

Revisiting Partitioning: Output-Adaptive ABE for Circuits from LWE

Jiaqi Cheng
UW-Madison*

Rishab Goyal
UW-Madison†

October 2, 2026

Abstract

We introduce Output-Adaptive Security, a new intermediate security notion for Attribute-Based Encryption (ABE) that bridges the gap between selective and full adaptive security. In our model, the adversary selectively commits to a policy function structure f but adaptively chooses a long target output value y during the challenge phase. We construct the first Output-Adaptive ABE scheme for general circuits from the standard Learning With Errors (LWE) assumption by revisiting classical partitioning techniques and applying them to the output space of circuit evaluation. Our result demonstrates that adaptivity for expressive policies can be achieved from standard lattices without complexity leveraging or heavy tools like witness encryption and indistinguishability obfuscation. Furthermore, we show that output-adaptivity is sufficient for many real-world applications, including biometric matching, weighted thresholds, and wildcard search, where system architecture is static while policy targets remain dynamic.

1 Introduction

Attribute-Based Encryption (ABE) [SW05, GPSW06] is a powerful concept in public-key cryptography, that goes beyond traditional one-to-one private communication to enable fine-grained access control based on user credentials. A central challenge in the study of ABE has been the quest for *adaptive security*. Unlike the weaker notion of *selective* security (which artificially forces the adversary to declare its target before seeing public parameters), adaptive security allows the adversary to corrupt keys and choose its target ciphertext dynamically, capturing the capabilities of real-world attackers.

While adaptive security is now well-understood in the realm of bilinear pairings [LOS⁺10, OT10, LW11b, LW11a, LW12, RW13, Att14, Att16, CGKW18, KW20, GLW21, RW22], due to the beautiful Dual System methodology by Waters [Wat09], the landscape for lattice-based ABE is quite underwhelming. Lattices offer the promise of post-quantum security and the ability to support general circuits (beyond the logarithmic depth limit implied by pairings) [BGG⁺14]. Yet, for over a decade, these two desirable properties: *expressiveness* (general circuits) and *full security* (adaptivity), have appeared mutually exclusive under standard lattice assumptions.

The Current Landscape. The state of lattice-based ABE can be summarized as a dilemma between security and assumptions:

*Email: jiaqicheng@cs.wisc.edu.

†Email: rishab@cs.wisc.edu.

Selective Security for Circuits: We have many ABE constructions for general circuits [GVW13, BGG⁺14, BV16, GVW15, Agr17] from standard Learning With Errors (LWE) [Reg05], but they are strictly limited to selective security.¹ More recently, Abram et al. [AMR25] showed that the adaptive-LWE assumption proposed by Quach, Wee, and Wichs [QWW18], which was introduced as a potential route toward adaptive security for BGG+-style constructions, does not hold for a broad range of parameters. This result highlights a concrete obstacle to extending the standard BGG+ methodology to adaptive settings, although it does not constitute a direct attack on the BGG+ scheme itself. Recent work of Brakerski and Medina [BM24] further investigates limitations on adaptive security for attribute-based encryption and provides additional evidence for the difficulty of achieving adaptive security within the BGG+ framework.

Adaptive Security for Weak Predicates: Partitioning-based strategies [BB04, ABB10] and lattice-based generalizations [Tsa19, KNY20, GLW21] of Gentry’s “tagging” methodology [Gen06] are the only two successful approaches to design adaptively-secure ABE from standard LWE. However, both of these only go so far to support either Identity-Based predicates or very weak predicates (e.g., $O(1)$ -CNFs, subset predicates, inner-product predicates). In particular, they fail to capture even $\mathbf{AC}^0$ circuits.

General Circuits via Heavy Machinery: If we desire adaptive security for general circuits, then the only approach is to rely on generic approaches such as Trojan methods [Wat15, ABSV15] or Dual-Mode Encryption [WW24, GY25]. However, these rely on non-falsifiable assumptions or heavy tools like Indistinguishability Obfuscation [BGI⁺01], Functional Encryption [SW05, BSW11] and Witness Encryption [GGSW13], for which we currently lack constructions from standard lattice assumptions.

