorem 5.1 (Ring-RTLWE from Ring-LWE). Let λ be a security parameter and let $n, q, m, \hat{m}, L, \chi, \chi_R$ be lattice parameters as in [Assumption 5.1](#). Denote $w := \lceil \log q \rceil$ and let g be the number of distinct prime ideal factors of $q\mathcal{R}$. Suppose the following conditions hold:

1. (Ring-LHL width ceiling.) $\chi_R \leq \frac{\sqrt{\pi}}{2} \cdot \frac{\sqrt{q}}{\sqrt{\ln 4}}$.
2. (Ring-LHL entropy condition.) One of the following:
 - (Unconditional.) $\hat{m}L \log \chi_R \geq 2 \log(\hat{m}(L-1)/\varepsilon) + (L+n) \log q + \log g + \hat{m}L$, where ε is any negligible function; or
 - (Conditional on [Assumption B.2](#).) $\hat{m}L \log \chi_R \geq 2 \log(\hat{m}(L-1)/\varepsilon) + (L+1) \log q + \log g + \hat{m}L$.
3. (Trapdoor quality condition.) $\chi \geq \tau \cdot \omega(\sqrt{\log n})$, where $\tau = O(\chi_R \cdot n^{3/2} \cdot \hat{m}^{3/2} \cdot L^{3/2})$ is the trapdoor quality from [Corollary B.2](#).
4. (Gaussian noise.) $\chi \geq \omega(\sqrt{\log n})$.

Then, if the Ring-LWE assumption $\text{RLWE}_{n,m+\hat{m}L,q,\chi}$ holds, the assumption $\text{RTRLWE}_{n,m,\hat{m},q,\chi,\chi_R,L}$ also holds.

We prove the above theorem in [Section B.5](#).

5.3 Our KP-ABE from Ring-LWE

We obtain the construction by using the *statistical* instantiation of the Ring-LWE trapdoor [[Pei10](#), [EBB14](#), [SSTX09](#)] as stated in [Theorem B.2](#), i.e. $m = 2k + \omega(\log n)$, so that $\mathbf{A}$ is statistically close to uniform over $R_q^{1 \times m}$. As explained in [Remark B.7](#), the cheaper computational instantiation ($m = k + 2$) is not sound here: the hybrid that replaces $\mathbf{A}$ by a uniform vector precedes the invocation of Ring-LWE and must be statistical, and a computationally uniform $\mathbf{A}$ would additionally leak the high-order bits of s through $\mathbf{c}_1$.

Construction. Let $k = \lceil \log q \rceil$, $\bar{m} = \omega(\log n) + k$, $m = \bar{m} + k$.

Setup(1^λ): It outputs mpk and msk as follows:

$$\begin{aligned} (\mathbf{A}, \mathbf{T}_\mathbf{A}) &\leftarrow \text{RingTrapGen}(1^n, q), p \xleftarrow{\$} R_q \\ \text{mpk} &= (\mathbf{A} \in R_q^{1 \times m}, p), \text{msk} = \mathbf{T}_\mathbf{A} \in R_q^{\bar{m} \times k}. \end{aligned}$$

Enc(mpk, X, μ): It parses $X = \{x_1, \dots, x_t\}$, and outputs ct as follows.

$$\begin{aligned} s &\xleftarrow{\$} R_q, \mathbf{e}_1 \leftarrow \chi^{m(t+1)}, e_2 \leftarrow \chi \\ D &= [H(x_1) \cdots H(x_t)] \in R_q^{1 \times mt}, \\ \text{ct} &= (\mathbf{c}_1^T, c_2) = (s[\mathbf{A}|D] + \mathbf{e}_1^T, sp + e_2 + \mu \lfloor q/2 \rfloor). \end{aligned}$$

KeyGen($\text{mpk}, \text{msk}, Y$): It parses $Y = \{y_1, \dots, y_u\}$, where it must hold that $u \leq L$. It computes, for $i \in [u]$,

$$\begin{aligned} p_i &\xleftarrow{\$} R_q \text{ if } i < u, \quad p_u = p - \sum_{j \in [u-1]} p_j, \\ \mathbf{k}_i &\in R_q^{2m} \leftarrow \text{RingSampleLeft}([\mathbf{A}|H(y_i)], \mathbf{T}_\mathbf{A}, p_i, \chi'). \end{aligned}$$

Then it outputs $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u)$.

Dec($\text{mpk}, \text{ct}, \text{sk}, X, Y$): It parses $\text{ct} = (\mathbf{c}_1^T, c_2)$, where $\mathbf{c}_1^T = [c_{1,0}^T \cdots c_{1,t}^T] \in R_q^{1 \times m(t+1)}$, $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u)$, and $X = \{x_1, \dots, x_t\}$, $Y = \{y_1, \dots, y_u\}$. If $Y \subseteq X$, it computes $\rho: [u] \rightarrow [t]$ such that $y_i = x_{\rho(i)}$ and outputs $\left\lfloor \frac{2}{q} v \right\rceil$, where

$$v := c_2 - \sum_{i \in [u]} [c_{1,0}^T | c_{1,\rho(i)}^T] \mathbf{k}_i.$$

Parameter sizes and comparison with LWE based construction. The parameter sizes of LWE based KP-ABE construction are $|\text{mpk}| = (2nk + 1)nk$, $|\text{ct}| = (2nk(t+1) + 1)k$, and $|\text{sk}| = (4nku)k$, where $|\text{ct}|$ encodes 1-bit of message. In contrast, the parameter sizes of our Ring-LWE based KP-ABE construction are $|\text{mpk}| = (m+1)nk$, $|\text{ct}| = (m(t+1) + 1)nk$, and $|\text{sk}| = (2mu)nk$, where ct encodes n -bit of message. Thus, with $m = 2k + \omega(\log n)$, the mpk is $n \times$ shorter and the ct is $n \times$ shorter (per message bit), while the sk sizes are comparable.

Parameter bounds for correctness. Recall from Dec that the decryption outputs $\left\lfloor \frac{2}{q}v \right\rfloor$, where, $v := c_2 - \sum_{i \in [u]} [\mathbf{c}_{1,0}^T | \mathbf{c}_{1,\rho(i)}^T] \mathbf{k}_i$. Simplifying v , we get that

$$v = \mu \lfloor q/2 \rfloor + e_2 - \sum_{i \in [u]} [\mathbf{e}_{1,0}^T | \mathbf{e}_{1,\rho(i)}^T] \mathbf{k}_i.$$

Let $\tilde{e} := e_2 - \sum_{i \in [u]} [\mathbf{e}_{1,0}^T | \mathbf{e}_{1,\rho(i)}^T] \mathbf{k}_i$. Thus, for correctness, we need to choose parameters such that $|\tilde{e}|_\infty < q/4$ holds with overwhelming probability. Observe that

$$\begin{aligned} |\tilde{e}|_\infty &\leq |e_2|_\infty + u \cdot |\mathbf{e}_1|_\infty \cdot |\mathbf{k}_i|_\infty \cdot (2m) \cdot n. \\ &\leq \sqrt{\lambda}\chi + 2mun(\sqrt{\lambda}\chi)(\sqrt{\lambda}\chi'), \\ &\leq \sqrt{\lambda}\chi + 2mLn\lambda\chi\chi'. \quad (\text{since } u \leq L) \end{aligned}$$

It suffices to choose q such that $q \geq 5|\tilde{e}|_\infty \geq 5(\sqrt{\lambda}\chi + 2mLn\lambda\chi\chi')$.

Correctness. For correctness, we have the following theorem.

Theorem 5.2. *Suppose $q \geq 5(\sqrt{\lambda}\chi + 2mLn\lambda\chi\chi')$. Then, our scheme is correct.*

Proof. In decryption, suppose $\text{ct} = (\mathbf{c}_1^\top, c_2)$ and $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u)$ are a ciphertext and a secret key for X and Y , respectively, such that $Y \subseteq X$. Write $\mathbf{c}_1^\top = [\mathbf{c}_{1,0}^\top | \dots | \mathbf{c}_{1,t}^\top]$ and $\mathbf{e}_1^\top = [\mathbf{e}_{1,0}^\top | \dots | \mathbf{e}_{1,t}^\top]$. In decryption, we have

$$\begin{aligned} v &= c_2 - \sum_{i \in [u]} [\mathbf{c}_{1,0}^\top | \mathbf{c}_{1,\rho(i)}^\top] \mathbf{k}_i \\ &= sp + e_2 + \mu \lfloor q/2 \rfloor - \sum_{i \in [u]} \left(s[\mathbf{A}|H(y_i)] + [\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,\rho(i)}^\top] \right) \mathbf{k}_i \\ &= \mu \lfloor q/2 \rfloor + e_2 - \underbrace{\sum_{i \in [u]} [\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,\rho(i)}^\top] \mathbf{k}_i}_{\tilde{e}}, \end{aligned}$$

where we used $[\mathbf{A}|H(y_i)] \mathbf{k}_i = p_i$ for every $i \in [u]$ and $\sum_{i \in [u]} p_i = p$. From the parameter setting and **Lemma 2.2**, applied coefficientwise, we have $|e_2|_\infty \leq \sqrt{\lambda}\chi$ and $|\mathbf{e}_1|_\infty \leq \sqrt{\lambda}\chi$ with overwhelming probability. Thanks to the property of ideal lattice trapdoors, the distribution of $\mathbf{k}_i$ is statistically close to $[\mathbf{A}|H(y_i)]_{\chi'}^{-1}(p_i)$. Since the marginal distribution of p_i is uniform in R_q , the marginal distribution of $\mathbf{k}_i$ is statistically close to $D_{\mathcal{R}, \chi'}^{\text{coeff}, 2m}$ from **Lemma B.5**. Thus, we have $|\mathbf{k}_i|_\infty \leq \sqrt{\lambda}\chi'$ with overwhelming probability, again from **Lemma 2.2**. In summary, we have, with overwhelming probability,

$$\begin{aligned} |\tilde{e}|_\infty &\leq |e_2|_\infty + u |\mathbf{e}_1|_\infty |\mathbf{k}_i|_\infty \cdot 2m \cdot n \\ &\leq \sqrt{\lambda}\chi + 2mun\lambda\chi\chi' \\ &\leq \sqrt{\lambda}\chi + 2mLn\lambda\chi\chi'. \end{aligned}$$

Thus, $|\tilde{e}|_\infty \leq q/5$ with overwhelming probability and correctness holds. $\square$

Parameter bounds for security. For the security of the scheme (**Theorem 5.3**), we need to set $m = 2k + \omega(\log n)$ and $\chi' = \tau \cdot \omega(\sqrt{\log n})$ (where $\tau = O(\sqrt{n \log q})$). For $\text{RLWE}_{n, m(t+1)+1, q, \chi}$ to hold, we also need $n \geq \frac{\log_2(q/\chi)}{4 \log_2(\delta)}$, where δ is the root Hermite factor and $\delta < 1.006$ corresponds to at least 100 bits of security [CN11].

Empirical parameter choices. We choose $n = O(\lambda)$, $\chi = O(1)$, $\chi' = O((nk \log(k))^{0.5})$, $q = O(Ln^{2.5} \cdot (\log(Ln))^{1.5} \cdot (\log n)^{0.5})$, $m = 2k + \omega(\log n)$ (where $k = \lceil \log q \rceil$).

Security Analysis. For security, we have the following theorem.

Theorem 5.3. Suppose $m = 2k + \omega(\log n)$ (where $k = \lceil \log q \rceil$), $\chi' = \tau \cdot \omega(\sqrt{\log n})$, $\tau = O(\sqrt{n \log q})$, the hash function H is modeled as a random oracle, t is the size of the attribute X for the challenge query, and $\text{RLWE}_{n, m(t+1)+1, q, \chi}$ holds. Then, the above KP-ABE scheme is selectively secure as per [Definition 2.1](#).

Proof. We prove the theorem by a series of hybrids described as follows.

- H_0 : This is the original game. In this game, $\mathcal{A}$ first outputs a challenge attribute $X = (x_1, \dots, x_t)$. Then, the challenger computes

$$(\mathbf{A}, \mathbf{T}_{\mathbf{A}}) \leftarrow \text{RingTrapGen}(1^n, q), p \leftarrow R_q$$

$$\text{mpk} = (\mathbf{A} \in R_q^{1 \times m}, p), \quad \text{msk} = \mathbf{T}_{\mathbf{A}} \in R_q^{\overline{m} \times k}$$

and gives mpk to $\mathcal{A}$. When $\mathcal{A}$ asks for a secret key for $Y = (y_1, \dots, y_u)$, the challenger computes

$$p_i \leftarrow R_q \quad \text{if } i < u, \quad p_u = p - \sum_{j \in [u-1]} p_j$$

$$\mathbf{k}_i \leftarrow \text{RingSampleLeft}([\mathbf{A}|H(y_i)], \mathbf{T}_{\mathbf{A}}, p_i, \chi') \quad \text{for } i \in [u]$$

and gives $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u)$ to $\mathcal{A}$. For the challenge, $\mathcal{A}$ outputs μ_0, μ_1 . Then, the challenger chooses a challenge bit $b \leftarrow \{0, 1\}$ and gives ct to $\mathcal{A}$ where

$$s \xleftarrow{\$} R_q, \mathbf{e}_1 \leftarrow \chi^{m(t+1)}, \mathbf{e}_2 \leftarrow \chi$$

$$D = [H(x_1) | \dots | H(x_t)] \in R_q^{1 \times mt},$$

$$\text{ct} = (\mathbf{c}_1^T, c_2) = (s[\mathbf{A}|D] + \mathbf{e}_1^T, sp + c_2 + \mu_b \lfloor q/2 \rfloor).$$

The post-challenge key queries are the same as the pre-challenge queries. Finally, $\mathcal{A}$ outputs b' . During the game, the challenger simulates the random oracle as follows. It maintains a dictionary $\mathcal{D}$ that is initially empty. When H is invoked on input x , the challenger checks if $\mathcal{D}(x) = \perp$. If this is the case, the challenger samples $D \leftarrow R_q^{1 \times m}$, updates $\mathcal{D}(x) = D$, and returns D . Otherwise, it returns $\mathcal{D}(x)$.

- H_1 : This is the same as H_0 except that the challenger simulates the random oracle as follows. Right after $\mathcal{A}$ outputs the challenge attribute $X = (x_1, \dots, x_t)$, the challenger computes $D_i \leftarrow R_q^{1 \times m}$ and updates $\mathcal{D}(x_i) = D_i$ for all $i \in [t]$. During the game, when H is invoked on input x , the challenger checks if $\mathcal{D}(x) = \perp$. If this is the case, the challenger samples $(D, \mathbf{T}_D) \leftarrow \text{RingTrapGen}(1^n, q)$, updates $\mathcal{D}(x) = D$, and returns D . Otherwise, it returns $\mathcal{D}(x)$. From the trapdoor distribution property of ideal lattice trapdoors ([Theorem B.1](#), [Theorem B.2](#)), D is statistically close to uniform, and thus H_0 and H_1 are statistically indistinguishable given $m = 2k + \omega(\log n)$.
- H_2 : This is the same as H_1 except the followings. In this hybrid, the challenger samples $\mathbf{A} \leftarrow R_q^{1 \times m}$ instead of using RingTrapGen for mpk . For all secret key queries on $Y = (y_1, \dots, y_u)$, let $Y_X = Y \cap X$ and $Y_{\bar{X}} = Y \setminus X$. The challenger computes

$$p_i \leftarrow R_q \quad \text{for } i : y_i \in Y_X,$$

$$p_i \leftarrow R_q \quad \text{conditioned on } \sum_{i \in [u]} p_i = p \quad \text{for } i : y_i \in Y_{\bar{X}},$$

$$\mathbf{k}_i \leftarrow \begin{cases} [\mathbf{A}|H(y_i)]_{\chi'}^{-1}(p_i) & (i : y_i \in Y_X) \\ \text{RingSampleLeft}([\mathbf{A}|H(y_i)], \mathbf{T}_{H(y_i)}, p_i, \chi') & (i : y_i \in Y_{\bar{X}}) \end{cases}$$

and gives $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u)$ to $\mathcal{A}$. Note that the sampling of $\mathbf{k}_i$ for $i : y_i \in Y_X$ is inefficient. It is easy to see that $H_1 \approx_s H_2$ since $\mathbf{A}$ is statistically close to uniform if $m = 2k + \omega(\log n)$, p_i for $i \in [u]$ are uniformly distributed conditioned on $\sum_{i \in [u]} p_i = p$ in both hybrids, and, for all $i \in [u]$,

$$\text{RingSampleLeft}([\mathbf{A}|H(y_i)], \mathbf{T}_{H(y_i)}, p_i, \chi')$$

$$\approx_s \text{RingSampleLeft}([\mathbf{A}|H(y_i)], \mathbf{T}_{\mathbf{A}}, p_i, \chi')$$

$$\approx_s [\mathbf{A}|H(y_i)]_{\chi'}^{-1}(p_i)$$

which follows from the preimage sampling property of ideal lattice trapdoors ([Theorem B.1](#)) if $\chi' \geq \tau \cdot \omega(\sqrt{\log n})$ and $\tau = O(\sqrt{n \log q})$.

- H_3 : This game is the same as H_2 except the followings. For all secret key queries on $Y = (y_1, \dots, y_u)$, let $Y_X = Y \cap X$, $Y_{\bar{X}} = Y \setminus X$. The challenger computes

$$\begin{aligned} \mathbf{k}_i &\leftarrow D_{R_q, \chi'}^{2m}, p_i = [\mathbf{A}|H(y_{i_1})]\mathbf{k}_i \quad \text{for } i : y_i \in Y_X \\ p_i &\leftarrow R_q \quad \text{for } i : y_i \in Y_{\bar{X}} \text{ conditioned on } \sum_{i \in [u]} p_i = p \\ \mathbf{k}_i &\leftarrow \text{RingSampleLeft}([\mathbf{A}|H(y_i)], \mathbf{T}_{H(y_i)}, p_i, \chi') \quad \text{for } i : y_i \in Y_{\bar{X}} \end{aligned} \quad (8)$$

and gives $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u)$ to $\mathcal{A}$. We prove $H_2 \approx_s H_3$ in [Lemma 5.1](#).

- H_4 : This game is the same as H_3 except the followings. The challenger generates the challenge ciphertext as

$$\text{ct} = (\mathbf{c}_1^\top, c_2) \leftarrow R_q^{m(t+1)} \times R_q$$

We prove $H_3 \approx_c H_4$ in [Lemma 5.2](#).

It is obvious that $\mathcal{A}$ does not obtain the information on b , and thus its advantage is 0 in the last hybrid. This concludes the proof. $\square$

Lemma 5.1. $H_2 \approx_s H_3$ if $m \geq 2k + \omega(\log n)$ and $\chi' = \tau \cdot \omega(\sqrt{\log n})$, where $\tau = O(\sqrt{n \log q})$.

Proof. Let Q be the number of secret key queries. Due to the game condition, we have $Y_X \subsetneq Y$ for all Y in secret key queries. Hence, $p_i^{(j)}$ for all $i : y_i^{(j)} \in Y_X^{(j)}$, $j \in [Q]$ are uniformly and independently distributed in R_q . Recall that $\mathbf{A}$ is uniformly distributed in $R_q^{1 \times m}$. Applying [Lemma B.6](#) with $\ell_1 = |X|$ and $\ell_2 = Q$, it follows that the joint distribution of $(\mathbf{A}, \mathbf{k}_i^{(j)}, p_i^{(j)})$ for $i : y_i^{(j)} \in Y_X^{(j)}$, $j \in [Q]$ in H_2 is statistically close to the distribution of $(\mathbf{A}, \mathbf{k}_i^{(j)}, p_i^{(j)})$ for $i : y_i^{(j)} \in Y_X^{(j)}$, $j \in [Q]$ where we first sample $\mathbf{A} \leftarrow R_q^{1 \times m}$ and $\mathbf{k}_i^{(j)} \leftarrow D_{R_q, \chi'}^{2m}$ and set $p_i^{(j)} = [\mathbf{A}|H(y_i^{(j)})]\mathbf{k}_i^{(j)}$. Hence, we have $H_2 \approx_s H_3$. $\square$

Lemma 5.2. $H_3 \approx_c H_4$ under the $\text{RLWE}_{n, m(t+1)+1, q, \chi}$ assumption where t is the size of the attribute X for the challenge query.

Proof. We prove that if there is an adversary $\mathcal{A}$ that distinguishes H_3 and H_4 with non-negligible probability, we can construct a reduction $\mathcal{B}$ that breaks the $\text{RLWE}_{n, m(t+1)+1, q, \chi}$ assumption. We can construct $\mathcal{B}$ as follows.

- $\mathcal{A}$ declares the challenge attribute $X = (x_1, \dots, x_t)$ at the beginning of the game. $\mathcal{B}$ obtains a RLWE instance $(\hat{\mathbf{A}}, \mathbf{v}^\top)$ where $\mathbf{v}^\top = s\hat{\mathbf{A}} + \mathbf{e}^\top$ or $\mathbf{v}^\top \leftarrow R_q^{m(t+1)+1}$. $\mathcal{B}$ parses $\hat{\mathbf{A}} = [\mathbf{A}|D_1| \dots |D_t|p]$ and gives $\text{mpk} = (\mathbf{A}, p)$ to $\mathcal{A}$.
- Random oracle queries: $\mathcal{B}$ first sets $\mathcal{D}(x_i) = D_i$ for all $i \in [t]$, where $\mathcal{D}$ is the dictionary for the random oracle simulation. During the game, when H is invoked on input x , the challenger checks if $\mathcal{D}(x) = \perp$. If this is the case, the challenger samples $(D, T_D) \leftarrow \text{RingTrapGen}(1^n, q)$, updates $\mathcal{D}(x) = D$, and returns D . Otherwise, it just returns $\mathcal{D}(x)$.
- Secret key queries: when $\mathcal{A}$ queries $\mathcal{B}$ on $Y = (y_1, \dots, y_u)$, $\mathcal{B}$ defines $Y_X = Y \cap X$, $Y_{\bar{X}} = Y \setminus X$, and computes $\mathbf{k}_i$ for $i \in [u]$ as in [Eq. \(8\)](#). Then, returns $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u)$ to $\mathcal{A}$.
- Challenge: after $\mathcal{A}$ outputs the challenge messages (μ_0, μ_1) , $\mathcal{B}$ chooses $b \leftarrow \{0, 1\}$ and gives the challenge ciphertext $\text{ct} = \mathbf{v}^\top + (0^{m(t+1)}, \mu_b \lfloor q/2 \rfloor)$.
- Output: if $\mathcal{A}$ outputs b' at the end of the game, $\mathcal{B}$ outputs 1 if and only if $b = b'$.

Observe that $\mathcal{A}$'s view corresponds to H_3 if $\mathbf{v}^\top = s\hat{\mathbf{A}} + \mathbf{e}^\top$ and H_4 otherwise. This concludes the proof. $\square$

5.4 Our CP-ABE from Ring-LWE

We obtain the construction by using the *statistical* instantiation of the Ring-LWE trapdoor [[Pei10](#), [EBB14](#), [SSTX09](#)] as in [Section 5.3](#).

Construction. Let $k = \lceil \log q \rceil$, $\overline{m} = \omega(\log n) + k$, $m = \overline{m} + k$.

Setup(1^λ): It outputs $(\text{mpk}, \text{msk}) := (\mathbf{A} \in R_q^{1 \times m}, \mathbf{T}_\mathbf{A} \in R_q^{\overline{m} \times k}) \leftarrow \text{RingTrapGen}(1^n, q)$.

Enc($\text{mpk}, X, \mu \in \mathcal{R}_2$): It parses $X = \{x_1, \dots, x_t\}$, where it must hold that $t \leq L$, and outputs ct as follows.

$$\begin{aligned} s &\xleftarrow{\$} R_q, \mathbf{e}_1 \leftarrow \chi^{m(t+1)}, \mathbf{e}_2 \leftarrow \chi^{mt}, e_3 \leftarrow \chi, \\ \mathbf{R} &\leftarrow D_{\mathcal{R}, \chi_R}^{\text{coeff}, mt \times m(t-1)}, \\ \left[\underbrace{D_i}_m \mid \underbrace{B_i}_{m(2t-1)} \mid \underbrace{p_i}_1 \right] &:= [H_t(x_i)], \\ D &:= [D_1 \mid \dots \mid D_t] \in R_q^{1 \times mt}, \end{aligned}$$

$$\begin{aligned} p &:= \sum_{i \in [t]} p_i \in R_q, B := \sum_{i \in [t]} B_i \in R_q^{1 \times m(2t-1)}, \\ \text{ct} &= (\mathbf{c}_1^T, \mathbf{c}_2^T, c_3) = (s[\mathbf{A}|D] + \mathbf{e}_1^T, sB + \mathbf{e}_2^T [\mathbf{I}_{mt} | \mathbf{R}], sp + e_3 + \mu \lfloor q/2 \rfloor). \end{aligned}$$

KeyGen($\text{mpk}, \text{msk}, Y$): It parses $Y = \{y_1, \dots, y_u\}$. It computes, $\mathbf{r}_\ell \leftarrow D_{\mathbb{Z}, \chi''}^{m(2\ell-1)}$ for $\ell \in [L]$ and, for $(i, \ell) \in [u] \times [L]$,

$$\begin{aligned} \left[\underbrace{\widehat{D}_{i,\ell}}_m \mid \underbrace{\widehat{B}_{i,\ell}}_{m(2t-1)} \mid \underbrace{\widehat{p}_{i,\ell}}_1 \right] &:= H_\ell(y_i), \\ \mathbf{k}_{i,\ell} &\in R_q^{m(t+1)} \leftarrow \text{RingSampleLeft}([\mathbf{A} | \widehat{D}_{i,\ell}], \mathbf{T}_\mathbf{A}, \widehat{B}_{i,\ell} \mathbf{r}_\ell + \widehat{p}_{i,\ell}, \chi'). \end{aligned}$$

Then it outputs $\text{sk} = \{\mathbf{k}_{1,\ell}, \dots, \mathbf{k}_{u,\ell}, \mathbf{r}_\ell\}_{\ell \in [L]}$.

Dec($\text{mpk}, \text{ct}, \text{sk}, X, Y$): It parses $\text{ct} = (\mathbf{c}_1^T, \mathbf{c}_2^T, c_3)$, where $\mathbf{c}_1^T = [c_{1,0}^T \mid \dots \mid c_{1,t}^T] \in R_q^{1 \times m(t+1)}$, $\mathbf{c}_2^T \in R_q^{1 \times m(2t-1)}$, and $\text{sk} = \{\mathbf{k}_{1,\ell}, \dots, \mathbf{k}_{u,\ell}, \mathbf{r}_\ell\}_{\ell \in [L]}$, and $X = \{x_1, \dots, x_t\}$, $Y = \{y_1, \dots, y_u\}$. If $X \subseteq Y$, it computes $\rho : [t] \rightarrow [u]$ such that $x_i = y_{\rho(i)}$ and outputs $\left\lfloor \frac{2}{q} v \right\rfloor$, where

$$v := c_3 - \left(\sum_{i \in [t]} [c_{1,0}^T \mid c_{1,i}^T] \mathbf{k}_{\rho(i)} - \mathbf{c}_2^T \mathbf{r}_t \right).$$

Parameter sizes and comparison with LWE based construction. The parameter sizes of LWE based CP-ABE construction are $|\text{mpk}| = (2n^2k)k$, $|\text{ct}| = (2nk(3t+1))k$, and $|\text{sk}| = (2nk((t+1)uL + L^2))k$, where ct encodes 1-bit of message. In contrast, the parameter sizes of our Ring-LWE based CP-ABE construction are $|\text{mpk}| = mnk$, $|\text{ct}| = (m3t+1)nk$, and $|\text{sk}| = (m((t+1)uL + L^2))nk$, where ct encodes n -bit of message. Thus, with $m = 2k + \omega(\log n)$, the mpk is $n \times$ shorter and the ct is $n \times$ shorter (per message bit), while the sk sizes are comparable.

Parameter bounds for correctness. For correctness of the scheme ([Theorem 5.4](#)), we need to set q such that $q \geq 5(\sqrt{\lambda}\chi + mL(2\chi' + mL\chi''\chi_R)\lambda\chi)$.

Parameter bounds for security. For the security of the scheme, we need to set parameters subject to constraints (1)-(6) in [Theorem 3.4](#). Additionally, for $\text{RLWE}_{n,m(2t+1)+1,q,\chi}$ to hold, we need $n \geq \frac{\log_2(q/\chi)}{4\log_2(\delta)}$, where δ is the root Hermite factor and $\delta < 1.006$ corresponds to at least 100 bits of security [[CN11](#)].

Empirical parameter choices. By the [Assumption B.2](#), we can set $\chi_R = O(1)$. Further, we choose $n = O(\lambda)$, $\chi = O(n^{1.5}m^{1.5}L^{1.5}\sqrt{\log n})$, $\chi' = \chi'' = O((nk \log(k))^{0.5})$, $q = O(L^{3.5}n^3(\log(Ln))^4 \log(n))$, and $m = 2k + \omega(\log n)$ (where $k = \lceil \log q \rceil$).

Correctness. For correctness, we have the following theorem.

Theorem 5.4. Suppose $q \geq 5(\sqrt{\lambda}\chi + mL(2\chi' + mL\chi''\chi_R)\lambda\chi)$. Then, our scheme is correct.

