Threshold Batched Attribute-Based Encryption from Succinct LWE

Shayeef Murshid[1][0009-0008-7042-7911], Ramprasad Sarkar[2][0000-0002-9907-3319], and Mriganka Mandal[1][0000-0001-8691-4185]

[1] Cryptology and Security Research Unit
R C Bose Centre for Cryptology and Security
Indian Statistical Institute Kolkata, Kolkata, India-700108
{shayeef_r,mriganka}@isical.ac.in
[2] Department of Mathematics, Indian Institute of Information Technology Lucknow, Lucknow 226002, Uttar Pradesh, India
rpsarkar@iiitl.ac.in

Abstract. Batched identity-based encryption (BIBE) enables a single compact key to decrypt ciphertexts associated with a designated batch of identities. Batched attribute-based encryption (B-ABE) extends this functionality to attribute-based access control, while threshold batched attribute-based encryption (TB-ABE) further distributes the decryption capability among multiple parties. Existing constructions of TB-ABE rely on bilinear pairings, motivating post-quantum alternatives. In this work, we give, to the best of our knowledge, the first lattice-based candidate construction of TB-ABE. Our construction combines the succinct polynomial-digest mechanism underlying lattice-based BIBE with key-homomorphic evaluation for arithmetic-circuit policies and linear secret sharing. We prove security under the ℓ-succinct LWE assumption, assuming the required share-uniformity property of the underlying ramp threshold sharing scheme.

Keywords: Attribute-based encryption · Batched encryption · Threshold Cryptography · Lattice-Based Cryptography

1 Introduction

Attribute-based encryption (ABE) provides fine-grained access control by associating ciphertexts with attributes and secret keys with access policies, so that decryption is possible only when the attributes satisfy the policy encoded in the key. Beginning with the fuzzy identity-based encryption framework of Sahai *et al.* [22] and the key-policy formulation of Goyal *et al.* [17], subsequent works developed increasingly expressive constructions supporting rich access structures and general circuits [6,10,16,25]. Lattice-based constructions further provide a route toward post-quantum ABE from variants of the learning with errors (LWE) assumption [6, 16]. Traditional ABE, however, provides persistent authorization since a key for a policy f can decrypt every ciphertext whose attributes satisfy f, regardless of the batch to which the ciphertext belongs. In applications such as auditing, compliance, or selective data release, access may instead need to

be restricted to a dynamically chosen collection of ciphertexts. This motivates cryptographic authorization that is simultaneously policy dependent and batch scoped.

Batched threshold encryption (BTE) was introduced for encrypted-mempool applications. Choudhuri *et al.* [8] studied batch-wide threshold decryption with sublinear communication. Subsequently, Agarwal *et al.* [2] introduced efficiently thresholdizable batched identity-based encryption (BIBE). In BIBE, ciphertexts are associated with identities and batch labels, while a compact key is associated with a selected collection of identities. Decryption requires membership in that collection and agreement of the labels. Further pairing and lattice-based constructions appear in [1, 7, 15, 30]. This chronology distinguishes the original batched threshold-encryption objective from the subsequent identity-based formulation.

Batched IBE, nevertheless, still expresses authorization only through membership. Whether a ciphertext can be decrypted is determined by whether its identity belongs to a designated batch. By contrast, many natural applications require both conditions at once: the ciphertext must belong to a selected batch and its attributes must satisfy an expressive policy. This led to the introduction of batched attribute-based encryption (B-ABE) and its threshold counterpart, threshold batched attribute-based encryption (TB-ABE) [23]. In B-ABE, a single key is bound simultaneously to a public batch digest, an access policy, and a batch label, it decrypts precisely the ciphertexts represented by the batch whose attributes satisfy the policy. In TB-ABE, the corresponding decryption capability is further distributed among several authorities so that sufficiently many parties must cooperate before decryption becomes possible. Existing realizations of these primitives, however, are based on bilinear pairings. Obtaining analogous functionality from lattices is considerably more subtle than replacing the underlying group operations by lattice operations. Recent work of Xiong *et al.* [30] constructs lattice-based batched identity based encryption in the standard model from the succinct LWE assumption. Their construction shows how a batch can be represented by a succinct polynomial digest and how batch membership can be enforced through lattice preimage sampling. This motivates a natural question.

> *Can batched attribute-based encryption be realized from lattices while retaining succinct batch-dependent decryption capabilities and expressive access policies?*

Threshold cryptography [12, 20, 24] distributes decryption capability among multiple parties so that only sufficiently large coalitions can recover the plaintext. Das *et al.* [11] recently introduced attribute-based threshold encryption (ABTE), combining threshold decryption with attribute-based access control, and gave constructions from both bilinear pairings and LWE. Threshold techniques have also been developed for batched IBE. Agarwal *et al.* [2] introduced efficiently thresholdizable batched IBE, Gong *et al.* [15] obtained a pairing-based construction in the plain model, and Boneh *et al.* [7] gave a lattice-based approach using distributed lattice preimage sampling. Existing threshold batched-IBE schemes, however, provide only identity-membership authorization, while directly distributing lattice trapdoors that requires additional machinery for Gaussian

preimage sampling. This motivates threshold batched attribute-based encryption in which the batch remains succinct, decryption shares are bound simultaneously to the batch and an access policy, and threshold functionality is obtained without jointly executing the lattice trapdoor-sampling procedure. This leads to the following question.

Can batched attribute-based encryption from lattices preserve succinctness and threshold decryption capabilities?

1.1 Our Contributions

In this work, we provide an affirmative candidate construction for the above questions. We give a construction of threshold batched attribute-based encryption (TB-ABE) supporting polynomial-size arithmetic-circuit policies and prove selective security from succinct LWE, together with the lattice and secret-sharing properties required by our construction. We give, to the best of our knowledge, the first lattice-based candidate construction of TB-ABE. Ciphertexts are associated with an attribute vector $x \in \{0,1\}^k$ and a batch label tg, while a batch $S \subseteq \{0,1\}^k$ is deterministically compressed into a digest dig_S. A key is associated simultaneously with dig_S, a circuit policy f_λ, and the batch label. The master decryption target is distributed among N parties using a linear ramp-threshold secret-sharing scheme. Fewer than T fully challenge-compatible parties remain in the privacy regime, while strictly more than T valid partial decryption shares suffice for reconstruction.

We prove correctness by jointly bounding the noise from batch membership, homomorphic policy evaluation, lattice preimage sampling, smudging, and threshold reconstruction, showing that suitable parameters keep the total error below the decoding threshold while preserving succinct digests, ciphertexts, and key shares. We then establish selective security under the $2m^2$-succinct LWE assumption together with Gaussian-sampling and ramp-secret-sharing properties. Challenge-compatible shares are handled using share uniformity and syndrome-first Gaussian sampling, while challenge-incompatible queries are simulated using alternative trapdoors. Smudging removes secret-key dependence, after which succinct LWE and output masking hide the challenge message.

1.2 Related Work

Attribute-based encryption (ABE) originates from the fuzzy IBE construction of Sahai and Waters [22] and the key-policy framework of Goyal *et al.* [17]. Lattice-based ABE for increasingly expressive policy classes was developed in [6, 10, 16, 28], with the key-homomorphic evaluation techniques of [6] being particularly relevant to our construction. Batched IBE was introduced by Agarwal, Fernando, and Pinkas [2], followed by pairing-based threshold batched IBE in the plain model [15] and further work on batch and threshold batch encryption [1, 7, 9]. Sarkar *et al.* [23] introduced batched attribute-based encryption (B-ABE) and its threshold variant (TB-ABE), giving pairing-based constructions. In the lattice setting, Xiong *et al.* [30] gave the first batched IBE construction in the standard model from succinct LWE, providing the polynomial-digest and matrix-commitment machinery underlying the batching component of our scheme, while Boneh, Laufer, and Tas [7] developed lattice-based threshold batch IBE.

2 Technical Overview

This section provides a conceptual overview of the main ideas underlying our lattice-based construction of a succinct threshold batched attribute-based encryption scheme (TB-ABE).

2.1 Towards Threshold Batched Attribute-Based Encryption

A threshold batched attribute-based encryption scheme combines batching, predicate based authorization, and distributed decryption. A batch $S \subseteq \{0,1\}^k$ of B attributes is first compressed into a succinct digest $\mathsf{dig} \leftarrow \mathsf{Digest}(\mathsf{mpk}, S)$. Given this digest, an access policy f_λ, and a batch label tg, the key-generation algorithm produces a collection of secret-key shares $\{\mathsf{sk}_i\}_{i\in[N]}$ for a committee of N parties. A ciphertext is encrypted under an attribute x and the label tg. A committee member can form a valid partial decryption only when the supplied batch is consistent with the digest, x belongs to that batch, the ciphertext and key labels agree, and $f_\lambda(x) = 0$. Finally, any set of at least t_{rec} honest partial decryptions reconstructs the message, whereas at most t_{priv} fully challenge-compatible key shares are permitted in our security experiment. The algorithms comprising the TB-ABE syntax are as follows.

- **Setup**. On input the security parameter, batch size B, threshold T, committee size N, attribute length k, and message length h, outputs a master public key mpk and master secret key msk.
- **Enc**. On input mpk, a message $\mu \in \{0,1\}^h$, an attribute $x \in \{0,1\}^k$, and a batch label tg, outputs a ciphertext ct.
- **Digest**. Deterministically compresses a batch $S \subseteq \{0,1\}^k$ of size B into a succinct digest dig.
- **KeyGen**. On input msk, a digest dig, a policy f_λ, and a batch label tg, outputs secret-key shares $\{\mathsf{sk}_i\}_{i\in[N]}$.
- **PDec**. Given a secret-key share, the corresponding batch S, and a ciphertext, outputs a partial decryption share d_i when the batch, policy, and label checks succeed, and outputs $\perp$ otherwise.
- **Combine**. Given at least t_{rec} honest partial decryption shares, reconstructs the plaintext μ.

Correctness requires recovery of the original message except with negligible probability whenever $x \in S$, $f_\lambda(x) = 0$, the ciphertext and key use the same batch label, and at least t_{rec} honest committee shares are provided. Moreover, each secret-key share, digest and ciphertext remains succinct and does not explicitly encode the entire batch S. We consider selective security where the adversary fixes (x^*, tg^*) before setup. A query $(i, S, f_\lambda, \mathsf{tg})$ is fully challenge-compatible when $x^* \in S$, $f_\lambda(x^*) = 0$ and $\mathsf{tg} = \mathsf{tg}^*$, and such shares may be obtained for at most t_{priv} distinct committee members, while all other queries may be issued polynomially many times. Fully compatible queries are made first so that the challenger can fix the corresponding threshold shares and complete a consistent sharing. The adversary receives an encryption of μ_b under (x^*, tg^*) and must distinguish b with only negligible advantage.

2.2 Designing TB-ABE from Lattices

Boneh *et al.* [7] describe the conceptual derivation of batched identity-based encryption from key-policy ABE. An analogous combined predicate can be used for batched ABE. Given a batch S, policy f, and tag tg_0, define

$$g_{S,f,\mathsf{tg}_0}(x,\mathsf{tg}) = \begin{cases} 0, & x \in S,\ f(x) = 0,\ \mathsf{tg} = \mathsf{tg}_0, \\ 1, & \text{otherwise.} \end{cases}$$

An expressive key-policy ABE scheme can enforce this combined predicate, but directly embedding a batch-membership test may make the predicate description grow with $|S|$. Such a compiler does not by itself establish the specific succinct-digest, compact-key, or threshold trust guarantees considered here. Our lattice construction represents membership through a polynomial digest and combines it with policy and tag blocks in one short-preimage relation. Likewise, a threshold-ABE to threshold-batched-ABE translation must preserve the intended reconstruction thresholds and trust model; threshold ABE alone does not entail dealer-free batched key issuance.

Our construction combines four main ingredients. The lattice batch-membership mechanism uses a polynomial representation of a batch together with the matrix commitment mechanism [26, 29]. Informally, for a committed vector $\mathbf{z}$, its commitment $\mathbf{C_z}$ and verification/opening matrices satisfy

$$\mathbf{C_z V} = \mathbf{z} \otimes \mathbf{G} - \mathbf{A Z_z}.$$

The norm bounds for $\mathbf{V}$ and $\mathbf{Z_z}$ control the noise of the extracted digest component. See Lemma 10 for the formal statement. Key-homomorphic evaluation [6] binds a secret key to an arithmetic-circuit policy f_λ. A linear ramp-threshold secret-sharing scheme [4, 5] distributes a common matrix $\mathbf{D}$ among the N decryption parties. Finally, lattice trapdoor algorithms [6] provide short preimages for matrices that simultaneously encode all three authorization conditions. The central design object is the matrix $\mathbf{F}_{\mathsf{dig},f_\lambda,\mathsf{tg}} = [\mathbf{A} \mid \mathsf{dig} \mid \mathbf{L}_0 + \mathsf{tg}\mathbf{L}_1 \mid \mathbf{B}_{f_\lambda}]$, where $\mathbf{B}_{f_\lambda}$ is obtained by homomorphically evaluating f_λ on the public attribute matrices. Each threshold secret-key share consists of short preimages of that same matrix.

Succinct batch encoding. Attributes are first injectively embedded into $\mathbb{Z}_q$ via $\iota : \{0,1\}^k \to \mathbb{Z}_q$. For a batch S, define $P_{\iota(S)}(X) = \prod_{a \in \iota(S)}(X - a)$ and let $\gamma \leftarrow \mathsf{GenCoeff}(\iota(S))$ be the binary decomposition of its coefficient vector. The crucial property is $\mathsf{ID}(x)\gamma^\top = P_{\iota(S)}(\iota(x))$. Consequently, when $x \in S$, this value is zero, whereas for $x \notin S$, it is a nonzero element of $\mathbb{Z}_q$. The digest is the single matrix $\mathsf{dig}_S = -\mathbf{QV}(\gamma^\top \otimes \mathbf{I}_m)$, where $\mathbf{V}$ is the verification matrix of the matrix commitment scheme. Thus, the complete batch affects secret-key generation only through the succinct matrix dig_S.

Policy and batch-label binding. Setup samples public matrices $\mathbf{B}_1, \ldots, \mathbf{B}_k$ and $\mathbf{L}_0, \mathbf{L}_1$. For a policy f_λ, key generation computes $\mathbf{B}_{f_\lambda} \leftarrow \mathsf{Eval}_{\mathsf{pk}}(f_\lambda, \mathbf{B}_1, \ldots, \mathbf{B}_k)$. The block $\mathbf{B}_{f_\lambda}$ binds the key to the access policy, whereas $\mathbf{L}_0 + \mathsf{tg}\mathbf{L}_1$ binds it to the batch label. These blocks and the batch digest are placed together in $\mathbf{F}_{\mathsf{dig},f_\lambda,\mathsf{tg}}$, so a single short preimage simultaneously carries all three authorization requirements.

Threshold key generation. During setup a trusted authority samples $\mathbf{D} \in \mathbb{Z}_q^{\ell \times h}$ and computes its single ramp sharing $\{(\mathsf{psh}_i, \mathbf{D}_{i,1}, \ldots, \mathbf{D}_{i,y})\}_{i \in [N]}$. For a requested $(\mathsf{dig}_S, f_\lambda, \mathsf{tg})$, the same authority forms $\mathbf{F}_{\mathsf{dig}_S, f_\lambda, \mathsf{tg}}$ and samples short $\mathbf{R}_{i,j}$ such that $\mathbf{F}_{\mathsf{dig}_S, f_\lambda, \mathsf{tg}} \mathbf{R}_{i,j} = \mathbf{D}_{i,j} \pmod q$. The underlying shares of $\mathbf{D}$ are reused for successive authorization tuples; however, new lattice preimages must be issued to committee members whenever the batch digest, policy, or tag changes. There is no per-request renegotiation of the underlying secret sharing, but this is not a dealer-free key-issuance mechanism.