In summary, this leaves a tantalizing open question:

Can we design adaptively secure ABE from standard lattices for any predicate class that cannot be expressed as an $\mathbf{AC}^0$ circuit?

1.1 Our Contribution: Output-Adaptive ABE

In this work, we provide an affirmative answer to the above question. We introduce an intermediate security notion for ABE, that we call “Output-Adaptive Security”. It serves as a stepping stone to design (fully) adaptive ABE for a non-trivial class of predicates.

To understand output-adaptive security, consider a ciphertext-policy variant of ABE. Recall in CP-ABE, each ciphertext ct encodes a predicate policy ϕ and a secret key sk encodes an attribute string x . We consider a generalization of standard ABE syntax² where each policy ϕ is associated with a function f and a long (multi-bit) value y , and we say policy ϕ satisfies attribute x if $f(x) = y$. We refer to f as the ‘compute’ function and y as the policy ‘target’. Correctness states that a user with attribute x can decrypt a ciphertext ct if $f(x) = y$.

In a fully adaptive game, an adversary chooses both the challenge function f^* and target y^* adaptively during the challenge phase. In our “output-adaptive” game:

¹While some of these ABE schemes [BV16, Agr17] have been proven to be semi-adaptively secure [CW14], we do not distinguish such ABE schemes due to known generic transformations between selective and semi-adaptive security [GKW16].

²In standard ABE syntax, the value y is typically viewed as a zero/one-bit.

- The adversary declares the function f^* at the beginning (selectively).
- However, the adversary can choose the target value y^* *adaptively* during the challenge phase, after seeing the public parameters and making polynomially many secret key queries.

We design a CP-ABE scheme for circuits (i.e., the compute function can be represented by any polynomial-sized circuit), and prove it to be output-adaptive under the hardness of standard LWE assumption. Our approach also extends to the key-policy setting, with selective commitment to the challenge input and adaptive choice of the target output.

1.2 Why Output-Adaptivity Matters

At first glance, output-adaptivity might appear to be a theoretical compromise. However, we argue that for a vast class of real-world applications, it is functionally equivalent to full adaptive security. In many systems, the “computation” f represents the system architecture (which is static), while the “target” y represents the policy decision (which is dynamic).

We highlight four concrete scenarios where output-adaptivity captures the full threat model:

Standardized Metrics and Biometrics: In many deployments, the logic used to process attributes is standardized or hardcoded into firmware. For example, in a biometric access system, the feature extraction algorithm f (e.g., an iris-scanning logic) is fixed at deployment. Similarly, in financial systems, the credit-scoring algorithm f (e.g., FICO) is a public industry standard. While f is known selectively, the specific target y , the iris hash or the credit threshold, is chosen adaptively based on the specific user or market conditions being targeted. Output-adaptivity perfectly models this separation between static extraction and dynamic targeting.

IoT and Sensor Networks: In Industrial IoT (IIoT) settings, edge devices collect raw data x . The aggregation logic f , such as “compute the vibration variance over n samples”, is often defined by the fixed network topology or hardware constraints. However, the critical trigger value y (e.g., the specific threshold indicating a machine failure) is determined adaptively by the control center in real-time. Here, the selective constraint on f is harmless, as the network’s processing capabilities are fixed, while the adaptive choice of y provides the necessary security for dynamic emergency commands.

Privacy-Preserving Classification:

Consider a cloud-hosted medical database where patient records x are encrypted. A researcher may broadcast a query to decrypt only records classified as a specific risk level. The classification model f (e.g., a pre-trained Neural Network) is a public, high-complexity circuit deployed system-wide. While the model architecture is fixed, the researcher (or an adversary) might adaptively choose which specific diagnosis y to target. Since our scheme supports general circuits, it can secure these sophisticated ML-based policies where f is a deep circuit but the classification target y is dynamic.