Proof. In decryption, suppose $\text{ct} = (\mathbf{c}_1^\top, \mathbf{c}_2^\top, c_3)$ where $\mathbf{c}_1^\top = [\mathbf{c}_{1,0}^\top | \cdots | \mathbf{c}_{1,t}^\top]$, $\mathbf{sk} = \{\mathbf{k}_{1,\ell}, \dots, \mathbf{k}_{u,\ell}, \mathbf{r}_\ell\}_{\ell \in [L]}$ are a ciphertext and a secret key for X and Y , respectively, such that $X \subseteq Y$. In decryption, we have

$$\begin{aligned}
v &= c_3 - \left(\sum_{i \in [t]} [\mathbf{c}_{1,0}^\top | \mathbf{c}_{1,i}^\top] \mathbf{k}_{\rho(i),t} - \mathbf{c}_2^\top \mathbf{r}_t \right) \\
&= s \sum_{i \in [t]} p_i + e_3 + \mu \lfloor q/2 \rfloor - \sum_{i \in [t]} (s[\mathbf{A}|D_i] + [\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,i}^\top]) \mathbf{k}_{\rho(i),t} + \\
&\quad \left(s \sum_{i \in [t]} B_i + \mathbf{e}_2^\top [\mathbf{I}_{mt} | \mathbf{R}] \right) \mathbf{r}_t \\
&= s \sum_{i \in [t]} p_i + e_3 + \mu \lfloor q/2 \rfloor - \sum_{i \in [t]} (s(B_i \mathbf{r}_t + p_i) + [\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,i}^\top] \mathbf{k}_{\rho(i),t}) + \\
&\quad \left(s \sum_{i \in [t]} B_i + \mathbf{e}_2^\top [\mathbf{I}_{mt} | \mathbf{R}] \right) \mathbf{r}_t \\
&= \underbrace{\mu \lfloor q/2 \rfloor + e_3 - \sum_{i \in [t]} [\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,i}^\top] \mathbf{k}_{\rho(i),t} + \mathbf{e}_2^\top [\mathbf{I}_{mt} | \mathbf{R}] \mathbf{r}_t}_{\tilde{e}}
\end{aligned}$$

where $[D_i | B_i | p_i] = H_t(x_i)$, $\mathbf{e}_1^\top = [\mathbf{e}_{1,0}^\top | \cdots | \mathbf{e}_{1,t}^\top]$. From the parameter setting and [Lemma 2.2](#), we have $|e_3|_\infty \leq \sqrt{\lambda} \chi$, $||[\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,i}^\top]|_\infty \leq \sqrt{\lambda} \chi$, $|\mathbf{e}_2|_\infty \leq \sqrt{\lambda} \chi$, and $|\mathbf{r}_t|_\infty \leq \sqrt{\lambda} \chi''$ with overwhelming probability. Thanks to the property of ideal lattice trapdoors, the distribution of $\mathbf{k}_{\rho(i),t}$ is statistically close to $[\mathbf{A}|D_i]_{\chi'}^{-1}(B_i \mathbf{r}_t + p_i)$. Since the marginal distribution of p_i is uniform in R_q , the marginal distribution of $\mathbf{k}_{\rho(i),t}$ is statistically close to $D_{\mathcal{R}, \chi'}^{2m}$ from [Lemma 2.3](#). Thus, we have $|\mathbf{k}_{\rho(i),t}|_\infty \leq \sqrt{\lambda} \chi'$ with overwhelming probability again from [Lemma 2.2](#). In summary, we have, with overwhelming probability,

$$\begin{aligned}
|\tilde{e}| &= \left| e_3 - \sum_{i \in [t]} [\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,i}^\top] \mathbf{k}_{\rho(i),t} + \mathbf{e}_2^\top [\mathbf{I}_{mt} | \mathbf{R}] \mathbf{r}_t \right| \\
&\leq |e_3|_\infty + t |[\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,i}^\top]|_\infty |\mathbf{k}_{\rho(i),t}|_\infty \cdot 2m \cdot n \\
&\quad + (mt \cdot (1 + m(t-1)|\mathbf{R}|_\infty)) \cdot |\mathbf{e}_2^\top|_\infty |\mathbf{r}_t|_\infty \\
&\leq \sqrt{\lambda} \chi + 2mt \lambda \chi \chi' + m^2 t^2 \lambda^{1.5} \chi \chi'' \chi_R \\
&\leq \sqrt{\lambda} \chi + mL(2\chi' + mL\chi'' \chi_R) \lambda \chi
\end{aligned}$$

Thus, $|\tilde{e}| \leq q/5$ with overwhelming probability and correctness holds. $\square$

Security Analysis. For security, we have the following theorem.

Theorem 5.5. *Let $t \leq L$ be the size of the attribute X for the challenge query. Denote $k := \lceil \log q \rceil$ and $g \leq n$ be the number of distinct prime ideal factors of $q\mathcal{R}$. Suppose the following conditions hold:*

1. $m = \bar{m} + k = 2k + \omega(\log n)$.
2. $\chi_R \leq \frac{\sqrt{\pi}}{2} \cdot \frac{\sqrt{q}}{\sqrt{\ln 4}}$.
3. One of the following:
 - (Unconditional.) $mt \log \chi_R \geq 2 \log(m(t-1)/\varepsilon) + (t+n) \log q + \log g + mt$, where ε is any negligible function; or
 - (Conditional on [Assumption B.2](#).) $mt \log \chi_R \geq 2 \log(m(t-1)/\varepsilon) + (t+1) \log q + \log g + mt$.
4. $\chi \geq \tau \cdot \omega(\sqrt{\log n})$, where $\tau = O(\chi_R \cdot n^{3/2} \cdot m^{3/2} \cdot t^{3/2})$.
5. $\chi' = \tau' \cdot \omega(\sqrt{\log n})$, where $\tau' = O(\sqrt{n \log q})$.
6. $\chi'' = \tau'' \cdot \omega(\sqrt{\log n})$, where $\tau'' = O(\sqrt{n \log q})$.
7. the hash function H is modeled as a random oracle
8. $\text{RLWE}_{n, m(2t+1)+1, q, \chi}$ holds.

Then, our scheme satisfies selective security as per [Definition 2.1](#).

Proof. We prove the theorem by a series of hybrids described as follows.

- H_0 : This is the original game. In this game, $\mathcal{A}$ first outputs the challenge attribute $X = (x_1, \dots, x_t)$. Then, the challenger computes

$$(\text{mpk}, \text{msk}) = (\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \text{RingTrapGen}(1^n, q)$$

and gives mpk to $\mathcal{A}$. When $\mathcal{A}$ asks for a secret key for $Y = (y_1, \dots, y_u)$, the challenger computes $\mathbf{r}_\ell \leftarrow D_{\mathbb{Z}, \chi}^{m(2\ell-1)}$ for $\ell \in [L]$ and, for $(i, \ell) \in [u] \times [L]$,

$$\underbrace{[\hat{D}_{i,\ell}]_m}_m \mid \underbrace{[\hat{B}_{i,\ell}]_{m(2t-1)}}_{m(2t-1)} \mid \underbrace{[\hat{p}_{i,\ell}]_1}_1 := H_\ell(y_i),$$

$$\mathbf{k}_{i,\ell} \in R_q^{m(t+1)} \leftarrow \text{RingSampleLeft}([\mathbf{A}|\hat{D}_{i,\ell}], \mathbf{T}_\mathbf{A}, \hat{B}_{i,\ell}\mathbf{r}_\ell + \hat{p}_{i,\ell}, \chi').$$

and gives $\text{sk} = \{\mathbf{k}_{1,\ell}, \dots, \mathbf{k}_{u,\ell}, \mathbf{r}_\ell\}_{\ell \in [L]}$ to $\mathcal{A}$. For the challenge, $\mathcal{A}$ outputs μ_0, μ_1 . Then, the challenger chooses a challenge bit $b \leftarrow \{0, 1\}$ and gives ct to $\mathcal{A}$ where

$$s \xleftarrow{\$} R_q, \mathbf{e}_1 \leftarrow \chi^{m(t+1)}, \mathbf{e}_2 \leftarrow \chi^{mt}, e_3 \leftarrow \chi$$

$$\mathbf{R} \xleftarrow{\$} \{-1, 1\}^{mt \times m(t-1)}$$

$$\underbrace{[D_i]_m}_m \mid \underbrace{[B_i]_{m(2t-1)}}_{m(2t-1)} \mid \underbrace{[p_i]_1}_1 := [H_t(x_i)]$$

$$D := [D_1 \mid \dots \mid D_t] \in R_q^{1 \times mt},$$

$$p := \sum_{i \in [t]} p_i \in R_q, B := \sum_{i \in [t]} B_i \in R_q^{1 \times m(2t-1)}$$

$$\text{ct} = (\mathbf{c}_1^T, \mathbf{c}_2^T, c_3) = (s[\mathbf{A}|D] + \mathbf{e}_1^T, sB + \mathbf{e}_2^T[\mathbf{I}_{mt}|\mathbf{R}], sp + e_3 + \mu_b \lfloor q/2 \rfloor).$$

The post-challenge key queries are the same as the pre-challenge queries. Finally, $\mathcal{A}$ outputs b' . During the game, the challenger simulates the random oracle as follows. It maintains a dictionary $\mathcal{D}_\ell$ for $\ell \in [L]$ that are initially empty. When H_ℓ is invoked on input x , the challenger checks if $\mathcal{D}_\ell(x) = \perp$. If this is the case, the challenger samples $[D|B|p] \leftarrow R_q^{1 \times (2mL+1)}$, updates $\mathcal{D}_\ell(x) = [D|B|p]$, and returns $[D|B|p]$. Otherwise, it returns $\mathcal{D}_\ell(x)$.

- H_1 : This is the same as H_0 except that the challenger simulates the random oracle as follows. Right after $\mathcal{A}$ outputs the challenge attribute $X = (x_1, \dots, x_t)$, the challenger for all $i \in [t]$: computes $[D_i|B_i|p_i] \leftarrow R_q^{1 \times (2mt+1)}$ and updates $\mathcal{D}_t(x_i) = [D_i|B_i|p_i]$. During the game, when H_ℓ is invoked on input x , the challenger checks if $\mathcal{D}_\ell(x) = \perp$. If this is the case, the challenger samples $(D, \mathbf{T}_D) \leftarrow \text{RingTrapGen}(n, m, q)$, $[B|p] \leftarrow R_q^{1 \times ((2\ell-1)m+1)}$, updates $\mathcal{D}_\ell(x) = [D|B|p]$, and returns $\mathcal{D}_\ell(x)$. Otherwise, it returns $\mathcal{D}_\ell(x)$. From the trapdoor distribution property of ideal lattices trapdoors ([Theorem B.1](#), [Theorem B.2](#)). D is statistically close to uniform, and thus H_0 and H_1 are statistically indistinguishable given $m = 2k + \omega(\log n)$.
- H_2 : This is the same as H_1 except the followings. In this hybrid, the challenger samples $\mathbf{A} \leftarrow R_q^{1 \times m}$ instead of using TrapGen for mpk . For all secret key queries on $Y = (y_1, \dots, y_u)$, let $Y_X = Y \cap X$ and $Y_{\bar{X}} = Y \setminus X$. The challenger computes for $(i, \ell) \in [u] \times [L]$,

$$\mathbf{r}_\ell \leftarrow D_{\mathbb{Z}, \chi}^{m(2t-1)}, \underbrace{[\hat{D}_{i,\ell}]_m}_m \mid \underbrace{[\hat{B}_{i,\ell}]_{m(2t-1)}}_{m(2t-1)} \mid \underbrace{[\hat{p}_{i,\ell}]_1}_1 = H_\ell(y_i),$$

$$\mathbf{k}_{i,\ell} \leftarrow \begin{cases} [\mathbf{A}|\hat{D}_{i,\ell}]_{\chi'}^{-1}(\hat{B}_{i,\ell}\mathbf{r}_\ell + \hat{p}_{i,\ell}) & (i : y_i \in Y_X \wedge \ell = t) \\ \text{RingSampleLeft}([\mathbf{A}|\hat{D}_{i,\ell}], \mathbf{T}_{\hat{D}_{i,\ell}}, \hat{B}_{i,\ell}\mathbf{r}_\ell + \hat{p}_{i,\ell}, \chi') & (\text{otherwise}) \end{cases}$$

and gives $\text{sk} = \{\mathbf{k}_{1,\ell}, \dots, \mathbf{k}_{u,\ell}, \mathbf{r}_\ell\}_{\ell \in [L]}$ to $\mathcal{A}$. Note that the sampling of $\mathbf{k}_{i,t}$ for $i : y_i \in Y_X$ is inefficient. It is easy to see that $H_1 \approx_s H_2$ since $\mathbf{A}$ is statistically close to uniform if $m = 2k + \omega(\log n)$ and, for all $(i, \ell) \in [u] \times [L]$,

$$\begin{aligned} & \text{RingSampleLeft}([\mathbf{A}|\hat{D}_{i,\ell}], \mathbf{T}_{\hat{D}_{i,\ell}}, \hat{B}_{i,\ell}\mathbf{r}_\ell + \hat{p}_{i,\ell}, \chi') \\ & \approx_s \text{RingSampleLeft}([\mathbf{A}|\hat{D}_{i,\ell}], \mathbf{T}_\mathbf{A}, \hat{B}_{i,\ell}\mathbf{r}_\ell + \hat{p}_{i,\ell}, \chi') \\ & \approx_s [\mathbf{A}|\hat{D}_{i,\ell}]_{\chi'}^{-1}(\hat{B}_{i,\ell}\mathbf{r}_\ell + \hat{p}_{i,\ell}) \end{aligned}$$

which follows from the preimage sampling property of ideal lattice trapdoors ([Theorem B.1](#)) if $\chi' \geq \tau' \cdot \omega(\sqrt{\log n})$ and $\tau' = O(\sqrt{n \log q})$.

- H_3 : This game is the same as H_2 except the followings. For all secret key queries on $Y = (y_1, \dots, y_u)$, let $Y_X = Y \cap X$, $Y_{\bar{X}} = Y \setminus X$, and $Y_X = (y_{i_1}, \dots, y_{i_k})$. The challenger computes

$$\underbrace{[\hat{D}_{i,t}]_m}_{m} \mid \underbrace{[\hat{B}_{i,t}]_{m(2t-1)}}_{m(2t-1)} \mid \underbrace{[\hat{p}_{i,t}]_1}_1 = H_t(y_i) \quad \text{for } i \in [u], \quad (9)$$

$$\mathbf{k}_{i,t} \leftarrow D_{\mathbb{Z}, \chi'}^{2m} \quad \text{for } i : y_i \in Y_X$$

$$\mathbf{r}_t \leftarrow \begin{bmatrix} \hat{B}_{i_1,t} \\ \vdots \\ \hat{B}_{i_k,t} \end{bmatrix}_{\chi''}^{-1} \left(\begin{bmatrix} [\mathbf{A} | \hat{D}_{i_1,t}] \mathbf{k}_{i_1,t} - \hat{p}_{i_1,t} \\ \vdots \\ [\mathbf{A} | \hat{D}_{i_k,t}] \mathbf{k}_{i_k,t} - \hat{p}_{i_k,t} \end{bmatrix} \right) \quad (10)$$

$$\mathbf{k}_{i,t} \leftarrow \text{RingSampleLeft}([\mathbf{A} | \hat{D}_{i,t}], \mathbf{T}_{\hat{D}_{i,t}}, \hat{B}_{i,t} \mathbf{r}_t + \hat{p}_{i,t}, \chi') \quad \text{for } i : y_i \in Y_{\bar{X}}$$

and gives $\text{sk} = \{\mathbf{k}_{1,\ell}, \dots, \mathbf{k}_{u,\ell}, \mathbf{r}_\ell\}_{\ell \in [L]}$ to $\mathcal{A}$ where $\{\mathbf{k}_{1,\ell}, \dots, \mathbf{k}_{u,\ell}, \mathbf{r}_\ell\}_{\ell \in [L] \setminus \{t\}}$ is generated in the same way as in H_2 . We prove $H_2 \approx_s H_3$ in [Lemma 5.3](#).

- H_4 : This game is the same as H_3 except the followings. The challenger generates the challenge ciphertext as

$$\text{ct} = (\mathbf{c}_1^\top, \mathbf{c}_2^\top, c_3) \leftarrow R_q^{(t+1)m} \times R_q^{m(2t-1)} \times R_q$$

We prove $H_3 \approx_c H_4$ in [Lemma 5.4](#). □

Lemma 5.3. $H_2 \approx_s H_3$ if $m = 2k + \omega(\log n)$, $\chi' = \tau' \cdot \omega(\sqrt{\log n})$, and $\chi'' = \tau'' \cdot \omega(\sqrt{\log n})$, where $\tau' = O(\sqrt{n \log q})$ and $\tau'' = O(\sqrt{n \log q})$.

Proof. Let Q be the number of secret key queries, $Y^{(j)} = (y_1^{(j)}, \dots, y_{u^{(j)}}^{(j)})$ be the attribute for the j -th secret key query, let $Y_X^{(j)} = Y^{(j)} \cap X$, $|Y_X^{(j)}| = k^{(j)}$, and $Y_{\bar{X}}^{(j)} = (y_{i_1}^{(j)}, \dots, y_{i_k}^{(j)})$. For $y_{i_1}^{(j)}$ and $y_{i_k}^{(j)}$ in the previous sentence, we omit the index (j) for conciseness in some parts, e.g, i_k should be $i_k^{(j)}$ precisely. We sometimes omit (j) similarly in what follows.

Let $\mathbf{B} = \begin{bmatrix} B_1 \\ \vdots \\ B_t \end{bmatrix}$ and $\mathbf{p} = \begin{bmatrix} p_1 \\ \vdots \\ p_t \end{bmatrix}$ where $[D_i | B_i | p_i] = H_t(x_i)$. Thanks to [Lemma B.5](#), $\mathbf{Br}_t^{(1)}, \dots, \mathbf{Br}_t^{(Q)}$

where $\mathbf{r}_t^{(j)} \leftarrow D_{\mathbb{Z}, \chi''}^{m(2L-1)}$ for $j \in [Q]$ are distributed statistically close to uniform in $(R_q^t)^Q$, because $t \leq L$, $m = 2k + \omega(\log n)$, and $\chi'' = \tau'' \cdot \omega(\sqrt{\log n})$, where $\tau'' = O(\sqrt{n \log q})$. Hence, the joint distribution of $\hat{B}^{(j)} \mathbf{r}_t^{(j)} + \hat{p}^{(j)}$ for all $j \in [Q]$ are also statistically close to uniform random, where $[\hat{D}_i^{(j)} | \hat{B}_i^{(j)} | \hat{p}_i^{(j)}] = H_t(y_i^{(j)})$ for $i \in \{i_1, \dots, i_k\}$ and

$$\hat{\mathbf{B}}^{(j)} = \begin{bmatrix} \hat{B}_{i_1}^{(j)} \\ \vdots \\ \hat{B}_{i_k}^{(j)} \end{bmatrix}, \quad \hat{\mathbf{p}}^{(j)} = \begin{bmatrix} \hat{p}_{i_1}^{(j)} \\ \vdots \\ \hat{p}_{i_k}^{(j)} \end{bmatrix}$$

since $\hat{B}_i^{(j)} \in \{\mathbf{B}_1, \dots, \mathbf{B}_t\}$ and $\hat{p}_i^{(j)} \in \{\mathbf{p}_1, \dots, \mathbf{p}_t\}$ for all $i : y_i^{(j)} \in Y_X^{(j)}, j \in [Q]$. Recall also that $\mathbf{A}$ is uniformly distributed in $R_q^{1 \times m}$. Thanks to [Lemma B.6](#) for $\ell_1 = |X|$ and $\ell_2 = Q$, the joint distribution of $(\mathbf{A}, \mathbf{k}_{i,t}^{(j)}, \hat{B}_i^{(j)} \mathbf{r}_t^{(j)} + \hat{p}_i^{(j)})$ for $i : y_i^{(j)} \in Y_X^{(j)}$ and $j \in [Q]$ in H_2 is statistically close to the distribution of $(\mathbf{A}, \mathbf{k}_{i,t}^{(j)}, \hat{B}_i^{(j)} \mathbf{r}_t^{(j)} + \hat{p}_i^{(j)})$ where we first sample $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{k}_{i,t}^{(j)} \leftarrow D_{\mathbb{Z}, \chi}^{2m}$ and then sample

$$\mathbf{B} \leftarrow R_q^{t \times m(2t-1)}, \mathbf{r}_t^{(j)} \leftarrow D_{\mathbb{Z}, \chi''}^{m(2t-1)}, \mathbf{p} \leftarrow R_q^t$$

conditioned on $[\mathbf{A} | \hat{D}_i^{(j)}] \mathbf{k}_{i,t}^{(j)} = \hat{B}_i^{(j)} \mathbf{r}_t^{(j)} + \hat{p}_i^{(j)}$ for $i : y_i \in Y_X^{(j)}$ and $j \in [Q]$. This is conceptually the same as sampling $\mathbf{B} \leftarrow R_q^{t \times m(2t-1)}$, $\mathbf{p} \leftarrow R_q^t$, and then sampling $\mathbf{r}_t^{(j)}$ for each secret key query as [Eq. \(10\)](#). Recall that the difference between H_2 and H_3 lies in $\mathbf{k}_{i,t}^{(j)}, \mathbf{r}_t^{(j)}$ for $i : y_i^{(j)} \in Y_X^{(j)}, j \in [Q]$. Thus, we have $H_2 \approx_s H_3$. □

Lemma 5.4. Let $t \leq L$ be the size of the attribute X for the challenge query. Denote $k := \lceil \log q \rceil$ and $g \leq n$ be the number of distinct prime ideal factors of $q\mathcal{R}$. $H_3 \approx_c H_4$ under the $\text{RLWE}_{n, m(2t+1)+1, q, \chi}$ assumption if the following conditions hold:

1. (Ring-LHL width ceiling.) $\chi_R \leq \frac{\sqrt{\pi}}{2} \cdot \frac{\sqrt{q}}{\sqrt{\ln 4}}$.
2. (Ring-LHL entropy condition.) *One of the following:*
 - (Unconditional.) $mt \log \chi_R \geq 2 \log(m(t-1)/\varepsilon) + (t+n) \log q + \log g + mt$, where ε is any negligible function; or
 - (Conditional on [Assumption B.2](#).) $mt \log \chi_R \geq 2 \log(m(t-1)/\varepsilon) + (t+1) \log q + \log g + mt$.
3. (Trapdoor quality condition.) $\chi \geq \tau \cdot \omega(\sqrt{\log n})$, where $\tau = O(\chi_R \cdot n^{3/2} \cdot m^{3/2} \cdot t^{3/2})$ is the trapdoor quality from [Corollary B.2](#).
4. (Gaussian noise.) $\chi \geq \omega(\sqrt{\log n})$.

Proof. We prove that if there is an adversary $\mathcal{A}$ that distinguishes H_3 and H_4 with non-negligible probability, we can construct a reduction $\mathcal{B}$ that breaks the $\text{RTRLWE}_{n,m+tm+1,m,q,\chi,\chi_R,t}$ assumption. Then, [Theorem 5.1](#) immediately implies [Lemma 5.4](#). We can construct $\mathcal{B}$ as follows.

- $\mathcal{A}$ declares the challenge attribute $X = (x_1, \dots, x_t)$ at the beginning of the game. $\mathcal{B}$ outputs $\mathbf{u}^\top = (\underbrace{1, \dots, 1}_t)$ for the security game for the related trapdoor LWE assumption and obtains $(\hat{\mathbf{A}}, \mathbf{B}, \mathbf{v}, \hat{\mathbf{v}}^\top)$ where $(\mathbf{v}^\top, \hat{\mathbf{v}}^\top) = (s\hat{\mathbf{A}}^\top + \mathbf{e}^\top, s \cdot \mathbf{u}^\top \mathbf{B} + \hat{\mathbf{e}}^\top)$ or $(\mathbf{v}^\top, \hat{\mathbf{v}}^\top) \leftarrow R_q^{m+tm+1} \times R_q^{m(2t-1)}$. $\mathcal{B}$ parses $\hat{\mathbf{A}} = [\mathbf{A}|D_1| \dots |D_t|p]$, $\mathbf{B} = \begin{bmatrix} B_1 \\ \vdots \\ B_t \end{bmatrix}$, and gives $\text{mpk} = \mathbf{A}$ to $\mathcal{A}$.
- Random oracle queries: $\mathcal{B}$ samples $p_1, \dots, p_t \leftarrow R_q$ conditioned on $\sum_{i \in [t]} p_i = p$. $\mathcal{B}$ sets $\mathcal{D}_t(x_i) = [D_i|B_i|p_i]$ for all $i \in [t]$, where $\mathcal{D}_t$ is the dictionary for the random oracle simulation. Note that since p is kept hidden from the adversary, $p_1, \dots, p_t$ are distributed randomly from the view point of $\mathcal{A}$. During the game, when H_t is invoked on input x , the challenger checks if $\mathcal{D}_t(x) = \perp$. If this is the case, the challenger samples $(D, \mathbf{T}_D) \leftarrow \text{RingTrapGen}(n, m, q)$, $[B|p] \leftarrow R_q^{1 \times ((2t-1)m+1)}$, updates $\mathcal{D}_t(x) = [D|B|p]$, and returns $[D|B|p]$. Otherwise, it just returns $\mathcal{D}_t(x)$. The simulation of H_ℓ for $\ell \neq t$ is the same as in H_0 .
- Secret key queries: when $\mathcal{A}$ queries $\mathcal{B}$ on $Y = (y_1, \dots, y_u)$, $\mathcal{B}$ defines $Y_X = Y \cap X$, $Y_{\bar{X}} = Y \setminus X$, $Y_X = (y_{i_1}, \dots, y_{i_k})$, $\mathbf{m}_i^\top = (0^{i-1}, 1, 0^{t-i}) \in \mathbb{Z}_q^t$ for $i \leq u$. Then, it computes

$$\begin{aligned} [\hat{D}_i|\hat{B}_i|\hat{p}_i] &= H_t(y_i) \quad \text{for } i \in [u] \\ \mathbf{k}_{i,t} &\leftarrow D_{\mathbb{Z},\chi}^{2m} \quad \text{for } i : y_i \in Y_X \end{aligned}$$

and makes a query $\text{tdO}(\mathbf{M}, \mathbf{t})$ and obtains $\mathbf{r}_t \in R_q^{m(2t-1)}$ as a reply, where $\phi : \{i_1, \dots, i_k\} \rightarrow [t]$ is an injective function such that $B_{\phi(i)} = \hat{B}_i$,

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_{\phi(i_1)}^\top \\ \vdots \\ \mathbf{m}_{\phi(i_k)}^\top \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} [\mathbf{A}|\hat{D}_{i_1}] \mathbf{k}_{i_1} - \hat{p}_{i_1} \\ \vdots \\ [\mathbf{A}|\hat{D}_{i_k}] \mathbf{k}_{i_k} - \hat{p}_{i_k} \end{bmatrix}$$

Finally, it computes $\mathbf{r}_\ell \leftarrow D_{\mathcal{R},\chi''}^{m(2\ell-1)}$ for $\ell \in [L] \setminus \{t\}$ and

$$\mathbf{k}_{i,\ell} \leftarrow \text{RingSampleLeft}([\mathbf{A}|\hat{D}_{i,\ell}], \mathbf{T}_{\hat{D}_{i,\ell}}, \hat{B}_{i,\ell} \mathbf{r}_\ell + \hat{p}_{i,\ell}, \chi'') \quad \text{for } i : y_i \in Y_{\bar{X}} \vee \ell \neq t$$

where $[\hat{D}_{i,\ell}|\hat{B}_{i,\ell}|\hat{p}_{i,\ell}] = H_\ell(y_i)$. Then, returns $\text{sk} = \{\mathbf{k}_{1,\ell}, \dots, \mathbf{k}_{u,\ell}, \mathbf{r}_\ell\}_{\ell \in [L]}$ to $\mathcal{A}$.

- Challenge: after $\mathcal{A}$ outputs the challenge messages (μ_0, μ_1) , $\mathcal{B}$ chooses $b \leftarrow \{0, 1\}$ and gives the challenge ciphertext $\text{ct} = (\mathbf{v}^\top + (0^{(t+1)m}, \mu_b \lfloor q/2 \rfloor), \hat{\mathbf{v}}^\top)$.
- Output: if $\mathcal{A}$ outputs b' at the end of the game, $\mathcal{B}$ outputs 1 if and only if $b = b'$.

Observe that $\mathcal{A}$'s view corresponds to H_3 if $(\mathbf{v}^\top, \hat{\mathbf{v}}^\top) = (s\hat{\mathbf{A}} + \mathbf{e}^\top, s \cdot \mathbf{u}^\top \mathbf{B} + \hat{\mathbf{e}}^\top)$ and H_4 otherwise. Additionally, if $\mathcal{A}$ is admissible, then $\mathcal{B}$ is admissible. This is because $Y_X \subsetneq X$ for all Y in secret key queries, and thus it is obvious that $k < t$ and thus $\{\mathbf{u}, \mathbf{m}_{\phi(i_1)}, \dots, \mathbf{m}_{\phi(i_k)}\}$ are linearly independent. This concludes the proof. $\square$

Looking ahead, the security proof relies on the RTRLWE assumption, and thus we need to set parameters such that RTRLWE holds (as per [Theorem 5.1](#)).

Table 2. Ring parameters

Scheme	n	$k = \lceil \log q \rceil$	χ	$\approx$ security of LWE
KP-ABE	2048	57	5.9	128
CP-ABE 1	2048	60	6.3	121
CP-ABE 2	4096	122	3.45e6	167

6 Implementation

We implemented Ring-LWE instantiations of the main ABE variants, namely the KP-ABE scheme of [Section 5.3](#) and the CP-ABE scheme of [Section 5.4](#). Our goal was to achieve practical performance (seconds to minutes) while maintaining a concrete security level of at least ≈ 100 bits.