Encryption and local authorization checks. Encryption produces five categories of components: base, attribute, batch, tag, and output. The base component $\mathbf{c}_{\mathsf{in}} = \mathbf{sA} + \mathbf{e}_0$ carries the underlying LWE sample. The attribute components $\{\mathbf{c}_a\}_{a \in [k]}$ allow $\mathsf{Eval}_{\mathsf{ct}}$ to produce an evaluated component for $f_\lambda(x)$. The batch component $\mathbf{c}_{\mathsf{bat}}$ contains a commitment to $\mathsf{ID}(x)$, while the label component $\mathbf{c}_{\mathsf{tg}}$ encrypts under $\mathbf{L}_0 + \mathsf{tg}\mathbf{L}_1$. Finally, $\mathbf{c}_{\mathsf{out}} = \mathbf{sD} + \mathbf{e}_{\mathsf{out}} + \lfloor q/2 \rfloor \mu$ carries the message. During partial decryption, the batch polynomial and matrix commitment cancel the batch component precisely when $x \in S$, while key-homomorphic evaluation gives $\mathbf{c}_{f_\lambda}^\top = \mathbf{sB}_{f_\lambda} + \mathsf{noise}$ whenever $f_\lambda(x) = 0$. Hence, the combined row vector satisfies $\overline{\mathbf{c}} = \mathbf{sF}_{\mathsf{dig}, f_\lambda, \mathsf{tg}} + \mathsf{noise}$, and multiplication by $\mathbf{R}_{i,j}$ produces $\mathbf{sD}_{i,j}$ plus bounded noise.

Threshold reconstruction. For any t_{rec} participating honest parties, the ramp secret-sharing reconstruction coefficients satisfy $\mathbf{D} = \sum_{i,j} \nu_{i,j} \mathbf{D}_{i,j}$. Therefore, combining the corresponding partial decryptions cancels $\mathbf{sD}$ from $\mathbf{c}_{\mathsf{out}}$, leaving $\lfloor q/2 \rfloor \mu$ plus bounded error. The small reconstruction coefficients of the secret-sharing scheme ensure that this error does not grow too quickly with the committee size, and parameters are chosen so that magnitude remains below $q/4$.

2.3 Correctness and Efficiency Guarantees of Construction

Correctness follows from the fact that the same lattice matrix appears in key generation and partial decryption. For an honest batch S containing x, $\mathsf{GenCoeff}$ algorithm gives $\mathsf{ID}(x)\gamma^\top = 0$. Combined with the matrix-commitment identity, this transforms the ciphertext's batch component into the term corresponding to the batch digest, namely $\mathbf{s} \cdot \mathsf{dig}_S$, plus an error arising from the original Gaussian noises χ and χ'. Similarly, when $f_\lambda(x) = 0$, correctness of $\mathsf{Eval}_{\mathsf{ct}}$ converts the evaluated policy component into $\mathbf{sB}_{f_\lambda}$ plus bounded evaluation noise. Together with the base and label components, partial decryption, therefore, operates on $\mathbf{sF}_{\mathsf{dig}_S, f_\lambda, \mathsf{tg}}$ plus bounded noise. Since every $\mathbf{R}_{i,j}$ is a preimage of $\mathbf{D}_{i,j}$, the i-th decryption share has the form $\mathbf{sD}_{i,j}$ plus bounded Gaussian and smudging errors. The reconstruction coefficients of the ramp scheme then combine any t_{rec} such values into $\mathbf{sD}$ plus a controlled error. Subtracting this quantity from $\mathbf{c}_{\mathsf{out}}$ cancels the common term $\mathbf{sD}$, leaving the encoded message and a total error bounded by the sum of the batched-IBE digest error, the attribute-evaluation error, the Gaussian preimage expansion, and the threshold-reconstruction error. Choosing q, according to Eq. (1), keeps every coordinate below $q/4$, and ordinary rounding recovers the message.

2.4 Achieving Security of TB-ABE Construction

We finally outline the selective-security proof of Theorem 3. Let x^* and tg^* denote the challenge attribute and batch label, respectively, fixed before setup.

Programming the challenge public parameters. We first program all public matrices that appear in the challenge ciphertext around the common base matrix $\mathbf{A}$. In particular, we set $\mathbf{Q} = \mathbf{A}\mathbf{U}_{\mathsf{bat}} - \mathbf{C}^*_{\mathsf{ID}}$, $\mathbf{L}_0 = \mathbf{A}\mathbf{U}_{\mathsf{tg}} - \mathsf{tg}^*\mathbf{L}_1$, $\mathbf{D} = \mathbf{A}\mathbf{U}_{\mathbf{D}}$, and $\mathbf{B}_a = \mathbf{A}\mathbf{S}^*_a - x^*_a\mathbf{G}$. The matrix leftover-hash lemma shows that these programmed matrices are jointly statistically indistinguishable from independently uniform public parameters, while retaining the correlated noise terms required by the attribute ciphertexts. Once this programming is in place, all challenge components can be expressed as transformations of the same base LWE sample $\mathbf{c}^*_{\mathsf{in}}$. Noise rerandomization then replaces the batch, tag and output components by ReRand evaluations of that sample.

Simulating fully challenge-compatible threshold shares. The main new difficulty compared with ordinary batched IBE is that the adversary may obtain keys satisfying all three challenge conditions for as many as t_{priv} parties. For such queries, none of the authorization conditions fails, and hence no alternative trapdoor can be derived from a failed condition. Instead, the stronger compatible-share property specified in Section 3.7 is assumed to make their private syndrome components jointly uniform independently of the shared matrix $\mathbf{D}$. The simulator, therefore, reverses the order of generation, first sampling $\mathbf{R}_{i,j} \leftarrow D_{\mathbb{Z}^{4m\times h},\sigma}$ and then defining $\mathbf{D}_{i,j} := \mathbf{F}_{S,f_\lambda,\mathsf{tg}^*}\mathbf{R}_{i,j}$. The Gaussian syndrome-first sampling lemma shows that this is statistically indistinguishable from first choosing a uniform $\mathbf{D}_{i,j}$ and then sampling its short preimage. After all fully compatible shares have been fixed, an efficient conditional-sharing sampler is required to complete a globally consistent sharing of $\mathbf{D}$.

Alternative trapdoors for challenge-incompatible queries. Every remaining key query violates at least one challenge condition, and each failure exposes a different trapdoor. If $x^* \notin S$, the batch polynomial gives $P_{\iota(S)}(\iota(x^*)) \neq 0$, allowing the digest block to be written as $\mathsf{dig}_S = \mathbf{A}\mathbf{R}_S + \alpha_S\mathbf{G}$ with $\alpha_S \in \mathbb{Z}^*_q$. The public gadget trapdoor, therefore, yields a trapdoor for the digest block. If $\mathsf{tg} \neq \mathsf{tg}^*$, the programmed identity $\mathbf{L}_0 + \mathsf{tg}\mathbf{L}_1 = \mathbf{A}\mathbf{U}_{\mathsf{tg}} + (\mathsf{tg} - \mathsf{tg}^*)\mathbf{L}_1$ allows the simulator to use the hidden trapdoor for $\mathbf{L}_1$. Finally, if $f_\lambda(x^*) \neq 0$, the simulation property of key-homomorphic evaluation gives $\mathbf{B}_{f_\lambda} = \mathbf{A}\mathbf{S}_{f_\lambda} - f_\lambda(x^*)\mathbf{G}$, so the gadget trapdoor again applies. In every case, the resulting trapdoor is for exactly the same matrix $\mathbf{F}_{S,f_\lambda,\mathsf{tg}}$ used in the real scheme. Canonical Gaussian preimage sampling, therefore, makes the indistinguishable simulated keys.

Removing secret keys from partial-decryption queries. After all key-generation queries have been simulated without using the master trapdoor $\mathbf{T}_{\mathbf{A}}$, the remaining challenge is to simulate the partial-decryption oracle. For every valid queried ciphertext, correctness gives $d_{i,j} = \mathbf{s}\mathbf{D}_{i,j} + \boldsymbol{\eta}_{i,j} + \mathbf{e}^{\mathsf{sm}}_{i,j}$ for a bounded lattice error $\boldsymbol{\eta}_{i,j}$. Since the smudging distribution is chosen much wider than this error, the smudging lemma allows us to replace the entire error term by fresh uniform smudging noise. Consequently, the oracle can answer using only the public encryption randomness and the secret-sharing value $\mathbf{D}_{i,j}$, and no longer needs a secret key or $\mathbf{T}_{\mathbf{A}}$.

Invoking succinct LWE and masking the message. At this point the simulator has eliminated every use of $\mathbf{T}_{\mathbf{A}}$, so the base challenge component $\mathbf{c}^*_{\mathsf{in}} = \mathbf{s}\mathbf{A} + \mathbf{e}_0$ can be embedded directly into a $2m^2$-succinct LWE challenge. The assumption allows this component to be replaced by a uniform vector while preserving the auxiliary commitment information $(\mathbf{W}, \mathbf{T})$. Since every other challenge component is now an efficiently computable transformation of $\mathbf{c}^*_{\mathsf{in}}$, the reduction simulates the entire challenge without

knowing the LWE secret. Finally, $\mathbf{c}^*_{\mathsf{out}}$ is a rerandomized image of this uniform vector under the short matrix $\mathbf{U_D}$. The final leftover-hash/output-masking argument makes this value statistically uniform in $\mathbb{Z}_q^{1\times h}$, so adding $\lfloor q/2 \rfloor \mu_b$ has no effect on its distribution. The final hybrid is, therefore, independent of the challenge bit. This joint structure allows batching, attribute-based authorization and threshold decryption to be realized within a single succinct lattice framework.

3 Preliminaries

This section provides an overview of the cryptographic tools used in this work.

Notations. Bold uppercase letters (e.g., $\mathbf{A}, \mathbf{B}$) denote matrices, bold lowercase letters (e.g., $\mathbf{u}, \mathbf{v}$) denote vectors, and non-bold letters denote scalars. For a finite set S, $s \leftarrow S$ denotes uniform sampling from S, and $|S|$ its cardinality. We use $\approx_s$ and $\approx_c$ for statistical and computational indistinguishability, respectively, and $\Delta(X, Y)$ for statistical distance. For $x \in 0,1^n$, $x[w]$ denotes its w-th bit, and $\mathbb{Z}_q^* := \mathbb{Z}_q \setminus \{0\}$. For a matrix $\mathbf{R}$, $||\mathbf{R}||$, $|\mathbf{R}|_2$, and $\widetilde{\mathbf{R}}$ denote its ℓ_∞ norm, ℓ_2 norm, and Gram–Schmidt orthogonalization, respectively. When taking norms, elements of $\mathbb{Z}_q$ are identified with their canonical representatives in $(-q/2, q/2]$, and $\lfloor \cdot \rceil$ denotes component-wise rounding to the nearest integer. PPT refers to probabilistic polynomial-time algorithms using classical computation.

3.1 Lattice Preliminaries

For a matrix $\mathbf{A} \in \mathbb{Z}_q^{\ell\times m}$, define $\Lambda_q^{\perp}(\mathbf{A}) := \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{Ax} = \mathbf{0} \bmod q\}$. For $\mathbf{u} \in \mathbb{Z}_q^{\ell}$, define the affine coset $\Lambda_q^{\mathbf{u}}(\mathbf{A}) := \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{Ax} = \mathbf{u} \bmod q\}$. For a target matrix $\mathbf{U} = (\mathbf{u}_1|\cdots|\mathbf{u}_d) \in \mathbb{Z}_q^{\ell\times d}$, we write $D_\sigma(\Lambda_q^{\mathbf{U}}(\mathbf{A}))$ for the distribution on matrices whose j-th column is sampled independently from $D_\sigma(\Lambda_q^{\mathbf{u}_j}(\mathbf{A}))$. We write $\mathcal{D}_{\mathbb{Z},\sigma}$ for the discrete Gaussian distribution over $\mathbb{Z}$ with parameter $\sigma > 0$, and $\mathcal{D}_{\mathbb{Z}^d,\sigma}$ for its d-fold product distribution. The quantities $\chi, \chi' > 0$ used later denote Gaussian widths. Thus, for example, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^{1\times m},\chi}$ means that the coordinates of $\mathbf{e}$ are independent discrete Gaussians of parameter χ.

Definition 1 (Learning with Errors [21]). *Fix integers ℓ, m, a prime q, and a noise distribution $\mathcal{D}$ over $\mathbb{Z}$. The $(\ell, m, q, \mathcal{D})$-LWE problem is to distinguish the distributions $(\mathbf{A}, \mathbf{sA} + \mathbf{e})$ and $(\mathbf{A}, \mathbf{u})$, where $\mathbf{A} \leftarrow \mathbb{Z}_q^{\ell\times m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^{1\times\ell}$, $\mathbf{e} \leftarrow \mathcal{D}^m$, and $\mathbf{u} \leftarrow \mathbb{Z}_q^{1\times m}$ are independently sampled. We use $m = \Theta(\ell \log q)$ throughout.*

Gadget Matrices. Let $\kappa := \lceil \log q \rceil$ and define $\mathbf{g}' := (1, 2, \ldots, 2^{\kappa-1}) \in \mathbb{Z}_q^{1\times\kappa}$. For every positive integer d, define the unpadded gadget matrix $\mathbf{G}_d := \mathbf{I}_d \otimes \mathbf{g}' \in \mathbb{Z}_q^{d\times d\kappa}$. For the lattice row dimension ℓ, we use the padded gadget matrix $\mathbf{G} := [\mathbf{G}_\ell \mid \mathbf{0}] \in \mathbb{Z}_q^{\ell\times m}$ whenever $m > \ell\kappa$. The lattice $\Lambda_q^{\perp}(\mathbf{G})$ admits a publicly known short basis $\mathbf{T_G}$. For every d, let $\mathbf{G}_d^{-1}$ denote the deterministic binary decomposition map satisfying $\mathbf{G}_d\mathbf{G}_d^{-1}(\mathbf{U}) = \mathbf{U}$ for every conformable matrix $\mathbf{U}$.

Batch Digest Coefficients. The algorithm GenCoeff compresses a batch $S \subseteq \mathbb{Z}_q$ of size B by forming the polynomial $P_S(X) = \prod_{a\in S}(X - a) = \sum_{i=0}^{B} y_i X^i$ and gadget-decomposing its coefficient vector to obtain $\gamma := \left(\mathbf{G}_{B+1}^{-1}((\mathbf{y}')^\top)\right)^\top \in \{0,1\}^{1\times\kappa(B+1)}$,

where $\mathbf{y}' := (y_0, \ldots, y_B) \in \mathbb{Z}_q^{1\times(B+1)}$ and $\mathbf{G}_{B+1} := \mathbf{I}_{B+1} \otimes \mathbf{g}' \in \mathbb{Z}_q^{(B+1)\times\kappa(B+1)}$. For $\mathsf{ID}(a) = (1, a, \ldots, a^B) \otimes \mathbf{g}'$, gadget decomposition gives $\mathsf{ID}(a)\gamma^\top = P_S(a)$, so $\mathsf{ID}(a)\gamma^\top = 0$ when $a \in S$, while for $a \notin S$, lies in $\mathbb{Z}_q^*$.