Architectural Stability (Polynomial Class Equivalence): From a theoretical perspective, if the set of admissible functions $\mathcal{F}$ is of polynomial size, output-adaptivity implies *full adaptive security* via a standard complexity-leveraging argument. Essentially, the reduction algorithm can guess the challenge function, and only incur a polynomial security loss in the reduction

as long as the set of admissible functions $\mathcal{F}$ is polynomial-sized. This makes our notion a generalization of several primitives. For instance, Identity-Based Encryption (IBE) corresponds to the case where f is the identity function ($|\mathcal{F}| = 1$). Similarly, adaptively secure ABE for the class of (weighted) threshold predicates and wildcard predicates can also be instantiated using our output-adaptive ABE approach. Our work thus provides a unified framework that scales from simple adaptively-secure IBE to complex, circuit-based ABE.

1.3 Technical Summary

BGG+ baseline and its algebraic strength. We begin from the standard [BGG⁺14] template for lattice ABE over circuits. **Setup** samples a public base matrix $\mathbf{A}_0 \in \mathbb{Z}_q^{n \times m}$ together with a private trapdoor $\mathbf{T}_{\mathbf{A}_0}$ (kept in master secret key `msk`), and public matrices $\{\mathbf{A}_i\}_{i \in [\ell]}$. For an attribute x , **KeyGen** hardwires x into a universal circuit $\mathcal{U}_x$ and computes

$$\mathbf{H}_{\mathcal{U}_x} := \text{EvalF}(\mathcal{U}_x, \{\mathbf{A}_i\}_{i \in [\ell]}), \quad \mathbf{A}_{\mathcal{U}_x} := [\mathbf{A}_1 | \cdots | \mathbf{A}_\ell] \cdot \mathbf{H}_{\mathcal{U}_x}.$$

It then samples a short key $\mathbf{e}_x$ such that $[\mathbf{A}_0 | \mathbf{A}_{\mathcal{U}_x} - \mathbf{G}] \cdot \mathbf{e}_x = \mathbf{u}$. To encrypt message μ under a boolean policy ϕ , the ciphertext consists of

$$\left(\mathbf{s}^T [\mathbf{A}_0 | \mathbf{A}_1 - \phi[1]\mathbf{G} | \cdots | \mathbf{A}_L - \phi[L]\mathbf{G}] + \text{noise}, \mathbf{s}^T \mathbf{u} + \mu \lfloor q/2 \rfloor + \text{noise} \right).$$

Its correctness is driven by key-homomorphic evaluation:

$$\begin{aligned} [\mathbf{A}_1 - \phi[1]\mathbf{G} | \cdots | \mathbf{A}_L - \phi[L]\mathbf{G}] \cdot \mathbf{H}_{\mathcal{U}_x, \phi} &= [\mathbf{A}_1 | \cdots | \mathbf{A}_L] \cdot \mathbf{H}_{\mathcal{U}_x} - \phi(x)\mathbf{G} \\ &= \mathbf{A}_{\mathcal{U}_x} - \phi(x)\mathbf{G}, \end{aligned}$$

where $\mathbf{H}_{\mathcal{U}_x, \phi} := \text{EvalFX}(\mathcal{U}_x, \phi, \{\mathbf{A}_i - \phi[i]\mathbf{G}\})$ has low norm. Here, $\mathcal{U}_x$ is the universal circuit which takes as input the policy ϕ and outputs 1 if and only if $\phi(x) = 1$. This identity is precisely what gives BGG+ circuit expressiveness in the lattice setting. It guarantees that the secret key generated for attribute x using the trapdoor for $\mathbf{A}_0$ can be correctly combined with a ciphertext `ct` encrypted under policy ϕ as long as $\phi(x) = 1$.³

In the pure BGG+ style lattice ABE proofs for general circuits, the adversary commits to ϕ^* at the start. The reduction then embeds ϕ^* into public matrices by setting

$$\mathbf{A}_i = \mathbf{A}_0 \mathbf{R}_i + \phi^*[i]\mathbf{G} \quad (i \in [L]).$$

For a key query x , the queried decryption matrix is $[\mathbf{A}_0 | \mathbf{A}_{\mathcal{U}_x} - \mathbf{G}]$. If $\phi^*(x) = 0$, it can rewrite the second block as $\mathbf{A}_0 \mathbf{R}' + \mathbf{G}$ (for suitable $\mathbf{R}'$ derived from $\mathbf{H}_{\mathcal{U}_x, f^*}$), and answer via standard lattice trapdoor sampling. The simulator's trapdoor is effectively disabled only for the attributes x such that $\phi^*(x) = 1$ (this is exactly the challenge region). Hence, simulability is pre-partitioned by the early commitment to ϕ^* .