To this end, we generated new prime moduli that admit efficient Number Theoretic Transform (NTT) arithmetic and, for a prescribed bit length k , maximize the number of leading ones, thereby accelerating uniform rejection sampling. The NTT/FFT enabled us to carry out all operations in the frequency domain, for both ring and field elements, and motivated the development of specialized algorithms for transposition and inversion in this representation. For additional performance gains, all elements were stored in bit-reversed frequency-domain form, which required corresponding adaptations of the relevant algorithms.

For Gaussian sampling, we extended the inversion-based method [\[Pei10\]](#) by incorporating binary search while preserving constant-time execution. This was achieved by truncating the output space and analyzing the resulting range size accordingly. For key generation and preimage sampling, we relied on the algorithms of [\[Pei10, EBB14, SSTX09\]](#) to generate the appropriate statistical trapdoor of size $\overline{m} \times k$. To generate short preimages with spherical distribution, we calculated the lower triangular matrix B_2 ([\[Pei10\]](#), Algorithm 2, offline phase) via Cholesky factorization at the **Setup** phase of our algorithms and included it in the **msk**. In addition, to accommodate the increased standard deviation of the sampled preimages while maintaining an adequate Ring-LWE security level of at least ≈ 100 bits, as well as satisfying our correctness and security requirements, we conducted a careful analysis of several combinations of system parameters.

6.1 System Parameters

Instantiating the schemes required selecting concrete parameters for the modulus q , lattice dimension n , and Gaussian standard deviation χ , such that multiple correctness and security constraints were satisfied. We fixed the security parameter $\lambda = 128$ and searched for $q, n, \chi, \chi_R, \chi', \chi''$ such that they satisfy the various correctness and security inequalities. For the asymptotic inequalities, we dropped the O, Ω notation and assumed constant factors ≈ 1 (with the exception of the CP-ABE2 scheme and χ ; see next paragraph). For the resulting parameter choices of n, q, χ , we used the **lattice-estimator** software [\[APS15\]](#) to assess the concrete security of the LWE assumption against state-of-the-art attacks. This required high enough dimension $n = 2048$; a parameter large enough that allows us to ignore constant factors of asymptotic bounds.

For KP-ABE, we achieved the required security target with a modulus of size $\lceil \log q \rceil = 57$ bits, which enabled us to set the maximum number of attributes per conjunction to $L = 32$. On the other hand, for CP-ABE, we considered two parameter regimes. The first uses a modulus below 64 bits, obtained by relaxing the constraints on χ of the reduction $\text{RTRLWE} \leftarrow \text{RLWE}$ ([Theorem B.3](#)), yielding practical performance. The second uses a 122-bit modulus together with a significantly larger χ , providing a more conservative security margin and adhering strictly to the conditions of the reduction. For both of the CP-ABE implementations we set $L = 8$ as the maximum number of attributes per conjunction.

The selected parameters and their concrete LWE security from **lattice-estimator** are summarized in [Table 2](#). This is the assumed security of the LWE instantiation based on the fastest state-of-the-art attack. The security of the corresponding Ring-LWE instantiation is expected to be slightly lower (up to ≈ 15 bits for various sets of parameters) [\[Ogi26\]](#).

For the 122-bit modulus, arithmetic was implemented using 128-bit integer types with specialized modular exponentiation routines, which incurs a noticeable performance overhead (along with the dependence of m and $\mathbf{T}_A$ from k). In addition, the standard C++ double-precision implementation of the error function (**erf**) did not provide sufficient precision for the binary search used in the Gaussian inversion method [\[Pei10\]](#). Consequently, we reverted to a native C++ Gaussian implementation (**std::normal_distribution**).

Table 3. Timing results (in milliseconds (ms) and seconds (sec)), and key / ciphertexts sizes (in MB = 10^6 bytes). See Remark 6.1 for —.

# of Attrs.	1	2	3	4
KeyGen				
KP-ABE	443 ms	920 ms	1326 ms	1765 ms
CP-ABE 1	6.3 sec	12.7 sec	19.9 sec	25.6 sec
CP-ABE 2	—	—	—	—
Enc				
KP-ABE	195 ms	366 ms	428 ms	545 ms
CP-ABE 1	331 ms	1110 ms	2434 ms	4267 ms
CP-ABE 2	22 sec	440 sec	1265 sec	2498 sec
Dec				
KP-ABE	5.6 ms	13.7 ms	16.9 ms	22.5 ms
CP-ABE 1	7.5 ms	15.9 ms	25.0 ms	33.5 ms
CP-ABE 2	2.4 sec	5.5 sec	8.6 sec	11.8 sec

# of Attrs.	1	2	1	2
	Key Size		Ciphertext Size	
KP-ABE	4.10 MB	8.20 MB	4.12 MB	6.17 MB
CP-ABE 1	34.45 MB	68.83 MB	6.46 MB	12.90 MB
CP-ABE 2	—	—	50.41 MB	100.76 MB

Remark 6.1 (CP-ABE2 key generation). The lower triangular matrix B_2 for the pre-image generation contains $m(m+1)/2$ elements of the field $\mathcal{K}_{2n} = \mathbb{Q}[x]/(x^n + 1)$. For KP-ABE (resp. CP-ABE1) it is of modest size ≈ 258 MB (resp. ≈ 283 MB). However, for CP-ABE2 the matrices $\mathbf{T}_A$ and B_2 were several GB (≈ 1.07 GB and ≈ 2.16 GB, resp.) and the memory accesses slowed down the **Setup** and **KeyGen** algorithms excessively. Due to this fact, we did not include the generation of proper **msk** or **KeyGen** in our benchmarks, and we measured the performance of **Dec** with mocked elements of the appropriate size. We assume that in typical deployments the key generation algorithms will be carried out by the trusted authority’s powerful infrastructure.

6.2 Environment and Benchmark Configuration

All experiments were conducted on a MacBook Pro running macOS Tahoe (Version 26.4), equipped with an Apple M4 Pro 14-core CPU and 48 GB of RAM. Benchmarks were executed from an administrative shell with elevated priority (**nice** value -10) to reduce scheduling noise.

We measured the execution times of the **KeyGen**, **Enc**, and **Dec** algorithms over attribute sets of size 1 to 4 with encryption message a random 256-bit AES key. Attribute sets were constructed incrementally as $\{\text{Attr0}\}$, $\{\text{Attr0}, \text{Attr1}\}$, $\dots$, $\{\text{Attr0}, \dots, \text{Attr3}\}$, with corresponding access policies defined as conjunctions of all included attributes. Each function was run 10 times with identical inputs, and we report the average execution time. Timing results and keys / ciphertext sizes are shown in Table 3.

6.3 Conclusion

Our experimental evaluation demonstrates that large-universe, post-quantum ABE is feasible for practical deployment. All core algorithms can be executed on commodity user devices within seconds to minutes, without requiring specialized hardware. Among these, **KeyGen** is the most computationally intensive due to the cost of Gaussian and preimage sampling; however, in typical deployment scenarios this operation can be offloaded to the infrastructure of a trusted authority. Prior works on DNF-based ABE from LWE [ABV⁺12, ABN⁺22, DKW21, BGG⁺14, CW23, CW24] primarily focus on integer-LWE constructions, often without concrete implementations and without native support for large universes, which complicates direct empirical comparison. We therefore provide an asymptotic comparison in Table 1.

The closest implementation-oriented work is [DDP⁺18], which realizes a Ring-LWE variant of [BGG⁺14] using the techniques of [GPR⁺18]. That work emphasizes extensive software and hardware optimizations, including GPU acceleration, for a KP-ABE system. Its approach to general policies relies on homomorphic circuit evaluation, as in [BGG⁺14], within a small-universe setting where performance depends on the total number of attributes ℓ . Due to differences in hardware (e.g., NVIDIA Titan X/Xp

GPUs), system design, attribute universe size, and evaluation methodology, direct comparison with our results is not meaningful. They report timings for encryption and homomorphic evaluation (the EvalPK and EvalCT algorithms, used in place of ABE decryption) for selected parameter sets (ℓ, k, n) , omitting preprocessing and memory transfer costs via specialized CUDA streams. The reported runtimes range from a few milliseconds to approximately 40 seconds, indicating that incorporating similar optimization techniques into our implementation is a promising direction for future work.

Another important future direction is an instantiation from the computational trapdoor by [MP12], which will improve the performance of our schemes. Our security proof relies on the fact that the trapdoor matrix is statistically close to uniform, and thus we cannot use the computational trapdoor straightforwardly as in [EBB14, GM18, GPR⁺18].

7 Open Science

We provide the code corresponding to our implementations at <https://anonymous.4open.science/r/ccs26-rlwe-0893/>. Owing to the proprietary nature of the underlying library, the released material has been adapted into pseudocode and focuses on the principal algorithms underlying our constructions. It therefore omits generic, though necessary, components responsible for attribute and policy parsing, satisfiability checking, secret sharing, hybrid encryption, and key serialization/storage.

More specifically, the released material includes the header and/or implementation files for the following components:

- Base types for ring coefficients and low-level arithmetic operations.
- The ring class, its parameters, and associated functionality.
- The ring element class, together with related functions and operators.
- The field element class, the field element spectrum class, and their operators.
- Uniform and Gaussian distributions over ring elements.
- Trapdoor generation and preimage sampling procedures.
- A script for generating suitable prime moduli and other LWE parameters.
- Implementations of the Ring-LWE KP-ABE and CP-ABE schemes.

Acknowledgements

We thank Mehdi Tibouchi from NTT Social Informatics Laboratories, Japan for helpful discussions regarding the implementation of the statistical lattice trapdoors.

References

- ABD⁺21. Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Kyber: Algorithm specifications and supporting documentation, version 3.01, 2021. Accessed: 2026-01-10.
- ABGW17. Miguel Ambrona, Gilles Barthe, Romain Gay, and Hoeteck Wee. Attribute-based encryption in the generic group model: Automated proofs and new constructions. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 647–664. ACM Press, October / November 2017.
- ABN⁺22. Shweta Agrawal, Rajarshi Biswas, Ryo Nishimaki, Keita Xagawa, Xiang Xie, and Shota Yamada. Cryptanalysis of boyen’s attribute-based encryption scheme in TCC 2013. *DCC*, 90(10):2301–2318, 2022.
- ABV⁺12. Shweta Agrawal, Xavier Boyen, Vinod Vaikuntanathan, Panagiotis Voulgaris, and Hoeteck Wee. Functional encryption for threshold functions (or fuzzy ibe) from lattices. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 280–297. Springer, Berlin, Heidelberg, May 2012.
- AC17. Shashank Agrawal and Melissa Chase. FAME: Fast attribute-based message encryption. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 665–682. ACM Press, October / November 2017.
- AFV11. Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 21–40. Springer, Berlin, Heidelberg, December 2011.
- AG21. Miguel Ambrona and Romain Gay. Multi-authority ABE, revisited. Cryptology ePrint Archive, Report 2021/1381, 2021.

- Ajt99. Miklós Ajtai. Generating hard instances of the short basis problem. In Jiri Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *ICALP 99*, volume 1644 of *LNCS*, pages 1–9. Springer, Berlin, Heidelberg, July 1999.
- AKY24. Shweta Agrawal, Simran Kumari, and Shota Yamada. Attribute based encryption for turing machines from lattices. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 352–386. Springer, Cham, August 2024.
- AMR25. Damiano Abram, Giulio Malavolta, and Lawrence Roy. Key-homomorphic computations for RAM: Fully succinct randomised encodings and more. In *CRYPTO 2025, Part III*, *LNCS*, pages 236–268. Springer, Cham, August 2025.
- AMY19. Shweta Agrawal, Monosij Maitra, and Shota Yamada. Attribute based encryption (and more) for nondeterministic finite automata from LWE. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 765–797. Springer, Cham, August 2019.
- AMYY25. Shweta Agrawal, Anuja Modi, Anshu Yadav, and Shota Yamada. Evasive lwe: Attacks, variants and obfustopia. Cryptology ePrint Archive, Paper 2025/375, 2025.
- AP09. Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. In Susanne Albers and Jean-Yves Marion, editors, *STACS 2009*, volume 3 of *LIPIcs*, pages 75–86. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, 2009.
- APS15. Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. Cryptology ePrint Archive, Paper 2015/046, 2015.
- ARYY23. Shweta Agrawal, Mélissa Rossi, Anshu Yadav, and Shota Yamada. Constant input attribute based (and predicate) encryption from evasive and tensor LWE. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part IV*, volume 14084 of *LNCS*, pages 532–564. Springer, Cham, August 2023.
- AT20. Nuttapong Attrapadung and Junichi Tomida. Unbounded dynamic predicate compositions in ABE from standard assumptions. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part III*, volume 12493 of *LNCS*, pages 405–436. Springer, Cham, December 2020.
- Att14. Nuttapong Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 557–577. Springer, Berlin, Heidelberg, May 2014.
- Att16. Nuttapong Attrapadung. Dual system encryption framework in prime-order groups via computational pair encodings. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 591–623. Springer, Berlin, Heidelberg, December 2016.
- Att19. Nuttapong Attrapadung. Unbounded dynamic predicate compositions in attribute-based encryption. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 34–67. Springer, Cham, May 2019.
- AY20a. Shweta Agrawal and Shota Yamada. CP-ABE for circuits (and more) in the symmetric key setting. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part I*, volume 12550 of *LNCS*, pages 117–148. Springer, Cham, November 2020.
- AY20b. Shweta Agrawal and Shota Yamada. Optimal broadcast encryption from pairings and LWE. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 13–43. Springer, Cham, May 2020.
- AYY22. Shweta Agrawal, Anshu Yadav, and Shota Yamada. Multi-input attribute based encryption and predicate encryption. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 590–621. Springer, Cham, August 2022.
- BGG⁺14. Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 533–556. Springer, Berlin, Heidelberg, May 2014.
- Boy13. Xavier Boyen. Attribute-based functional encryption on lattices. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 122–142. Springer, Berlin, Heidelberg, March 2013.
- BSW07. John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy*, pages 321–334. IEEE Computer Society Press, May 2007.
- BÜW24. Chris Brzuska, Akin Ünäl, and Ivy K. Y. Woo. Evasive LWE assumptions: Definitions, classes, and counterexamples. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part IV*, volume 15487 of *LNCS*, pages 418–449. Springer, Singapore, December 2024.
- BV16. Zvika Brakerski and Vinod Vaikuntanathan. Circuit-ABE from LWE: Unbounded attributes and semi-adaptive security. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 363–384. Springer, Berlin, Heidelberg, August 2016.
- BV20. Zvika Brakerski and Vinod Vaikuntanathan. Lattice-inspired broadcast encryption and succinct ciphertext-policy ABE. Cryptology ePrint Archive, Report 2020/191, 2020.
- CCG⁺23. Jie Chen, Qiaohan Chu, Ying Gao, Jianting Ning, and Luping Wang. Improved fully adaptive decentralized MA-ABE for NC1 from MDDH. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part V*, volume 14442 of *LNCS*, pages 3–32. Springer, Singapore, December 2023.

- CGKW18. Jie Chen, Junqing Gong, Lucas Kowalczyk, and Hoeteck Wee. Unbounded ABE via bilinear entropy expansion, revisited. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 503–534. Springer, Cham, April / May 2018.
- CGW15. Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system ABE in prime-order groups via predicate encodings. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 595–624. Springer, Berlin, Heidelberg, April 2015.
- Cha07. Melissa Chase. Multi-authority attribute based encryption. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 515–534. Springer, Berlin, Heidelberg, February 2007.
- CHK09. David Cash, Dennis Hofheinz, and Eike Kiltz. How to delegate a lattice basis. *Cryptology ePrint Archive*, Report 2009/351, 2009.
- CHKP10. David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 523–552. Springer, Berlin, Heidelberg, May / June 2010.
- CLW24. Valerio Cini, Russell W. F. Lai, and Ivy K. Y. Woo. Lattice-based multi-authority/client attribute-based encryption for circuits. *CiC*, 1(4):1, 2024.
- CLW25. Valerio Cini, Russell Lai, and Ivy Woo. Lattice-based multi-authority/client attribute-based encryption for circuits. *IACR Communications in Cryptology*, 1(4):1–67, 2025.
- CN11. Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better lattice security estimates. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 1–20. Springer, Berlin, Heidelberg, December 2011.
- CW23. Valerio Cini and Hoeteck Wee. ABE for circuits with poly(λ)-sized keys from LWE. In *64th FOCS*, pages 435–446. IEEE Computer Society Press, November 2023.
- CW24. Valerio Cini and Hoeteck Wee. Unbounded ABE for circuits from LWE, revisited. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part IV*, volume 15487 of *LNCS*, pages 238–267. Springer, Singapore, December 2024.
- CW25a. Valerio Cini and Hoeteck Wee. Faster abe for turing machines from circular evasive lwe. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 94–125. Springer, 2025.
- CW25b. Valerio Cini and Hoeteck Wee. Faster ABE for turing machines from circular evasive LWE. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part III*, volume 15603 of *LNCS*, pages 94–125. Springer, Cham, May 2025.
- DDP⁺18. Wei Dai, Yarkin Doroz, Yuriy Polyakov, Kurt Rohloff, Hadi Sajjadpour, Erkey Savas, and Berk Sunar. Implementation and evaluation of a lattice-based key-policy abe scheme. *Trans. Info. For. Sec.*, 13(5):1169–1184, May 2018.
- DHM⁺24. Fangqi Dong, Zihan Hao, Ethan Mook, Hoeteck Wee, and Daniel Wichs. Laconic function evaluation and ABE for RAMs from (ring-)LWE. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 107–142. Springer, Cham, August 2024.
- DJM⁺25. Nico Döttling, Abhishek Jain, Giulio Malavolta, Surya Mathialagan, and Vinod Vaikuntanathan. Simple and general counterexamples for private-coin evasive LWE. In *CRYPTO 2025, Part VII*, *LNCS*, pages 73–92. Springer, Cham, August 2025.
- DKW21. Pratish Datta, Ilan Komargodski, and Brent Waters. Decentralized multi-authority ABE for DNFs from LWE. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 177–209. Springer, Cham, October 2021.
- DKW23a. Pratish Datta, Ilan Komargodski, and Brent Waters. Decentralized multi-authority ABE for NC¹ from BDH. *Journal of Cryptology*, 36(2):6, April 2023.
- DKW23b. Pratish Datta, Ilan Komargodski, and Brent Waters. Fully adaptive decentralized multi-authority ABE. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 447–478. Springer, Cham, April 2023.
- dIPVA23. Antonio de la Piedra, Marloes Venema, and Greg Alpar. ACABELLA: Automated (crypt)analysis of attribute-based encryption leveraging linear algebra. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *ACM CCS 2023*, pages 3269–3283. ACM Press, November 2023.
- DRS04. Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *International conference on the theory and applications of cryptographic techniques*, pages 523–540. Springer, 2004.
- DTV25. Pratish Datta, Junichi Tomida, and Nikhil Vanjani. Fully adaptive decentralized MA-ABE: Simplified, optimized, ASP supported. *Cryptology ePrint Archive*, Paper 2025/1628, 2025.
- EBB14. Rachid El Bansarkhani and Johannes Buchmann. Improvement and efficient implementation of a lattice-based signature scheme. In Tanja Lange, Kristin Lauter, and Petr Lisoněk, editors, *Selected Areas in Cryptography – SAC 2013*, pages 48–67, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
- GGH⁺13. Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and Brent Waters. Attribute-based encryption for circuits from multilinear maps. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 479–499. Springer, Berlin, Heidelberg, August 2013.

- GGL24. Rachit Garg, Rishab Goyal, and George Lu. Dynamic collusion functional encryption and multi-authority attribute-based encryption. In Qiang Tang and Vanessa Teague, editors, *PKC 2024, Part II*, volume 14604 of *LNCS*, pages 69–104. Springer, Cham, April 2024.
- GKW17. Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In Chris Umans, editor, *58th FOCS*, pages 612–621. IEEE Computer Society Press, October 2017.
- GM18. Nicholas Genise and Daniele Micciancio. Faster Gaussian sampling for trapdoor lattices with arbitrary modulus. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 174–203. Springer, Cham, April / May 2018.
- GPR⁺18. Kamil Doruk Gür, Yuriy Polyakov, Kurt Rohloff, Gerard W Ryan, and Erkey Savas. Implementation and evaluation of improved gaussian sampling for lattice trapdoors. In *Proceedings of the 6th workshop on encrypted computing & applied homomorphic cryptography*, pages 61–71, 2018.
- GPSW06. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309.
- GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.
- GVW13. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 545–554. ACM Press, June 2013.
- GW20. Junqing Gong and Hoeteck Wee. Adaptively secure ABE for DFA from k -Lin and more. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 278–308. Springer, Cham, May 2020.
- GWV19. Junqing Gong, Brent Waters, and Hoeteck Wee. ABE for DFA from k -Lin. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 732–764. Springer, Cham, August 2019.
- HHY25. Tzu-Hsiang Huang, Wei-Hsiang Hung, and Shota Yamada. A note on obfuscation-based attacks on private-coin evasive LWE. Cryptology ePrint Archive, Paper 2025/421, 2025.
- HJL25. Yao-Ching Hsieh, Aayush Jain, and Huijia Lin. Lattice-based post-quantum iO from circular security with random opening assumption. In *CRYPTO 2025, Part VII*, *LNCS*, pages 3–38. Springer, Cham, August 2025.
- HLL23. Yao-Ching Hsieh, Huijia Lin, and Ji Luo. Attribute-based encryption for circuits of unbounded depth from lattices. In *64th FOCS*, pages 415–434. IEEE Computer Society Press, November 2023.
- HLL24. Yao-Ching Hsieh, Huijia Lin, and Ji Luo. A general framework for lattice-based ABE using evasive inner-product functional encryption. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 433–464. Springer, Cham, May 2024.
- JLL23. Aayush Jain, Huijia Lin, and Ji Luo. On the optimal succinctness and efficiency of functional encryption and attribute-based encryption. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 479–510. Springer, Cham, April 2023.
- JLWG25. Haoxiang Jin, Feng-Hao Liu, Zhedong Wang, and Dawu Gu. Discrete gaussians modulo sub-lattices: New leftover hash lemmas for discrete gaussians. In *IACR International Conference on Public-Key Cryptography*, pages 301–330. Springer, 2025.
- KSW08. Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 146–162. Springer, Berlin, Heidelberg, April 2008.
- KW19. Lucas Kowalczyk and Hoeteck Wee. Compact adaptively secure ABE for NC^1 from k -Lin. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 3–33. Springer, Cham, May 2019.
- KW20. Lucas Kowalczyk and Hoeteck Wee. Compact adaptively secure ABE for NC^1 from k -Lin. *Journal of Cryptology*, 33(3):954–1002, July 2020.
- LL20a. Huijia Lin and Ji Luo. Compact adaptively secure ABE from k -Lin: Beyond NC^1 and towards NL. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 247–277. Springer, Cham, May 2020.
- LL20b. Huijia Lin and Ji Luo. Succinct and adaptively secure ABE for ABP from k -Lin. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part III*, volume 12493 of *LNCS*, pages 437–466. Springer, Cham, December 2020.
- LLL22. Hanjun Li, Huijia Lin, and Ji Luo. ABE for circuits with constant-size secret keys and adaptive security. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 680–710. Springer, Cham, November 2022.
- LOS⁺10. Allison B. Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 62–91. Springer, Berlin, Heidelberg, May / June 2010.

- LPR10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Berlin, Heidelberg, May / June 2010.
- LPR13. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-LWE cryptography. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 35–54. Springer, Berlin, Heidelberg, May 2013.
- LS15. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.*, 75(3):565–599, 2015.
- LVV⁺23. Watson Ladd, Tanya Verma, Marloes Venema, Armando Faz-Hernández, Brendan McMillion, Avani Wildani, and Nick Sullivan. Portunus: Re-imagining access control in distributed systems. In *2023 USENIX Annual Technical Conference (USENIX ATC 23)*, pages 35–52, 2023.
- LW10. Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 455–479. Springer, Berlin, Heidelberg, February 2010.
- LW11a. Allison B. Lewko and Brent Waters. Decentralizing attribute-based encryption. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 568–588. Springer, Berlin, Heidelberg, May 2011.
- LW11b. Allison B. Lewko and Brent Waters. Unbounded HIBE and attribute-based encryption. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 547–567. Springer, Berlin, Heidelberg, May 2011.
- LW12. Allison B. Lewko and Brent Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 180–198. Springer, Berlin, Heidelberg, August 2012.
- MP12. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Berlin, Heidelberg, April 2012.
- Ogi26. Tabitha Ogilvie. On the concrete hardness gap between MLWE and LWE. Cryptology ePrint Archive, Paper 2026/279, 2026.
- OSW07. Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based encryption with non-monotonic access structures. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM CCS 2007*, pages 195–203. ACM Press, October 2007.
- OT10. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 191–208. Springer, Berlin, Heidelberg, August 2010.
- OT12. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure unbounded inner-product and attribute-based encryption. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 349–366. Springer, Berlin, Heidelberg, December 2012.
- OT13. Tatsuaki Okamoto and Katsuyuki Takashima. Decentralized attribute-based signatures. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 125–142. Springer, Berlin, Heidelberg, February / March 2013.
- OT20. Tatsuaki Okamoto and Katsuyuki Takashima. Decentralized attribute-based encryption and signatures. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 103(1):41–73, 2020.
- Pei10. Chris Peikert. An efficient and parallel gaussian sampler for lattices. In *Proceedings of the 30th Annual Conference on Advances in Cryptology*, CRYPTO’10, page 80–97, Berlin, Heidelberg, 2010. Springer-Verlag.
- Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.
- RVV24. Doreen Riepel, Marloes Venema, and Tanya Verma. ISABELLA: Improving structures of attribute-based encryption leveraging linear algebra. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024*, pages 4628–4642. ACM Press, October 2024.
- RW15. Yannis Rouselakis and Brent Waters. Efficient statically-secure large-universe multi-authority attribute-based encryption. In Rainer Böhme and Tatsuaki Okamoto, editors, *FC 2015*, volume 8975 of *LNCS*, pages 315–332. Springer, Berlin, Heidelberg, January 2015.
- RW22. Doreen Riepel and Hoeteck Wee. FABEO: Fast attribute-based encryption with optimal security. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 2491–2504. ACM Press, November 2022.
- SBKL⁺20. Léo Lucas Shi Bai, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS–dilithium: Algorithm specification and supporting documentation (version 3.1). Round-3 submission to the NIST Post-Quantum Cryptography Standardization Project, 2020. <https://cryptojedi.org/papers/#dilithiumnistr3>.
- SSTX09. Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 617–635. Springer, Berlin, Heidelberg, December 2009.

- SW05. Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Berlin, Heidelberg, May 2005.
- Tha22. Justin Thaler. Proofs, arguments, and zero-knowledge. *Found. Trends Priv. Secur.*, 4(2-4):117–660, 2022.
- TKN20. Junichi Tomida, Yuto Kawahara, and Ryo Nishimaki. Fast, compact, and expressive attribute-based encryption. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 3–33. Springer, Cham, May 2020.
- Tsa19. Rotem Tsabary. Fully secure attribute-based encryption for t-CNF from LWE. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 62–85. Springer, Cham, August 2019.
- Ven23. Marloes Venema. A practical compiler for attribute-based encryption: New decentralized constructions and more. In Mike Rosulek, editor, *CT-RSA 2023*, volume 13871 of *LNCS*, pages 132–159. Springer, Cham, April 2023.
- Wat09. Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, Berlin, Heidelberg, August 2009.
- Wat11. Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 53–70. Springer, Berlin, Heidelberg, March 2011.
- Wat12. Brent Waters. Functional encryption for regular languages. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 218–235. Springer, Berlin, Heidelberg, August 2012.
- Wee14. Hoeteck Wee. Dual system encryption via predicate encodings. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 616–637. Springer, Berlin, Heidelberg, February 2014.
- Wee21a. Hoeteck Wee. ABE for DFA from LWE against bounded collusions, revisited. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part II*, volume 13043 of *LNCS*, pages 288–309. Springer, Cham, November 2021.
- Wee21b. Hoeteck Wee. Broadcast encryption with size $N^{1/3}$ and more from k -lin. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 155–178, Virtual Event, August 2021. Springer, Cham.
- Wee22. Hoeteck Wee. Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 217–241. Springer, Cham, May / June 2022.
- Wee24. Hoeteck Wee. Circuit ABE with $\text{poly}(\text{depth}, \lambda)$ -sized ciphertexts and keys from lattices. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 178–209. Springer, Cham, August 2024.
- Wee25. Hoeteck Wee. Almost optimal KP and CP-ABE for circuits from succinct LWE. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part III*, volume 15603 of *LNCS*, pages 34–62. Springer, Cham, May 2025.
- WW24. Brent Waters and Daniel Wichs. Adaptively secure attribute-based encryption from witness encryption. In Elette Boyle and Mohammad Mahmoody, editors, *TCC 2024, Part III*, volume 15366 of *LNCS*, pages 65–90. Springer, Cham, December 2024.
- WWW22. Brent Waters, Hoeteck Wee, and David J. Wu. Multi-authority ABE from lattices without random oracles. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 651–679. Springer, Cham, November 2022.
- Xag13. Keita Xagawa. Improved (hierarchical) inner-product encryption from lattices. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 235–252. Springer, Berlin, Heidelberg, February / March 2013.

A Our CP-ABE and MA-ABE with Shorter Secret Keys

In this section, we present variants of our CP-ABE (Section 3.2) and MA-ABE (Section 4) schemes with shorter secret keys and larger ciphertexts.

A.1 CP-ABE

Construction. Let $n, m, q \in \mathbb{N}$ and $\chi \in \mathbb{R}^+$ be lattice parameters that depend on the security parameter λ , where $q > 2$ is a prime. Let the message space $\mathcal{M} = \{0, 1\}$. Let L be the maximum size of the set used in Enc. Let $H : \{0, 1\}^* \rightarrow \mathbb{Z}_q^{n \times (2mL+1)}$ be a hash function.