3.2 Lattice Trapdoors

Lemma 1 (Lattice Trapdoor Algorithms [6]). *Let λ be a security parameter and let ℓ, m, q be lattice parameters. There exist efficient algorithms with the following properties:*

- $(\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \mathsf{TrapGen}(1^\ell, 1^m, q)$. *For $m = \Theta(\ell \log q)$, the algorithm outputs a full-rank matrix $\mathbf{A} \in \mathbb{Z}_q^{\ell\times m}$ that is statistically close to uniform, together with a basis $\mathbf{T}_\mathbf{A} \in \mathbb{Z}^{m\times m}$ of $\Lambda_q^\perp(\mathbf{A})$ satisfying $\|\widetilde{\mathbf{T}}_\mathbf{A}\| = O(\sqrt{\ell \log q})$, except with negligible probability.*
- $\mathbf{X} \leftarrow \mathsf{SamplePre}(\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{U}, \sigma)$. *Given $\mathbf{U} \in \mathbb{Z}_q^{\ell\times d}$ and $\sigma \geq \|\widetilde{\mathbf{T}}_\mathbf{A}\|\omega(\sqrt{\log m})$, the algorithm outputs $\mathbf{X} \in \mathbb{Z}^{m\times d}$ distributed statistically close to $D_\sigma(\Lambda_q^\mathbf{U}(\mathbf{A}))$.*
- $\mathbf{X} \leftarrow \mathsf{SampleRight}(\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{B}, \mathbf{U}, \sigma)$. *For $\mathbf{A} \in \mathbb{Z}_q^{\ell\times m_1}$, $\mathbf{B} \in \mathbb{Z}_q^{\ell\times m_2}$, and $\mathbf{U} \in \mathbb{Z}_q^{\ell\times d}$, the algorithm outputs $\mathbf{X} \in \mathbb{Z}^{(m_1+m_2)\times d}$ distributed statistically close to $D_\sigma(\Lambda_q^\mathbf{U}([\mathbf{A} \mid \mathbf{B}]))$ for a sufficiently large Gaussian parameter σ.*
- $\mathbf{T}_{[\mathbf{A}|\mathbf{B}]} \leftarrow \mathsf{ExtendRight}(\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{B})$. *Given $\mathbf{A} \in \mathbb{Z}_q^{\ell\times m_1}$, a trapdoor for $\mathbf{A}$, and an arbitrary conformable matrix $\mathbf{B} \in \mathbb{Z}_q^{\ell\times m_2}$, the algorithm outputs a trapdoor for $[\mathbf{A} \mid \mathbf{B}]$.*
- $\mathbf{T}_{[\mathbf{A}|\mathbf{AR}+\mathbf{B}]} \leftarrow \mathsf{ExtendLeft}(\mathbf{A}, \mathbf{B}, \mathbf{T}_\mathbf{B}, \mathbf{R})$. *Let $\mathbf{A} \in \mathbb{Z}_q^{\ell\times m_1}$, $\mathbf{B} \in \mathbb{Z}_q^{\ell\times m_2}$, and $\mathbf{R} \in \mathbb{Z}^{m_1\times m_2}$, and let $\mathbf{T}_\mathbf{B}$ be a short basis of $\Lambda_q^\perp(\mathbf{B})$. Then the algorithm outputs a trapdoor for $[\mathbf{A} \mid \mathbf{AR} + \mathbf{B}]$ satisfying $\|\widetilde{\mathbf{T}}_{[\mathbf{A}|\mathbf{AR}+\mathbf{B}]}\| \leq \|\widetilde{\mathbf{T}}_\mathbf{B}\|(1 + \|\mathbf{R}\|_2)$ up to the standard trapdoor-sampling factors.*

3.3 Auxiliary Lattice Lemmas

Lemma 2 (Gaussian Tail Bound). *Let $\sigma > 0$ and let $\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z}^d,\sigma}$. For every $\lambda \in \mathbb{N}$, $\Pr\left[\|\mathbf{x}\| > \sigma\sqrt{d\lambda}\right] \leq d \cdot 2^{-\lambda}$. In particular, for $d = \mathsf{poly}(\lambda)$, the failure probability is negligible. We occasionally write $c_\mathsf{G} := \sqrt{\lambda}$ and use the shorthand $\|\mathbf{x}\| \leq c_\mathsf{G}\sigma\sqrt{d}$.*

Lemma 3 (Gaussian Matrix Norm Bound [6]). *Let $\mathbf{A} \in \mathbb{Z}_q^{\ell\times M}$ and $\mathbf{U} \in \mathbb{Z}_q^{\ell\times d}$. If $\mathbf{R} \leftarrow D_\sigma(\Lambda_q^\mathbf{U}(\mathbf{A}))$, then, except with negligible probability, $\|\mathbf{R}\|_2 \leq \sigma\sqrt{Md}$. Moreover, if $\mathbf{S} \leftarrow \{\pm 1\}^{m\times m}$, then $\|\mathbf{S}\|_2 \leq 20\sqrt{m}$ except with negligible probability.*

Encryption Randomness Let $B_0 := c_\mathsf{G}\chi\sqrt{m}$, $B_1 := c_\mathsf{G}\chi'\sqrt{m}$, and $B_\mathsf{out} := c_\mathsf{G}\chi'\sqrt{h}$. We call encryption randomness $r = (\mathbf{s}, \mathbf{e}_0, \mathbf{S}_1, \ldots, \mathbf{S}_k, \mathbf{e}_\mathsf{bat}, \mathbf{e}_\mathsf{tg}, \mathbf{e}_\mathsf{out})$ *well formed* if $\|\mathbf{e}_0\| \leq B_0$, $\|\mathbf{e}_\mathsf{bat}\|, \|\mathbf{e}_\mathsf{tg}\| \leq B_1$, $\|\mathbf{e}_\mathsf{out}\| \leq B_\mathsf{out}$, and $\mathbf{S}_a \in \{\pm 1\}^{m\times m}$ with $\|\mathbf{S}_a\|_2 \leq 20\sqrt{m}$ for all $a \in [k]$. Let $\mathcal{R}_\mathsf{Enc}$ denote the set of such randomness, which contains honestly sampled encryption randomness except with negligible probability by Lemmas 2 and 3. We now recall two versions of the leftover hash lemma [3, 13, 18].

Lemma 4 (Leftover Hash Lemma). *Let λ be a security parameter, and let ℓ, m, q be lattice parameters. Let $\mathcal{P}$ be a distribution over $\mathbb{Z}^m$ satisfying $H_\infty(\mathcal{P}) \geq 2\ell \log q$. Then, $(\mathbf{A}, \mathbf{Ar} \bmod q) \approx_s (\mathbf{A}, \mathbf{u})$, where $\mathbf{A} \leftarrow \mathbb{Z}_q^{\ell \times m}$, $\mathbf{r} \leftarrow \mathcal{P}$, and $\mathbf{u} \leftarrow \mathbb{Z}_q^{\ell}$.*

Lemma 5 (Matrix Leftover Hash Lemma). *Suppose $m > (\ell + 1)\log_2 q + \omega(\log \ell)$ and $q > 2$ is prime. Let $d = \mathsf{poly}(\lambda)$, $\mathbf{S} \leftarrow \{\pm 1\}^{m \times d}$, $\mathbf{A} \leftarrow \mathbb{Z}_q^{\ell \times m}$, and $\mathbf{B} \leftarrow \mathbb{Z}_q^{\ell \times d}$ be independently uniform. Then, for every fixed row vector $\mathbf{e} \in \mathbb{Z}_q^{1 \times m}$,*

$$(\mathbf{A}, \mathbf{AS}, \mathbf{eS}) \approx_s (\mathbf{A}, \mathbf{B}, \mathbf{eS}).$$

Lemma 6 (Noise Rerandomization [19]). *Let $\chi > \max\{\omega(\sqrt{\log m}),\ \omega(\sqrt{\log d})\}$. There exists an efficient algorithm ReRand such that, for $\mathbf{v} \in \mathbb{Z}_q^{1 \times m}$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^{1 \times m}, \chi}$, $\mathbf{R} \in \mathbb{Z}^{m \times d}$ and $\rho > \|\mathbf{R}\|_2$,*

$$\left(\mathbf{R}, \mathbf{x} := \mathbf{v} + \mathbf{e}, \mathsf{ReRand}(\mathbf{R}, \mathbf{x}, \rho)\right) \approx_s \left(\mathbf{R}, \mathbf{x}, \mathbf{vR} + \mathbf{e}'\right),$$

where $\mathbf{e}' \leftarrow \mathcal{D}_{\mathbb{Z}^{1 \times d}, 2\chi\rho}$. In particular, taking $\rho = \chi'/(2\chi)$ gives output noise of Gaussian width χ', provided $\chi'/(2\chi) > \|\mathbf{R}\|_2$.

Lemma 7 (Smudging Lemma). *Let $B_1, B_2 \in \mathbb{N}$ and let $e \in [-B_1, B_1]$ be fixed. Let E_1, E_2 be independently and uniformly distributed over $[-B_2, B_2]$. Then, $\Delta(E_1 + e, E_2) < B_1/B_2$.*

Lemma 8 (Gaussian Samples [14]). *Let ℓ, m, q, σ be lattice parameters satisfying $\sigma \geq \log m$, $m \geq 2\ell \log q$, and let q be prime. Then, for all but a $q^{-\ell}$ fraction of matrices $\mathbf{A} \in \mathbb{Z}_q^{\ell \times m}$, the following holds,*

$$\left\{(\mathbf{x}, \mathbf{Ax}) : \mathbf{x} \leftarrow D_{\mathbb{Z}^m, \sigma}\right\} \approx_s \left\{(\mathbf{x}, \mathbf{y}) : \mathbf{y} \leftarrow \mathbb{Z}_q^{\ell},\ \mathbf{x} \leftarrow D_{\sigma}(\Lambda_q^{\mathbf{y}}(\mathbf{A}))\right\}.$$

Corollary 1 (Output Masking). *Let $\mathbf{A} \leftarrow \mathbb{Z}_q^{\ell \times m}$, $\mathbf{U} \leftarrow \{0,1\}^{m \times h}$, and set $\mathbf{D} := \mathbf{AU}$. Let $\mathbf{c} \leftarrow \mathbb{Z}_q^{1 \times m}$ be uniform and define $\mathbf{z} := \mathsf{ReRand}\left(\mathbf{U}, \mathbf{c}, \frac{\chi'}{2\chi}\right)$. Then, it follows that $(\mathbf{A}, \mathbf{D}, \mathbf{c}, \mathbf{z}) \approx_s (\mathbf{A}, \mathbf{D}, \mathbf{c}, \mathbf{u})$, where $\mathbf{u} \leftarrow \mathbb{Z}_q^{1 \times h}$ is independent uniform.*

3.4 Key-Homomorphic Evaluation

Lemma 9 (Key-Homomorphic Evaluation [6]). *Let $\mathcal{F}_{\mathsf{ckt}} = \{\mathcal{F}_{\mathsf{ckt},\lambda}\}_\lambda$ be a family of polynomial-size arithmetic circuits over $\mathbb{Z}_q$. Let $d = d(\lambda)$ and $r = r(\lambda)$ be upper bounds on circuit depth and gate fan-in, respectively, and let $p = p(\lambda)$ denote the upper bound at least one input to every multiplication gate. There exist deterministic polynomial-time algorithms $\mathsf{Eval}_{\mathsf{pk}}$, $\mathsf{Eval}_{\mathsf{ct}}$, and $\mathsf{Eval}_{\mathsf{sim}}$ with the following properties:*

- $\mathbf{B}_f \leftarrow \mathsf{Eval}_{\mathsf{pk}}(f, \mathbf{B}_1, \ldots, \mathbf{B}_z)$ *outputs* $\mathbf{B}_f \in \mathbb{Z}_q^{\ell \times m}$.
- *For $\mathbf{s} \in \mathbb{Z}_q^{\ell}$ and $\delta > 0$, define $E_{\mathbf{s},\delta}(x, \mathbf{B}) := \{(x\mathbf{G} + \mathbf{B})^\top \mathbf{s} + \mathbf{e} : \|\mathbf{e}\| \leq \delta\}$. If $\mathbf{c}_i \in E_{\mathbf{s},\delta}(x_i, \mathbf{B}_i)$ for all $i \in [z]$, then $\mathbf{c}_f \leftarrow \mathsf{Eval}_{\mathsf{ct}}(f, ((x_i, \mathbf{B}_i, \mathbf{c}_i))_{i=1}^z)$ belongs to $E_{\mathbf{s},\Delta}(f(x), \mathbf{B}_f)$, where $\mathbf{B}_f = \mathsf{Eval}_{\mathsf{pk}}(f, \mathbf{B}_1, \ldots, \mathbf{B}_z)$ and $\Delta < \delta\, \alpha_{\mathcal{F}}(\ell)$, with $\alpha_{\mathcal{F}}(\ell) = O((p^{r-1}m)^d \sqrt{m})$.*
- *Given $\mathbf{B}_i = \mathbf{AS}_i - x_i^* \mathbf{G}$, the simulation algorithm $\mathbf{S}_f \leftarrow \mathsf{Eval}_{\mathsf{sim}}(f, ((x_i^*, \mathbf{S}_i))_{i=1}^z, \mathbf{A})$ satisfies $\mathbf{AS}_f - f(x^*)\mathbf{G} = \mathbf{B}_f$, where $\mathbf{B}_f = \mathsf{Eval}_{\mathsf{pk}}(f, \mathbf{AS}_1 - x_1^*\mathbf{G}, \ldots, \mathbf{AS}_z - x_z^*\mathbf{G})$. If $\mathbf{S}_1, \ldots, \mathbf{S}_z \leftarrow \{\pm 1\}^{m \times m}$, then $\|\mathbf{S}_f\|_2 < \alpha_{\mathcal{F}}(\ell)$ except with negligible probability.*

3.5 Matrix Commitments

Lemma 10 (Matrix Commitment [26, 29]). *Let λ be a security parameter, and let ℓ, m, q be lattice parameters. Let* $\mathsf{pp} := (\mathbf{A}, \mathbf{W}, \mathbf{T})$*, where* $\mathbf{A} \in \mathbb{Z}_q^{\ell \times m}$*,* $\mathbf{W} \in \mathbb{Z}_q^{2m^2\ell \times m}$*, and* $\mathbf{T} \in \mathbb{Z}^{(2m^2+1)m \times 2m^3}$ *satisfy* $[\mathbf{I}_{2m^2} \otimes \mathbf{A} \mid \mathbf{W}]\mathbf{T} = \mathbf{I}_{2m^2} \otimes \mathbf{G}$*. There exist efficient algorithms* $(\mathsf{Com}, \mathsf{Ver}, \mathsf{Open})$ *with the following properties.*

- $\mathbf{C}_{\mathbf{x}} \leftarrow \mathsf{Com}(\mathsf{pp}, \mathbf{x})$*, on input* $\mathbf{x} \in \mathbb{Z}_q^d$*, outputs* $\mathbf{C}_{\mathbf{x}} \in \mathbb{Z}_q^{\ell \times m}$*.*
- $\mathbf{V} \leftarrow \mathsf{Ver}(\mathsf{pp}, 1^d)$ *outputs* $\mathbf{V} \in \mathbb{Z}_q^{m \times dm}$*.*
- $\mathbf{Z}_{\mathbf{x}} \leftarrow \mathsf{Open}(\mathsf{pp}, \mathbf{x})$ *outputs* $\mathbf{Z}_{\mathbf{x}} \in \mathbb{Z}_q^{m \times dm}$*.*

For every $d \in \mathbb{N}$ *and every* $\mathbf{x} \in \mathbb{Z}_q^d$*, these matrices satisfy* $\mathbf{C}_{\mathbf{x}}\mathbf{V} = \mathbf{x} \otimes \mathbf{G} - \mathbf{A}\mathbf{Z}_{\mathbf{x}}$*. Moreover,* $\|\mathbf{V}\| \leq O(\|\mathbf{T}\| m^4 \log q)$ *and* $\|\mathbf{Z}_{\mathbf{x}}\| \leq O(\|\mathbf{T}\| \log d\, m^7 \log q)$*.*

3.6 Hardness Assumptions

ℓ-Succinct LWE Assumption [27, 29] Let ℓ, m, n, q, χ be lattice and noise parameters with $m \geq 2n \log q$. Sample $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^{1 \times n}$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^{1 \times m}, \chi}$, $\mathbf{u} \leftarrow \mathbb{Z}_q^{1 \times m}$, and $\mathbf{W} \leftarrow \mathbb{Z}_q^{n\ell \times m}$. Let $\mathbf{T}$ be a short preimage satisfying $[\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}]\mathbf{T} = \mathbf{I}_\ell \otimes \mathbf{G}$. The ℓ-succinct LWE assumption states that $(\mathbf{A}, \mathbf{sA} + \mathbf{e}, \mathbf{W}, \mathbf{T}) \approx_c (\mathbf{A}, \mathbf{u}, \mathbf{W}, \mathbf{T})$.