Setup(1^λ): It outputs mpk and msk as follows:

$$(\text{mpk}, \text{msk}) = (\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \text{TrapGen}(1^n, m, q)$$

$\text{Enc}(\text{mpk}, X, \mu)$: It parses $X = \{x_1, \dots, x_t\}$, where it must hold that $t < L$, and outputs ct as follows.

$$\begin{aligned} \mathbf{s} &\leftarrow \mathbb{Z}_q^n, \mathbf{e}_1 \leftarrow D_{\mathbb{Z}, \chi}^{(t+1)m}, \mathbf{e}_2 \leftarrow D_{\mathbb{Z}, \chi}^{mL}, \mathbf{e}_3 \leftarrow D_{\mathbb{Z}, \chi}, \mathbf{R} \leftarrow \{-1, 1\}^{mL \times m(L-1)} \\ \underbrace{[\mathbf{D}_i]}_m \mid \underbrace{[\mathbf{B}_i]}_{m(2L-1)} \mid \underbrace{[\mathbf{p}_i]}_1 &= H(x_i), \mathbf{D} = [\mathbf{D}_1 \mid \dots \mid \mathbf{D}_t] \\ \text{ct} &= (\mathbf{c}_1^\top, \mathbf{c}_2^\top, c_3) \\ &= \left(\mathbf{s}^\top [\mathbf{A} \mid \mathbf{D}] + \mathbf{e}_1^\top, \mathbf{s}^\top \sum_{i \in [t]} \mathbf{B}_i + \mathbf{e}_2^\top [\mathbf{I}_{mL} \mid \mathbf{R}], \mathbf{s}^\top \sum_{i \in [t]} \mathbf{p}_i + e_3 + \mu \lfloor q/2 \rfloor \right) \end{aligned}$$

$\text{KeyGen}(\text{mpk}, \text{msk}, Y)$: It parses $Y = \{y_1, \dots, y_u\}$. It computes $\mathbf{r} \leftarrow D_{\mathbb{Z}, \chi}^{m(2L-1)}$ and, for $i \in [u]$,

$$\underbrace{[\hat{\mathbf{D}}_i]}_m \mid \underbrace{[\hat{\mathbf{B}}_i]}_{m(2L-1)} \mid \underbrace{[\hat{\mathbf{p}}_i]}_1 = H(y_i), \mathbf{k}_i \leftarrow \text{SampleLeft}([\mathbf{A} \mid \hat{\mathbf{D}}_i], \mathbf{T}_\mathbf{A}, \hat{\mathbf{B}}_i \mathbf{r} + \hat{\mathbf{p}}_i, \chi)$$

Then, it outputs $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u, \mathbf{r})$.

$\text{Dec}(\text{mpk}, \text{ct}, \text{sk}, X, Y)$: It parses $\text{ct} = (\mathbf{c}_1^\top, \mathbf{c}_2^\top, c_3)$ where $\mathbf{c}_1^\top = [\mathbf{c}_{1,0}^\top \mid \dots \mid \mathbf{c}_{1,t}^\top] \in (\mathbb{Z}_q^m)^{t+1}$, $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u, \mathbf{r})$, $X = \{x_1, \dots, x_t\}$, and $Y = \{y_1, \dots, y_u\}$. If $X \subseteq Y$, it computes $\rho: [t] \rightarrow [u]$ such that $x_i = y_{\rho(i)}$ and outputs

$$\left\lfloor \frac{2}{q} \cdot \left(c_3 - \left(\sum_{i \in [t]} [\mathbf{c}_{1,0}^\top \mid \mathbf{c}_{1,i}^\top] \mathbf{k}_{\rho(i)} - \mathbf{c}_2^\top \mathbf{r} \right) \right) \right\rfloor$$

Otherwise, output $\perp$.

Parameter Selection. From [Theorem A.1](#) and [Theorem A.2](#), we can choose parameters as follows:

$$n = \lambda, \quad q = O(n^8 L^7), \quad m = O(n \log q), \quad \chi = O(m^2 L^2 \log n)$$

Observe that we can choose q, χ as polynomials in λ . Hence, our scheme is secure under the LWE with a polynomial modulus-to-noise ratio.

Correctness. For correctness, we have the following theorem.

Theorem A.1. Suppose $q \geq 5(\sqrt{\lambda}\chi + mL(2 + 2mL - m)\lambda\chi^2)$, $m = O(n \log q)$, and $\chi = \omega(\sqrt{\log m})$. Then, our scheme is correct.

Proof. In decryption, suppose $\text{ct} = (\mathbf{c}_1^\top, \mathbf{c}_2^\top, c_3)$ where $\mathbf{c}_1^\top = [\mathbf{c}_{1,0}^\top \mid \dots \mid \mathbf{c}_{1,t}^\top]$, $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u, \mathbf{r})$ are a ciphertext and a secret key for X and Y , respectively, such that $X \subseteq Y$. In decryption, we have

$$\begin{aligned} &c_3 - \left(\sum_{i \in [t]} [\mathbf{c}_{1,0}^\top \mid \mathbf{c}_{1,i}^\top] \mathbf{k}_{\rho(i)} - \mathbf{c}_2^\top \mathbf{r} \right) \\ &= \mathbf{s}^\top \sum_{i \in [t]} \mathbf{p}_i + e_3 + \mu \lfloor q/2 \rfloor - \sum_{i \in [t]} (\mathbf{s}^\top [\mathbf{A} \mid \mathbf{D}_i] + [\mathbf{e}_{1,0}^\top \mid \mathbf{e}_{1,i}^\top]) \mathbf{k}_{\rho(i)} + \left(\mathbf{s}^\top \sum_{i \in [t]} \mathbf{B}_i + \mathbf{e}_2^\top [\mathbf{I}_{mL} \mid \mathbf{R}] \right) \mathbf{r} \\ &= \mathbf{s}^\top \sum_{i \in [t]} \mathbf{p}_i + e_3 + \mu \lfloor q/2 \rfloor - \sum_{i \in [t]} (\mathbf{s}^\top (\mathbf{B}_i \mathbf{r} + \mathbf{p}_i) + [\mathbf{e}_{1,0}^\top \mid \mathbf{e}_{1,i}^\top] \mathbf{k}_{\rho(i)}) + \left(\mathbf{s}^\top \sum_{i \in [t]} \mathbf{B}_i + \mathbf{e}_2^\top [\mathbf{I}_{mL} \mid \mathbf{R}] \right) \mathbf{r} \\ &= \mu \lfloor q/2 \rfloor + e_3 - \underbrace{\sum_{i \in [t]} [\mathbf{e}_{1,0}^\top \mid \mathbf{e}_{1,i}^\top] \mathbf{k}_{\rho(i)} + \mathbf{e}_2^\top [\mathbf{I}_{mL} \mid \mathbf{R}] \mathbf{r}}_{\tilde{e}} \end{aligned}$$

where $[\mathbf{D}_i \mid \mathbf{B}_i \mid \mathbf{p}_i] = H(x_i)$, $\mathbf{e}_1^\top = [\mathbf{e}_{1,0}^\top \mid \dots \mid \mathbf{e}_{1,t}^\top]$. From the parameter setting and [Lemma 2.2](#), we have $|e_3| \leq \sqrt{\lambda}\chi$, $\|[\mathbf{e}_{1,0}^\top \mid \mathbf{e}_{1,i}^\top]\| \leq \sqrt{\lambda}\chi$, $\|\mathbf{e}_2\| \leq \sqrt{\lambda}\chi$, and $\|\mathbf{r}\| \leq \sqrt{\lambda}\chi$ with overwhelming probability. Thanks to the property of lattice trapdoors, the distribution of $\mathbf{k}_{\rho(i)}$ is statistically close to $[\mathbf{A} \mid \mathbf{D}_i]_\chi^{-1} (\mathbf{B}_i \mathbf{r} + \mathbf{p}_i)$. Since the marginal distribution of $\mathbf{p}_i$ is uniform in $\mathbb{Z}_q^n$, the marginal distribution of $\mathbf{k}_{\rho(i)}$ is statistically

close to $D_{\mathbb{Z},\chi}^{2m}$ from [Lemma 2.3](#). Thus, we have $\|\mathbf{k}_{\rho(i)}\| \leq \sqrt{\lambda}\chi$ with overwhelming probability again from [Lemma 2.2](#). In summary, we have, with overwhelming probability,

$$\begin{aligned} |\tilde{e}| &= \left| e_3 - \sum_{i \in [t]} [\mathbf{e}_{1,0}^\top | \mathbf{e}_{1,i}^\top] \mathbf{k}_{\rho(i)} + \mathbf{e}_2^\top [\mathbf{I}_{mL} | \mathbf{R}] \mathbf{r} \right| \leq \sqrt{\lambda}\chi + 2mt\lambda\chi^2 + m^2L(2L-1)\lambda\chi^2 \\ &\leq \sqrt{\lambda}\chi + mL(2+2mL-m)\lambda\chi^2 \end{aligned}$$

Thus, $|\tilde{e}| \leq q/5$ with overwhelming probability and correctness holds. $\square$

Security Analysis. For security, we have the following theorem.

Theorem A.2. Suppose $m = O(n \log q)$, $\chi = 2m^2L^2 \cdot \omega(\sqrt{\log n})$, the hash function H is modeled as a random oracle, t is the size of the attribute X for the challenge query, and $\text{LWE}_{n,m(t+L+1)+1,q,\chi}$ holds. Then, our scheme satisfies selective security as per [Definition 2.1](#).

Proof. We prove the theorem by a series of hybrids described as follows.

- H_0 : This is the original game. In this game, $\mathcal{A}$ first outputs the challenge attribute $X = (x_1, \dots, x_t)$. Then, the challenger computes

$$(\text{mpk}, \text{msk}) = (\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \text{TrapGen}(1^n, m, q)$$

and gives mpk to $\mathcal{A}$. When $\mathcal{A}$ asks for a secret key for $Y = (y_1, \dots, y_u)$, the challenger computes $\mathbf{r} \leftarrow D_{\mathbb{Z},\chi}^{m(2L-1)}$ and for $i \in [u]$,

$$\left[\underbrace{\widehat{\mathbf{D}}_i}_m \mid \underbrace{\widehat{\mathbf{B}}_i}_{m(2L-1)} \mid \underbrace{\widehat{\mathbf{p}}_i}_1 \right] = H(y_i), \quad \mathbf{k}_i \leftarrow \text{SampleLeft}([\mathbf{A} | \widehat{\mathbf{D}}_i], \mathbf{T}_\mathbf{A}, \widehat{\mathbf{B}}_i \mathbf{r} + \widehat{\mathbf{p}}_i, \chi)$$

and gives $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u, \mathbf{r})$ to $\mathcal{A}$. For the challenge, $\mathcal{A}$ outputs μ_0, μ_1 . Then, the challenger chooses a challenge bit $\beta \leftarrow \{0, 1\}$ and gives ct to $\mathcal{A}$ where

$$\begin{aligned} \mathbf{s} &\leftarrow \mathbb{Z}_q^n, \mathbf{e}_1 \leftarrow D_{\mathbb{Z},\chi}^{(t+1)m}, \mathbf{e}_2 \leftarrow D_{\mathbb{Z},\chi}^{mL}, e_3 \leftarrow D_{\mathbb{Z},\chi}, \mathbf{R} \leftarrow \{-1, 1\}^{mL \times m(L-1)} \\ \left[\underbrace{\mathbf{D}_i}_m \mid \underbrace{\mathbf{B}_i}_{m(2L-1)} \mid \underbrace{\mathbf{p}_i}_1 \right] &= H(x_i), \mathbf{D} = [\mathbf{D}_1 \mid \dots \mid \mathbf{D}_t] \\ \text{ct} &= (\mathbf{c}_1^\top, \mathbf{c}_2^\top, c_3) \\ &= \left(\mathbf{s}^\top [\mathbf{A} | \mathbf{D}] + \mathbf{e}_1^\top, \mathbf{s}^\top \sum_{i \in [t]} \mathbf{B}_i + \mathbf{e}_2^\top [\mathbf{I}_{mL} | \mathbf{R}], \mathbf{s}^\top \sum_{i \in [t]} \mathbf{p}_i + e_3 + \mu_\beta \lfloor q/2 \rfloor \right) \end{aligned}$$

The post-challenge key queries are the same as the pre-challenge queries. Finally, $\mathcal{A}$ outputs β' . During the game, the challenger simulates the random oracle as follows. It maintains a dictionary $\mathcal{D}$ that is initially empty. When H is invoked on input x , the challenger checks if $\mathcal{D}(x) = \perp$. If this is the case, the challenger samples $[\mathbf{D} | \mathbf{B} | \mathbf{p}] \leftarrow \mathbb{Z}_q^{n \times (2mL+1)}$, updates $\mathcal{D}(x) = [\mathbf{D} | \mathbf{B} | \mathbf{p}]$, and returns $[\mathbf{D} | \mathbf{B} | \mathbf{p}]$. Otherwise, it returns $\mathcal{D}(x)$.

- H_1 : This is the same as H_0 except that the challenger simulates the random oracle as follows. Right after $\mathcal{A}$ outputs the challenge attribute $X = (x_1, \dots, x_t)$, the challenger computes $[\mathbf{D}_i | \mathbf{B}_i | \mathbf{p}_i] \leftarrow \mathbb{Z}_q^{n \times (2mL+1)}$ and updates $\mathcal{D}(x_i) = [\mathbf{D}_i | \mathbf{B}_i | \mathbf{p}_i]$ for all $i \in [t]$. During the game, when H is invoked on input x , the challenger checks if $\mathcal{D}(x) = \perp$. If this is the case, the challenger samples $(\mathbf{D}, \mathbf{T}_\mathbf{D}) \leftarrow \text{TrapGen}(1^n, m, q)$, $[\mathbf{B} | \mathbf{p}] \leftarrow \mathbb{Z}_q^{n \times ((2L-1)m+1)}$, updates $\mathcal{D}(x) = [\mathbf{D} | \mathbf{B} | \mathbf{p}]$, and returns $\mathcal{D}(x)$. Otherwise, it returns $\mathcal{D}(x)$. Since the property of lattice trapdoors ([Lemma 2.1](#)), $\mathbf{D}$ is statistically close to uniform, and thus H_0 and H_1 are statistically indistinguishable given $m = O(n \log q)$.
- H_2 : This is the same as H_1 except the followings. In this hybrid, the challenger samples $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ instead of using TrapGen for mpk . For all secret key queries on $Y = (y_1, \dots, y_u)$, let $Y_X = Y \cap X$ and $Y_{\bar{X}} = Y \setminus X$. The challenger computes

$$\begin{aligned} \mathbf{r} &\leftarrow D_{\mathbb{Z},\chi}^{m(2L-1)}, \quad \left[\underbrace{\widehat{\mathbf{D}}_i}_m \mid \underbrace{\widehat{\mathbf{B}}_i}_{m(2L-1)} \mid \underbrace{\widehat{\mathbf{p}}_i}_1 \right] = H(y_i) \\ \mathbf{k}_i &\leftarrow \begin{cases} [\mathbf{A} | \widehat{\mathbf{D}}_i]_\chi^{-1} (\widehat{\mathbf{B}}_i \mathbf{r} + \widehat{\mathbf{p}}_i) & (i : y_i \in Y_X) \\ \text{SampleLeft}([\mathbf{A} | \widehat{\mathbf{D}}_i], \mathbf{T}_{\widehat{\mathbf{D}}_i}, \widehat{\mathbf{B}}_i \mathbf{r} + \widehat{\mathbf{p}}_i, \chi) & (i : y_i \in Y_{\bar{X}}) \end{cases} \end{aligned}$$

and gives $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u, \mathbf{r})$ to $\mathcal{A}$. Note that the sampling of $\mathbf{k}_i$ for $i : y_i \in Y_X$ is inefficient. It is easy to see that $H_1 \approx_s H_2$ since $\mathbf{A}$ is statistically close to uniform in H_1 and, for all $i \in [u]$,

$$\begin{aligned} \text{SampleLeft}([\mathbf{A}|\widehat{\mathbf{D}}_i], \mathbf{T}_{\widehat{\mathbf{D}}_i}, \widehat{\mathbf{B}}_i \mathbf{r} + \widehat{\mathbf{p}}_i, \chi) &\approx_s \text{SampleLeft}([\mathbf{A}|\widehat{\mathbf{D}}_i], \mathbf{T}_{\mathbf{A}}, \widehat{\mathbf{B}}_i \mathbf{r} + \widehat{\mathbf{p}}_i, \chi) \\ &\approx_s [\mathbf{A}|\widehat{\mathbf{D}}_i]_{\chi}^{-1} (\widehat{\mathbf{B}}_i \mathbf{r} + \widehat{\mathbf{p}}_i) \end{aligned}$$

which follows from [Lemma 2.1](#) if $m = O(n \log q)$.

- H_3 : This game is the same as H_2 except the followings. For all secret key queries on $Y = (y_1, \dots, y_u)$, let $Y_X = Y \cap X$, $Y_{\bar{X}} = Y \setminus X$, and $Y_X = (y_{i_1}, \dots, y_{i_k})$. The challenger computes

$$\begin{aligned} \underbrace{[\widehat{\mathbf{D}}_i]}_m \mid \underbrace{[\widehat{\mathbf{B}}_i]}_{m(2L-1)} \mid \underbrace{[\widehat{\mathbf{p}}_i]}_1 &= H(y_i) \quad \text{for } i \in [u], \quad \mathbf{k}_i \leftarrow D_{\mathbb{Z}, \chi}^{2m} \quad \text{for } i : y_i \in Y_X \\ \mathbf{r} &\leftarrow \begin{bmatrix} \widehat{\mathbf{B}}_{i_1} \\ \vdots \\ \widehat{\mathbf{B}}_{i_k} \end{bmatrix}_{\chi}^{-1} \left(\begin{bmatrix} [\mathbf{A}|\widehat{\mathbf{D}}_{i_1}] \mathbf{k}_{i_1} - \widehat{\mathbf{p}}_{i_1} \\ \vdots \\ [\mathbf{A}|\widehat{\mathbf{D}}_{i_k}] \mathbf{k}_{i_k} - \widehat{\mathbf{p}}_{i_k} \end{bmatrix} \right) \\ \mathbf{k}_i &\leftarrow \text{SampleLeft}([\mathbf{A}|\widehat{\mathbf{D}}_i], \mathbf{T}_{\widehat{\mathbf{D}}_i}, \widehat{\mathbf{B}}_i \mathbf{r} + \widehat{\mathbf{p}}_i, \chi) \quad \text{for } i : y_i \in Y_{\bar{X}} \end{aligned} \tag{11}$$

and gives $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u, \mathbf{r})$ to $\mathcal{A}$. We prove $H_2 \approx_s H_3$ in [Lemma A.1](#).

- H_4 : This game is the same as H_3 except the followings. The challenger generates the challenge ciphertext as

$$\text{ct} = (\mathbf{c}_1^{\top}, \mathbf{c}_2^{\top}, c_3) \leftarrow \mathbb{Z}_q^{(t+1)m} \times \mathbb{Z}_q^{m(2L-1)} \times \mathbb{Z}_q$$

We prove $H_3 \approx_c H_4$ in [Lemma A.2](#).

It is obvious that $\mathcal{A}$ does not obtain the information on β , and thus its advantage is 0 in the last hybrid. This concludes the proof. $\square$

Lemma A.1. $H_2 \approx_s H_3$ if $m = O(n \log q)$ and $\chi = \omega(\sqrt{\log m})$.

Proof. Let Q be the number of secret key queries, $Y^{(j)} = (y_1^{(j)}, \dots, y_{u^{(j)}}^{(j)})$ be the attribute for the j -th secret key query, let $Y_X^{(j)} = Y^{(j)} \cap X$, $|Y_X^{(j)}| = k^{(j)}$, and $Y_X^{(j)} = (y_{i_1}^{(j)}, \dots, y_{i_k}^{(j)})$. For $y_{i_1}^{(j)}$ and $y_{i_k}^{(j)}$ in the previous sentence, we omit the index (j) for conciseness in some parts, e.g, i_k should be $i_{k^{(j)}}$ precisely. We sometimes omit (j) similarly in what follows.

Let $\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_t \end{bmatrix}$ and $\mathbf{p} = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_t \end{bmatrix}$ where $[\mathbf{D}_i|\mathbf{B}_i|\mathbf{p}_i] = H(x_i)$. Thanks to [Lemma 2.3](#), $\mathbf{B}^{(1)}, \dots, \mathbf{B}^{(Q)}$ where $\mathbf{r}^{(j)} \leftarrow D_{\mathbb{Z}, \chi}^{m(2L-1)}$ for $j \in [Q]$ are distributed statistically close to uniform in $(\mathbb{Z}_q^{nt})^Q$, because $t \leq L$, $m = O(n \log q)$, and $\chi = \omega(\sqrt{\log m})$. Hence, the joint distribution of $\widehat{\mathbf{B}}^{(j)} \mathbf{r}^{(j)} + \widehat{\mathbf{p}}^{(j)}$ for all $j \in [Q]$ are also statistically close to uniform random, where $[\widehat{\mathbf{D}}_i^{(j)}|\widehat{\mathbf{B}}_i^{(j)}|\widehat{\mathbf{p}}_i^{(j)}] = H(y_i^{(j)})$ for $i \in \{i_1, \dots, i_k\}$ and

$$\widehat{\mathbf{B}}^{(j)} = \begin{bmatrix} \widehat{\mathbf{B}}_{i_1}^{(j)} \\ \vdots \\ \widehat{\mathbf{B}}_{i_k}^{(j)} \end{bmatrix}, \quad \widehat{\mathbf{p}}^{(j)} = \begin{bmatrix} \widehat{\mathbf{p}}_{i_1}^{(j)} \\ \vdots \\ \widehat{\mathbf{p}}_{i_k}^{(j)} \end{bmatrix}$$

By definition, $\widehat{\mathbf{B}}_i^{(j)} \in \{\mathbf{B}_1, \dots, \mathbf{B}_t\}$ and $\widehat{\mathbf{p}}_i^{(j)} \in \{\mathbf{p}_1, \dots, \mathbf{p}_t\}$ for all $i : y_i^{(j)} \in Y_X^{(j)}$, $j \in [Q]$. Recall also that $\mathbf{A}$ is uniformly distributed in $\mathbb{Z}_q^{n \times m}$. Thanks to [Lemma 3.1](#) for $\ell_1 = |X|$ and $\ell_2 = Q$, the joint distribution of $(\mathbf{A}, \mathbf{k}_i^{(j)}, \widehat{\mathbf{B}}^{(j)} \mathbf{r}^{(j)} + \widehat{\mathbf{p}}^{(j)})$ for $i : y_i^{(j)} \in Y_X^{(j)}$ and $j \in [Q]$ in H_2 is statistically close to the distribution of $(\mathbf{A}, \mathbf{k}_i^{(j)}, \widehat{\mathbf{B}}^{(j)} \mathbf{r}^{(j)} + \widehat{\mathbf{p}}^{(j)})$ where we first sample $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{k}_i^{(j)} \leftarrow D_{\mathbb{Z}, \chi}^{2m}$ and then sample

$$\mathbf{B} \leftarrow \mathbb{Z}_q^{nt \times m(2L-1)}, \mathbf{r}^{(j)} \leftarrow D_{\mathbb{Z}, \chi}^{m(2L-1)}, \mathbf{p} \leftarrow \mathbb{Z}_q^{nt}$$

conditioned on $[\mathbf{A}|\widehat{\mathbf{D}}_i^{(j)}]\mathbf{k}_i^{(j)} = \widehat{\mathbf{B}}_i^{(j)} \mathbf{r}^{(j)} + \widehat{\mathbf{p}}_i^{(j)}$ for $i : y_i \in Y_X^{(j)}$ and $j \in [Q]$. This is conceptually the same as sampling $\mathbf{B} \leftarrow \mathbb{Z}_q^{nt \times m(2L-1)}$, $\mathbf{p} \leftarrow \mathbb{Z}_q^{nt}$, and then sampling $\mathbf{r}^{(j)}$ for each secret key query as [Eq. \(11\)](#). Recall that the difference between H_2 and H_3 lies in $\mathbf{k}_i^{(j)}, \mathbf{r}^{(j)}$ for $i : y_i^{(j)} \in Y_X^{(j)}$, $j \in [Q]$. Thus, we have $H_2 \approx_s H_3$. $\square$

Lemma A.2. $H_3 \approx_c H_4$ under the $\text{LWE}_{n,m(t+L+1)+1,q,\chi}$ assumption if $\chi = 2m^2L^2 \cdot \omega(\sqrt{\log n})$ and t is the size of the attribute X for the challenge query.

Proof. We prove that if there is an adversary $\mathcal{A}$ that distinguishes H_3 and H_4 with non-negligible probability, we can construct a reduction $\mathcal{B}$ that breaks the $\text{RTLWE}_{n,tm+m+1,m,q,\chi,L}$ assumption. Then, Lemma 2.4 immediately implies Lemma A.2. We can construct $\mathcal{B}$ as follows.

- $\mathcal{A}$ declares the challenge attribute $X = (x_1, \dots, x_t)$ at the beginning of the game. $\mathcal{B}$ outputs $\mathbf{u}^\top = (\underbrace{1, \dots, 1}_t, \underbrace{0, \dots, 0}_{L-t})$ for the security game for the related trapdoor LWE assumption and obtains $(\hat{\mathbf{A}}, \mathbf{B}, \mathbf{v}^\top, \hat{\mathbf{v}}^\top)$ where $(\mathbf{v}^\top, \hat{\mathbf{v}}^\top) = (\mathbf{s}^\top \hat{\mathbf{A}} + \mathbf{e}^\top, \mathbf{s}^\top (\mathbf{u}^\top \otimes \mathbf{I}_n) \mathbf{B} + \hat{\mathbf{e}}^\top)$ or $(\mathbf{v}^\top, \hat{\mathbf{v}}^\top) \leftarrow \mathbb{Z}_q^{tm+m+1} \times \mathbb{Z}_q^{m(2L-1)}$. $\mathcal{B}$ parses $\hat{\mathbf{A}} = [\mathbf{A}|\mathbf{D}_1|\dots|\mathbf{D}_t|\mathbf{p}]$, $\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_L \end{bmatrix}$, and gives $\text{mpk} = \mathbf{A}$ to $\mathcal{A}$.
- Random oracle queries: $\mathcal{B}$ samples $\mathbf{p}_1, \dots, \mathbf{p}_t \leftarrow \mathbb{Z}_q^n$ conditioned on $\sum_{i \in [t]} \mathbf{p}_i = \mathbf{p}$. $\mathcal{B}$ sets $\mathcal{D}(x_i) = [\mathbf{D}_i|\mathbf{B}_i|\mathbf{p}_i]$ for all $i \in [t]$, where $\mathcal{D}$ is the dictionary for the random oracle simulation. Note that $\mathbf{p}$ is kept hidden from the adversary, $\mathbf{p}_1, \dots, \mathbf{p}_t$ are distributed randomly from the view point of $\mathcal{A}$. During the game, when H is invoked on input x , the challenger checks if $\mathcal{D}(x) = \perp$. If this is the case, the challenger samples $(\mathbf{D}, \mathbf{T}_\mathbf{D}) \leftarrow \text{TrapGen}(1^n, m, q)$, $[\mathbf{B}|\mathbf{p}] \leftarrow \mathbb{Z}_q^{n \times ((2L-1)m+1)}$, updates $\mathcal{D}(x) = [\mathbf{D}|\mathbf{B}|\mathbf{p}]$, and returns $[\mathbf{D}|\mathbf{B}|\mathbf{p}]$. Otherwise, it just returns $\mathcal{D}(x)$.
- Secret key queries: when $\mathcal{A}$ queries $\mathcal{B}$ on $Y = (y_1, \dots, y_u)$, $\mathcal{B}$ defines $Y_X = Y \cap X$, $Y_{\bar{X}} = Y \setminus X$, $Y_X = (y_{i_1}, \dots, y_{i_k})$, $\mathbf{m}_i^\top = (0^{i-1}, 1, 0^{L-i}) \in \mathbb{Z}_q^L$ for $i \leq u$. Then, it computes

$$\begin{aligned} [\hat{\mathbf{D}}_i|\hat{\mathbf{B}}_i|\hat{\mathbf{p}}_i] &= H(y_i) \quad \text{for } i \in [u] \\ \mathbf{k}_i &\leftarrow D_{\mathbb{Z}, \chi}^{2m} \quad \text{for } i : y_i \in Y_X \end{aligned}$$

and makes a query $\text{tdO}(\mathbf{M}, \mathbf{t})$ and obtains $\mathbf{r} \in \mathbb{Z}_q^{m(2L-1)}$ as a reply, where $\phi : \{i_1, \dots, i_k\} \rightarrow [t]$ is an injective function such that $\mathbf{B}_{\phi(i)} = \hat{\mathbf{B}}_i$,

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_{\phi(i_1)}^\top \\ \vdots \\ \mathbf{m}_{\phi(i_k)}^\top \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} [\mathbf{A}|\hat{\mathbf{D}}_{i_1}] \mathbf{k}_{i_1} - \hat{\mathbf{p}}_{i_1} \\ \vdots \\ [\mathbf{A}|\hat{\mathbf{D}}_{i_k}] \mathbf{k}_{i_k} - \hat{\mathbf{p}}_{i_k} \end{bmatrix}$$

Finally, it computes

$$\mathbf{k}_i \leftarrow \text{SampleLeft}([\mathbf{A}|\hat{\mathbf{D}}_i], \mathbf{T}_{\hat{\mathbf{D}}_i}, \hat{\mathbf{B}}_i \mathbf{r} + \hat{\mathbf{p}}_i, \chi) \quad \text{for } i : y_i \in Y_{\bar{X}}$$

and returns $\text{sk} = (\mathbf{k}_1, \dots, \mathbf{k}_u, \mathbf{r})$ to $\mathcal{A}$.