3.7 Ramp Threshold Secret Sharing

We recall the notion of a ramp threshold access structure, which allows a gap between the unauthorized and authorized regimes [4, 5].

Definition 2 (Ramp Threshold Access Structure). *A partial access structure over n parties is a pair $\Gamma = (\Gamma_0, \Gamma_1)$, where $\Gamma_0, \Gamma_1 \subseteq 2^{[n]}$ are non-empty and satisfy $B \notin A$ for every $A \in \Gamma_0$ and $B \in \Gamma_1$. Sets in Γ_1 are authorized and sets in Γ_0 are unauthorized. For $0 < \tau_p < \tau_c \leq 1$, the (τ_p, τ_c)-ramp threshold access structure consists of all subsets of size at most $\tau_p n$ in Γ_0 and all subsets of size at least $\tau_c n$ in Γ_1.*

Syntax. A ramp-threshold secret sharing scheme is a pair of PPT algorithms $(\mathsf{Share}, \mathsf{Reconst})$. The algorithm $\mathsf{Share}(1^\lambda, n, \tau_p, \tau_c, s)$ outputs $(\mathsf{sh}_1, \ldots, \mathsf{sh}_n)$, while $\mathsf{Reconst}(\{\mathsf{sh}_i\}_{i \in S})$ outputs either the shared secret s or $\perp$.

Definition 3 (Ramp Secret-Sharing Correctness). *A ramp-threshold secret sharing scheme is correct if, for every $S \subseteq [n]$ satisfying $|S| \geq \tau_c n$,* $\Pr[\mathsf{Reconst}(\{\mathsf{sh}_i\}_{i \in S}) = s] \geq 1 - \mathsf{negl}(\lambda)$*, where* $(\mathsf{sh}_1, \ldots, \mathsf{sh}_n) \leftarrow \mathsf{Share}(1^\lambda, n, \tau_p, \tau_c, s)$*.*

Definition 4 (Share Uniformity). *A ramp-threshold secret sharing scheme satisfies share uniformity if, for every secret s and every $J \subseteq [n]$ with $|J| \leq \tau_p n$, the joint distribution $(\{\mathsf{sh}_i\}_{i \in J})$ is identical to the corresponding uniform distribution over the share space, independently of s. In particular, the shares of any such set J reveal no information about the secret.*

Theorem 1 (Black-Box Ramp Threshold Secret Sharing). *For every pair of constants $0 < \tau_p < \tau_c < 1$, there exists a black-box (τ_p, τ_c)-ramp-threshold secret sharing scheme with the following properties.*

- Share *is linear. The share of party i decomposes as $\mathsf{sh}_i = (\mathsf{psh}_i, \widehat{\mathsf{sh}}_i)$, where $\widehat{\mathsf{sh}}_i = (\widehat{\mathsf{sh}}_{i,1}, \ldots, \widehat{\mathsf{sh}}_{i,y})$.*
- $\mathsf{Reconst}$ *decomposes into $\mathsf{psh} \leftarrow \mathsf{Reconst}_1(\{\mathsf{psh}_i\}_{i \in S})$ and $s \leftarrow \mathsf{Reconst}_2(\mathsf{psh}, \{\widehat{\mathsf{sh}}_i\}_{i \in S})$, where $s = \sum_{i \in S} \sum_{j \in [y]} \nu_{i,j} \widehat{\mathsf{sh}}_{i,j}$, for reconstruction coefficients satisfying $|\nu_{i,j}| \leq \nu(n)$ for a fixed polynomial $\nu(\cdot)$, while each secret share has size $O(1)$ independent of n. For fixed constants $0 < \tau_p < \tau_c < 1$, define*

$$t_{\mathsf{priv}} := \lfloor \tau_p N \rfloor, \qquad t_{\mathsf{rec}} := \lceil \tau_c N \rceil.$$

4 Threshold Batched Attribute-Based Encryption

This section defines the notion of threshold batched attribute-based encryption.

4.1 System Model of TB-ABE

Let λ be a security parameter. A threshold batched attribute-based encryption scheme is a tuple of efficient algorithms $\prod_{\mathsf{TB\text{-}ABE}} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Digest}, \mathsf{KeyGen}, \mathsf{PDec}, \mathsf{Combine})$ with message space $\mathcal{M}$, attribute space $\{0,1\}^k$, policy space $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ where each $f_\lambda \in \mathcal{F}_\lambda$ is a function $f_\lambda : \{0,1\}^k \to \{0,1\}$, and batch label space Γ. We adopt the convention that an attribute x satisfies a policy f_λ if $f_\lambda(x) = 0$, and fails to satisfy it otherwise. We assume throughout that the master public key mpk is implicitly available to every algorithm.

$\mathsf{Setup}(1^\lambda, 1^B, 1^T, 1^N) \to (\mathsf{mpk}, \mathsf{msk})$. On input the security parameter $\lambda \in \mathbb{N}$, an upper bound B on the batch size, a threshold T, the size N of the decryption committee, the algorithm outputs a master public key mpk and a master secret key msk.

$\mathsf{Enc}(\mathsf{mpk}, m, x, \mathsf{tg}) \to \mathsf{ct}$. On input the master public key mpk, a message $m \in \mathcal{M}$, an attribute vector $x \in \{0,1\}^k$, and a batch label $\mathsf{tg} \in \Gamma$, the encryption algorithm outputs a ciphertext ct.

$\mathsf{Digest}(\mathsf{mpk}, S) \to \mathsf{dig}$. On input mpk and a batch of attribute vectors $S \subseteq \{0,1\}^k$ with $|S| = B$, the digest algorithm outputs a digest dig. This algorithm is deterministic.

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{dig}, f_\lambda, \mathsf{tg}) \to \{\mathsf{sk}_i\}_{i \in [N]}$. On input msk, a digest dig, an access policy $f_\lambda \in \mathcal{F}_\lambda$, and a batch label $\mathsf{tg} \in \Gamma$, the key-generation algorithm outputs secret key shares $\mathsf{sk}_1, \ldots, \mathsf{sk}_N$, each bound to $(\mathsf{dig}, f_\lambda, \mathsf{tg})$.

$\mathsf{PDec}(\mathsf{mpk}, \mathsf{sk}_i, S, \mathsf{ct}) \to d_i$ or $\perp$. On input mpk, a secret key share sk_i, a batch S, and a ciphertext ct, the partial decryption algorithm outputs a partial decryption share d_i, or $\perp$ on failure.

$\mathsf{Combine}(\mathsf{mpk}, \mathsf{ct}, \{d_i\}_{i \in U}) \to m$ or $\perp$. On input mpk, a ciphertext ct, and a set of partial decryption shares $\{d_i\}_{i \in U}$ for $U \subseteq [N]$, the combination algorithm outputs a message $m \in \mathcal{M}$, or $\perp$ on failure.

4.2 Correctness and Efficiency of TB-ABE

We require the scheme to satisfy the following correctness and efficiency properties.

1. **Decryption correctness.** For every $\lambda, B, T, N, k \in \mathbb{N}$, $m \in \mathcal{M}$, $\mathsf{tg} \in \Gamma$, every batch $S \subseteq \{0,1\}^k$ with $|S| = B$, every $x \in S$ and $f_\lambda \in \mathcal{F}_\lambda$ with $f_\lambda(x) = 0$, and every $U \subseteq [N]$ with $|U| \geq t_{\mathsf{rec}}$,

$$\Pr\left[\begin{array}{l|l} \mathsf{Combine}(\mathsf{mpk}, & (\mathsf{mpk},\mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^B, 1^T, 1^N), \\ \mathsf{ct}, \{d_i\}_{i\in U}) = m & \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, m, x, \mathsf{tg}), \\ & \mathsf{dig} \leftarrow \mathsf{Digest}(\mathsf{mpk}, S), \\ & \{\mathsf{sk}_i\}_{i\in[N]} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{dig}, f_\lambda, \mathsf{tg}), \\ & d_i \leftarrow \mathsf{PDec}(\mathsf{mpk}, \mathsf{sk}_i, S, \mathsf{ct}) \ \ \forall i \in U \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

2. **Succinctness.** There exists a universal polynomial $\mathsf{poly}(\cdot)$ such that, for all $(\mathsf{mpk}, \mathsf{msk})$ in the support of Setup, all dig in the support of Digest, all $f_\lambda \in \mathcal{F}_\lambda$, and all $\mathsf{tg} \in \Gamma$, the size of each key share sk_i, the size of ct, and the size of dig are bounded by $\mathsf{poly}(\lambda, \log N, \log B)$.

4.3 Selective Security of TB-ABE

For a security parameter λ and a bit $b \in \{0,1\}$, the query-selective security experiment $\mathsf{Exp}^{\mathsf{tb\text{-}abe}}_{\mathcal{A}}(\lambda, b)$ between a challenger and an adversary $\mathcal{A}$ proceeds as follows. At the beginning, $\mathcal{A}(1^\lambda)$ outputs a batch-size bound B, a threshold $T = t_{\mathsf{rec}} \leq N$, a committee size N, an attribute length k, a message space $\mathcal{M}$, a challenge attribute $x^* \in \{0,1\}^k$, and a challenge batch label $\mathsf{tg}^* \in \Gamma$.

- **Setup:** The challenger computes $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^B, 1^T, 1^N)$ and gives mpk to $\mathcal{A}$. It initializes $\mathsf{Keys} \leftarrow \emptyset$, where Keys is a dictionary caching key shares generated for a given $(S, f_\lambda, \mathsf{tg})$ triple.
- **Query declaration:** Before making any oracle queries, $\mathcal{A}$ declares a set $Q^* = \{(i, S_i^*, f_i^*)\}_{i\in C^*}$ where $C^* \subseteq [N]$, $|C^*| \leq t_{\mathsf{priv}}$, the indices in C^* are distinct, $|S_i^*| = B$, $x^* \in S_i^*$, and $f_i^*(x^*) = 0$ for every $i \in C^*$. These are the only fully challenge-compatible tuples for which $\mathcal{A}$ may obtain a secret-key share or a partial decryption of the challenge ciphertext defined by the oracles below.
- $O_{\mathsf{KeyGen}}(i, S, f_\lambda, \mathsf{tg})$**:** On input a party $i \in [N]$, a batch $S \subseteq \{0,1\}^k$ with $|S| = B$, a policy $f_\lambda \in \mathcal{F}_\lambda$, and a batch label $\mathsf{tg} \in \Gamma$, proceed as follows.
  - If $\mathsf{tg} = \mathsf{tg}^*$, $x^* \in S$, and $f_\lambda(x^*) = 0$, then the query is fully challenge compatible. If $(i, S, f_\lambda) \notin Q^*$, return $\perp$.
  - If $\mathsf{Keys}[(S, f_\lambda, \mathsf{tg})] = \emptyset$, compute $\mathsf{dig} \leftarrow \mathsf{Digest}(\mathsf{mpk}, S)$ and $\{\mathsf{sk}_j\}_{j\in[N]} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{dig}, f_\lambda, \mathsf{tg})$, and store $\mathsf{Keys}[(S, f_\lambda, \mathsf{tg})] := \{\mathsf{sk}_j\}_{j\in[N]}$.
  - Set $\{\mathsf{sk}_j\}_{j\in[N]} := \mathsf{Keys}[(S, f_\lambda, \mathsf{tg})]$ and return sk_i.
- $O_{\mathsf{PDec}}(i, S, f_\lambda, \mathsf{tg}, \mathsf{ct}, \mu, x, r)$**:** On input a party $i \in [N]$, a batch $S \subseteq \{0,1\}^k$ with $|S| = B$, a policy $f_\lambda \in \mathcal{F}_\lambda$, a batch label $\mathsf{tg} \in \Gamma$, a ciphertext ct, a message $\mu \in \mathcal{M}$, an attribute x, and encryption randomness r, the challenger proceeds as follows.

- If the challenge ciphertext ct^* has already been issued and $\mathsf{ct} = \mathsf{ct}^*$, then require $\mathsf{tg} = \mathsf{tg}^*$, $x^* \in S$, $f_\lambda(x^*) = 0$, and $(i, S, f_\lambda) \in Q^*$. If any of these conditions fails, return $\perp$.
- Otherwise, require $r \in \mathcal{R}_{\mathsf{Enc}}$ and verify that $\mathsf{ct} = \mathsf{Enc}(\mathsf{mpk}, \mu, x, \mathsf{tg}; r)$. If either condition fails, return $\perp$.
- If $\mathsf{Keys}[(S, f_\lambda, \mathsf{tg})] = \emptyset$, compute $\mathsf{dig} \leftarrow \mathsf{Digest}(\mathsf{mpk}, S)$ and $\{\mathsf{sk}_j\}_{j\in[N]} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{dig}, f_\lambda, \mathsf{tg})$, and store $\mathsf{Keys}[(S, f_\lambda, \mathsf{tg})] := \{\mathsf{sk}_j\}_{j\in[N]}$.
- Set $\{\mathsf{sk}_j\}_{j\in[N]} := \mathsf{Keys}[(S, f_\lambda, \mathsf{tg})]$ and return $\mathsf{PDec}(\mathsf{mpk}, \mathsf{sk}_i, S, \mathsf{ct})$.

• **Challenge phase:** This phase occurs at most once. The adversary submits two messages $m_0, m_1 \in \mathcal{M}$. The challenger computes $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, m_b, x^*, \mathsf{tg}^*)$ and returns ct^* to $\mathcal{A}$. The adversary may continue to access the above oracles subject to the same restrictions.

• **Output phase:** The adversary finally outputs a bit $b' \in \{0, 1\}$, which is taken as the output of the experiment.