- Challenge: after $\mathcal{A}$ outputs the challenge messages (μ_0, μ_1) , $\mathcal{B}$ chooses $\beta \leftarrow \{0, 1\}$ and gives the challenge ciphertext $\text{ct} = (\mathbf{v}^\top + (0^{(t+1)m}, \mu_\beta \lfloor q/2 \rfloor), \hat{\mathbf{v}}^\top)$.
- Output: if $\mathcal{A}$ outputs β' at the end of the game, $\mathcal{B}$ outputs 1 if and only if $\beta = \beta'$.

Observe that $\mathcal{A}$'s view corresponds to H_3 if $(\mathbf{v}^\top, \hat{\mathbf{v}}^\top) = (\mathbf{s}^\top \hat{\mathbf{A}} + \mathbf{e}^\top, \mathbf{s}^\top (\mathbf{u}^\top \otimes \mathbf{I}_n) \mathbf{B} + \hat{\mathbf{e}}^\top)$ and H_4 otherwise. Additionally, if $\mathcal{A}$ is admissible, then $\mathcal{B}$ is admissible. This is because $Y_X \subsetneq X$ for all Y in secret key queries, and thus it is obvious that $k < t$ and thus $\{\mathbf{u}, \mathbf{m}_{\phi(i_1)}, \dots, \mathbf{m}_{\phi(i_k)}\}$ are linearly independent. This concludes the proof. $\square$

A.2 MA-ABE

Construction. Let $n, m, q \in \mathbb{N}$ and $\chi \in \mathbb{R}^+$ be lattice parameters that depend on the security parameter λ , where $q > 2$ is a prime. Let the message space $\mathcal{M} = \{0, 1\}$. Let L be the maximum size of the set used in Enc . Let $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^{n \times (2mL+1)}$ be a hash function. Let $H_2 : \mathcal{G} \rightarrow \mathbb{Z}_q^{m(2L-1)}$ be a hash function modeled as a random oracle, the outputs of which are distributed according to a discrete Gaussian distribution, i.e., for each gid , $H_2(\text{gid})$ is sampled from $D_{\mathbb{Z}, \chi}^{m(2L-1)}$. Waters, Wee, and Wu showed how to implement such a random oracle from a standard random oracle that outputs uniform distribution [WWW22]. Hence, we directly use H_2 in our construction.

GSetup(1^λ): It outputs $\text{gp} = (n, m, q, L, \chi, H_1, H_2)$.

ASetup(α): It outputs mpk and msk as follows. It runs $(\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \text{TrapGen}(1^n, m, q)$ and output these as $(\text{mpk}, \text{msk}) = (\mathbf{A}, (\alpha, \mathbf{A}, \mathbf{T}_\mathbf{A}))$.
Enc($X, \{\text{mpk}_\alpha\}_{\alpha \in A_X}, \mu$): It parses $X = \{(\alpha_1, x_1), \dots, (\alpha_t, x_t)\}$ and $\text{mpk}_\alpha = \mathbf{A}_\alpha$, where it must hold that $t < L$, and outputs $\text{ct} = (\{\mathbf{c}_{1,i}^\top\}_{i \in [t]}, \mathbf{c}_2^\top, c_3)$ as follows.⁹

$$\begin{aligned} \mathbf{s}_\alpha &\leftarrow \mathbb{Z}_q^n, \tilde{\mathbf{e}}_{1,\alpha} \leftarrow D_{\mathbb{Z},\chi}^m \quad \text{for } \alpha \in A_X \\ \hat{\mathbf{e}}_{1,i} &\leftarrow D_{\mathbb{Z},\chi}^m, \underbrace{[\mathbf{D}_i]}_m \mid \underbrace{[\mathbf{B}_i]}_{m(2L-1)} \mid \underbrace{[\mathbf{p}_i]}_1 = H_1((\alpha_i, x_i)) \quad \text{for } i \in [t] \\ \mathbf{e}_{1,i}^\top &= [\tilde{\mathbf{e}}_{1,\alpha_i}^\top | \hat{\mathbf{e}}_{1,i}^\top], \mathbf{e}_2 \leftarrow D_{\mathbb{Z},\chi}^{mL}, \mathbf{e}_3 \leftarrow D_{\mathbb{Z},\chi}, \mathbf{R} \leftarrow \{-1, 1\}^{mL \times m(L-1)} \\ \mathbf{c}_{1,i}^\top &= \mathbf{s}_{\alpha_i}^\top [\mathbf{A}_{\alpha_i} | \mathbf{D}_i] + \mathbf{e}_{1,i}^\top \\ \mathbf{c}_2^\top &= \sum_{i \in [t]} \mathbf{s}_{\alpha_i}^\top \mathbf{B}_i + \mathbf{e}_2^\top [\mathbf{I}_{mL} | \mathbf{R}], \quad c_3 = \sum_{i \in [t]} \mathbf{s}_{\alpha_i}^\top \mathbf{p}_i + e_3 + \mu \lfloor q/2 \rfloor \end{aligned}$$

KeyGen($\text{msk}, \text{gid}, y$): It parses $\text{msk} = (\alpha, \mathbf{A}, \mathbf{T}_\mathbf{A})$, computes $\mathbf{r} = H_2(\text{gid})$, and

$$\underbrace{[\hat{\mathbf{D}}]}_m \mid \underbrace{[\hat{\mathbf{B}}]}_{m(2L-1)} \mid \underbrace{[\hat{\mathbf{p}}]}_1 = H_1((\alpha, y)), \quad \mathbf{k} \leftarrow \text{SampleLeft}([\mathbf{A} | \hat{\mathbf{D}}], \mathbf{T}_\mathbf{A}, \hat{\mathbf{B}}\mathbf{r} + \hat{\mathbf{p}}, \chi)$$

Then, it outputs $\text{sk} = (\mathbf{k}, \text{gid})$.

Dec($\text{ct}, \{\text{sk}_{\alpha,y}\}_{(\alpha,y) \in Y}, X, Y$): It parses $\text{ct} = (\{\mathbf{c}_{1,i}^\top\}_{i \in [t]}, \mathbf{c}_2^\top, c_3)$, $\text{sk}_{\alpha,y} = (\mathbf{k}_{\alpha,y}, \text{gid}_{\alpha,y})$, $X = \{(\alpha_1, x_1), \dots, (\alpha_t, x_t)\}$, and $Y = \{(\alpha'_1, y_1), \dots, (\alpha'_t, y_t)\}$. If there exists $(\alpha, y), (\alpha', y') \in Y$ such that $\text{gid}_{\alpha,y} \neq \text{gid}_{\alpha',y'}$, it outputs $\perp$. If $X \subseteq Y$, it defines $\rho: [t] \rightarrow Y$ such that $\rho(i) = (\alpha_i, x_i)$ and outputs

$$\left\lfloor \frac{2}{q} \cdot \left(c_3 - \left(\sum_{i \in [t]} \mathbf{c}_{1,i}^\top \mathbf{k}_{\rho(i)} - \mathbf{c}_2^\top H_2(\text{gid}) \right) \right) \right\rfloor$$

Otherwise, output $\perp$.

Parameter Selection. From [Theorem A.3](#) and [Theorem A.4](#), we can choose parameters as follows:

$$n = \lambda, \quad q = O(n^8 L^7), \quad m = O(n \log q), \quad \chi = O(m^2 L^2 \log n)$$

Observe that we can choose q, χ as polynomials in λ . Hence, our scheme is secure under the LWE with a polynomial modulus-to-noise ratio.

Correctness. For correctness, we have the following theorem.

Theorem A.3. Suppose $q \geq 5(\sqrt{\lambda}\chi + mL(2 + 2mL - m)\lambda\chi^2)$, $m = O(n \log q)$, and $\chi = \omega(\sqrt{\log m})$. Then, our scheme is correct.

Proof. In decryption, suppose $\text{ct} = (\{\mathbf{c}_{1,i}^\top\}_{i \in [t]}, \mathbf{c}_2^\top, c_3)$ and $\{\text{sk}_{\alpha,y} = (\mathbf{k}_{\alpha,y}, \text{gid})\}_{(\alpha,y) \in Y}$ are a ciphertext and a secret key for $X \in \mathcal{X}$ and $(Y, \text{gid}) \in \mathbb{Y} \times \mathcal{GU}$, respectively, such that $X \subseteq Y$. In decryption, we have

$$\begin{aligned} &c_3 - \left(\sum_{i \in [t]} \mathbf{c}_{1,i}^\top \mathbf{k}_{\rho(i)} - \mathbf{c}_2^\top H_2(\text{gid}) \right) \\ &= \sum_{i \in [t]} \mathbf{s}_{\alpha_i}^\top \mathbf{p}_i + e_3 + \mu \lfloor q/2 \rfloor - \sum_{i \in [t]} (\mathbf{s}_{\alpha_i}^\top [\mathbf{A}_{\alpha_i} | \mathbf{D}_i] + \mathbf{e}_{1,i}^\top) \mathbf{k}_{\rho(i)} + \left(\sum_{i \in [t]} \mathbf{s}_{\alpha_i}^\top \mathbf{B}_i + \mathbf{e}_2^\top [\mathbf{I}_{mL} | \mathbf{R}] \right) \mathbf{r} \\ &= \sum_{i \in [t]} \mathbf{s}_{\alpha_i}^\top \mathbf{p}_i + e_3 + \mu \lfloor q/2 \rfloor - \sum_{i \in [t]} (\mathbf{s}_{\alpha_i}^\top (\mathbf{B}_i \mathbf{r} + \mathbf{p}_i) + \mathbf{e}_{1,i}^\top \mathbf{k}_{\rho(i)}) + \left(\sum_{i \in [t]} \mathbf{s}_{\alpha_i}^\top \mathbf{B}_i + \mathbf{e}_2^\top [\mathbf{I}_{mL} | \mathbf{R}] \right) \mathbf{r} \\ &= \mu \lfloor q/2 \rfloor + e_3 - \underbrace{\sum_{i \in [t]} \mathbf{e}_{1,i}^\top \mathbf{k}_{\rho(i)} + \mathbf{e}_2^\top [\mathbf{I}_{mL} | \mathbf{R}] \mathbf{r}}_{\tilde{e}} \end{aligned}$$

⁹ Note that the first m elements of $\mathbf{c}_{1,i}^\top$ may be redundant if $\alpha_i = \alpha_j$ for some $i \neq j$, but we use this notation for convenience.

where $[\mathbf{D}_i|\mathbf{B}_i|\mathbf{p}_i] = H_1((\alpha_i, x_i))$, $\mathbf{r} = H_2(\text{gid})$. From the parameter setting and Lemma 2.2, we have $|e_3| \leq \sqrt{\lambda\chi}$, $\|\mathbf{e}_{1,i}\| \leq \sqrt{\lambda\chi}$, $\|\mathbf{e}_2\| \leq \sqrt{\lambda\chi}$, and $\|\mathbf{r}\| \leq \sqrt{\lambda\chi}$ with overwhelming probability. Thanks to the property of lattice trapdoors, the distribution of $\mathbf{k}_{\rho(i)}$ is statistically close to $[\mathbf{A}_{\alpha_i}|\mathbf{D}_i]_{\chi}^{-1}(\mathbf{B}_i\mathbf{r} + \mathbf{p}_i)$. Since the marginal distribution of $\mathbf{p}_i$ is uniform in $\mathbb{Z}_q^n$, the marginal distribution of $\mathbf{k}_{\rho(i)}$ is statistically close to $D_{\mathbb{Z},\chi}^{2m}$ from Lemma 2.3. Thus, we have $\|\mathbf{k}_{\rho(i)}\| \leq \sqrt{\lambda\chi}$ with overwhelming probability again from Lemma 2.2. In summary, we have, with overwhelming probability,

$$\begin{aligned} |\tilde{e}| &= \left| e_3 - \sum_{i \in [t]} \mathbf{e}_{1,i}^\top \mathbf{k}_{\rho(i)} + \mathbf{e}_2^\top [\mathbf{I}_{mL}|\mathbf{R}]\mathbf{r} \right| \leq \sqrt{\lambda\chi} + 2mt\lambda\chi^2 + m^2L(2L-1)\lambda\chi^2 \\ &\leq \sqrt{\lambda\chi} + mL(2+2mL-m)\lambda\chi^2 \end{aligned}$$

Thus, $|\tilde{e}| \leq q/5$ with overwhelming probability and correctness holds. $\square$

Security Analysis. For security, we have the following theorem.

Theorem A.4. Suppose $m = O(n \log q)$, $\chi = 2m^2L^2 \cdot \omega(\sqrt{\log n})$, the hash function H is modeled as a random oracle, t is the size of the challenge attribute X , t' is the number of non-corrupted authorities related to the challenge attribute X , and $\text{LWE}_{n,m(t+t'+L)+1,q,\chi}$ holds. Then, our scheme satisfies static security as per Definition 2.2.

Proof. We prove the theorem by a series of hybrids described as follows.

- H_0 : This is the original game. In this game, $\mathcal{A}$ first outputs the challenge attribute $X = \{(\alpha_1^*, x_1^*), \dots, (\alpha_t^*, x_t^*)\}$, the challenge messages μ_0, μ_1 , a set of corrupted authorities $\mathcal{C} \subseteq \mathcal{AU}$ and their master public keys $\{\text{mpk}_\alpha\}_{\alpha \in \mathcal{C}}$ where $\text{mpk}_\alpha = \mathbf{A}_\alpha$, a set of non-corrupted authorities $\mathcal{N} \subseteq \mathcal{AU}$, secret key queries $\{\text{gid}, Y_{\text{gid}}\}_{\text{gid} \in \mathcal{Q}}$ where $\mathcal{Q} \subseteq \mathcal{GU}$ and $Y_{\text{gid}} \in \mathbb{Y}$ for all $\text{gid} \in \mathcal{Q}$. Then, the challenger computes for $\alpha \in \mathcal{N}$,

$$(\mathbf{A}_\alpha, \mathbf{T}_{\mathbf{A}_\alpha}) \leftarrow \text{TrapGen}(1^n, m, q), \quad (\text{mpk}_\alpha, \text{msk}_\alpha) = (\mathbf{A}_\alpha, (\alpha, \mathbf{A}_\alpha, \mathbf{T}_{\mathbf{A}_\alpha}))$$

It computes the challenge ciphertext $\text{ct}^* = (\{\mathbf{c}_{1,i}^\top\}_{i \in [t]}, \mathbf{c}_2^\top, c_3)$ as $b \leftarrow \{0, 1\}$ and

$$\begin{aligned} \mathbf{s}_\alpha &\leftarrow \mathbb{Z}_q^n, \quad \tilde{\mathbf{e}}_{1,\alpha} \leftarrow D_{\mathbb{Z},\chi}^m \quad \text{for } \alpha \in A_X \\ \hat{\mathbf{e}}_{1,i} &\leftarrow D_{\mathbb{Z},\chi}^m, \quad [\underbrace{\mathbf{D}_i}_m \mid \underbrace{\mathbf{B}_i}_{m(2L-1)} \mid \underbrace{\mathbf{p}_i}_1] = H_1((\alpha_i^*, x_i^*)) \quad \text{for } i \in [t] \\ \mathbf{e}_{1,i}^\top &= [\tilde{\mathbf{e}}_{1,\alpha_i^*}^\top | \hat{\mathbf{e}}_{1,i}^\top], \quad \mathbf{e}_2 \leftarrow D_{\mathbb{Z},\chi}^{mL}, \quad e_3 \leftarrow D_{\mathbb{Z},\chi}, \quad \mathbf{R} \leftarrow \{-1, 1\}^{mL \times m(L-1)} \\ \mathbf{c}_{1,i}^\top &= \mathbf{s}_{\alpha_i^*}^\top [\mathbf{A}_{\alpha_i^*} | \mathbf{D}_i] + \mathbf{e}_{1,i}^\top \\ \mathbf{c}_2^\top &= \sum_{i \in [t]} \mathbf{s}_{\alpha_i^*}^\top \mathbf{B}_i + \mathbf{e}_2^\top [\mathbf{I}_{mL} | \mathbf{R}], \quad c_3 = \sum_{i \in [t]} \mathbf{s}_{\alpha_i^*}^\top \mathbf{p}_i + e_3 + \mu_b \lfloor q/2 \rfloor \end{aligned}$$

It computes secret keys $\text{sk}_{\alpha,y,\text{gid}}$ for $(\alpha, y) \in Y_{\text{gid}}, \text{gid} \in \mathcal{Q}$ as $\mathbf{r} = H_2(\text{gid})$ and

$$\begin{aligned} [\underbrace{\hat{\mathbf{D}}}_m \mid \underbrace{\hat{\mathbf{B}}}_{m(2L-1)} \mid \underbrace{\hat{\mathbf{p}}}_1] &= H_1((\alpha, y)), \quad \mathbf{k} \leftarrow \text{SampleLeft}([\mathbf{A}_\alpha | \hat{\mathbf{D}}], \mathbf{T}_{\mathbf{A}_\alpha}, \hat{\mathbf{B}}\mathbf{r} + \hat{\mathbf{p}}, \chi) \\ \text{sk}_{\alpha,y,\text{gid}} &= (\mathbf{k}, \text{gid}) \end{aligned}$$

Then, it gives $(\{\text{mpk}_\alpha\}_{\alpha \in \mathcal{N}}, \text{ct}^*, \{\text{sk}_{\alpha,y,\text{gid}}\}_{(\alpha,y) \in Y_{\text{gid}}, \text{gid} \in \mathcal{Q}})$ to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs b' . During the game, the challenger simulates the random oracle as follows. It maintains a dictionary $\mathcal{D}_1, \mathcal{D}_2$ that are initially empty. When H_1 is invoked on input x , the challenger checks if $\mathcal{D}_1(x) = \perp$. If this is the case, the challenger samples $[\mathbf{D}|\mathbf{B}|\mathbf{p}] \leftarrow \mathbb{Z}_q^{n \times (2mL+1)}$, updates $\mathcal{D}_1(x) = [\mathbf{D}|\mathbf{B}|\mathbf{p}]$, and returns $[\mathbf{D}|\mathbf{B}|\mathbf{p}]$. Otherwise, it returns $\mathcal{D}_1(x)$. Similarly, when H_2 is invoked on input gid , the challenger checks if $\mathcal{D}_2(\text{gid}) = \perp$. If this is the case, the challenger samples $\mathbf{r} \leftarrow D_{\mathbb{Z},\chi}^{m(2L-1)}$, updates $\mathcal{D}_2(\text{gid}) = \mathbf{r}$, and returns $\mathbf{r}$. Otherwise, it returns $\mathcal{D}_2(\text{gid})$.

- H_1 : This is the same as H_0 except that the challenger simulates the random oracle as follows. Let $X = \{(\alpha_1^*, x_1^*), \dots, (\alpha_t^*, x_t^*)\}$ be the challenge attribute that $\mathcal{A}$ outputs at the beginning of the game. The challenger computes $[\mathbf{D}_i|\mathbf{B}_i|\mathbf{p}_i] \leftarrow \mathbb{Z}_q^{n \times (2mL+1)}$ and updates $\mathcal{D}_1((\alpha_i^*, x_i^*)) = [\mathbf{D}_i|\mathbf{B}_i|\mathbf{p}_i]$ for all

$i \in [t]$. During the game, when H_1 is invoked on input x , the challenger checks if $\mathcal{D}_1(x) = \perp$. If this is the case, the challenger samples $(\mathbf{D}, \mathbf{T}_\mathbf{D}) \leftarrow \text{TrapGen}(1^n, m, q)$, $[\mathbf{B}|\mathbf{p}] \leftarrow \mathbb{Z}_q^{n \times ((2L-1)m+1)}$, updates $\mathcal{D}_1(x) = [\mathbf{D}|\mathbf{B}|\mathbf{p}]$, and returns $\mathcal{D}_1(x)$. Otherwise, it returns $\mathcal{D}_1(x)$. Since the property of lattice trapdoors (Lemma 2.1), $\mathbf{D}$ is statistically close to uniform, and thus H_0 and H_1 are statistically indistinguishable given $m = O(n \log q)$.

- H_2 : This is the same as H_1 except the followings. In this hybrid, the challenger samples $\mathbf{A}_\alpha \leftarrow \mathbb{Z}_q^{n \times m}$ for $\alpha \in \mathcal{N}$ instead of using TrapGen for mpk_α . For secret key queries on Y_{gid} for all $\text{gid} \in \mathcal{Q}$, let $Y_{\text{gid},X} = Y_{\text{gid}} \cap X$ and $Y_{\text{gid},\bar{X}} = Y_{\text{gid}} \setminus X$. The challenger computes

$$\begin{aligned} \mathbf{r}_{\text{gid}} &= H_2(\text{gid}), \quad \underbrace{[\hat{\mathbf{D}}_{\alpha,y}]_m}_m \mid \underbrace{[\hat{\mathbf{B}}_{\alpha,y}]_{m(2L-1)}}_{m(2L-1)} \mid \underbrace{[\hat{\mathbf{p}}_{\alpha,y}]_1}_1 = H_1((\alpha, y)) \\ \mathbf{k}_{\alpha,y,\text{gid}} &\leftarrow \begin{cases} [\mathbf{A}_\alpha | \hat{\mathbf{D}}_{\alpha,y}]_\chi^{-1} (\hat{\mathbf{B}}_{\alpha,y} \mathbf{r}_{\text{gid}} + \hat{\mathbf{p}}_{\alpha,y}) & ((\alpha, y) \in Y_{\text{gid},X}) \\ \text{SampleLeft}([\mathbf{A}_\alpha | \hat{\mathbf{D}}_{\alpha,y}], \mathbf{T}_{\hat{\mathbf{D}}_{\alpha,y}}, \hat{\mathbf{B}}_{\alpha,y} \mathbf{r}_{\text{gid}} + \hat{\mathbf{p}}_{\alpha,y}, \chi) & ((\alpha, y) \in Y_{\text{gid},\bar{X}}) \end{cases} \end{aligned}$$

and gives $\text{sk}_{\alpha,y,\text{gid}} = (\mathbf{k}_{\alpha,y,\text{gid}}, \text{gid})$ for $(\alpha, y) \in Y_{\text{gid}}$, $\text{gid} \in \mathcal{Q}$ to $\mathcal{A}$. Note that the sampling of $\mathbf{k}_{\alpha,y,\text{gid}}$ for $(\alpha, y) \in Y_X$ is inefficient. It is easy to see that $H_1 \approx_s H_2$ since $\mathbf{A}_\alpha$ for $\alpha \in \mathcal{N}$ is statistically close to uniform in H_1 and, for all $(\alpha, y) \in Y_{\text{gid}}$,

$$\begin{aligned} &\text{SampleLeft}([\mathbf{A}_\alpha | \hat{\mathbf{D}}_{\alpha,y}], \mathbf{T}_{\hat{\mathbf{D}}_{\alpha,y}}, \hat{\mathbf{B}}_{\alpha,y} \mathbf{r}_{\text{gid}} + \hat{\mathbf{p}}_{\alpha,y}, \chi) \\ &\approx_s \text{SampleLeft}([\mathbf{A}_\alpha | \hat{\mathbf{D}}_{\alpha,y}], \mathbf{T}_{\mathbf{A}_\alpha}, \hat{\mathbf{B}}_{\alpha,y} \mathbf{r}_{\text{gid}} + \hat{\mathbf{p}}_{\alpha,y}, \chi) \\ &\approx_s [\mathbf{A}_\alpha | \hat{\mathbf{D}}_{\alpha,y}]_\chi^{-1} (\hat{\mathbf{B}}_{\alpha,y} \mathbf{r}_{\text{gid}} + \hat{\mathbf{p}}_{\alpha,y}) \end{aligned}$$

which follows from Lemma 2.1 if $m = O(n \log q)$.

- H_3 : This game is the same as H_2 except the followings. For secret key queries on Y_{gid} for all $\text{gid} \in \mathcal{Q}$, let $Y_{\text{gid},X} = Y_{\text{gid}} \cap X = ((\alpha_1, y_1), \dots, (\alpha_k, y_k))$ and $Y_{\text{gid},\bar{X}} = Y_{\text{gid}} \setminus X$. The challenger computes for all $\text{gid} \in \mathcal{Q}$,

$$\begin{aligned} &\underbrace{[\hat{\mathbf{D}}_{\alpha,y}]_m}_m \mid \underbrace{[\hat{\mathbf{B}}_{\alpha,y}]_{m(2L-1)}}_{m(2L-1)} \mid \underbrace{[\hat{\mathbf{p}}_{\alpha,y}]_1}_1 = H_1((\alpha, y)) \quad \text{for } (\alpha, y) \in Y_{\text{gid}} \\ \mathbf{k}_{\alpha,y,\text{gid}} &\leftarrow D_{\mathbb{Z},\chi}^{2m} \quad \text{for } (\alpha, y) \in Y_{\text{gid},X} \\ \mathbf{r}_{\text{gid}} &\leftarrow \begin{bmatrix} \hat{\mathbf{B}}_{\alpha_1,y_1} \\ \vdots \\ \hat{\mathbf{B}}_{\alpha_k,y_k} \end{bmatrix}_\chi^{-1} \left(\begin{bmatrix} [\mathbf{A}_{\alpha_1} | \hat{\mathbf{D}}_{\alpha_1,y_1}] \mathbf{k}_{\alpha_1,y_1} - \hat{\mathbf{p}}_{\alpha_1,y_1} \\ \vdots \\ [\mathbf{A}_{\alpha_k} | \hat{\mathbf{D}}_{\alpha_k,y_k}] \mathbf{k}_{\alpha_k,y_k} - \hat{\mathbf{p}}_{\alpha_k,y_k} \end{bmatrix} \right) \\ \mathbf{k}_{\alpha,y,\text{gid}} &\leftarrow \text{SampleLeft}([\mathbf{A}_\alpha | \hat{\mathbf{D}}_{\alpha,y}], \mathbf{T}_{\hat{\mathbf{D}}_{\alpha,y}}, \hat{\mathbf{B}}_{\alpha,y} \mathbf{r}_{\text{gid}} + \hat{\mathbf{p}}_{\alpha,y}, \chi) \quad \text{for } (\alpha, y) \in Y_{\text{gid},\bar{X}} \end{aligned} \tag{12}$$

and sets $\text{sk}_{\alpha,y,\text{gid}} = (\mathbf{k}_{\alpha,y,\text{gid}}, \text{gid})$ and $H_2(\text{gid}) = \mathbf{r}_{\text{gid}}$ right after $\mathcal{A}$ outputs the challenge instance. We prove $H_2 \approx_s H_3$ in Lemma A.3.

- H_4 : This game is the same as H_3 except the followings. The challenger generates the challenge ciphertext $\text{ct} = (\{\mathbf{c}_{1,i}^\top\}_{i \in [t]}, \mathbf{c}_2^\top, c_3)$ as

$$\begin{aligned} \tilde{\mathbf{c}}_{1,\alpha} &\leftarrow \mathbb{Z}_q^m \quad \text{for } \alpha \in A_X \cap \mathcal{N}, \quad \hat{\mathbf{c}}_{1,i} \leftarrow \mathbb{Z}_q^m \quad \text{for } i : \alpha_i^* \in \mathcal{N} \\ \mathbf{c}_{1,i}^\top &= [\tilde{\mathbf{c}}_{1,\alpha_i^*}^\top | \hat{\mathbf{c}}_{1,i}^\top] \quad \text{for } i : \alpha_i^* \in \mathcal{N}, \quad \mathbf{c}_{1,i}^\top = \mathbf{s}_{\alpha_i^*}^\top [\mathbf{A}_{\alpha_i^*} | \mathbf{D}_i] + \mathbf{e}_{1,i}^\top \quad \text{for } i : \alpha_i^* \in \mathcal{C} \\ (\mathbf{c}_2^\top, c_3) &\leftarrow \mathbb{Z}_q^{m(2L-1)} \times \mathbb{Z}_q \end{aligned}$$

We prove $H_3 \approx_c H_4$ in Lemma A.4.

It is obvious that $\mathcal{A}$ does not obtain the information on b , and thus its advantage is 0 in the last hybrid. This concludes the proof. $\square$

Lemma A.3. $H_2 \approx_s H_3$ if $m = O(n \log q)$ and $\chi = \omega(\sqrt{\log m})$.

Proof. Let $Y_{\text{gid},X} = Y_{\text{gid}} \cap X$ and $Y_{\text{gid},\bar{X}} = Y_{\text{gid}} \setminus X$. Let $\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_t \end{bmatrix}$ and $\mathbf{p} = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_t \end{bmatrix}$ where $[\mathbf{D}_i | \mathbf{B}_i | \mathbf{p}_i] = H_1((\alpha_i^*, x_i^*))$. Thanks to Lemma 2.3, $\{\mathbf{B} \mathbf{r}_{\text{gid}}\}_{\text{gid} \in \mathcal{Q}}$ where $\mathbf{r}_{\text{gid}} = H_2(\text{gid})$ is distributed statistically close

to uniform in $(\mathbb{Z}_q^{nt})^{|\mathcal{Q}|}$, because $t \leq L$, $m = O(n \log q)$, and $\chi = \omega(\sqrt{\log m})$. Hence, the joint distribution of $\{\widehat{\mathbf{B}}_{\alpha,y} \mathbf{r}_{\text{gid}} + \widehat{\mathbf{p}}_{\alpha,y}\}$ for all $(\alpha, y) \in Y_{\text{gid},X}, \text{gid} \in \mathcal{Q}$ are also statistically close to uniform random in $(\mathbb{Z}_q^n)^{\sum_{\text{gid} \in \mathcal{Q}} |Y_{\text{gid},X}|}$, where $[\widehat{\mathbf{B}}_{\alpha,y} | \widehat{\mathbf{D}}_{\alpha,y} | \widehat{\mathbf{p}}_{\alpha,y}] = H_1((\alpha, y))$ (recall that $\widehat{\mathbf{B}}_{\alpha,y} \in \{\mathbf{B}_1, \dots, \mathbf{B}_t\}$ for all $(\alpha, y) \in Y_{\text{gid},X}$). Recall also that $\mathbf{A}_\alpha$ is uniformly distributed in $\mathbb{Z}_q^{n \times m}$ for $\alpha \in \mathcal{N}$. Thanks to [Lemma 3.1](#) (by applying it for each α), the joint distribution of $(\mathbf{A}_\alpha, \mathbf{k}_{\alpha,y,\text{gid}}, \widehat{\mathbf{B}}_{\alpha,y} \mathbf{r}_{\text{gid}} + \widehat{\mathbf{p}}_{\alpha,y})$ for $(\alpha, y) \in Y_{\text{gid},X}, \text{gid} \in \mathcal{Q}$ in $\mathbf{H}_2$ is statistically close to the distribution of $(\mathbf{A}_\alpha, \mathbf{k}_{\alpha,y,\text{gid}}, \widehat{\mathbf{B}}_{\alpha,y} \mathbf{r}_{\text{gid}} + \widehat{\mathbf{p}}_{\alpha,y})$ where we first sample $\mathbf{A}_\alpha \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{k}_{\alpha,y,\text{gid}} \leftarrow D_{\mathbb{Z},\chi}^{2m}$ and then sample

$$\widehat{\mathbf{B}}_{\alpha,y} \leftarrow \mathbb{Z}_q^{n \times m(2L-1)}, \mathbf{r}_{\text{gid}} \leftarrow D_{\mathbb{Z},\chi}^{m(2L-1)}, \widehat{\mathbf{p}}_{\alpha,y} \leftarrow \mathbb{Z}_q^{nt}$$

conditioned on $[\mathbf{A}_\alpha | \widehat{\mathbf{D}}_{\alpha,y}] \mathbf{k}_{\alpha,y,\text{gid}} = \widehat{\mathbf{B}}_{\alpha,y} \mathbf{r}_{\text{gid}} + \widehat{\mathbf{p}}_{\alpha,y}$ for $(\alpha, y) \in Y_{\text{gid},X}, \text{gid} \in \mathcal{Q}$. This is conceptually the same as sampling $\widehat{\mathbf{B}}_{\alpha,y} \leftarrow \mathbb{Z}_q^{n \times m(2L-1)}, \widehat{\mathbf{p}}_{\alpha,y} \leftarrow \mathbb{Z}_q^n$ for $(\alpha, y) \in \bigcup_{\text{gid} \in \mathcal{Q}} Y_{\text{gid},X}$, and then sampling $\mathbf{r}_{\text{gid}}$ for each $\text{gid} \in \mathcal{Q}$ as [Eq. \(12\)](#). Recall that the difference between $\mathbf{H}_2$ and $\mathbf{H}_3$ lies in $\mathbf{k}_{\alpha,y,\text{gid}}, \mathbf{r}_{\text{gid}}$ for $(\alpha, y) \in Y_{\text{gid},X}, \text{gid} \in \mathcal{Q}$. Thus, we have $\mathbf{H}_2 \approx_s \mathbf{H}_3$. $\square$

Lemma A.4. $\mathbf{H}_3 \approx_c \mathbf{H}_4$ under the $\text{LWE}_{n,m(t+t'+L)+1,q,\chi}$ assumption if $\chi = 2m^2 L^2 \cdot \omega(\sqrt{\log n})$, t is the size of the challenge attribute X , and t' is the number of non-corrupted authorities related to the challenge attribute X .

Proof. We prove that if there is an adversary $\mathcal{A}$ that distinguishes $\mathbf{H}_3$ and $\mathbf{H}_4$ with non-negligible probability, we can construct a reduction $\mathcal{B}$ that breaks the $\text{RTLWE}_{n,(t+t')m+1,m,q,\chi,L}$ assumption. Then, [Lemma 2.4](#) immediately implies [Lemma A.4](#). We can construct $\mathcal{B}$ as follows. After $\mathcal{A}$ outputs the challenge instance, $\mathcal{B}$ generates the input $(\{\text{mpk}_\alpha\}_{\alpha \in \mathcal{N}}, \text{ct}^*, \{\text{sk}_{\alpha,y,\text{gid}}\}_{(\alpha,y) \in Y_{\text{gid}}, \text{gid} \in \mathcal{Q}})$ to $\mathcal{A}$ as follows:

- Non-corrupted master public keys: $\mathcal{A}$ declares the challenge attribute $X = \{(\alpha_1^*, x_1^*), \dots, (\alpha_t^*, x_t^*)\}$ and non-corrupted authorities $\mathcal{N}$ at the beginning of the game. Let $A_X \cap \mathcal{N} = \{\alpha_1, \dots, \alpha_{t'}\}$. $\mathcal{B}$ outputs $\mathbf{u} = (u_1, \dots, u_L)$ where $u_i = 1$ if $\alpha_i^* \in \mathcal{N}$ and $u_i = 0$ otherwise, for the security game for the related trapdoor LWE assumption. Then, it obtains $(\widehat{\mathbf{A}}, \mathbf{B}, \mathbf{v}^\top, \widehat{\mathbf{v}}^\top)$ where $(\mathbf{v}^\top, \widehat{\mathbf{v}}^\top) = (\mathbf{s}^\top \widehat{\mathbf{A}} + \mathbf{e}^\top, \mathbf{s}^\top (\mathbf{u}^\top \otimes \mathbf{I}_n) \mathbf{B} + \widehat{\mathbf{e}}^\top)$ or $(\mathbf{v}^\top, \widehat{\mathbf{v}}^\top) \leftarrow \mathbb{Z}_q^{tm+m+1} \times \mathbb{Z}_q^{m(2L-1)}$. $\mathcal{B}$ parses

$$\widehat{\mathbf{A}} = [\mathbf{A}_{\alpha_1} | \dots | \mathbf{A}_{\alpha_{t'}} | \mathbf{D}_1 | \dots | \mathbf{D}_t | \mathbf{p}], \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_L \end{bmatrix}$$

and sets $\text{mpk}_{\alpha_i} = \mathbf{A}_{\alpha_i}$ for $i \in [t']$. For $\alpha \in \mathcal{N} \setminus A_X$, $\mathcal{B}$ chooses $\text{mpk}_\alpha = \mathbf{A}_\alpha \leftarrow \mathbb{Z}_q^{n \times m}$.

- Challenge ciphertext: $\mathcal{A}$ outputs corrupted master public keys $\text{mpk}_\alpha = \mathbf{A}_\alpha$ for $\alpha \in \mathcal{C}$. $\mathcal{B}$ parses $\mathbf{v}^\top = [\mathbf{v}_{\alpha_1}^\top | \dots | \mathbf{v}_{\alpha_{t'}}^\top | \widehat{\mathbf{v}}_1^\top | \dots | \widehat{\mathbf{v}}_t^\top | v]$. $\mathcal{B}$ samples $b \leftarrow \{0, 1\}$, $\mathbf{p}_i \leftarrow \mathbb{Z}_q^n$ conditioned on $\sum_{i: \alpha_i^* \in \mathcal{N}} \mathbf{p}_i = \mathbf{p}$ for $i: \alpha_i^* \in \mathcal{N}$ and $\mathbf{p}_i \leftarrow \mathbb{Z}_q^n$ for $i: \alpha_i^* \in \mathcal{C}$. Then, $\mathcal{B}$ computes $\text{ct} = (\{\mathbf{c}_{1,i}^\top\}_{i \in [t]}, \mathbf{c}_2^\top, c_3)$ as follows.

$$\begin{aligned} \mathbf{s}_\alpha &\leftarrow \mathbb{Z}_q^n \quad \text{for } \alpha \in A_X, \quad \widehat{\mathbf{e}}_{1,\alpha} \leftarrow D_{\mathbb{Z},\chi}^m \quad \text{for } \alpha \in A_X \cap \mathcal{C} \\ \widehat{\mathbf{e}}_{1,i} &\leftarrow D_{\mathbb{Z},\chi}^m \text{ for } i: \alpha_i^* \in \mathcal{C} \\ \mathbf{c}_{1,i}^\top &= \begin{cases} \mathbf{s}_{\alpha_i^*} [\mathbf{A}_{\alpha_i^*} | \mathbf{D}_i] + [\mathbf{v}_{\alpha_i^*}^\top | \widehat{\mathbf{v}}_i^\top] & (i: \alpha_i^* \in \mathcal{N}) \\ \mathbf{s}_{\alpha_i^*} [\mathbf{A}_{\alpha_i^*} | \mathbf{D}_i] + [\widehat{\mathbf{e}}_{\alpha_i^*}^\top | \widehat{\mathbf{e}}_i^\top] & (i: \alpha_i^* \in \mathcal{C}) \end{cases} \\ \mathbf{c}_2^\top &= \sum_{i \in [t]} \mathbf{s}_{\alpha_i^*} \mathbf{B}_i + \widehat{\mathbf{v}}^\top, \quad c_3 = \sum_{i \in [t]} \mathbf{s}_{\alpha_i^*} \mathbf{p}_i + v + \mu_b \lfloor q/2 \rfloor \end{aligned}$$

- Random oracle programming for H_1 : $\mathcal{B}$ sets $\mathcal{D}_1((\alpha_i^*, x_i^*)) = [\mathbf{D}_i | \mathbf{B}_i | \mathbf{p}_i]$ for all $i \in [t]$. Note that $\mathbf{p}$ is kept hidden from the adversary, $\mathbf{p}_1, \dots, \mathbf{p}_t$ are distributed randomly from the view point of $\mathcal{A}$. During the game, when H_1 is invoked on input x , the challenger checks if $\mathcal{D}_1(x) = \perp$. If this is the case, the challenger samples $(\mathbf{D}, \mathbf{T}_\mathbf{D}) \leftarrow \text{TrapGen}(1^n, m, q)$, $[\mathbf{B} | \mathbf{p}] \leftarrow \mathbb{Z}_q^{n \times ((2L-1)m+1)}$, updates $\mathcal{D}_1(x) = [\mathbf{D} | \mathbf{B} | \mathbf{p}]$, and returns $[\mathbf{D} | \mathbf{B} | \mathbf{p}]$. Otherwise, it just returns $\mathcal{D}_1(x)$.
- Secret keys: For each Y_{gid} in the challenge instance, $\mathcal{B}$ simulates $\text{sk}_{\alpha,y,\text{gid}} = (\mathbf{k}_{\alpha,y,\text{gid}}, \text{gid})$ as follows. Let $Y_{\text{gid},X} = Y_{\text{gid}} \cap X = ((\alpha_1, y_1), \dots, (\alpha_k, y_k))$ and $Y_{\text{gid},\bar{X}} = Y_{\text{gid}} \setminus X$, $\mathbf{m}_i^\top = (0^{i-1}, 1, 0^{L-i}) \in \mathbb{Z}_q^L$ for $i \leq L$. Then, it computes for all $\text{gid} \in \mathcal{Q}$,

$$\begin{aligned} [\widehat{\mathbf{D}}_{\alpha,y} | \widehat{\mathbf{B}}_{\alpha,y} | \widehat{\mathbf{p}}_{\alpha,y}] &= H_1(\alpha, y) \quad \text{for } (\alpha, y) \in Y_{\text{gid}} \\ \mathbf{k}_{\alpha,y,\text{gid}} &\leftarrow D_{\mathbb{Z},\chi}^{2m} \quad \text{for } (\alpha, y) \in Y_{\text{gid},X} \end{aligned}$$

and makes a query $\text{tdO}(\mathbf{M}, \mathbf{t})$ and obtains $\mathbf{r}_{\text{gid}} \in \mathbb{Z}_q^{m(2L-1)}$ as a reply, where $\phi : [k] \rightarrow [L]$ is an injective function such that $\mathbf{B}_{\phi(i)} = \widehat{\mathbf{B}}_{\alpha_i, y_i}$,

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_{\phi(1)}^\top \\ \vdots \\ \mathbf{m}_{\phi(k)}^\top \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} [\mathbf{A}_{\alpha_1} | \widehat{\mathbf{D}}_{\alpha_1, y_1}] \mathbf{k}_{\alpha_1, y_1, \text{gid}} - \widehat{\mathbf{p}}_{\alpha_1, y_1} \\ \vdots \\ [\mathbf{A}_{\alpha_k} | \widehat{\mathbf{D}}_{\alpha_k, y_k}] \mathbf{k}_{\alpha_k, y_k, \text{gid}} - \widehat{\mathbf{p}}_{\alpha_k, y_k} \end{bmatrix}$$

Finally, it computes

$$\mathbf{k}_{\alpha, y, \text{gid}} \leftarrow \text{SampleLeft}([\mathbf{A}_\alpha | \widehat{\mathbf{D}}_{\alpha, y}], \mathbf{T}_{\widehat{\mathbf{D}}_{\alpha, y}}, \widehat{\mathbf{B}}_{\alpha, y} \mathbf{r}_{\text{gid}} + \widehat{\mathbf{p}}_{\alpha, y}, \chi) \quad \text{for } (\alpha, y) \in Y_{\text{gid}, \bar{X}}$$

- Random oracle programming for H_2 : $\mathcal{B}$ sets $\mathcal{D}_2(\text{gid}) = \mathbf{r}_{\text{gid}}$ for all $\text{gid} \in \mathcal{Q}$. The rest of the simulation of H_2 is the same as the challenger.

At the end of the game, if $\mathcal{A}$ outputs b' at the end of the game, $\mathcal{B}$ outputs 1 if and only if $b = b'$. Observe that $\mathcal{A}$'s view corresponds to H_3 if $(\mathbf{v}^\top, \widehat{\mathbf{v}}^\top) = (\mathbf{s}^\top \widehat{\mathbf{A}} + \mathbf{e}^\top, \mathbf{s}^\top (\mathbf{u}^\top \otimes \mathbf{I}_n) \mathbf{B} + \widehat{\mathbf{e}}^\top)$ and H_4 otherwise. Especially, $\mathcal{B}$ implicitly defines $\mathbf{s}'_\alpha = \mathbf{s} + \mathbf{s}_\alpha$ for $\alpha \in A_X \cap \mathcal{N}$. Additionally, if $\mathcal{A}$ is admissible, then $\mathcal{B}$ is admissible. This is because $Y_{\text{gid}, X} \subseteq X \cap \mathcal{N}$ for all $\text{gid} \in \mathcal{Q}$, which follows from the game condition, and thus it is obvious that $k < t'$ and thus $\{\mathbf{u}, \mathbf{m}_{\phi(i_1)}, \dots, \mathbf{m}_{\phi(i_k)}\}$ are linearly independent. This concludes the proof. $\square$

B Ring-RTLWE and Ring-LHL with Auxiliary Information

B.1 Preliminaries

The Ring Setting. Throughout, let λ be a security parameter and let $N = N(\lambda)$ be a power of two. We work with the $2N$ -th cyclotomic ring of integers

$$\mathcal{R} = \mathbb{Z}[x]/(x^N + 1),$$

whose elements are polynomials of degree less than N with integer coefficients. For an integer modulus $q = q(\lambda)$, let

$$R_q = \mathcal{R}/q\mathcal{R} = \mathbb{Z}_q[x]/(x^N + 1).$$

Throughout this document, we assume that q is a prime that splits completely in $\mathcal{R}$, i.e., $q\mathcal{R}$ factors as

$$q\mathcal{R} = \mathfrak{q}_1 \cdot \mathfrak{q}_2 \cdots \mathfrak{q}_N$$

where each $\mathfrak{q}_i$ is a prime ideal of norm $\mathcal{N}(\mathfrak{q}_i) = q$. In the notation of JLWG25[JLWG25], this corresponds to the parameters $f = 1$, $g = N$, $e = 1$. Such q are typically called *NTT-friendly* primes, and they admit efficient ring arithmetic via the Number Theoretic Transform.

For an element $a = \sum_{i=0}^{N-1} a_i x^i \in \mathcal{R}$, we write $\phi(a) = (a_0, \dots, a_{N-1}) \in \mathbb{Z}^N$ for its coefficient embedding, and define the coefficient norm

$$\|a\|_{\text{coeff}} := \|\phi(a)\|_\infty = \max_{0 \leq i < N} |a_i|.$$

We extend $\|\cdot\|_{\text{coeff}}$ to vectors and matrices over $\mathcal{R}$ entrywise (i.e., as the maximum coefficient norm of any entry).

For a Gaussian width parameter $\sigma > 0$, we write $D_{\mathcal{R}, \sigma}^{\text{coeff}}$ for the discrete Gaussian distribution over $\mathcal{R}$ where each coefficient is independently sampled from the discrete Gaussian $D_{\mathbb{Z}, \sigma}$ over the integers. We extend this notation to vectors and matrices over $\mathcal{R}$ in the natural way: $D_{\mathcal{R}, \sigma}^{\text{coeff}, m}$ samples a vector of m independent ring elements, each from $D_{\mathcal{R}, \sigma}^{\text{coeff}}$.

Ring-LWE. We recall the Ring Learning with Errors assumption [LPR10, LPR13].

Assumption B.1 (Ring-LWE). *Let λ be a security parameter, and let $N = N(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$, and $\chi = \chi(\lambda)$ be lattice parameters. The $\text{RLWE}_{N, m, q, \chi}$ assumption states that the following two distributions are computationally indistinguishable:*

$$\left\{ (\mathbf{a}, \mathbf{a} \cdot \mathbf{s} + \mathbf{e}) : \mathbf{a} \xleftarrow{\$} R_q^m, \mathbf{s} \xleftarrow{\$} R_q, \mathbf{e} \leftarrow D_{\mathcal{R}, \chi}^{\text{coeff}, m} \right\}$$

and

$$\left\{ (\mathbf{a}, \mathbf{u}) : \mathbf{a} \xleftarrow{\$} R_q^m, \mathbf{u} \xleftarrow{\$} R_q^m \right\}.$$

Remark B.1. We have stated Ring-LWE for the case where the secret s is a single ring element (rank 1). The natural generalization to higher rank is the Module Learning with Errors (Module-LWE) assumption, where the secret is a vector of ring elements. We do not need this generalization in what follows.

Min-Entropy and Bounded Leakage. We recall the standard fact that conditioning on a random variable with bounded support reduces min-entropy by at most the logarithm of the support size.

Lemma B.1 ([JLWG25, Lemma 2.1]). *Let X, Y, Z be (correlated) random variables where the support of Z has size at most 2^ℓ . Then*

$$H_\infty(X | Y, Z) \geq H_\infty(X | Y) - \ell.$$

Min-Entropy of Discrete Gaussians Modulo Prime Ideals. We recall the min-entropy bound for discrete Gaussians modulo prime ideal factors of $q\mathcal{R}$ from JLWG25. We state the version specialized to NTT-friendly q , where every prime ideal factor $\mathfrak{q}$ of $q\mathcal{R}$ has norm q .

Lemma B.2 ([JLWG25, Corollary 4], specialized). *Let $\mathcal{R}$ be the $2N$ -th cyclotomic ring of integers and let q be an NTT-friendly prime, so that every prime ideal factor $\mathfrak{q}$ of $q\mathcal{R}$ has norm $\mathcal{N}(\mathfrak{q}) = q$. Let $\sigma > 0$ be a Gaussian parameter satisfying*

$$\sigma \leq \frac{\sqrt{\pi}}{2} \cdot \frac{\sqrt{q}}{\sqrt{\ln 4}}.$$

Let $S := D_{\mathcal{R}, \sigma}^{\text{coeff}} \bmod \mathfrak{q}$. Then

$$H_\infty(S) \geq \log \sigma - 1.$$

Remark B.2. The ceiling on σ in Lemma B.2 comes from JLWG25's Claim 4.2, which establishes $\rho_\sigma(\mathbb{Z}_q^t) \geq \sigma^t/2$ when $\sigma \leq \frac{\sqrt{\pi}}{2} \cdot \frac{\sqrt{q}}{\sqrt{\ln(4t)}}$. The original statement in JLWG25 uses $\ln(4t)$ where t is the norm exponent of the ideal factor $\mathfrak{q}$ (so $\mathcal{N}(\mathfrak{q}) = q^t$). In our NTT-friendly setting, $t = 1$, and the bound becomes $\sigma \leq \frac{\sqrt{\pi}}{2} \cdot \frac{\sqrt{q}}{\sqrt{\ln 4}}$.

For applications where the lemma is invoked with a vector of m ring elements, one can apply the tighter version of Claim 4.2 (which extends naturally to higher dimensions and replaces $\ln 4$ with $\ln(4m)$ in the denominator). This is a constant-factor optimization that does not affect any of the main results below.

Algebraic Leftover Hash Lemma over Rings. The following is the main entropy-based leftover hash lemma from JLWG25, specialized to NTT-friendly q and ring rank $n = 1$. It relates the statistical distance between $(\mathbf{a}, \mathbf{a}^\top S, \text{aux})$ and the uniform distribution to the conditional min-entropy of S modulo each prime ideal factor.

Lemma B.3 ([JLWG25, Theorem 5.3], specialized to NTT-friendly q). *Let $\mathcal{R}$ be the $2N$ -th cyclotomic ring of integers, let q be an NTT-friendly prime, and let $n, m \geq 1$ be parameters. Let (S, aux) be correlated random variables with S supported on $\mathcal{R}^m$, such that for every prime ideal factor $\mathfrak{q}$ of $q\mathcal{R}$,*

$$H_\infty(S \bmod \mathfrak{q} | \text{aux}) \geq m \log \sigma - \delta$$

for some $\sigma, \delta > 0$. Let

$$\begin{aligned} \mathcal{D}_0 &= \left\{ (\mathbf{A}, \mathbf{A}\mathbf{r}, \text{aux}) : \mathbf{A} \xleftarrow{\$} R_q^{n \times m}, \mathbf{r} \leftarrow S \right\}, \\ \mathcal{D}_1 &= \left\{ (\mathbf{A}, \mathbf{u}, \text{aux}) : \mathbf{A} \xleftarrow{\$} R_q^{n \times m}, \mathbf{u} \xleftarrow{\$} R_q^n, \mathbf{r} \leftarrow S \right\}. \end{aligned}$$

Then for any $\varepsilon > 0$ with $\varepsilon < 2^{(\delta-1)/2}$, if

$$m \log \sigma \geq 2 \log(1/\varepsilon) + n \log q + \log g + \delta,$$

where $g = N$ is the number of distinct prime ideal factors of $q\mathcal{R}$, we have $\text{SD}(\mathcal{D}_0, \mathcal{D}_1) \leq \varepsilon$.

B.2 Ring-LHL with Auxiliary Linear Information

We now present the main lemma of this section: a Ring-LHL statement that retains its conclusion even when an auxiliary linear function of the Gaussian secret is leaked. This is the ring analog of the auxiliary form of the Leftover Hash Lemma from Lemma 3.14 of [WWW22], which they used in the security proof of their generalized related-trapdoor LWE assumption.

Lemma B.4 (Ring-LHL with Auxiliary Linear Information). *Let λ be a security parameter, let $\mathcal{R}$ be the $2N$ -th cyclotomic ring of integers for $N = N(\lambda)$ a power of two, and let $q = q(\lambda)$ be an NTT-friendly prime. Let $n, m, k = \text{poly}(\lambda)$ be parameters, $\sigma > 0$ a Gaussian width, and $\varepsilon > 0$ a target distance. Suppose:*

1. (Width upper bound.) $\sigma \leq \frac{\sqrt{\pi}}{2} \cdot \frac{\sqrt{q}}{\sqrt{\ln 4}}.$
2. (Width lower bound / entropy condition.)

$$m \log \sigma \geq 2 \log(k/\varepsilon) + (n + N) \log q + \log g + m,$$

where $g = N$ is the number of distinct prime ideal factors of $q\mathcal{R}$.

Then for any fixed vector $\mathbf{e} \in \mathcal{R}^m$, the following two distributions are within statistical distance ε :

$$\begin{aligned} \mathcal{D}_0 &= \left\{ (\mathbf{A}, \mathbf{A}\mathbf{R}, \mathbf{e}^\top \mathbf{R}) : \mathbf{A} \xleftarrow{\$} R_q^{n \times m}, \mathbf{R} \leftarrow D_{\mathcal{R}, \sigma}^{\text{coeff}, m \times k} \right\}, \\ \mathcal{D}_1 &= \left\{ (\mathbf{A}, \mathbf{U}, \mathbf{e}^\top \mathbf{R}) : \mathbf{A} \xleftarrow{\$} R_q^{n \times m}, \mathbf{U} \xleftarrow{\$} R_q^{n \times k}, \mathbf{R} \leftarrow D_{\mathcal{R}, \sigma}^{\text{coeff}, m \times k} \right\}. \end{aligned}$$

Remark B.3. Lemma B.4 is quantified over *any* fixed vector $\mathbf{e} \in \mathcal{R}^m$, with no norm bound. This is because the proof handles the leakage $\mathbf{e}^\top \mathbf{R}$ via the bounded-support lemma (Lemma B.1), which only uses the worst-case support size of the leakage and is insensitive to the specific structure of $\mathbf{e}$. As a consequence, the lemma is well-suited for security reductions in which an adversary adaptively chooses $\mathbf{e}$ before invoking the LHL: the lemma's conclusion holds uniformly over all such choices.

Proof of Lemma B.4. The proof proceeds in three steps. First, we use a hybrid argument over the columns of $\mathbf{R}$ to reduce to the single-column ($k = 1$) case. Second, we bound the conditional min-entropy of a single column of $\mathbf{R}$ given the corresponding leakage entry. Third, we apply Lemma B.3 to conclude.

Step 1: Reduction to a single column.

For $j = 0, 1, \dots, k$, define the hybrid distribution

$$\mathcal{H}_j = \left\{ (\mathbf{A}, [\mathbf{u}_1, \dots, \mathbf{u}_j, \mathbf{A}\mathbf{R}_{*,j+1}, \dots, \mathbf{A}\mathbf{R}_{*,k}], \mathbf{e}^\top \mathbf{R}) : \mathbf{A} \xleftarrow{\$} R_q^{n \times m}, \mathbf{R} \leftarrow D_{\mathcal{R}, \sigma}^{\text{coeff}, m \times k}, \mathbf{u}_1, \dots, \mathbf{u}_j \xleftarrow{\$} R_q^n \right\},$$

where $\mathbf{R}_{*,\ell}$ denotes the ℓ -th column of $\mathbf{R}$. Then $\mathcal{H}_0 = \mathcal{D}_0$ and $\mathcal{H}_k = \mathcal{D}_1$, so by the triangle inequality,

$$\text{SD}(\mathcal{D}_0, \mathcal{D}_1) \leq \sum_{j=1}^k \text{SD}(\mathcal{H}_{j-1}, \mathcal{H}_j).$$

It therefore suffices to show $\text{SD}(\mathcal{H}_{j-1}, \mathcal{H}_j) \leq \varepsilon/k$ for each $j \in [k]$.

Fix j . The distributions $\mathcal{H}_{j-1}$ and $\mathcal{H}_j$ differ only in the j -th column of the second component: it is real ($\mathbf{A}\mathbf{R}_{*,j}$) in $\mathcal{H}_{j-1}$ and uniform ($\mathbf{u}_j$) in $\mathcal{H}_j$. All other columns of $\mathbf{R}$ are sampled identically and independently in both hybrids, and the auxiliary $\mathbf{e}^\top \mathbf{R}$ decomposes column-wise: its ℓ -th entry is $(\mathbf{e}^\top \mathbf{R})_\ell = \mathbf{e}^\top \mathbf{R}_{*,\ell}$, which depends only on the ℓ -th column of $\mathbf{R}$.

We may therefore condition on the values of all columns $\mathbf{R}_{*,\ell}$ for $\ell \neq j$ without affecting the statistical distance between $\mathcal{H}_{j-1}$ and $\mathcal{H}_j$. After conditioning, the only random quantities are $\mathbf{A}$, $\mathbf{R}_{*,j}$, and (in $\mathcal{H}_j$) $\mathbf{u}_j$, and the only entry of the auxiliary information that depends on $\mathbf{R}_{*,j}$ is $(\mathbf{e}^\top \mathbf{R})_j = \mathbf{e}^\top \mathbf{R}_{*,j}$. The other entries of the auxiliary information are determined by the conditioning and may be treated as constants.

Therefore, it suffices to prove

$$\text{SD}((\mathbf{A}, \mathbf{A}\mathbf{r}, \mathbf{e}^\top \mathbf{r}), (\mathbf{A}, \mathbf{u}, \mathbf{e}^\top \mathbf{r})) \leq \varepsilon/k,$$

where $\mathbf{A} \xleftarrow{\$} R_q^{n \times m}$, $\mathbf{r} \leftarrow D_{\mathcal{R}, \sigma}^{\text{coeff}, m}$, and $\mathbf{u} \xleftarrow{\$} R_q^n$.

Step 2: Bounding the conditional min-entropy.

We bound $H_\infty(\mathbf{r} \bmod \mathfrak{q} \mid \mathbf{e}^\top \mathbf{r})$ for an arbitrary prime ideal factor $\mathfrak{q}$ of $q\mathcal{R}$.