We say that $\prod_{\mathsf{TB\text{-}ABE}}$ is secure if, for every efficient admissible adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{tb\text{-}abe}}_{\mathcal{A}}(\lambda) := \left| \Pr[\mathsf{Exp}^{\mathsf{tb\text{-}abe}}_{\mathcal{A}}(\lambda, 0) = 1] - \Pr[\mathsf{Exp}^{\mathsf{tb\text{-}abe}}_{\mathcal{A}}(\lambda, 1) = 1] \right| \leq \mathsf{negl}(\lambda).$$

5 Lattice Based Construction of TB-ABE

In this section, we provide a construction of a threshold batched attribute-based encryption scheme. Let λ be a security parameter, k be an attribute-length parameter, and h be a message-length parameter. We define the following building blocks.

- Let $\ell, m, q, \chi, \chi', \sigma_0, \sigma, B_{\mathsf{sm}}$ be lattice and noise parameters fixed by λ, with $q \geq 2^k$, and let $\kappa := \lceil \log q \rceil$ denote the length of the binary decomposition with respect to the gadget matrix $\mathbf{G}$. Let $\mathbf{g}' = (1, 2, \ldots, 2^{\kappa-1}) \in \mathbb{Z}_q^{1\times\kappa}$.
- Let $\mathcal{X} = \{0, 1\}^k$ be the attribute space, $\mathcal{M} = \{0, 1\}^h$ be the message space, and $\Gamma = \mathbb{Z}_q$ be the batch label space.
- Let $\iota(x) := \sum_{a=1}^{k} x_a\, 2^{a-1} \in \mathbb{Z}_q$ denote the canonical embedding of an attribute vector $x = (x_1, \ldots, x_k) \in \{0, 1\}^k$ into $\mathbb{Z}_q$, extended coordinatewise to subsets $S \subseteq \{0, 1\}^k$. Since $q \geq 2^k$, ι is injective on $\{0, 1\}^k$.
- Fix constants $0 < \tau_p < \tau_c < 1$ and put $t_{\mathsf{priv}} = \lfloor \tau_p N \rfloor$, $t_{\mathsf{rec}} = \lceil \tau_c N \rceil$, with $t_{\mathsf{priv}} < t_{\mathsf{rec}}$. The parameter T in the interface denotes the reconstruction threshold t_{rec}.

The algorithms of TB-ABE are as follows.

$\mathsf{Setup}(1^\lambda, 1^B, 1^T, 1^N)$:
- Initialize the lattice and noise parameters $\ell, m, q, \chi, \chi', \sigma_0, \sigma, B_{\mathsf{sm}}$.
- Compute $(\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \mathsf{TrapGen}(1^\ell, 1^m, q)$.
- Sample $\mathbf{W} \leftarrow \mathbb{Z}_q^{2m^2\ell\times m}$ and compute $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_{2m^2} \otimes \mathbf{A} \mid \mathbf{W}], \mathbf{I}_{2m^2} \otimes \mathbf{T}_\mathbf{A}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma_0)$. Set $\mathsf{pp}_{\mathsf{com}} := (\mathbf{A}, \mathbf{W}, \mathbf{T})$.

- Sample independently $\mathbf{Q}, \mathbf{L}_0, \mathbf{L}_1, \mathbf{B}_1, \ldots, \mathbf{B}_k \leftarrow \mathbb{Z}_q^{\ell \times m}$ and $\mathbf{D} \leftarrow \mathbb{Z}_q^{\ell \times h}$.
- Compute $\big((\mathsf{psh}_1, \mathbf{D}_{1,1}, \ldots, \mathbf{D}_{1,y}), \ldots, (\mathsf{psh}_N, \mathbf{D}_{N,1}, \ldots, \mathbf{D}_{N,y})\big) \leftarrow \mathsf{Share}(1^\lambda, N, \tau_p, \tau_c, \mathbf{D})$.
- Output $\mathsf{mpk} := (\mathbf{A}, \mathbf{W}, \mathbf{T}, \mathbf{Q}, \mathbf{L}_0, \mathbf{L}_1, \mathbf{B}_1, \ldots, \mathbf{B}_k, \mathbf{D})$ and $\mathsf{msk} := \big(\mathbf{T}_\mathbf{A}, \{(\mathsf{psh}_i, \mathbf{D}_{i,1}, \ldots, \mathbf{D}_{i,y})\}_{i=1}^N\big)$.

$\mathsf{Digest}(\mathsf{mpk}, S)$:

- Parse $S = \{x_1, \ldots, x_B\} \subseteq \{0,1\}^k$, let $\iota(S) = \{\iota(x_1), \ldots, \iota(x_B)\} \subseteq \mathbb{Z}_q$.
- Compute $\gamma \leftarrow \mathsf{GenCoeff}(\iota(S))$, a binary row vector of length $\kappa(B+1)$ and $\mathbf{V} \leftarrow \mathsf{Ver}(\mathsf{pp}_{\mathsf{com}}, 1^{\kappa(B+1)})$.
- Output $\mathsf{dig}_S := -\mathbf{QV}(\gamma^\top \otimes \mathbf{I}_m) \in \mathbb{Z}_q^{\ell \times m}$.

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{dig}, f_\lambda, \mathsf{tg})$:

- Compute $\mathbf{B}_{f_\lambda} \leftarrow \mathsf{Eval}_{\mathsf{pk}}(f_\lambda, \mathbf{B}_1, \ldots, \mathbf{B}_k)$.
- Define $\mathbf{F}_{\mathsf{dig}, f_\lambda, \mathsf{tg}} := [\mathbf{A} \mid \mathsf{dig} \mid \mathbf{L}_0 + \mathsf{tg}\,\mathbf{L}_1 \mid \mathbf{B}_{f_\lambda}]$.
- Using $\mathbf{T}_\mathbf{A}$ and repeated applications of $\mathsf{ExtendRight}$, derive a trapdoor $\mathbf{T}_{\mathsf{dig}, f_\lambda, \mathsf{tg}}$ for $\mathbf{F}_{\mathsf{dig}, f_\lambda, \mathsf{tg}}$.
- For every $i \in [N]$ and $j \in [y]$, sample $\mathbf{R}_{i,j} \leftarrow \mathsf{SamplePre}(\mathbf{F}_{\mathsf{dig}, f_\lambda, \mathsf{tg}}, \mathbf{T}_{\mathsf{dig}, f_\lambda, \mathsf{tg}}, \mathbf{D}_{i,j}, \sigma) \in \mathbb{Z}^{4m \times h}$.
- Output $\{\mathsf{sk}_i\}_{i \in [N]}$, where $\mathsf{sk}_i := (f_\lambda, \mathsf{tg}, \mathsf{dig}, \mathsf{psh}_i, \mathbf{R}_{i,1}, \ldots, \mathbf{R}_{i,y})$.

$\mathsf{Enc}(\mathsf{mpk}, \mu, x, \mathsf{tg})$:

- Parse $x = (x_1, \ldots, x_k) \in \{0,1\}^k$ and $\mu \in \{0,1\}^h$.
- Sample $\mathbf{s} \leftarrow \mathbb{Z}_q^{1 \times \ell}$, $\mathbf{e}_0 \leftarrow \mathcal{D}_{\mathbb{Z}^{1 \times m}, \chi}$ and $\mathbf{S}_1, \ldots, \mathbf{S}_k \leftarrow \{\pm 1\}^{m \times m}$.
- Set $\mathbf{c}_{\mathsf{in}} := \mathbf{sA} + \mathbf{e}_0 \in \mathbb{Z}_q^{1 \times m}$.
- For every $a \in [k]$, set $\mathbf{c}_a := \mathbf{s}(x_a \mathbf{G} + \mathbf{B}_a) + \mathbf{e}_0 \mathbf{S}_a \in \mathbb{Z}_q^{1 \times m}$.
- Define $\mathsf{ID}(x) := (1, \iota(x), \iota(x)^2, \ldots, \iota(x)^B) \otimes \mathbf{g}' \in \mathbb{Z}_q^{1 \times \kappa(B+1)}$, and compute $\mathbf{C}_{\mathsf{ID}} \leftarrow \mathsf{Com}(\mathsf{pp}_{\mathsf{com}}, \mathsf{ID}(x)) \in \mathbb{Z}_q^{\ell \times m}$.
- Sample $\mathbf{e}_{\mathsf{bat}}, \mathbf{e}_{\mathsf{tg}} \leftarrow \mathcal{D}_{\mathbb{Z}^{1 \times m}, \chi'}$ and $\mathbf{e}_{\mathsf{out}} \leftarrow \mathcal{D}_{\mathbb{Z}^{1 \times h}, \chi'}$.
- Set $\mathbf{c}_{\mathsf{bat}} := \mathbf{s}(\mathbf{Q} + \mathbf{C}_{\mathsf{ID}}) + \mathbf{e}_{\mathsf{bat}} \in \mathbb{Z}_q^{1 \times m}$, $\mathbf{c}_{\mathsf{tg}} := \mathbf{s}(\mathbf{L}_0 + \mathsf{tg}\,\mathbf{L}_1) + \mathbf{e}_{\mathsf{tg}} \in \mathbb{Z}_q^{1 \times m}$, and $\mathbf{c}_{\mathsf{out}} := \mathbf{sD} + \mathbf{e}_{\mathsf{out}} + \lfloor q/2 \rfloor \mu \in \mathbb{Z}_q^{1 \times h}$.
- Output $\mathsf{ct} := (x, \mathsf{tg}, \mathbf{c}_{\mathsf{in}}, \mathbf{c}_1, \ldots, \mathbf{c}_k, \mathbf{c}_{\mathsf{bat}}, \mathbf{c}_{\mathsf{tg}}, \mathbf{c}_{\mathsf{out}})$.

$\mathsf{PDec}(\mathsf{mpk}, \mathsf{sk}_i, S, \mathsf{ct})$:

- Parse $\mathsf{sk}_i = (f_\lambda, \mathsf{tg}, \mathsf{dig}, \mathsf{psh}_i, \mathbf{R}_{i,1}, \ldots, \mathbf{R}_{i,y})$, where $\mathbf{R}_{i,j} \in \mathbb{Z}^{4m \times h}$ for every $j \in [y]$, and parse $\mathsf{ct} = (x, \mathsf{tg}', \mathbf{c}_{\mathsf{in}}, \mathbf{c}_1, \ldots, \mathbf{c}_k, \mathbf{c}_{\mathsf{bat}}, \mathbf{c}_{\mathsf{tg}'}, \mathbf{c}_{\mathsf{out}})$.
- If $\mathsf{tg}' \neq \mathsf{tg}$ or $x \notin S$, output $\perp$.
- Compute $\mathsf{dig}_S \leftarrow \mathsf{Digest}(\mathsf{mpk}, S)$. If $\mathsf{dig}_S \neq \mathsf{dig}$, output $\perp$.
- For the batch S, recompute $\gamma \leftarrow \mathsf{GenCoeff}(\iota(S)) \in \{0,1\}^{1 \times \kappa(B+1)}$ and $\mathbf{V} \leftarrow \mathsf{Ver}(\mathsf{pp}_{\mathsf{com}}, 1^{\kappa(B+1)}) \in \mathbb{Z}_q^{m \times \kappa(B+1)m}$, and compute $\mathbf{Z}_x \leftarrow \mathsf{Open}(\mathsf{pp}_{\mathsf{com}}, \mathsf{ID}(x)) \in \mathbb{Z}_q^{m \times \kappa(B+1)m}$.
- Define $\mathbf{Y}_S := \gamma^\top \otimes \mathbf{I}_m \in \mathbb{Z}_q^{\kappa(B+1)m \times m}$ and compute

$$\mathbf{c}_{\mathsf{dig}} := [\mathbf{c}_{\mathsf{in}} \mid \mathbf{c}_{\mathsf{bat}}] \begin{pmatrix} -\mathbf{Z}_x \\ -\mathbf{V} \end{pmatrix} \mathbf{Y}_S \in \mathbb{Z}_q^{1 \times m}.$$

- Compute $\mathbf{c}_{f_\lambda} \leftarrow \mathsf{Eval}_{\mathsf{ct}}\big(f_\lambda, \{x_a, \mathbf{B}_a, \mathbf{c}_a^\top\}_{a=1}^k\big) \in \mathbb{Z}_q^{m\times 1}$.
- If $f_\lambda(x) \neq 0$, output $\perp$.
- Define $\overline{\mathbf{c}} := [\mathbf{c}_{\mathsf{in}} \mid \mathbf{c}_{\mathsf{dig}} \mid \mathbf{c}_{\mathsf{tg}} \mid \mathbf{c}_{f_\lambda}^\top] \in \mathbb{Z}_q^{1\times 4m}$.
- For every $j \in [y]$, sample $\mathbf{e}_{i,j}^{\mathsf{sm}} \leftarrow [-B_{\mathsf{sm}}, B_{\mathsf{sm}}]^h$ and compute $d_{i,j} := \overline{\mathbf{c}}\,\mathbf{R}_{i,j} + \mathbf{e}_{i,j}^{\mathsf{sm}} \in \mathbb{Z}_q^{1\times h}$.
- Output $d_i = (\mathsf{psh}_i, d_{i,1}, \ldots, d_{i,y})$.

$\mathsf{Combine}(\mathsf{mpk}, \mathsf{ct}, \{d_i\}_{i\in U})$:

- If $|U| < t_{\mathsf{rec}}$, output $\perp$.
- Choose any subset $U_0 \subseteq U$ of size t_{rec}.
- Compute $\mathsf{psh} \leftarrow \mathsf{Reconst}_1(\{\mathsf{psh}_i\}_{i\in U_0})$ and derive coefficients $\{\nu_{i,j}\}_{i\in U_0, j\in[y]}$ such that $\mathbf{D} = \sum_{i\in U_0}\sum_{j\in[y]} \nu_{i,j}\mathbf{D}_{i,j}$.
- Compute $\mathbf{v} := \mathbf{c}_{\mathsf{out}} - \sum_{i\in U_0}\sum_{j\in[y]} \nu_{i,j} d_{i,j} \pmod q \in \mathbb{Z}_q^{1\times h}$.
- Output componentwise $\mu := \left\lfloor \frac{2}{q}\mathbf{v} \right\rceil \bmod 2 \in \{0,1\}^h$.

6 Analysis of TB-ABE Construction

Theorem 2 (Correctness). *Assume $q \geq 2^k$ and suppose the lattice and noise parameters satisfy the correctness condition in Eq* (1)*. Then the construction of Section 5 satisfies the decryption correctness property of Definition 4.2.*

Proof. Fix $\lambda, B, T, N, k, h \in \mathbb{N}$, a message $\mu \in \{0,1\}^h$, a tag $\mathsf{tg} \in \Gamma$, a batch $S \subseteq \{0,1\}^k$ with $|S| = B$, an attribute $x \in S$, a policy $f_\lambda \in \mathcal{F}_\lambda$ satisfying $f_\lambda(x) = 0$, and a set $U \subseteq [N]$ with $|U| \geq T$. Let $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^B, 1^T, 1^N)$, $\mathsf{dig}_S \leftarrow \mathsf{Digest}(\mathsf{mpk}, S)$, $\{\mathsf{sk}_i\}_{i\in[N]} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{dig}_S, f_\lambda, \mathsf{tg})$, and $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mu, x, \mathsf{tg})$. For every $i \in U$, let $d_i \leftarrow \mathsf{PDec}(\mathsf{mpk}, \mathsf{sk}_i, S, \mathsf{ct})$, and let $U_0 \subseteq U$ with $|U_0| = T$ be the subset used by $\mathsf{Combine}$.