Since $\mathbf{r}$ consists of m independent samples from $D_{\mathcal{R},\sigma}^{\text{coeff}}$, the reduction $\mathbf{r} \bmod \mathfrak{q}$ consists of m independent samples from $D_{\mathcal{R},\sigma}^{\text{coeff}} \bmod \mathfrak{q}$. By [Lemma B.2](#) and the upper bound on σ from condition 1 of the lemma, each such sample has min-entropy at least $\log \sigma - 1$. Summing over the m independent entries:

$$H_\infty(\mathbf{r} \bmod \mathfrak{q}) \geq m \log \sigma - m.$$

Now we account for the leakage. The auxiliary information $\mathbf{e}^\top \mathbf{r}$ is a single ring element, lying in R_q (or, more precisely, taking values in R_q once we view the inner product modulo q). The support of $\mathbf{e}^\top \mathbf{r}$ is therefore at most $|R_q| = q^N$. By [Lemma B.1](#),

$$H_\infty(\mathbf{r} \bmod \mathfrak{q} \mid \mathbf{e}^\top \mathbf{r}) \geq H_\infty(\mathbf{r} \bmod \mathfrak{q}) - N \log q \geq m \log \sigma - m - N \log q.$$

We can therefore set $\delta = m + N \log q$ in the application of [Lemma B.3](#).

Step 3: Applying [Lemma B.3](#).

With $\delta = m + N \log q$ and target distance $\varepsilon' = \varepsilon/k$, the condition required by [Lemma B.3](#) becomes

$$m \log \sigma \geq 2 \log(k/\varepsilon) + n \log q + \log g + m + N \log q = 2 \log(k/\varepsilon) + (n + N) \log q + \log g + m,$$

which is exactly condition 2 of our lemma.

We must also verify the side condition $\varepsilon' < 2^{(\delta-1)/2}$, i.e., $\varepsilon/k < 2^{(m+N \log q-1)/2}$. Since ε is at most a polynomial in λ (in fact, typically negligible) and the right-hand side is exponentially large in $m + N \log q$, this side condition holds trivially.

Therefore, by [Lemma B.3](#), the per-column distance is at most ε/k . Summing over k columns via Step 1 yields

$$\text{SD}(\mathcal{D}_0, \mathcal{D}_1) \leq k \cdot (\varepsilon/k) = \varepsilon,$$

which completes the proof. $\square$

Remark B.4. The width upper bound (condition 1) and lower bound (condition 2) of [Lemma B.4](#) are nearly tight against each other. Substituting the upper bound $\log \sigma \approx \frac{1}{2} \log q$ into the lower bound gives the parameter constraint

$$\frac{m}{2} \log q \gtrsim (n + N) \log q,$$

or equivalently $m \gtrsim 2(n + N)$. In the $n = 1$ case (plain Ring-LWE, single output ring element), this gives $m \geq 2N + 2$, which is the minimum for which the lemma applies. More generally, for larger n (e.g., $n = L$ as needed in some security reductions), the required m grows linearly with $n + N$; when n dominates N (which happens when $n \gtrsim N$), the constraint on m becomes $m \geq 2n$ with essentially no dependence on N for small n . This tradeoff between m and n reflects the entropy budget: each additional unit of ring rank n in the output costs one ring element of entropy in the source.

Conjectural Improvement: Per-Ideal Leakage Bound. The $(n + N) \log q$ term in condition 2 of [Lemma B.4](#) arises from two separate sources: $n \log q$ is the inherent cost of making n ring elements of output uniform (from JLWG25 Theorem 5.3), and $N \log q$ is the entropy lost to the leakage $\mathbf{e}^\top \mathbf{r}$ (from the bounded-leakage bound of [\[DRS04\]](#), [Lemma B.1](#)). The [\[DRS04\]](#) bound charges the full support size $|R_q| = q^N$ of the ring-valued leakage, which is likely pessimistic: the leakage $\mathbf{e}^\top \mathbf{r} \bmod q$ decomposes via CRT into N components (one per prime ideal factor $\mathfrak{q}_i$), and each component carries only $\log q$ bits that are “relevant” to the collision probability of $\mathbf{r} \bmod \mathfrak{q}_i$. The remaining $(N - 1) \log q$ bits are “cross-ideal” leakage that intuitively should not significantly affect the collision probability at ideal $\mathfrak{q}_i$. We formalize this intuition as a conjecture:

Assumption B.2 (Per-Ideal Leakage Bound). *Let $\mathcal{R} = \mathbb{Z}[x]/(x^N + 1)$ be the $2N$ -th cyclotomic ring of integers, q an NTT-friendly prime, and $\mathfrak{q}_i$ any prime ideal factor of $q\mathcal{R}$ with $\mathcal{N}(\mathfrak{q}_i) = q$. Let $\mathbf{r} \leftarrow (D_{\mathcal{R},\sigma}^{\text{coeff}})^m$ be a vector of independent discrete Gaussian ring elements with $\sigma \leq \frac{\sqrt{\pi}}{2} \cdot \frac{\sqrt{q}}{\sqrt{\ln 4}}$, and let $\mathbf{e} \in \mathcal{R}^m$ be any fixed vector. Then:*

$$\text{Col}(\mathbf{r} \bmod \mathfrak{q}_i \mid \mathbf{e}^\top \mathbf{r} \bmod q) \leq q \cdot \text{Col}(\mathbf{r} \bmod \mathfrak{q}_i).$$

Remark B.5. [Assumption B.2](#) asserts that conditioning on the full ring-valued leakage $\mathbf{e}^\top \mathbf{r} \bmod q$ increases the collision probability modulo any single prime ideal $\mathfrak{q}_i$ by at most a factor of q (the “same-ideal” leakage cost), rather than the q^N factor given by [\[DRS04\]](#). This is plausible because the CRT decomposition $R_q \cong \mathbb{F}_q^N$ maps the leakage into N independent-looking $\mathbb{F}_q$ -components, of which only the i -th is directly relevant to $\mathbf{r} \bmod \mathfrak{q}_i$. A formal proof would require showing that the cross-ideal correlations induced by the coefficient-to-CRT transformation are benign for collision probability purposes; we leave this as an interesting question for future work.

Under [Assumption B.2](#), we obtain an improved version of the Ring-LHL:

Corollary B.1 (Ring-LHL with Auxiliary Linear Information, conditional improvement).
Assume that [Assumption B.2](#) holds. Then [Lemma B.4](#) holds with condition 2 replaced by:

$$m \log \sigma \geq 2 \log(k/\varepsilon) + (n+1) \log q + \log g + m.$$

Proof. Instead of applying the [\[DRS04\]](#) bound ([Lemma B.1](#)) to bound the conditional min-entropy, we work directly with conditional collision probability inside the algebraic LHL ([Lemma B.3](#)).

As in the proof of [Lemma B.4](#), we reduce to a single column via the hybrid argument (Step 1) and consider the per-column problem: show $(\mathbf{A}, \mathbf{A}\mathbf{r}, \mathbf{e}^\top \mathbf{r})$ is close to $(\mathbf{A}, \mathbf{u}, \mathbf{e}^\top \mathbf{r})$.

The algebraic LHL (specifically, the generalization of JLWG25 Lemma 5.1 to include auxiliary information) bounds the statistical distance via:

$$\text{SD}(\mathcal{D}_0, \mathcal{D}_1) \leq \frac{1}{2} \sqrt{\sum_{\mathbf{q} | qR} \mathcal{N}(\mathbf{q})^n \cdot \text{Col}(S_{\mathbf{q}} | \mathbf{e}^\top \mathbf{r}) - 1}.$$

For a general ideal divisor $\mathbf{q} = \mathbf{q}_1^{a_1} \cdots \mathbf{q}_N^{a_N}$ (with $a_i \in \{0, 1\}$ in the unramified NTT-friendly case), the conditional collision probability factors as:

$$\text{Col}(S_{\mathbf{q}} | \mathbf{e}^\top \mathbf{r}) = \text{Col}\left(\mathbf{r} \bmod \prod_i \mathbf{q}_i^{a_i} \mid \mathbf{e}^\top \mathbf{r} \bmod q\right).$$

By submultiplicativity of collision probability (conditioning on additional information can only increase collision probability, but the product over independent CRT components is bounded by the product of per-component bounds):

$$\text{Col}(S_{\mathbf{q}} | \mathbf{e}^\top \mathbf{r}) \leq \prod_{i: a_i=1} \text{Col}(S_{\mathbf{q}_i} | \mathbf{e}^\top \mathbf{r}).$$

Applying [Assumption B.2](#) to each factor:

$$\text{Col}(S_{\mathbf{q}_i} | \mathbf{e}^\top \mathbf{r}) \leq q \cdot \text{Col}(S_{\mathbf{q}_i}).$$

Substituting into the sum:

$$\begin{aligned} & \sum_{\mathbf{q} | qR} \mathcal{N}(\mathbf{q})^n \cdot \text{Col}(S_{\mathbf{q}} | \mathbf{e}^\top \mathbf{r}) \\ & \leq \sum_{(a_1, \dots, a_N) \in \{0, 1\}^N} \prod_{i=1}^N q^{na_i} \cdot (q \cdot \text{Col}(S_{\mathbf{q}_i}))^{a_i} \\ & = \prod_{i=1}^N (1 + q^{n+1} \cdot \text{Col}(S_{\mathbf{q}_i})). \end{aligned}$$

This is the same product structure as in JLWG25 Theorem 5.3, but with q^n replaced by q^{n+1} . By JLWG25 Corollary 4, $\text{Col}(S_{\mathbf{q}_i}) \leq 2^{-H_\infty(S_{\mathbf{q}_i})} \leq 2\sigma^{-m}$, so each factor is $1 + 2q^{n+1}\sigma^{-m}$. Setting $\theta = q^{n+1}/\sigma^m$ and following JLWG25's analysis (their proof of Theorem 5.3), we obtain

$$\text{SD}(\mathcal{D}_0, \mathcal{D}_1) \leq \sqrt{N} \cdot \theta \leq \varepsilon/k$$

provided $m \log \sigma \geq 2 \log(k/\varepsilon) + (n+1) \log q + \log g + m$. Summing over k columns via the hybrid argument gives the final bound. $\square$

Remark B.6. With the improved condition, the binding constraint on m becomes $m \gtrsim 2(n+1)$ (substituting $\log \sigma \approx \frac{1}{2} \log q$), which is independent of the ring dimension N . For the application to Ring-RTLWE with $n = L$, this gives $\hat{m} \geq 2 + 2/L$, so $\hat{m} = 3$ suffices for any $L \geq 2$. Consequently, the Gaussian width χ_R can be a constant (independent of N and L), and the modulus q remains polynomial in λ and L regardless of the ring dimension.

B.3 Ring Preimage Sampling

In this subsection, we present the ring analogs of the preimage sampling and delegation lemmas.

Lemma B.5 (Ring Preimage Sampling). *Let λ be a security parameter, let $N = N(\lambda)$ be a power of two, and let $q = q(\lambda)$ be an NTT-friendly prime. Let $m_0 = m_0(N, q) = O(\log q)$ be as in [Theorem B.1](#), and let $\tau = O(\sqrt{N \log q})$ be the corresponding trapdoor quality. Suppose $m \geq m_0$ and*

$$\chi \geq \tau \cdot \omega(\sqrt{\log N}).$$

Then the distributions

$$(\mathbf{a}, \mathbf{x}, \mathbf{a} \cdot \mathbf{x}) \quad \text{and} \quad (\mathbf{a}, \mathbf{a}_\chi^{-1}(u), u)$$

are statistically indistinguishable, where $\mathbf{a} \leftarrow R_q^{1 \times m}$, $\mathbf{x} \leftarrow D_{\mathcal{R}, \chi}^{\text{coeff}, m}$, and $u \leftarrow R_q$.

Lemma B.6 (Ring Preimage Delegation). *Let λ, N, q, m_0, τ be as in [Lemma B.5](#). Let $\ell_1, \ell_2 = \text{poly}(\lambda)$, suppose $m \geq m_0$, and $\chi \geq \tau \cdot \omega(\sqrt{\log N})$ (with the τ defined in [Theorem B.1](#)). Then the distributions*

$$(\mathbf{a}, \{\mathbf{b}_i, \mathbf{X}_i, \mathbf{Y}_i\}_{i \in [\ell_1]}) \quad \text{and} \quad (\mathbf{a}, \{\mathbf{b}_i, \hat{\mathbf{X}}_i, \hat{\mathbf{Y}}_i\}_{i \in [\ell_1]})$$

are statistically indistinguishable, where

$$\begin{aligned} \mathbf{a} &\leftarrow R_q^{1 \times m}, & \mathbf{b}_i &\leftarrow R_q^{1 \times m}, \\ \mathbf{X}_i &\leftarrow D_{\mathcal{R}, \chi}^{\text{coeff}, 2m \times \ell_2}, & \mathbf{Y}_i &= [\mathbf{a} \mid \mathbf{b}_i] \cdot \mathbf{X}_i, \\ \hat{\mathbf{Y}}_i &\leftarrow R_q^{1 \times \ell_2}, & \hat{\mathbf{X}}_i &\leftarrow [\mathbf{a} \mid \mathbf{b}_i]_\chi^{-1}(\hat{\mathbf{Y}}_i). \end{aligned}$$

Proof. By a simple hybrid argument over the ℓ_2 columns of $\mathbf{X}$ (each column is independent), [Lemma B.5](#) implies that

$$(\hat{\mathbf{a}}, \mathbf{X}, \hat{\mathbf{a}} \cdot \mathbf{X}) \quad \text{and} \quad (\hat{\mathbf{a}}, \hat{\mathbf{a}}_\chi^{-1}(\mathbf{Y}), \mathbf{Y})$$

are statistically indistinguishable, where $\hat{\mathbf{a}} \leftarrow R_q^{1 \times \hat{m}}$, $\mathbf{X} \leftarrow D_{\mathcal{R}, \chi}^{\text{coeff}, \hat{m} \times \ell_2}$, $\mathbf{Y} \leftarrow R_q^{1 \times \ell_2}$, and $\hat{m} \geq 2m_0$.

For each $i \in [\ell_1]$, set $\hat{\mathbf{a}} = [\mathbf{a} \mid \mathbf{b}_i]$ (so $\hat{m} = 2m \geq 2m_0$), $\mathbf{X} = \mathbf{X}_i$, and $\mathbf{Y} = \hat{\mathbf{Y}}_i$. Note that the marginal distribution of $[\mathbf{a} \mid \mathbf{b}_i]$ is uniform over $R_q^{1 \times 2m}$, matching the distribution of $\hat{\mathbf{a}}$. Applying the indistinguishability ℓ_1 times via a hybrid argument yields the lemma. $\square$

B.4 Ring Gadget Matrix and Ring Trapdoors

In this subsection, we introduce the ring analogs of the gadget matrix and trapdoor sampling tools from [\[MP12, WWW22, Pei10, EBB14, SSTX09\]](#), which we will use in subsequent constructions. We focus on the power-of-two cyclotomic setting introduced in [Section B.1](#) and work throughout in the coefficient embedding.

The Ring Gadget Vector. In the integer setting, the gadget matrix is $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}^\top$ where $\mathbf{g}^\top = [1, 2, 4, \dots, 2^{w-1}]$ and $w = \lceil \log q \rceil$. The factor $\mathbf{I}_n$ is needed to give the matrix the right “per-dimension” structure for an n -dimensional integer LWE secret. In the ring setting, the secret is a single ring element, which already carries N scalars-worth of structure under the coefficient embedding, so the analogous gadget object is much smaller.

Definition B.1 (Ring Gadget Vector). Let $w := \lceil \log q \rceil$. The *ring gadget vector* is

$$\mathbf{g} := [1, 2, 4, \dots, 2^{w-1}] \in R_q^{1 \times w},$$

where each entry is the corresponding integer viewed as a constant ring element (i.e., a polynomial whose constant coefficient is the indicated integer and whose other coefficients are zero).

The corresponding inverse function decomposes any ring element into a vector of w binary ring elements:

Definition B.2 (Ring Gadget Decomposition). The function $\mathbf{g}^{-1} : R_q \rightarrow \mathcal{R}^w$ takes a ring element $a \in R_q$ and returns the vector $(a^{(0)}, a^{(1)}, \dots, a^{(w-1)}) \in \mathcal{R}^w$ where $a^{(i)} \in \mathcal{R}$ has each coefficient equal to the i -th bit of the corresponding coefficient of a . By construction:

1. $\mathbf{g} \cdot \mathbf{g}^{-1}(a) = a$ for all $a \in R_q$.

2. $\|g^{-1}(a)\|_{\text{coeff}} \leq 1$ for all $a \in R_q$.

We extend g^{-1} to matrices over R_q entrywise: for $H \in R_q^{k \times t}$, $g^{-1}(H) \in \mathcal{R}^{kw \times t}$ is obtained by replacing each entry of H with its w -element decomposition stacked vertically.

Ring Trapdoor Sampling. We now state the ring analog of trapdoor sampling, following the framework of [MP12] as adapted to the ring setting (briefly outlined in MP12 [MP12] itself, with explicit parameters in subsequent works [EBB14, SSTX09]).

Theorem B.1 (Ring Trapdoor Generation and Preimage Sampling). *Let N be a power of two, $\mathcal{R} = \mathbb{Z}[x]/(x^N + 1)$, and q an NTT-friendly prime. There exist efficient algorithms (RingTrapGen, RingSamplePre) with the following syntax:*

- RingTrapGen($1^N, q, m$) $\rightarrow (\mathbf{a}, \text{td}_{\mathbf{a}})$: On input the ring dimension N , modulus q , and number of samples m , outputs a row vector $\mathbf{a} \in R_q^{1 \times m}$ together with a trapdoor $\text{td}_{\mathbf{a}}$.
- RingSamplePre($\mathbf{a}, \text{td}_{\mathbf{a}}, u, s$) $\rightarrow \mathbf{x}$: On input a row vector $\mathbf{a}$, a trapdoor $\text{td}_{\mathbf{a}}$, a target $u \in R_q$, and a Gaussian width s , outputs a column vector $\mathbf{x} \in \mathcal{R}^m$.

Moreover, there exists a polynomial $m_0 = m_0(N, q) = O(\log(Nq))$ such that for all $m \geq m_0$, the algorithms satisfy the following properties:

- (**Trapdoor distribution.**) The vector $\mathbf{a}$ output by RingTrapGen is statistically close to uniform over $R_q^{1 \times m}$.
- (**Trapdoor quality.**) The trapdoor $\text{td}_{\mathbf{a}}$ is a τ -trapdoor for some $\tau = O(\sqrt{N \log q})$.
- (**Preimage sampling.**) For any target $u \in R_q$ and any $s \geq \tau \cdot \omega(\sqrt{\log N})$, the distribution of RingSamplePre($\mathbf{a}, \text{td}_{\mathbf{a}}, u, s$) is statistically close to the discrete Gaussian distribution $D_{\mathcal{R}_m, s}^{\text{coeff}}$ conditioned on $\mathbf{a} \cdot \mathbf{x} = u \bmod q$.

For a row vector $\mathbf{a} \in R_q^{1 \times m}$ and a target $u \in R_q$, we write $\mathbf{a}_s^{-1}(u)$ for the random variable distributed as the discrete Gaussian preimage above. We extend this to matrix targets entrywise.

Ring Gadget Trapdoors. The key property of the gadget vector is that it admits a trivial public trapdoor (since one can directly bit-decompose), and more importantly, this trapdoor can be transferred to any vector that is related to $\mathbf{g}$ via a small linear transformation. The following is the ring analog of [WWW22, Theorem 3.11].

Theorem B.2 (Ring Gadget Trapdoor). *The ring gadget vector $\mathbf{g} \in R_q^{1 \times w}$ has a public τ -trapdoor with $\tau = O(1)$ (given by direct gadget decomposition).*

Moreover, suppose $\mathbf{a} \cdot \mathbf{R} = h \cdot \mathbf{g}$ for some row vector $\mathbf{a} \in R_q^{1 \times m_0}$, matrix $\mathbf{R} \in \mathcal{R}^{m_0 \times w}$, and an invertible element $h \in R_q$. Then $(\mathbf{R}, h)$ can be used as a τ -trapdoor for $\mathbf{a}$ in the sense of [MP12, Definition 5.2], where $\tau = s_1(\mathbf{R})$ is the largest singular value of $\mathbf{R}$ (viewed as a $\mathbb{Z}$ -linear map by expanding each ring element into its N coefficients in the coefficient embedding). It satisfies Theorem B.1 in the following two ways:

- Under the Ring Leftover Hash Lemma, the distribution of $\mathbf{a}$ is statistically close to uniform over $R_q^{1 \times m_0}$ for $m_0 = 2w + \omega(\log N)$. [MP12, EBB14, SSTX09]
- Under the Ring-LWE assumption, the distribution of $\mathbf{a}$ is computationally indistinguishable from uniform over $R_q^{1 \times m_0}$ for $m_0 = w + 2$. [MP12]

Remark B.7 (Which instantiation we use).

All constructions in this paper use the *statistical* instantiation, i.e. $m_0 = 2w + \omega(\log N)$, and we set $m = 2k + \omega(\log N)$ with $k = \lceil \log q \rceil$ in Section 5.3 and Section 5.4 accordingly. The computational instantiation ($m_0 = w + 2$) is cheaper by roughly a factor of two, but it is not sound in our setting for two reasons. First, in the security proofs of Theorem 5.3 and Theorem 5.5 the hybrid that replaces $\mathbf{a}$ by a uniform vector precedes the invocation of Ring-LWE (resp. RTRLWE) on the challenge ciphertext, and the intervening hybrids are statistical—they invoke Lemma B.4, preimage sampling, and, in the CP-ABE case, an inefficient sampling step that is only later simulated by the RTRLWE oracle. A $\mathbf{a}$ that is merely computationally uniform does not compose with these steps. Second, under the computational instantiation $\mathbf{a}$ retains a fixed public structure, so the ciphertext component $s[\mathbf{a} \mid D] + \mathbf{e}^\top$ leaks the high-order bits of the secret s , since the noise $\mathbf{e}$ masks only the low-order bits.

Remark B.8. **Theorem B.2** defines the trapdoor quality τ as the singular value of $\mathbf{R}$ viewed as an integer matrix (via the coefficient embedding), matching [MP12, Definition 5.2] directly. The ring structure enters only through the fact that a ring matrix $\mathbf{R} \in \mathcal{R}^{p \times q}$ expands to an integer matrix of dimensions $pN \times qN$, so “dimensions” become larger by a factor of N when relating singular values to entrywise norms. In particular, the standard bound $s_1(\mathbf{M}) \leq \sqrt{\text{rows} \cdot \text{cols}} \cdot \|\mathbf{M}\|_{\max}$ applied to a ring matrix picks up an extra factor of N compared to the integer setting because the integer dimensions have an N factor in each direction. This is the mechanism by which “ring-multiplication-related” N factors appear in quality bounds under the coefficient embedding. Working in the canonical embedding instead reduces these N factors to $\sqrt{N}$, at the cost of additional notational overhead.

The ring analog of [WWW22] Corollary 3.12 follows by the same argument as in the integer setting, with the following minor modifications:

- The gadget matrix $\mathbf{G}$ is replaced by the ring gadget vector $\mathbf{g}$, and “ $\mathbf{I}_n$ ” factors collapse since the secret is a ring element rather than an n -dimensional vector.
- The notion of “full rank” for matrices over R_q is interpreted as “invertible modulo every prime ideal factor $\mathfrak{q}_i$ of $q\mathcal{R}$.” Since R_q is not a field (it is a product of N copies of $\mathbb{F}_q$ via the Chinese Remainder Theorem when q is NTT-friendly), this is a stronger condition than invertibility over $\mathbb{Z}_q$ alone.
- Norms are tracked in the coefficient embedding, and ring multiplication picks up the N factor noted in **Remark B.8**.

Corollary B.2 (Ring Gadget Trapdoor for Tensored Targets). *Let $\mathbf{H} \in R_q^{k \times t}$ be a matrix that is full row rank in the ring sense, i.e., $\mathbf{H}$ has full row rank modulo every prime ideal factor $\mathfrak{q}_i$ of $q\mathcal{R}$ (equivalently, the rows of $\mathbf{H}$ are linearly independent over each $\mathbb{F}_q$ obtained by reducing modulo $\mathfrak{q}_i$). Suppose*

$$\mathbf{A} \cdot \mathbf{R} = \mathbf{H} \otimes \mathbf{g}$$

for some $\mathbf{A} \in R_q^{k \times m_0}$ and $\mathbf{R} \in \mathcal{R}^{m_0 \times tw}$ sampled from $D_{\mathcal{R}, \chi_R}^{\text{coeff}, m_0 \times tw}$, where $\mathbf{H} \otimes \mathbf{g} \in R_q^{k \times tw}$ denotes the matrix obtained by replacing each entry h_{ij} of $\mathbf{H}$ with the w -vector $h_{ij} \cdot \mathbf{g}$. Then $(\mathbf{R}, \mathbf{H})$ can be used as a τ -trapdoor for $\mathbf{A}$, with quality

$$\tau = O(\chi_R tw N (\sqrt{m_0 N} + \sqrt{tw N} + \sqrt{\lambda}))$$

with probability $1 - 2^{-\Omega(\lambda)}$ over the choice of $\mathbf{R}$.

Proof. We follow the structure of the proof of [WWW22, Corollary 3.12], adapted to the ring setting. The key idea is to construct an explicit small matrix $\bar{\mathbf{R}}$ such that $\mathbf{A} \cdot \bar{\mathbf{R}} = \mathbf{I}_k \otimes \mathbf{g}$ (the standard ring gadget structure for k rows). Once this is done, the pair $(\bar{\mathbf{R}}, \mathbf{I}_k)$ serves as a trapdoor for $\mathbf{A}$ in the standard sense of **Theorem B.2**, and the quality bound on $\bar{\mathbf{R}}$ propagates to the claimed bound on $(\mathbf{R}, \mathbf{H})$.

Step 1: Factor the tensored gadget.

First, observe that we can write

$$\mathbf{H} \otimes \mathbf{g} = \mathbf{H} \cdot (\mathbf{I}_t \otimes \mathbf{g}),$$

where $\mathbf{I}_t \otimes \mathbf{g} \in R_q^{t \times tw}$ is the block-diagonal matrix whose i -th row is $\mathbf{g}$ in columns $(i-1)w+1$ through iw and zero elsewhere. This is a direct algebraic identity: each entry h_{ij} of $\mathbf{H}$ “picks out” the j -th block of w columns and scales it by h_{ij} , and summing over j gives exactly the w -vector $\sum_j h_{ij} \mathbf{g}$, which is the i -th row of $\mathbf{H} \otimes \mathbf{g}$.

Step 2: Construct a right inverse for $\mathbf{H}$.

Since $\mathbf{H}$ has full row rank modulo every prime ideal factor $\mathfrak{q}_i$ of $q\mathcal{R}$, by the Chinese Remainder Theorem we can construct a right inverse $\mathbf{H}^* \in R_q^{t \times k}$ with $\mathbf{H}\mathbf{H}^* = \mathbf{I}_k$ as follows. For each $i \in [N]$ (recalling $g = N$ in the NTT-friendly setting), reduce $\mathbf{H}$ modulo $\mathfrak{q}_i$ to get $\mathbf{H}^{(i)} \in \mathbb{F}_q^{k \times t}$, which is full row rank by hypothesis. Choose any right inverse $(\mathbf{H}^{(i)})^* \in \mathbb{F}_q^{t \times k}$ with $\mathbf{H}^{(i)}(\mathbf{H}^{(i)})^* = \mathbf{I}_k$ (which exists by linear algebra over the field $\mathbb{F}_q$, since $k \leq t$). Combine these via CRT to obtain $\mathbf{H}^* \in R_q^{t \times k}$ with $\mathbf{H}^* \bmod \mathfrak{q}_i = (\mathbf{H}^{(i)})^*$ for each i . Then $\mathbf{H}\mathbf{H}^* = \mathbf{I}_k$ in $R_q^{k \times k}$, since this equation holds modulo every $\mathfrak{q}_i$.

Step 3: Construct the small trapdoor matrix $\bar{\mathbf{R}}$.

Define

$$\bar{\mathbf{R}} := \mathbf{R} \cdot \mathbf{g}^{-1}(\mathbf{H}^* \otimes \mathbf{g}) \in \mathcal{R}^{m_0 \times kw},$$

where $\mathbf{g}^{-1}$ is applied entrywise: $\mathbf{H}^* \otimes \mathbf{g} \in R_q^{t \times kw}$ is the tensored target, and $\mathbf{g}^{-1}(\mathbf{H}^* \otimes \mathbf{g}) \in \mathcal{R}^{tw \times kw}$ is its gadget decomposition. Note that the entries of $\mathbf{g}^{-1}(\mathbf{H}^* \otimes \mathbf{g})$ have coefficient norm at most 1 by **Definition B.2**.

We compute:

$$\begin{aligned}
\mathbf{A} \cdot \bar{\mathbf{R}} &= \mathbf{A} \cdot \mathbf{R} \cdot \mathbf{g}^{-1}(\mathbf{H}^* \otimes \mathbf{g}) \\
&= (\mathbf{H} \otimes \mathbf{g}) \cdot \mathbf{g}^{-1}(\mathbf{H}^* \otimes \mathbf{g}) && \text{(by hypothesis)} \\
&= \mathbf{H} \cdot (\mathbf{I}_t \otimes \mathbf{g}) \cdot \mathbf{g}^{-1}(\mathbf{H}^* \otimes \mathbf{g}) && \text{(by Step 1)} \\
&= \mathbf{H} \cdot (\mathbf{H}^* \otimes \mathbf{g}) && \text{(since } (\mathbf{I}_t \otimes \mathbf{g}) \cdot \mathbf{g}^{-1}(\cdot) = (\cdot) \text{)} \\
&= (\mathbf{H}\mathbf{H}^*) \otimes \mathbf{g} && \text{(distributivity of } \otimes \text{ over matrix product)} \\
&= \mathbf{I}_k \otimes \mathbf{g}. && \text{(by Step 2)}
\end{aligned}$$

So $\mathbf{A} \cdot \bar{\mathbf{R}} = \mathbf{I}_k \otimes \mathbf{g}$, which is exactly the form of a standard ring gadget trapdoor relation. Applying [Theorem B.2](#) (with $h = 1$ playing the role of the invertible element, applied to each of the k rows of $\mathbf{A}$ independently), the pair $(\bar{\mathbf{R}}, \mathbf{I}_k)$ is a valid trapdoor for $\mathbf{A}$.