Let $\gamma \leftarrow \mathsf{GenCoeff}(\iota(S))$ and $\mathbf{Y}_S := \gamma^\top \otimes \mathbf{I}_m$. By $\mathsf{GenCoeff}$ algorithm, $\mathsf{ID}(x)\gamma^\top = P_{\iota(S)}(\iota(x))$. Since $x \in S$ and ι is injective, $\iota(x) \in \iota(S)$, and hence $\mathsf{ID}(x)\gamma^\top = 0$. Recall that $\mathbf{c}_{\mathsf{in}} = \mathbf{sA} + \mathbf{e}_0$ and $\mathbf{c}_{\mathsf{bat}} = \mathbf{s}(\mathbf{Q} + \mathbf{C}_{\mathsf{ID}}) + \mathbf{e}_{\mathsf{bat}}$. Therefore,

$$\begin{aligned}\mathbf{c}_{\mathsf{dig}} &= [\mathbf{c}_{\mathsf{in}} \mid \mathbf{c}_{\mathsf{bat}}] \begin{pmatrix} -\mathbf{Z}_x \\ -\mathbf{V} \end{pmatrix} \mathbf{Y}_S \\ &= \mathbf{s}\big(-\mathbf{A}\mathbf{Z}_x - (\mathbf{Q} + \mathbf{C}_{\mathsf{ID}})\mathbf{V}\big)\mathbf{Y}_S + \mathbf{e}_{\mathsf{dig}},\end{aligned}$$

where $\mathbf{e}_{\mathsf{dig}} := [\mathbf{e}_0 \mid \mathbf{e}_{\mathsf{bat}}]\binom{-\mathbf{Z}_x}{-\mathbf{V}}\mathbf{Y}_S$. By Lemma 10, $\mathbf{C}_{\mathsf{ID}}\mathbf{V} = \mathsf{ID}(x) \otimes \mathbf{G} - \mathbf{A}\mathbf{Z}_x$, and thus, $\mathbf{c}_{\mathsf{dig}} = -\mathbf{sQVY}_S - \mathbf{s}(\mathsf{ID}(x) \otimes \mathbf{G})\mathbf{Y}S + \mathbf{e}_{\mathsf{dig}} = \mathbf{s}\mathsf{dig}S - \mathbf{s}\big(\mathsf{ID}(x)\gamma^\top\big)\mathbf{G} + \mathbf{e}_{\mathsf{dig}} = \mathbf{s}\mathsf{dig}S + \mathbf{e}_{\mathsf{dig}}$. By Lemma 2, except with negligible probability, $\|\mathbf{e}_0\| \leq c_{\mathsf{G}}\chi\sqrt{m}$ and $\|\mathbf{e}_{\mathsf{bat}}\| \leq c_{\mathsf{G}}\chi'\sqrt{m}$. Together with the bounds on $\mathbf{V}$ and $\mathbf{Z}_x$ from Lemma 10 and $\|\mathbf{Y}_S\|_2 = \|\gamma\|_2 \leq \sqrt{\kappa(B+1)}$, this yields $\|\mathbf{e}_{\mathsf{dig}}\| \leq (\chi + \chi')\, O\Big(\|\mathbf{T}\| m^8 B \log B \log^2 q\Big) =: \Delta_{\mathsf{dig}}$. For every $a \in [k]$, $\mathbf{c}_a = \mathbf{s}(x_a\mathbf{G} + \mathbf{B}_a) + \mathbf{e}_0\mathbf{S}_a$. By Lemmas 2 and 3, $\|\mathbf{S}_a\|_2 \leq 20\sqrt{m}$ and hence $\|\mathbf{e}_0\mathbf{S}_a\| \leq 20c_{\mathsf{G}}\chi m$, except with negligible probability. The correctness property of $\mathsf{Eval}_{\mathsf{ct}}$ from Lemma 9 therefore gives $\mathbf{c}_{f_\lambda}^\top = \mathbf{s}\big(f_\lambda(x)\mathbf{G} + \mathbf{B}_{f_\lambda}\big) +$

$\mathbf{e}_{f_\lambda}^\top, \|\mathbf{e}_{f_\lambda}\| \le 20 c_{\mathsf{G}} \alpha_{\mathcal{F}}(\ell) \chi m =: \Delta_f$. Since $f_\lambda(x) = 0$, we have $\mathbf{c}_{f_\lambda}^\top = \mathbf{s}\mathbf{B}_{f_\lambda} + \mathbf{e}_{f_\lambda}^\top$. It follows, using $\mathbf{c}_{\mathsf{tg}} = \mathbf{s}(\mathbf{L}_0 + \mathsf{tg}\mathbf{L}_1) + \mathbf{e}_{\mathsf{tg}}$, that $\overline{\mathbf{c}} = \mathbf{s}\mathbf{F}_{\mathsf{dig}_S, f_\lambda, \mathsf{tg}} + \mathbf{e}_{\mathsf{ct}}$, where $\mathbf{e}_{\mathsf{ct}} := [\mathbf{e}_0 \mid \mathbf{e}_{\mathsf{dig}} \mid \mathbf{e}_{\mathsf{tg}} \mid \mathbf{e}_{f_\lambda}^\top]$. Again by Lemma 2, $\|\mathbf{e}_{\mathsf{tg}}\| \le c_{\mathsf{G}} \chi' \sqrt{m}$ except with negligible probability. Consequently, $\|\mathbf{e}_{\mathsf{ct}}\| \le \Delta_{\mathsf{ct}} := c_{\mathsf{G}}(\chi + \chi')\sqrt{m} + \Delta_{\mathsf{dig}} + \Delta_f$, and, explicitly, $\Delta_{\mathsf{ct}} \le c_{\mathsf{G}}(\chi + \chi')\sqrt{m} + (\chi + \chi') O\left(\|\mathbf{T}\| m^8 B \log B \log^2 q\right) + 20 c_{\mathsf{G}} \alpha_{\mathcal{F}}(\ell) \chi m$. For every $i \in [N]$ and $j \in [y]$, key generation ensures $\mathbf{F}_{\mathsf{dig}_S, f_\lambda, \mathsf{tg}} \mathbf{R}_{i,j} = \mathbf{D}_{i,j} \bmod q$ by Lemma 1. Hence, $d_{i,j} = \mathbf{s}\mathbf{D}_{i,j} + \mathbf{e}_{\mathsf{ct}}\mathbf{R}_{i,j} + \mathbf{e}_{i,j}^{\mathsf{sm}}$. Since $\mathbf{R}_{i,j} \in \mathbb{Z}^{4m \times h}$ is statistically close to a discrete-Gaussian preimage matrix of parameter σ, Lemma 3 gives $\|\mathbf{R}_{i,j}\|_2 \le \sigma\sqrt{4mh} = 2\sigma\sqrt{mh}$ except with negligible probability. Moreover, $\|\mathbf{e}_{i,j}^{\mathsf{sm}}\| \le B_{\mathsf{sm}}\sqrt{h}$. Therefore, $\left\|\mathbf{e}_{\mathsf{ct}}\mathbf{R}_{i,j} + \mathbf{e}_{i,j}^{\mathsf{sm}}\right\| \le 2\sigma\sqrt{mh}\,\Delta_{\mathsf{ct}} + B_{\mathsf{sm}}\sqrt{h} =: \Delta_{\mathsf{share}}$. Since $|U_0| = t_{\mathsf{rec}} \ge \tau_c N$, correctness of the ramp sharing scheme (Definition 3), together with the linear reconstruction and small-coefficient properties of Theorem 1, gives coefficients $\{\nu_{i,j}\}_{i \in U_0, j \in [y]}$ satisfying $\mathbf{D} = \sum_{i \in U_0} \sum_{j \in [y]} \nu_{i,j} \mathbf{D}_{i,j}$ and $|\nu_{i,j}| \le \nu(N)$. Consequently, $\sum_{i \in U_0} \sum_{j \in [y]} \nu_{i,j} d_{i,j} = \mathbf{s}\mathbf{D} + \mathbf{e}_{\mathsf{rec}}$, where $\mathbf{e}_{\mathsf{rec}} := \sum_{i \in U_0} \sum_{j \in [y]} \nu_{i,j}(\mathbf{e}_{\mathsf{ct}}\mathbf{R}_{i,j} + \mathbf{e}_{i,j}^{\mathsf{sm}})$, and therefore $\|\mathbf{e}_{\mathsf{rec}}\| \le t_{\mathsf{rec}} y \nu(N) \left(2\sigma\sqrt{mh}\,\Delta_{\mathsf{ct}} + B_{\mathsf{sm}}\sqrt{h}\right)$. Finally, $\mathbf{c}_{\mathsf{out}} = \mathbf{s}\mathbf{D} + \mathbf{e}_{\mathsf{out}} + \lfloor q/2 \rfloor \mu$, where Lemma 2 gives $\|\mathbf{e}_{\mathsf{out}}\| \le c_{\mathsf{G}} \chi' \sqrt{h}$ except with negligible probability. Thus, the value computed by $\mathsf{Combine}$ is $\mathbf{v} = \left\lfloor \frac{q}{2} \right\rfloor \mu + \mathbf{e}_{\mathsf{out}} - \mathbf{e}_{\mathsf{rec}}$. In particular, the total error has norm at most $c_{\mathsf{G}} \chi' \sqrt{h} + t_{\mathsf{rec}} y \nu(N) \left(2\sigma\sqrt{mh}\,\Delta_{\mathsf{ct}} + B_{\mathsf{sm}}\sqrt{h}\right)$. Accordingly, we require

$$q > 4 \left[c_{\mathsf{G}} \chi' \sqrt{h} + t_{\mathsf{rec}} y \nu(N) \left(2\sigma\sqrt{mh}\,\Delta_{\mathsf{ct}} + B_{\mathsf{sm}}\sqrt{h}\right) \right]. \tag{1}$$

Under Eq (1), every coordinate of $\mathbf{e}_{\mathsf{out}} - \mathbf{e}_{\mathsf{rec}}$ has magnitude strictly less than $q/4$, so componentwise rounding of $(2/q)\mathbf{v}$ modulo 2 recovers μ. All Gaussian-tail, Gaussian-matrix, trapdoor-sampling, and secret-sharing correctness events invoked above fail with only negligible probability. Since there are polynomially many such events, a union bound gives $\Pr\left[\mathsf{Combine}(\mathsf{mpk}, \mathsf{ct}, \{d_i\}_{i \in U}) = \mu\right] \ge 1 - \mathsf{negl}(\lambda)$, which proves the claim.

Theorem 3 (Selective Security of TB-ABE). *Assume the $2m^2$-succinct LWE assumption (Assumption 3.6) and a linear ramp-threshold secret sharing scheme satisfying share uniformity (Definition 4). Then, the construction given in Section 5 is selectively secure (Definition 4.3).*

Proof. Fix an admissible PPT adversary $\mathcal{A}$ and its selectively chosen challenge attribute $x^* \in \{0,1\}^k$ and challenge tag $\mathsf{tg}^* \in \Gamma$. Let $\mathsf{ID}^* := (1, \iota(x^*), \ldots, \iota(x^*)^B) \otimes \mathbf{g}'$ and $\mathbf{C}_{\mathsf{ID}}^* \leftarrow \mathsf{Com}(\mathsf{pp}_{\mathsf{com}}, \mathsf{ID}^*)$. After receiving mpk, let $Q^* = \{(i, S_i^*, f_i^*)\}_{i \in C^*}$ denote the query set declared by $\mathcal{A}$, where $|C^*| \le t_{\mathsf{priv}}$, $x^* \in S_i^*$, and $f_i^*(x^*) = 0$ for every $i \in C^*$. We consider the following sequence of hybrids.

Hyb_0. This is the real selective-security experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{tb\text{-}abe}}(\lambda, b)$.

Hyb_1. Sample $\mathbf{U}_{\mathsf{bat}}, \mathbf{U}_{\mathsf{tg}} \leftarrow \{0,1\}^{m \times m}$, $\mathbf{U}_{\mathbf{D}} \leftarrow \{0,1\}^{m \times h}$, $\mathbf{S}_1^*, \ldots, \mathbf{S}_k^* \leftarrow \{\pm 1\}^{m \times m}$, and $(\mathbf{L}_1, \mathbf{T}_{\mathbf{L}_1}) \leftarrow \mathsf{TrapGen}(1^\ell, 1^m, q)$. Program $\mathbf{Q} := \mathbf{A}\mathbf{U}_{\mathsf{bat}} - \mathbf{C}_{\mathsf{ID}}^*$, $\mathbf{L}_0 := \mathbf{A}\mathbf{U}_{\mathsf{tg}} - \mathsf{tg}^*\mathbf{L}_1$, and $\mathbf{D} := \mathbf{A}\mathbf{U}_{\mathbf{D}}$ and $\mathbf{B}_a := \mathbf{A}\mathbf{S}_a^* - x_a^*\mathbf{G}$ for every $a \in [k]$. The matrices $\mathbf{S}_a^*$ are also used in the challenge encryption. All secret shares and keys are otherwise generated honestly.

Hyb$_2$. Here, for every $(i, S_i^*, f_i^*) \in Q^*$, compute $\mathsf{dig}_i^* \leftarrow \mathsf{Digest}(\mathsf{mpk}, S_i^*)$ and define $\mathbf{F}_i^* := \mathbf{F}_{\mathsf{dig}_i^*, f_i^*, \mathsf{tg}^*}$, namely $\mathbf{F}_i^* = [\mathbf{A} \mid \mathsf{dig}_i^* \mid \mathbf{L}_0 + \mathsf{tg}^* \mathbf{L}_1 \mid \mathbf{B}_{f_i^*}]$. For every $j \in [y]$, sample $\mathbf{R}_{i,j}^* \leftarrow D_{\mathbb{Z}^{4m \times h}, \sigma}$ and define $\mathbf{D}_{i,j} := \mathbf{F}_i^* \mathbf{R}_{i,j}^* \bmod q$. The corresponding public share psh_i is sampled uniformly from its share space. After these values have been fixed for all $i \in C^*$, the challenger completes the ramp sharing of $\mathbf{D}$ conditioned on the prescribed shares $\{(\mathsf{psh}_i, \mathbf{D}_{i,1}, \ldots, \mathbf{D}_{i,y})\}_{i \in C^*}$. Whenever a declared tuple $(i, S_i^*, f_i^*) \in Q^*$ is subsequently queried, the challenger returns the programmed key share $\mathsf{sk}_i^* = (f_i^*, \mathsf{tg}^*, \mathsf{dig}_i^*, \mathsf{psh}_i, \mathbf{R}_{i,1}^*, \ldots, \mathbf{R}_{i,y}^*)$. All remaining key material is generated honestly using $\mathbf{T}_\mathbf{A}$.