Step 4: Bound the trapdoor quality.

By [Theorem B.2](#), the quality of the trapdoor $(\bar{\mathbf{R}}, \mathbf{I}_k)$ is $\tau = s_1(\bar{\mathbf{R}})$, the largest singular value of $\bar{\mathbf{R}}$ (viewed as a $\mathbb{Z}$ -linear map $\mathbb{Z}^{kwN} \rightarrow \mathbb{Z}^{m_0N}$ by expanding each ring element into its N coefficients). We bound $s_1(\bar{\mathbf{R}})$ via submultiplicativity:

$$s_1(\bar{\mathbf{R}}) = s_1(\mathbf{R} \cdot \mathbf{g}^{-1}(\mathbf{H}^* \otimes \mathbf{g})) \leq s_1(\mathbf{R}) \cdot s_1(\mathbf{g}^{-1}(\mathbf{H}^* \otimes \mathbf{g})).$$

We bound the two factors separately.

Bounding $s_1(\mathbf{R})$ via MP12 Lemma 2.9. Since $\mathbf{R}$ is sampled from $D_{\mathcal{R}, \chi_R}^{\text{coeff}, m_0 \times tw}$, each of its $m_0 \cdot tw \cdot N$ scalar coefficients is independently drawn from $D_{\mathbb{Z}, \chi_R}$, which is 0-subgaussian with parameter χ_R . Applying [\[MP12, Lemma 2.9\]](#) to $\mathbf{R}$ viewed as an integer matrix of dimensions $(m_0N) \times (twN)$:

$$s_1(\mathbf{R}) \leq C \cdot \chi_R \cdot \left(\sqrt{m_0N} + \sqrt{twN} + \sqrt{\lambda} \right)$$

with probability $1 - 2^{-\Omega(\lambda)}$, for some universal constant $C > 0$.

Bounding $s_1(\mathbf{g}^{-1}(\mathbf{H}^* \otimes \mathbf{g}))$ via the entrywise norm. The matrix $\mathbf{g}^{-1}(\mathbf{H}^* \otimes \mathbf{g})$ is a deterministic matrix over $\mathcal{R}$ of ring dimensions $tw \times kw$ (equivalently, integer dimensions $(twN) \times (kwN)$ after expansion). By [Definition B.2](#), each of its scalar coefficients is at most 1 in absolute value. Using the standard bound $s_1(\mathbf{M}) \leq \|\mathbf{M}\|_F \leq \sqrt{p \cdot q} \cdot \|\mathbf{M}\|_{\max}$ for an integer matrix of dimensions $p \times q$ and using $k \leq t$:

$$s_1(\mathbf{g}^{-1}(\mathbf{H}^* \otimes \mathbf{g})) \leq \sqrt{(twN)(kwN)} \cdot 1 = wN\sqrt{tk} \leq wNt.$$

Combining. Multiplying the two bounds:

$$\tau = s_1(\bar{\mathbf{R}}) \leq O(\chi_R(\sqrt{m_0N} + \sqrt{twN} + \sqrt{\lambda})) \cdot twN.$$

Simplifying, we get that

$$\tau = O(\chi_R twN(\sqrt{m_0N} + \sqrt{twN} + \sqrt{\lambda})).$$

Note in particular that the dependence on N is $N^{3/2}$: one factor of $\sqrt{N}$ comes from the Gaussian concentration on $\mathbf{R}$ (via MP12 Lemma 2.9), and one factor of N comes from the entrywise-to-operator-norm bound on the deterministic matrix $\mathbf{g}^{-1}(\mathbf{H}^* \otimes \mathbf{g})$ (where the expansion from the ring dimension adds a factor of N to the integer-matrix dimensions). $\square$

Remark B.9. The “full rank in the ring sense” condition in [Corollary B.2](#) is more restrictive than the integer-setting condition of [\[WWW22, Corollary 3.12\]](#). In the integer setting, $\mathbb{Z}_q$ is a field (when q is prime), so “full rank” is unambiguous. In the ring setting with NTT-friendly q , $R_q \cong \mathbb{F}_q^N$ via CRT, and a matrix over R_q is full rank if and only if it is full rank in each of the N component fields simultaneously. Matrices that are full rank over $\mathbb{Z}_q$ may fail to be full rank over R_q if their entries happen to lie in a common prime ideal factor.

For applications where $\mathbf{H}$ has $\{0, 1\}$ -valued entries (i.e., is an integer matrix lifted to R_q), the ring full rank condition coincides with the integer full rank condition over $\mathbb{Z}$, since integer-valued matrices behave uniformly across all the $\mathbb{F}_q$ components.

B.5 Ring Generalized Related-Trapdoor LWE Assumption

In this subsection, we state the ring analog of the generalized related-trapdoor LWE assumption introduced by Waters, Wee, and Wu [WWW22, Assumption 4.1]. This is an intermediate assumption used as a building block for ring-based ABE constructions, and it will be shown (in a subsequent subsection) to be implied by Ring-LWE via a hybrid argument that uses the tools developed in Section B.2 and Section B.3.

Setup and Notation. Throughout this subsection, $\mathcal{R}$ is the $2N$ -th cyclotomic ring of integers, q is an NTT-friendly prime (so $q\mathcal{R}$ factors into N prime ideals of norm q), and all vectors and matrices are over $\mathcal{R}$ or R_q . We use the row-vector convention: for a matrix $\mathbf{B} \in R_q^{L \times c}$ and a vector $\mathbf{u} \in R_q^L$, the product $\mathbf{u}^\top \mathbf{B} \in R_q^c$ is the standard left-multiplication.

We write $\mathbf{g} \in R_q^{1 \times w}$ for the ring gadget vector (Definition B.1) with $w = \lceil \log q \rceil$. For a vector $\mathbf{u} \in \mathcal{R}^L$, we write $\mathbf{U}^\perp \in \mathbb{Z}^{L \times (L-1)}$ for a fixed choice of basis for the kernel of $\mathbf{u}^\top$ as a $\mathbb{Z}$ -linear map $\mathbb{Z}^L \rightarrow \mathbb{Z}$ — that is, an integer matrix whose columns span $\{\mathbf{v} \in \mathbb{Z}^L : \mathbf{u}^\top \mathbf{v} = 0\}$ and which has bounded coefficient norm $\|\mathbf{U}^\perp\|_{\text{coeff}} \leq 1$. Such a basis exists whenever $\mathbf{u} \in \{0, 1\}^L$ with at least one nonzero entry; for example, when $\mathbf{u} = (1, 1, \dots, 1)$, the matrix

$$\mathbf{U}^\perp = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ -1 & 0 & \cdots & 0 \\ 0 & -1 & \cdots & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & -1 \end{pmatrix} \in \{-1, 0, 1\}^{L \times (L-1)}$$

is one such choice.

The Assumption. The assumption is presented in Assumption 5.1.

Comparison with [WWW22] Assumption 4.1. Assumption 5.1 is the direct ring analog of [WWW22, Assumption 4.1], with the following changes:

- *Ring rank.* The “integer LWE dimension” n of [WWW22] is collapsed into the ring structure: the LWE secret is a single ring element $s \in R_q$ (ring rank 1) instead of an n -dimensional vector. Consequently, $\mathbf{a}$ is a vector of m ring elements (rather than an $n \times m$ matrix of integer LWE samples), and $\mathbf{B}$ has L ring rows (rather than nL integer rows).
- *Distribution of $\mathbf{R}$.* In [WWW22], $\mathbf{R}$ is sampled uniformly from $\{-1, 1\}^{\hat{m}L \times \hat{m}(L-1)}$ with $\|\mathbf{R}\| = 1$. In the ring setting, $\mathbf{R}$ is sampled from a discrete Gaussian $D_{\mathcal{R}, \chi_R}^{\text{coeff}}$ at each entry. This is forced by the Ring-LHL of Section B.2, which applies to Gaussian (not uniform-binary) distributions. As a consequence, the trapdoor quality bound in the ring setting has an additional χ_R factor compared to [WWW22], which propagates to the required noise width χ .
- *Rank condition on $\overline{\mathbf{M}}$.* In [WWW22], “full rank” over $\mathbb{Z}_q$ is unambiguous since $\mathbb{Z}_q$ is a field. In the ring setting, R_q is a product of N copies of $\mathbb{F}_q$ via CRT, and “full rank in the ring sense” means full rank modulo every prime ideal factor simultaneously (see Remark B.9). For integer-valued $\mathbf{M}$ (e.g., $\{0, 1\}$ -valued matrices arising from Boolean policies), the two notions coincide. The restriction $\mathbf{u} \in \{0, 1\}^L$ (with at least one nonzero entry), which is inherited from [WWW22], also ensures that (i) the kernel basis $\mathbf{U}^\perp$ can be chosen with small coefficient norm (in fact, $\{-1, 0, 1\}$ -valued), and (ii) $\mathbf{u}$ is “totally invertible” in the ring sense (each component of $\mathbf{u}$ is a unit in every $\mathbb{F}_q$ component obtained by reducing modulo a prime ideal factor), which simplifies the linear-algebra argument in the forthcoming reduction.
- *Parameters m and $\hat{m}$.* These play distinct roles: m is the number of “extra” Ring-LWE samples in the challenge (via $\mathbf{a}$), while $\hat{m}$ is the gadget-related width inside $\mathbf{B}$. Both are polynomial in λ , and their values are constrained by the reduction to Ring-LWE (to be stated in the next subsection): roughly, $\hat{m} \geq 2 + 2N/L$ (from Ring-LHL) and the Ring-LWE hardness is invoked with $m + \hat{m}L$ samples.

Remark B.10 (Role of χ_R). The Gaussian width χ_R for sampling $\mathbf{R}$ is a separate parameter from the noise width χ . It is constrained on the upper end by the ceiling of Lemma B.2 ($\chi_R \leq \frac{\sqrt{\pi q}}{2\sqrt{\ln 4}}$) and on the lower end by the entropy condition of Lemma B.4 (condition 2, applied with $n = L$). The choice of χ_R then determines the trapdoor quality bound (via Corollary B.2), which in turn determines the minimum χ needed for the trapdoor-based preimage sampling in the oracle response.

B.6 Reduction of Ring-RTLWE to Ring-LWE

In this subsection, we prove the main theorem of this section: the Ring Generalized Related-Trapdoor LWE assumption ([Assumption 5.1](#)) is implied by the standard Ring-LWE assumption ([Assumption B.1](#)), via a hybrid argument that mirrors the integer-setting proof of [[WWW22](#), Theorem 4.2].

Main Theorem.

Theorem B.3 (Ring-RTLWE from Ring-LWE). *Let λ be a security parameter and let $N, q, m, \hat{m}, L, \chi, \chi_R$ be lattice parameters as in [Assumption 5.1](#). Suppose the following conditions hold:*

1. (Ring-LHL width ceiling.) $\chi_R \leq \frac{\sqrt{\pi}}{2} \cdot \frac{\sqrt{q}}{\sqrt{\ln 4}}$.
2. (Ring-LHL entropy condition.) *One of the following:*
 - (Unconditional.) $\hat{m}L \log \chi_R \geq 2 \log(\hat{m}(L-1)/\varepsilon) + (L+N) \log q + \log g + \hat{m}L$, where ε is any negligible function; or
 - (Conditional on [Assumption B.2](#).) $\hat{m}L \log \chi_R \geq 2 \log(\hat{m}(L-1)/\varepsilon) + (L+1) \log q + \log g + \hat{m}L$.
3. (Trapdoor quality condition.) $\chi \geq \tau \cdot \omega(\sqrt{\log N})$, where $\tau = O(\chi_R \cdot N^{3/2} \cdot \hat{m}^{3/2} \cdot L^{3/2})$ is the trapdoor quality from [Corollary B.2](#).
4. (Gaussian noise.) $\chi \geq \omega(\sqrt{\log N})$.

Then, if the Ring-LWE assumption $\text{RLWE}_{N,m+\hat{m}L,q,\chi}$ holds, the assumption $\text{RTRLWE}_{N,m,\hat{m},q,\chi,\chi_R,L}$ also holds.

Remark B.11 (Parameter regimes). The two variants of condition 2 yield different parameter regimes:

- *Unconditional.* The binding constraint is $\hat{m} \geq 2 + 2N/L$, which requires $\hat{m} = \Theta(N/L)$ when $N \gg L$. This imposes $\chi_R = 2^{O(N/L)}$, and polynomiality of all parameters requires $L = \Omega(N)$.
- *Conditional on [Assumption B.2](#).* The binding constraint is $\hat{m} \geq 2 + 2/L$, so $\hat{m} = 3$ suffices for any $L \geq 2$. The Gaussian width χ_R can be a constant independent of N , and all parameters are polynomial in λ and L without restriction.

Hybrid Sequence. Fix an adversary $\mathcal{A}$ for $\text{RTRLWE}_{N,m,\hat{m},q,\chi,\chi_R,L}$. We define a sequence of hybrid experiments $H_0, H_1, \dots, H_5$ such that H_0 is the $b = 0$ (real) experiment and H_5 is the $b = 1$ (random) experiment. Consecutive hybrids are shown to be indistinguishable, either statistically or under the Ring-LWE assumption, establishing the theorem.

- H_0 (real): This is the $b = 0$ experiment of [Assumption 5.1](#). The challenger samples $\mathbf{a} \xleftarrow{\$} R_q^m$, $\mathbf{B} \xleftarrow{\$} R_q^{L \times \hat{m}(2L-1)}$, $\mathbf{s} \xleftarrow{\$} R_q$, $\mathbf{R} \leftarrow D_{\mathbf{R}, \chi_R}^{\text{coeff}, \hat{m}L \times \hat{m}(L-1)}$, and Gaussian error terms $\mathbf{e}_1, \hat{\mathbf{e}}_0$. The challenge is $(\mathbf{a}, \mathbf{B}, \mathbf{s}\mathbf{a} + \mathbf{e}_1, \mathbf{s}^\top \mathbf{u}^\top \mathbf{B} + \hat{\mathbf{e}}_0^\top [\mathbf{I} \mid \mathbf{R}])$. Oracle queries are answered by inefficient (statistical) preimage sampling on $\mathbf{M}\mathbf{B}$.
- H_1 : Same as H_0 , except that $\mathbf{B}$ is now sampled with the gadget-structured form

$$\mathbf{B} = [\hat{\mathbf{A}} \mid \hat{\mathbf{A}}\mathbf{R} + \mathbf{U}^\perp \otimes \mathbf{g}],$$

where $\hat{\mathbf{A}} \xleftarrow{\$} R_q^{L \times \hat{m}L}$ and $\mathbf{R}$ is the same Gaussian matrix as in H_0 . Oracle queries are still answered by inefficient preimage sampling.

- H_2 : Same as H_1 , except that:
 - $\hat{\mathbf{A}}$ is sampled by the “solve-for-one-block” trick: let i be the smallest index with $u_i = 1$, sample $\hat{\mathbf{A}}_j \xleftarrow{\$} R_q^{1 \times \hat{m}L}$ uniformly for $j \neq i$, sample $\mathbf{D} \xleftarrow{\$} R_q^{1 \times \hat{m}L}$ uniformly, and set $\hat{\mathbf{A}}_i := \mathbf{D} - \sum_{j \neq i} u_j \hat{\mathbf{A}}_j$. This ensures $\mathbf{u}^\top \hat{\mathbf{A}} = \mathbf{D}$ while keeping $\hat{\mathbf{A}}$ uniformly distributed.
 - Oracle queries are answered *efficiently* using the explicit ring trapdoor $(-\mathbf{R}, \mathbf{I})^\top$ for $\mathbf{M}\mathbf{B}$, via the relation from [Corollary B.2](#).
- H_3 : Same as H_2 , except that the LWE-like components of the challenge are replaced with uniform randomness: $\mathbf{a}$ is replaced with a uniform $\mathbf{v} \xleftarrow{\$} R_q^m$, and $\mathbf{D}$ (the solve-for-one-block quantity used in building $\hat{\mathbf{A}}$) is sampled independently of the LWE secret — equivalently, the quantities $\mathbf{s}\mathbf{a} + \mathbf{e}_1$ and $\mathbf{s}\mathbf{D} + \hat{\mathbf{e}}_0$ are replaced by fresh uniform elements.
- H_4 : Same as H_3 , except that $\hat{\mathbf{A}}$ is sampled uniformly (reversing the solve-for-one-block trick) and oracle queries are answered by inefficient preimage sampling (reversing the efficient trapdoor-based sampling).

- H_5 (random): Same as H_4 , except that $\mathbf{B}$ is sampled uniformly from $R_q^{L \times \hat{m}(2L-1)}$ and the $(\mathbf{u}^\top \mathbf{B}, \mathbf{R})$ -based second component of the challenge is replaced with fresh uniform randomness. This is the $b = 1$ experiment of [Assumption 5.1](#).

The indistinguishability of consecutive hybrids is established by the following lemmas, proved below.

Lemma B.7. H_0 and H_1 are statistically indistinguishable, assuming conditions 1 and 2 of [Theorem B.3](#) hold.

Lemma B.8. H_1 and H_2 are statistically indistinguishable, assuming condition 3 of [Theorem B.3](#) holds.

Lemma B.9. H_2 and H_3 are computationally indistinguishable under the $\text{RLWE}_{N,m+\hat{m}L,q,\chi}$ assumption.

Lemma B.10. H_3 and H_4 are statistically indistinguishable, assuming condition 3 of [Theorem B.3](#) holds.

Lemma B.11. H_4 and H_5 are statistically indistinguishable, assuming conditions 1 and 2 of [Theorem B.3](#) hold.

Proof of [Theorem B.3](#). By [Lemmas B.7](#) to [B.11](#), the hybrid chain

$$H_0 \stackrel{s}{\approx} H_1 \stackrel{s}{\approx} H_2 \stackrel{c}{\approx} H_3 \stackrel{s}{\approx} H_4 \stackrel{s}{\approx} H_5$$

implies that H_0 and H_5 are computationally indistinguishable under $\text{RLWE}_{N,m+\hat{m}L,q,\chi}$. Since H_0 is the $b = 0$ experiment and H_5 is the $b = 1$ experiment of [Assumption 5.1](#), this establishes that $\text{RTRLWE}_{N,m,\hat{m},q,\chi,\chi_R,L}$ holds. $\square$

Proofs of the Hybrid Lemmas.

Proof of [Lemma B.7](#). We apply [Lemma B.4](#) (Ring-LHL with Auxiliary Linear Information) with parameters $n = L$ (the ring rank of $\hat{\mathbf{A}}$), $m = \hat{m}L$ (the number of rows of $\mathbf{R}$), $k = \hat{m}(L-1)$ (the number of columns of $\mathbf{R}$), Gaussian width $\sigma = \chi_R$, and leakage vector $\mathbf{e} = \hat{\mathbf{e}}_0$.

Applying the lemma, the joint distribution

$$(\hat{\mathbf{A}}, \hat{\mathbf{A}}\mathbf{R}, \hat{\mathbf{e}}_0^\top \mathbf{R})$$

is statistically close to

$$(\hat{\mathbf{A}}, \mathbf{U}', \hat{\mathbf{e}}_0^\top \mathbf{R})$$

where $\mathbf{U}' \text{SampleLeft} R_q^{L \times \hat{m}(L-1)}$ is uniform and independent. This holds provided condition 1 (width ceiling on χ_R) and condition 2 are satisfied: in the unconditional variant, we invoke [Lemma B.4](#) directly; in the conditional variant, we invoke [Corollary B.1](#) (which relies on [Assumption B.2](#)).

Now observe that in H_1 , $\mathbf{B} = [\hat{\mathbf{A}} \mid \hat{\mathbf{A}}\mathbf{R} + \mathbf{U}^\perp \otimes \mathbf{g}]$. Applying the statistical closeness above, this is statistically close to the distribution in which $\hat{\mathbf{A}}\mathbf{R}$ is replaced by a uniform $\mathbf{U}'$ (independent of $\hat{\mathbf{A}}$), giving $\mathbf{B} = [\hat{\mathbf{A}} \mid \mathbf{U}' + \mathbf{U}^\perp \otimes \mathbf{g}]$. Since $\mathbf{U}'$ is uniform, so is $\mathbf{U}' + \mathbf{U}^\perp \otimes \mathbf{g}$, and therefore $\mathbf{B}$ is uniform on $R_q^{L \times \hat{m}(2L-1)}$ — matching the distribution in H_0 .

The closeness also applies to the joint distribution with $\hat{\mathbf{e}}_0^\top \mathbf{R}$, which appears in the challenge as the right half of $\hat{\mathbf{e}}^\top = \hat{\mathbf{e}}_0^\top [\mathbf{I} \mid \mathbf{R}]$. All other components of the challenge are independent of $\mathbf{R}$ and $\hat{\mathbf{A}}$ given the conditioning, so the full challenge distributions of H_0 and H_1 are statistically close. Oracle queries depend only on $\mathbf{B}$ and use the same inefficient sampling procedure in both hybrids, so they do not introduce any distinguishing advantage. $\square$

Proof of [Lemma B.8](#). The change from H_1 to H_2 consists of two simultaneous modifications.

Sampling $\hat{\mathbf{A}}$. In H_2 , $\hat{\mathbf{A}}$ is constructed as $\hat{\mathbf{A}}_i = \mathbf{D} - \sum_{j \neq i} u_j \hat{\mathbf{A}}_j$ with $\hat{\mathbf{A}}_j$ and $\mathbf{D}$ uniform and independent for $j \neq i$. Since a uniform-minus-uniform combination is uniform, the marginal distribution of $\hat{\mathbf{A}}$ is identical to that of H_1 . Furthermore, by construction, $\mathbf{u}^\top \hat{\mathbf{A}} = u_i \hat{\mathbf{A}}_i + \sum_{j \neq i} u_j \hat{\mathbf{A}}_j = \hat{\mathbf{A}}_i + \sum_{j \neq i} u_j \hat{\mathbf{A}}_j = \mathbf{D}$ (using $u_i = 1$). Thus the marginal distribution of $(\hat{\mathbf{A}}, \mathbf{u}^\top \hat{\mathbf{A}})$ is the same as the joint distribution $(\hat{\mathbf{A}}, \mathbf{D})$ where both are uniform and independent, matching H_1 .

Oracle queries. We verify that the efficient trapdoor-based preimage sampling in H_2 is statistically close to the inefficient preimage sampling in H_1 . The key observation is the structural identity:

$$\mathbf{M}\mathbf{B} \cdot \begin{pmatrix} -\mathbf{R} \\ \mathbf{I}_{\hat{m}(L-1)} \end{pmatrix} = \mathbf{M} \cdot [\hat{\mathbf{A}}\mathbf{R} + \mathbf{U}^\perp \otimes \mathbf{g}] - \mathbf{M}\hat{\mathbf{A}}\mathbf{R} = \mathbf{M}(\mathbf{U}^\perp \otimes \mathbf{g}) = (\mathbf{M}\mathbf{U}^\perp) \otimes \mathbf{g}.$$

When $\overline{\mathbf{M}} = \begin{pmatrix} \mathbf{M} \\ \mathbf{u}^\top \end{pmatrix}$ is full rank in the ring sense, $\mathbf{M}\mathbf{U}^\perp$ is also full rank in the ring sense (since $\mathbf{U}^\perp$ is a $\mathbb{Z}$ -basis for the kernel of $\mathbf{u}^\top$ and the rank condition behaves the same way over each $\mathbb{F}_q$ component via CRT). By [Corollary B.2](#), the pair $(-\mathbf{R}, \mathbf{M}\mathbf{U}^\perp)$ is a τ -trapdoor for $\mathbf{M}\mathbf{B}$ with $\tau \leq O(\chi_R N^{3/2} \hat{m}^{3/2} L^{3/2})$ (using $k, t = O(L)$ in the corollary statement).

By [Theorem B.1](#), ring preimage sampling (i.e., `RingSamplePre`) with Gaussian width $\chi \geq \tau \cdot \omega(\sqrt{\log N})$ (condition 3 of [Theorem B.3](#)) produces outputs statistically close to the ideal Gaussian preimage distribution. Thus the oracle responses in $\mathbf{H}_2$ are statistically close to those in $\mathbf{H}_1$. $\square$

Proof of [Lemma B.9](#). We construct a reduction $\mathcal{B}$ that uses $\mathcal{A}$'s distinguishing advantage between $\mathbf{H}_2$ and $\mathbf{H}_3$ to distinguish Ring-LWE with $m + \hat{m}L$ samples.

On input a Ring-LWE challenge $(\mathbf{d}, \mathbf{z}) \in R_q^{m+\hat{m}L} \times R_q^{m+\hat{m}L}$ (where $\mathbf{z}$ is either $s\mathbf{d} + \mathbf{e}'$ or uniformly random), $\mathcal{B}$ parses $\mathbf{d} = (\mathbf{a}, \mathbf{D})$ with $\mathbf{a} \in R_q^m$ and $\mathbf{D} \in R_q^{\hat{m}L}$, and correspondingly $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2)$ with $\mathbf{z}_1 \in R_q^m$ and $\mathbf{z}_2 \in R_q^{\hat{m}L}$.

$\mathcal{B}$ samples $\mathbf{R} \leftarrow D_{\mathcal{R}, \chi_R}^{\text{coeff}, \hat{m}L \times \hat{m}(L-1)}$ and $\hat{\mathbf{A}}_j \xleftarrow{\$} R_q^{1 \times \hat{m}L}$ for $j \neq i$ (the standard solve-for-one-block construction), setting $\hat{\mathbf{A}}_i = \mathbf{D} - \sum_{j \neq i} u_j \hat{\mathbf{A}}_j$. It constructs $\mathbf{B} = [\hat{\mathbf{A}} \mid \hat{\mathbf{A}}\mathbf{R} + \mathbf{U}^\perp \otimes \mathbf{g}]$ exactly as in $\mathbf{H}_2$.

$\mathcal{B}$ outputs the challenge $(\mathbf{a}, \mathbf{B}, \mathbf{z}_1, \mathbf{z}_2^\top [\mathbf{I}_{\hat{m}L} \mid \mathbf{R}])$ to $\mathcal{A}$, and answers oracle queries using the trapdoor $(-\mathbf{R}, \mathbf{I})$ just as in $\mathbf{H}_2$ (note that $\mathcal{B}$ knows $\mathbf{R}$ since it sampled it itself).

Simulation analysis. When the Ring-LWE challenge is real ($\mathbf{z} = s\mathbf{d} + \mathbf{e}'$), we have $\mathbf{z}_1 = s\mathbf{a} + \mathbf{e}_1$ (matching the form of $\mathbf{H}_2$'s third component), and

$$\mathbf{z}_2^\top [\mathbf{I} \mid \mathbf{R}] = (s\mathbf{D} + \hat{\mathbf{e}}_0)^\top [\mathbf{I} \mid \mathbf{R}] = s\mathbf{D}^\top [\mathbf{I} \mid \mathbf{R}] + \hat{\mathbf{e}}_0^\top [\mathbf{I} \mid \mathbf{R}].$$

Using $\mathbf{D} = \mathbf{u}^\top \hat{\mathbf{A}}$ and $\mathbf{u}^\top \mathbf{U}^\perp = \mathbf{0}$:

$$\mathbf{u}^\top \mathbf{B} = [\mathbf{u}^\top \hat{\mathbf{A}} \mid \mathbf{u}^\top \hat{\mathbf{A}}\mathbf{R} + \mathbf{u}^\top (\mathbf{U}^\perp \otimes \mathbf{g})] = [\mathbf{D}^\top \mid \mathbf{D}^\top \mathbf{R}],$$

so $\mathbf{D}^\top [\mathbf{I} \mid \mathbf{R}] = \mathbf{u}^\top \mathbf{B}$, and therefore $\mathbf{z}_2^\top [\mathbf{I} \mid \mathbf{R}] = s\mathbf{u}^\top \mathbf{B} + \hat{\mathbf{e}}_0^\top [\mathbf{I} \mid \mathbf{R}]$, exactly matching $\mathbf{H}_2$.

When the Ring-LWE challenge is uniform, $\mathbf{z}_1$ and $\mathbf{z}_2$ are uniform and independent, and the challenge $(\mathbf{a}, \mathbf{B}, \mathbf{z}_1, \mathbf{z}_2^\top [\mathbf{I} \mid \mathbf{R}])$ matches $\mathbf{H}_3$ (where the LWE-like components are replaced with uniform).

Thus $\mathcal{B}$'s distinguishing advantage against Ring-LWE equals $\mathcal{A}$'s advantage between $\mathbf{H}_2$ and $\mathbf{H}_3$. $\square$

Proof of [Lemma B.10](#). Symmetric to the proof of [Lemma B.8](#), applied in reverse. The change from $\mathbf{H}_3$ to $\mathbf{H}_4$ reverses both the solve-for-one-block sampling of $\hat{\mathbf{A}}$ and the efficient trapdoor-based oracle sampling. The former is an exact distributional equivalence (same argument as in [Lemma B.8](#)), and the latter is statistically close by [Corollary B.2](#) and condition 3. $\square$

Proof of [Lemma B.11](#). Symmetric to the proof of [Lemma B.7](#), applied in reverse. The change from $\mathbf{H}_4$ to $\mathbf{H}_5$ reverses the Ring-LHL application, switching $\mathbf{B}$ from the gadget-structured form back to uniform. The hypotheses are identical (conditions 1 and 2). $\square$