Hyb$_3$. We change the challenge ciphertext. Let $\mathbf{c}_{\mathsf{in}}^* = \mathbf{s}\mathbf{A} + \mathbf{e}_0$. Since $x_a^* \mathbf{G} + \mathbf{B}_a = \mathbf{A}\mathbf{S}_a^*$, set $\mathbf{c}_a^* := \mathbf{c}_{\mathsf{in}}^* \mathbf{S}_a^*$ for every $a \in [k]$. Moreover, set

$$\mathbf{c}_{\mathsf{bat}}^* := \mathsf{ReRand}\left(\mathbf{U}_{\mathsf{bat}}, \mathbf{c}_{\mathsf{in}}^*, \frac{\chi'}{2\chi}\right), \qquad \mathbf{c}_{\mathsf{tg}}^* := \mathsf{ReRand}\left(\mathbf{U}_{\mathsf{tg}}, \mathbf{c}_{\mathsf{in}}^*, \frac{\chi'}{2\chi}\right),$$
$$\mathbf{c}_{\mathsf{out}}^* := \mathsf{ReRand}\left(\mathbf{U}_{\mathbf{D}}, \mathbf{c}_{\mathsf{in}}^*, \frac{\chi'}{2\chi}\right) + \left\lfloor \frac{q}{2} \right\rfloor \mu_b.$$

Hyb$_4$. Whenever key material for a challenge-incompatible tuple satisfying $x^* \notin S$ is generated, either through O_{KeyGen} or internally while answering a partial-decryption query, it is now generated using a trapdoor derived from the nonzero value $P_{\iota(S)}(\iota(x^*))$, instead of $\mathbf{T}_\mathbf{A}$.

Hyb$_5$. Whenever key material for a remaining challenge-incompatible tuple with $\mathsf{tg} \neq \mathsf{tg}^*$ is generated, either through O_{KeyGen} or internally while answering a partial-decryption query, it is now generated using the trapdoor $\mathbf{T}_{\mathbf{L}_1}$.

Hyb$_6$. Every remaining challenge-incompatible tuple necessarily satisfies $x^* \in S$, $\mathsf{tg} = \mathsf{tg}^*$, and $f_\lambda(x^*) \neq 0$. Whenever key material for such a tuple is generated, either through O_{KeyGen} or internally while answering a partial-decryption query, it is now generated using $\mathsf{Eval}_{\mathsf{sim}}$ and the public gadget trapdoor $\mathbf{T}_\mathbf{G}$.

Hyb$_7$. For a valid non-challenge query, the supplied well-formed encryption randomness determines the encryption secret $\mathbf{s}$. Instead of invoking PDec with a secret key, the challenger returns $d_i' = (\mathsf{psh}_i, d_{i,1}', \ldots, d_{i,y}')$, where $d_{i,j}' := \mathbf{s}\mathbf{D}_{i,j} + \mathbf{e}_{i,j}'$ and $\mathbf{e}_{i,j}' \leftarrow [-B_{\mathsf{sm}}, B_{\mathsf{sm}}]^h$. If $\mathsf{ct} = \mathsf{ct}^*$, the security experiment guarantees that the query corresponds to a declared tuple $(i, S_i^*, f_i^*) \in Q^*$. In this case, the challenger uses the programmed matrices $\{\mathbf{R}_{i,j}^*\}_{j \in [y]}$ from **Hyb$_2$** and returns $d_i^* = (\mathsf{psh}_i, d_{i,1}^*, \ldots, d_{i,y}^*)$, where $d_{i,j}^* := \overline{\mathbf{c}}^* \mathbf{R}_{i,j}^* + \mathbf{e}_{i,j}^{\mathsf{sm}}, \mathbf{e}_{i,j}^{\mathsf{sm}} \leftarrow [-B_{\mathsf{sm}}, B_{\mathsf{sm}}]^h$.

Hyb$_8$. This hybrid is identical to **Hyb$_7$**, except that $\mathbf{c}_{\mathsf{in}}^*$ in the challenge ciphertext is sampled uniformly from $\mathbb{Z}_q^{1 \times m}$. All remaining challenge components are computed from it exactly as in **Hyb$_3$**.

Hyb$_9$. This hybrid is identical to **Hyb$_8$**, except that $\mathbf{c}_{\mathsf{out}}^*$ is replaced by an independent uniform vector in $\mathbb{Z}_q^{1 \times h}$.

We now establish indistinguishability of adjacent hybrids.

Lemma 11. *For every $b \in \{0, 1\}$, $\mathbf{Hyb}_0(b) \approx_s \mathbf{Hyb}_1(b)$.*

Proof. Convert $\mathbf{U}_{\mathsf{bat}}, \mathbf{U}_{\mathsf{tg}}, \mathbf{U}_{\mathbf{D}}$ into sign matrices through the affine map $\widetilde{\mathbf{U}} = 2\mathbf{U} - \mathbf{J}$, and concatenate them with $\mathbf{S}_1^*, \ldots, \mathbf{S}_k^*$. The resulting matrix has polynomial width, so Lemma 5 implies that $\mathbf{AU}_{\mathsf{bat}}, \mathbf{AU}_{\mathsf{tg}}, \mathbf{AU}_{\mathbf{D}}$ and $\mathbf{AS}_1^*, \ldots, \mathbf{AS}_k^*$ are jointly statistically close to independent uniform matrices, while retaining the correlated challenge-error terms $\mathbf{e}_0\mathbf{S}_a^*$. The affine shifts by $-\mathbf{C}_{\mathsf{ID}}^*$, $-\mathsf{tg}^*\mathbf{L}_1$, and $-x_a^*\mathbf{G}$ preserve uniformity. By Lemma 10, $\mathbf{C}_{\mathsf{ID}}^*$ depends only on $(\mathbf{A}, \mathbf{W}, \mathbf{T})$ and ID^* and not on $\mathbf{Q}$. Finally, Lemma 1 implies that the marginal distribution of $\mathbf{L}_1$ produced by $\mathsf{TrapGen}$ is statistically close to uniform. Hence, the complete views differ by only negligible statistical distance.

Lemma 12. *For every $b \in \{0,1\}$, $\mathbf{Hyb}_1(b) \approx_s \mathbf{Hyb}_2(b)$.*

Proof. Let $Q^* = \{(i, S_i^*, f_i^*)\}_{i \in C^*}$ be the set declared by the adversary, before the query phase. By admissibility, $|C^*| \le T - 1 = \tau_p N$. By share uniformity (Definition 4), the joint tuple $\{(\mathsf{psh}_i, \mathbf{D}_{i,1}, \ldots, \mathbf{D}_{i,y})\}_{i \in C}$ is uniform over the corresponding share spaces and independent of $\mathbf{D}$. Fix a declared tuple $(i, S_i^*, f_i^*) \in Q^*$, and write $\mathbf{F}_i^* := \mathbf{F}_{S_i^*, f_i^*, \mathsf{tg}^*}$. Honest key generation samples $\mathbf{R}_{i,j}$ statistically close to $D_\sigma(\Lambda_q^{\mathbf{D}_{i,j}}(\mathbf{F}))$ by Lemma 1. Lemma 8 implies that the joint distribution obtained by choosing uniform $\mathbf{D}_{i,j}$ and subsequently sampling the corresponding Gaussian preimage is statistically close to the distribution generated by sampling $\mathbf{R}_{i,j} \leftarrow D_{\mathbb{Z}^{4m \times h}, \sigma}$ and setting $\mathbf{D}_{i,j} := \mathbf{FR}_{i,j} \bmod q$. This argument extends to all polynomially many queried matrices and columns. Once these shares are fixed, the challenger samples the remaining secret-sharing randomness conditioned on the prescribed shares and the shared matrix $\mathbf{D}$. Consequently, the complete secret-sharing state in $\mathbf{Hyb}_2$ is statistically indistinguishable from $\mathbf{Hyb}_1$.

Lemma 13. *For every $b \in \{0,1\}$, $\mathbf{Hyb}_2(b) \approx_s \mathbf{Hyb}_3(b)$.*

Proof. For the attribute components, the distributions are identical since $\mathbf{B}_a = \mathbf{AS}_a^* - x_a^*\mathbf{G}$ gives $\mathbf{c}_a^* = \mathbf{c}_{\mathsf{in}}^*\mathbf{S}_a^*$. Moreover, $\mathbf{Q} + \mathbf{C}_{\mathsf{ID}}^* = \mathbf{AU}_{\mathsf{bat}}$, $\mathbf{L}_0 + \mathsf{tg}^*\mathbf{L}_1 = \mathbf{AU}_{\mathsf{tg}}$, and $\mathbf{D} = \mathbf{AU}_{\mathbf{D}}$. The remaining replacements, therefore, follow directly from Lemma 6, provided $\chi > \max\{\omega(\sqrt{m}), \omega(\sqrt{h})\}$ and $\chi'/(2\chi)$ dominates the operator norms of $\mathbf{U}_{\mathsf{bat}}$, $\mathbf{U}_{\mathsf{tg}}$, and $\mathbf{U}_{\mathbf{D}}$.

Lemma 14. *For every $b \in \{0,1\}$, $\mathbf{Hyb}_3(b) \approx_s \mathbf{Hyb}_4(b)$.*

Proof. Consider a query with $x^* \notin S$. Let $\gamma \leftarrow \mathsf{GenCoeff}(\iota(S))$, $\mathbf{Y}_S := \gamma^\top \otimes \mathbf{I}_m$, $\mathbf{Z}^* \leftarrow \mathsf{Open}(\mathsf{pp}_{\mathsf{com}}, \mathsf{ID}^*)$, and $\mathbf{V} \leftarrow \mathsf{Ver}(\mathsf{pp}_{\mathsf{com}}, 1^{\kappa(B+1)})$. Using $\mathbf{Q} = \mathbf{AU}_{\mathsf{bat}} - \mathbf{C}_{\mathsf{ID}}^*$ and Lemma 10, $\mathsf{dig}_S = -\mathbf{QVY}_S = -\mathbf{A}(\mathbf{U}_{\mathsf{bat}}\mathbf{V} + \mathbf{Z}^*)\mathbf{Y}_S + (\mathsf{ID}^* \otimes \mathbf{G})\mathbf{Y}_S = \mathbf{AR}_S + \alpha_S\mathbf{G}$ where $\mathbf{R}_S := -(\mathbf{U}_{\mathsf{bat}}\mathbf{V} + \mathbf{Z}^*)\mathbf{Y}_S$ and, by $\mathsf{GenCoeff}$ algorithm, $\alpha_S = P_{\iota(S)}(\iota(x^*))$. Since $x^* \notin S$ and ι is injective, $\mathsf{GenCoeff}$ algorithm gives $\alpha_S \in \mathbb{Z}_q^*$. Thus, $\mathbf{T}_{\mathbf{G}}$ is also a trapdoor for $\alpha_S\mathbf{G}$, and Lemma 1 allows the challenger to apply $\mathsf{ExtendLeft}$ to obtain a trapdoor for $[\mathbf{A} \mid \mathsf{dig}_S]$, then append the tag and policy blocks using $\mathsf{ExtendRight}$. The resulting trapdoor is valid for $\mathbf{F}_{S, f_\lambda, \mathsf{tg}}$. Sampling with this trapdoor is statistically indistinguishable from honest key generation.

Lemma 15. *For every $b \in \{0,1\}$, $\mathbf{Hyb}_4(b) \approx_s \mathbf{Hyb}_5(b)$.*

Proof. Consider a remaining query with $x^* \in S$ and $\mathsf{tg} \neq \mathsf{tg}^*$. Since $\mathbf{L}_0 = \mathbf{AU}_{\mathsf{tg}} - \mathsf{tg}^*\mathbf{L}_1$, we have $\mathbf{L}_0 + \mathsf{tg}\mathbf{L}_1 = \mathbf{AU}_{\mathsf{tg}} + (\mathsf{tg} - \mathsf{tg}^*)\mathbf{L}_1$. Let $\mathbf{A}' := [\mathbf{A} \mid \mathsf{dig}_S]$ and $\mathbf{R}' := (\mathbf{U}_{\mathsf{tg}}^\top \mid \mathbf{0}^{m \times m})^\top$, so that $\mathbf{A}'\mathbf{R}' = \mathbf{AU}_{\mathsf{tg}}$. Since $\Gamma = \mathbb{Z}_q$ and $\mathsf{tg} - \mathsf{tg}^* \in \mathbb{Z}_q^*$, $\mathbf{T}_{\mathbf{L}_1}$ is a trapdoor for $(\mathsf{tg} - \mathsf{tg}^*)\mathbf{L}_1$. Lemma 1 therefore gives, via $\mathsf{ExtendLeft}$, a trapdoor for $[\mathbf{A} \mid \mathsf{dig}_S \mid \mathbf{L}_0 + \mathsf{tg}\mathbf{L}_1]$, after which $\mathbf{B}_{f_\lambda}$ is appended by $\mathsf{ExtendRight}$ proving the claim.

Lemma 16. *For every* $b \in \{0,1\}$, $\mathbf{Hyb}_5(b) \approx_s \mathbf{Hyb}_6(b)$.

Proof. Every remaining challenge-incompatible query satisfies $x^* \in S$, $\mathsf{tg} = \mathsf{tg}^*$, and $f_\lambda(x^*) \neq 0$. By Lemma 9, the challenger computes $\mathbf{S}_{f_\lambda} \leftarrow \mathsf{Eval}_{\mathsf{sim}}(f_\lambda, ((x_a^*, \mathbf{S}_a^*))_{a=1}^k, \mathbf{A})$ such that $\mathbf{B}_{f_\lambda} = \mathbf{A}\mathbf{S}_{f_\lambda} - f_\lambda(x^*)\mathbf{G}$, $\|\mathbf{S}_{f_\lambda}\|_2 < \alpha_{\mathcal{F}}(\ell)$ except with negligible probability. Set $\beta := -f_\lambda(x^*) \in \mathbb{Z}_q^*$. Then, $\mathbf{B}_{f_\lambda} = \mathbf{A}\mathbf{S}_{f_\lambda} + \beta\mathbf{G}$. Using $\mathbf{T}_{\mathbf{G}}$ and Lemma 1, $\mathsf{ExtendLeft}$ gives a trapdoor for $[\mathbf{A} \mid \mathbf{B}_{f_\lambda}]$, the digest and tag blocks are appended using $\mathsf{ExtendRight}$ and the columns are publicly permuted into the order of $\mathbf{F}_{S,f_\lambda,\mathsf{tg}^*}$. This then gives statistical indistinguishability from honest key generation.

Lemma 17. *For every* $b \in \{0,1\}$, $\mathbf{Hyb}_6(b) \approx_s \mathbf{Hyb}_7(b)$.

Proof. Consider first a valid partial-decryption query for $\mathsf{ct} \neq \mathsf{ct}^*$ with $r \in \mathcal{R}_{\mathsf{Enc}}$. By Theorem 2, $\overline{\mathbf{c}} = \mathbf{s}\mathbf{F}_{S,f_\lambda,\mathsf{tg}} + \mathbf{e}_{\mathsf{ct}}$, and hence $d_{i,j} = \mathbf{s}\mathbf{D}_{i,j} + \boldsymbol{\eta}_{i,j} + \mathbf{e}_{i,j}^{\mathsf{sm}}$, where $\boldsymbol{\eta}_{i,j} := \mathbf{e}_{\mathsf{ct}}\mathbf{R}_{i,j}$. The well-formedness of r, together with Lemma 3, gives $\|\boldsymbol{\eta}_{i,j}\| \leq B_{\mathsf{err}}$ except with negligible probability. Since the parameters satisfy $hB_{\mathsf{err}}/B_{\mathsf{sm}} = \mathsf{negl}(\lambda)$, Lemma 7 implies that this response is statistically indistinguishable from $d'_{i,j} = \mathbf{s}\mathbf{D}_{i,j} + \mathbf{e}'_{i,j}$, where $\mathbf{e}'_{i,j} \leftarrow [-B_{\mathsf{sm}}, B_{\mathsf{sm}}]^h$. For a permitted query on ct^*, query selectivity implies that $(i, S_i^*, f_i^*) \in Q^*$. The matrices $\{\mathbf{R}_{i,j}^*\}_{j\in[y]}$ were already fixed in $\mathbf{Hyb}_2$, and both hybrids return $d_{i,j}^* = \overline{\mathbf{c}}^*\mathbf{R}_{i,j}^* + \mathbf{e}_{i,j}^{\mathsf{sm}}$. Thus, challenge partial-decryption responses are identically distributed, while nonchallenge responses differ only negligibly by smudging. A union bound over polynomially many queries proves the claim.

Lemma 18. *Under Assumption 3.6, for every* $b \in \{0,1\}$, $\mathbf{Hyb}_7(b) \approx_c \mathbf{Hyb}_8(b)$.

Proof. Suppose $\mathcal{A}$ distinguishes the two hybrids with non-negligible advantage. We construct a reconstruction $\mathcal{B}$ against $2m^2$-succinct LWE. The challenger gives $\mathcal{B}$ a tuple $(\mathbf{A}, \mathbf{u}, \mathbf{W}, \mathbf{T})$, where either $\mathbf{u} = \mathbf{s}\mathbf{A} + \mathbf{e}_0$ or $\mathbf{u} \leftarrow \mathbb{Z}_q^{1\times m}$ is uniform. The reduction sets $\mathsf{pp}_{\mathsf{com}} = (\mathbf{A}, \mathbf{W}, \mathbf{T})$, computes $\mathbf{C}_{\mathsf{ID}}^*$, samples $\mathbf{U}_{\mathsf{bat}}, \mathbf{U}_{\mathsf{tg}} \leftarrow \{0,1\}^{m\times m}$, $\mathbf{U}_{\mathbf{D}} \leftarrow \{0,1\}^{m\times h}$ and $\mathbf{S}_1^*, \ldots, \mathbf{S}_k^* \leftarrow \{\pm 1\}^{m\times m}$, generates $(\mathbf{L}_1, \mathbf{T}_{\mathbf{L}_1}) \leftarrow \mathsf{TrapGen}(1^\ell, 1^m, q)$, and programs $\mathbf{Q} = \mathbf{A}\mathbf{U}_{\mathsf{bat}} - \mathbf{C}_{\mathsf{ID}}^*$, $\mathbf{L}_0 = \mathbf{A}\mathbf{U}_{\mathsf{tg}} - \mathsf{tg}^*\mathbf{L}_1$, $\mathbf{D} = \mathbf{A}\mathbf{U}_{\mathbf{D}}$, and $\mathbf{B}_a = \mathbf{A}\mathbf{S}_a^* - x_a^*\mathbf{G}$. After receiving mpk, $\mathcal{A}$ declares $Q^* = \{(i, S_i^*, f_i^*)\}_{i\in C^*}$ with $|C^*| \leq t_{\mathsf{priv}}$. For every declared tuple, $\mathcal{B}$ computes $\mathsf{dig}_i^* \leftarrow \mathsf{Digest}(\mathsf{mpk}, S_i^*)$ and $\mathbf{F}_i^* := \mathbf{F}_{\mathsf{dig}_i^*, f_i^*, \mathsf{tg}^*}$, samples $\mathbf{R}_{i,j}^* \leftarrow D_{\mathbb{Z}^{4m\times h},\sigma}$, and sets $\mathbf{D}_{i,j} := \mathbf{F}_i^*\mathbf{R}_{i,j}^* \bmod q$. By share uniformity, the remaining shares of $\mathbf{D}$ can be sampled conditioned on these prescribed values. Declared fully challenge-compatible key queries are answered using the programmed $\mathbf{R}_{i,j}^*$, while all challenge-incompatible keys are generated using the alternative trapdoors of $\mathbf{Hyb}_4$-$\mathbf{Hyb}_6$. Valid nonchallenge partial-decryption queries are answered as in $\mathbf{Hyb}_7$ by returning $d'_{i,j} = \mathbf{s}\mathbf{D}_{i,j} + \mathbf{e}'_{i,j}$ using the encryption secret $\mathbf{s}$ contained in the supplied well-formed randomness. At challenge time, on receiving $\mu_0, \mu_1 \in \mathcal{M}$, $\mathcal{B}$ samples $b \leftarrow \{0,1\}$, sets $\mathbf{c}_{\mathsf{in}}^* := \mathbf{u}$ and $\mathbf{c}_a^* := \mathbf{u}\mathbf{S}_a^*$ for every $a \in [k]$, and computes $\mathbf{c}_{\mathsf{bat}}^* := \mathsf{ReRand}(\mathbf{U}_{\mathsf{bat}}, \mathbf{u}, \chi'/(2\chi))$, $\mathbf{c}_{\mathsf{tg}}^* := \mathsf{ReRand}(\mathbf{U}_{\mathsf{tg}}, \mathbf{u}, \chi'/(2\chi))$, and $\mathbf{c}_{\mathsf{out}}^* := \mathsf{ReRand}(\mathbf{U}_{\mathbf{D}}, \mathbf{u}, \chi'/(2\chi)) + \lfloor q/2 \rfloor \mu_b$. For a permitted partial decryption of ct^*, the queried tuple belongs to Q^*, so $\mathcal{B}$ computes $\overline{\mathbf{c}}^*$ from the challenge ciphertext and returns $d_{i,j}^* = \overline{\mathbf{c}}^*\mathbf{R}_{i,j}^* + \mathbf{e}_{i,j}^{\mathsf{sm}}$. Hence, no oracle response requires either the hidden LWE secret or $\mathbf{T}_{\mathbf{A}}$. If $\mathbf{u} = \mathbf{s}\mathbf{A} + \mathbf{e}_0$, the complete view of $\mathcal{A}$ is distributed as in $\mathbf{Hyb}_7$, whereas if $\mathbf{u}$ is uniform it is distributed as in $\mathbf{Hyb}_8$. Therefore, a non-negligible distinguisher between the two hybrids contradicts Assumption 3.6.

Lemma 19. *For every* $b \in \{0,1\}$, $\boldsymbol{Hyb}_8(b) \approx_s \boldsymbol{Hyb}_9(b)$.

Proof. In $\mathbf{Hyb}_8$, $\mathbf{c}^*_{\mathsf{in}} \leftarrow \mathbb{Z}_q^{1\times m}$ is uniform and $\mathbf{c}^*_{\mathsf{out}} = \mathsf{ReRand}\left(\mathbf{U_D}, \mathbf{c}^*_{\mathsf{in}}, \frac{\chi'}{2\chi}\right) + \left\lfloor \frac{q}{2} \right\rfloor \mu_b$. By Lemma 6, this is statistically close to $\mathbf{rU_D} + \mathbf{e}' + \lfloor q/2 \rfloor \mu_b$ for uniform $\mathbf{r} \in \mathbb{Z}_q^{1\times m}$ and the corresponding Gaussian error $\mathbf{e}'$. By Lemma 1 and the choice of h providing the required entropy slack, $\mathbf{rU_D} + \mathbf{e}'$ is statistically close to an independent uniform $\mathbf{u}_{\mathsf{out}} \leftarrow \mathbb{Z}_q^{1\times h}$, jointly with $\mathbf{U_D}$ and $\mathbf{c}^*_{\mathsf{in}}$. Appending the other challenge components cannot increase statistical distance, and adding the fixed vector $\lfloor q/2 \rfloor \mu_b$ to a uniform vector preserves uniformity. Hence, $\mathbf{c}^*_{\mathsf{out}}$ may be replaced by an independent uniform vector.

In $\mathbf{Hyb}_9$, the only challenge-ciphertext component depending on μ_b is independent uniform. All other components depend only on x^*, tg^*, the public parameters, and fresh randomness. Hence, the adversary's complete view is independent of b, so $\Pr[\mathsf{Hyb}_9(0) = 1] = \Pr[\mathsf{Hyb}_9(1) = 1]$. Combining Lemmas 11–19 by the triangle inequality yields $\mathsf{Adv}^{\mathsf{tb\text{-}abe}}_{\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\mathsf{succLWE}}_{\mathcal{B}}(\lambda) + \mathsf{negl}(\lambda)$, where the negligible term collects the losses from Lemma 5, Lemma 8, Lemma 6, the trapdoor-sampling replacements, Lemma 7, and Lemma 1. This completes the proof.

Acknowledgments. This work is supported in part by the Anusandhan National Research Foundation (ANRF), Department of Science and Technology (DST), Government of India, under File No. EEQ/2023/000164, Defence Research and Development Organisation, Government of India, under File No. SAG/4600/TCID/CARS/C-90 and the Information Security Education and Awareness (ISEA) Project Phase-III initiatives of the Ministry of Electronics and Information Technology (MeitY) under Grant No. F.No. L-14017/1/2022-HRD.

References

1. Agarwal, A., Das, S., Gilkalaye, B.P., Rindal, P., Shoup, V.: Btx: Simple and efficient batch threshold encryption. Cryptology ePrint Archive (2026)
2. Agarwal, A., Fernando, R., Pinkas, B.: Efficiently-thresholdizable batched identity based encryption, with applications. In: Kalai, Y.T., Kamara, S.F. (eds.) CRYPTO 2025, Part III. LNCS, vol. 16002, pp. 69–100. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 17–21, 2025). https://doi.org/10.1007/978-3-032-01881-6_3
3. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer Berlin Heidelberg, Germany, French Riviera (May 30 – Jun 3, 2010). https://doi.org/10.1007/978-3-642-13190-5_28
4. Applebaum, B., Nir, O., Pinkas, B.: How to recover a secret with $o(n)$ additions. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023, Part I. LNCS, vol. 14081, pp. 236–262. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 20–24, 2023). https://doi.org/10.1007/978-3-031-38557-5_8
5. Ball, M., Çakan, A., Malkin, T.: Linear threshold secret-sharing with binary reconstruction. In: Tessaro, S. (ed.) ITC 2021. LIPIcs, vol. 199, pp. 12:1–12:22. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Seattle, WA, USA (Jul 23–26, 2021). https://doi.org/10.4230/LIPIcs.ITC.2021.12

6. Boneh, D., Gentry, C., Gorbunov, S., Halevi, S., Nikolaenko, V., Segev, G., Vaikuntanathan, V., Vinayagamurthy, D.: Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 533–556. Springer Berlin Heidelberg, Germany, Copenhagen, Denmark (May 11–15, 2014). https://doi.org/10.1007/978-3-642-55220-5_30
7. Boneh, D., Laufer, E., Tas, E.N.: Batch decryption without epochs and its application to encrypted mempools. Cryptology ePrint Archive (2025)
8. Choudhuri, A.R., Garg, S., Piet, J., Policharla, G.V.: Mempool privacy via batched threshold encryption: Attacks and defenses. In: 33rd USENIX Security Symposium (USENIX Security 24). pp. 3513–3529 (2024)
9. Choudhuri, A.R., Garg, S., Policharla, G.V., Wang, M.: Practical mempool privacy via one-time setup batched threshold encryption. In: 34th USENIX Security Symposium (USENIX Security 25). pp. 3477–3495 (2025)
10. Cini, V., Lai, R.W.F., Woo, I.K.Y.: Lattice-based multi-authority/client attribute-based encryption for circuits. CiC **1**(4), 1 (2024). https://doi.org/10.62056/ahmpgy4e-
11. Das, S., Datta, P., Partap, A., Sasmal, S., Zhandry, M.: Optimal threshold traitor tracing. Cryptology ePrint Archive, Report 2025/2154 (2025), https://eprint.iacr.org/2025/2154
12. Desmedt, Y., Frankel, Y.: Threshold cryptosystems. In: Brassard, G. (ed.) CRYPTO'89. LNCS, vol. 435, pp. 307–315. Springer, New York, USA, Santa Barbara, CA, USA (Aug 20–24, 1990). https://doi.org/10.1007/0-387-34805-0_28
13. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 523–540. Springer Berlin Heidelberg, Germany, Interlaken, Switzerland (May 2–6, 2004). https://doi.org/10.1007/978-3-540-24676-3_31
14. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC. pp. 197–206. ACM Press, Victoria, BC, Canada (May 17–20, 2008). https://doi.org/10.1145/1374376.1374407
15. Gong, J., Waters, B., Wee, H., Wu, D.J.: Threshold batched identity-based encryption from pairings in the plain model. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 548–578. Springer (2026)
16. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) 45th ACM STOC. pp. 545–554. ACM Press, Palo Alto, CA, USA (Jun 1–4, 2013). https://doi.org/10.1145/2488608.2488677
17. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Juels, A., Wright, R.N., De Capitani di Vimercati, S. (eds.) ACM CCS 2006. pp. 89–98. ACM Press, Alexandria, Virginia, USA (Oct 30 – Nov 3, 2006). https://doi.org/10.1145/1180405.1180418, available as Cryptology ePrint Archive Report 2006/309
18. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing **28**(4), 1364–1396 (1999)
19. Katsumata, S., Yamada, S.: Partitioning via non-linear polynomial functions: More compact IBEs from ideal lattices and bilinear maps. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 682–712. Springer Berlin Heidelberg, Germany, Hanoi, Vietnam (Dec 4–8, 2016). https://doi.org/10.1007/978-3-662-53890-6_23
20. Pedersen, T.P.: A threshold cryptosystem without a trusted party (extended abstract) (rump session). In: Davies, D.W. (ed.) EUROCRYPT'91. LNCS, vol. 547, pp. 522–526. Springer Berlin Heidelberg, Germany, Brighton, UK (Apr 8–11, 1991). https://doi.org/10.1007/3-540-46416-6_47
21. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC. pp. 84–93. ACM Press, Baltimore, MA, USA (May 22–24, 2005). https://doi.org/10.1145/1060590.1060603

22. Sahai, A., Waters, B.R.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer Berlin Heidelberg, Germany, Aarhus, Denmark (May 22–26, 2005). https://doi.org/10.1007/11426639_27
23. Sarkar, R., Murshid, S., Mandal, M.: Batched attribute-based encryption from bilinear pairings. Cryptology ePrint Archive (2026)
24. Shamir, A.: How to share a secret. Communications of the Association for Computing Machinery **22**(11), 612–613 (Nov 1979). https://doi.org/10.1145/359168.359176
25. Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer Berlin Heidelberg, Germany, Taormina, Italy (Mar 6–9, 2011). https://doi.org/10.1007/978-3-642-19379-8_4
26. Waters, B., Wu, D.J.: Silent threshold cryptography from pairings: Expressive policies in the plain model. Cryptology ePrint Archive, Report 2025/1547 (2025), https://eprint.iacr.org/2025/1547
27. Wee, H.: Circuit ABE with $\mathsf{poly}(\mathsf{depth}, \lambda)$-sized ciphertexts and keys from lattices. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part III. LNCS, vol. 14922, pp. 178–209. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 18–22, 2024). https://doi.org/10.1007/978-3-031-68382-4_6
28. Wee, H.: Almost optimal kp and cp-abe for circuits from succinct lwe. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 34–62. Springer (2025)
29. Wee, H.: Almost optimal KP and CP-ABE for circuits from succinct LWE. In: Fehr, S., Fouque, P.A. (eds.) EUROCRYPT 2025, Part III. LNCS, vol. 15603, pp. 34–62. Springer, Cham, Switzerland, Madrid, Spain (May 4–8, 2025). https://doi.org/10.1007/978-3-031-91131-6_2
30. Xiong, S., Zhang, Y., Chen, J.: Efficient batched IBE from lattices in the standard model. IACR Communications in Cryptology **3**(2) (2026). https://doi.org/10.62056/aksude-3