Challenges in extending the pure BGG+ framework to output adaptivity. Now we move to output-adaptiveness for policies $\phi = (f, y)$. A natural attempt is to keep the BGG+ design unchanged and jointly encode (f, y) , allowing adaptive choice of y^* at challenge time. This places the adaptive target in the same linear gadget channel as in the adaptive encodings considered by

³Here $\phi(x) = 1$ indicates successful decryption, whereas in [BGG⁺14] $\phi(x) = 0$ indicates successful decryption.

Quach, Wee, and Wicks [QWW18]. Abram et al. [AMR25] show that, when the adaptively chosen target is sufficiently long, such encodings can be vulnerable. At a high level, if the target-encoding term has the form $\mathbf{s}^T(\mathbf{A} - y \otimes \mathbf{G}) + \text{noise}$, then by choosing a crafted target y_i , given by the bit decomposition of $\mathbf{Y} = \mathbf{A}\mathbf{H} - 2^i\mathbf{I}$, one obtains

$$(\mathbf{s}^T(\mathbf{A} - y_i \otimes \mathbf{G}) + \text{noise})\mathbf{H} = 2^i\mathbf{s}^T + \text{noise}.$$

Taking the most significant bit reveals the i -th high bit of $\mathbf{s}$, and repeating this over $i \in [\log q]$ recovers $\mathbf{s}$. This suggests that directly encoding the adaptive target through the BGG+ linear channel is problematic. Our construction therefore takes a different route and encodes the computation and target separately.

Partitioning the target space with admissible hash. To achieve target-adaptive security, our construction extends the BGG+ encoding framework. Instead of encoding the entire predicate $\phi = (f, y)$ together, we split it into two coupled layers in our design:

- (i) a BGG+ style *computation layer* for evaluating f , and
- (ii) a *target-binding layer* for handling y through a separate matrix family to support target space partitioning.

At a high level, our partitioning approach relies on admissible hash functions, which were introduced in pairing-based identity-based encryption (IBE) [BB04, Wat05] and later adapted to lattices [ABB10, CHKP12].

Concretely, in addition to $\mathbf{A}_0, \mathbf{T}_{\mathbf{A}_0}$, and $\{\mathbf{A}_i\}_{i \in [L]}$, **Setup** samples and publishes $\mathbf{B}_0, \{\mathbf{B}_i\}_{i \in [\ell]}$, where ℓ is the target length. The ciphertext keeps the BGG+ structure for the function description f and adds a separate linear term for the target y . The separate linear term mirrors Agrawal-Boneh-Boyen (ABB) encodings developed originally for IBE. Specifically, for policy (f, y) and message μ , **Enc** outputs $(\mathbf{c}_0, \mathbf{c}_1, c_2)$ where

$$\begin{aligned} \mathbf{c}_0^T &= \mathbf{s}^T \cdot \underbrace{[\mathbf{A}_0 \mid \mathbf{A}_1 - f[1] \cdot \mathbf{G} \mid \cdots \mid \mathbf{A}_L - f[L] \cdot \mathbf{G}]}_{\text{BGG+}(f)} + \text{noise}, \\ \mathbf{c}_1^T &= \mathbf{s}^T \cdot \underbrace{\left(\sum_{i=1}^{\ell} y[i] \cdot \mathbf{B}_i + \mathbf{B}_0 \right)}_{\text{ABB}(y)} + \text{noise}. \end{aligned}$$

So $\mathbf{c}_0$ carries the function information, and $\mathbf{c}_1$ encodes the target. This separation is exactly what allows adaptive handling of y through partitioning. The last component c_2 is set such that $c_2 = \mathbf{s}^T \cdot \mathbf{u} + \mu \cdot \lfloor q/2 \rfloor + \text{noise}$.

Our intuition is that, in the security reduction, we can program the output matrices as $\mathbf{B}_i = \mathbf{A}_0\mathbf{R}'_i + h_i\mathbf{B}_0$ for each $i \in [\ell]$, where $\mathbf{R}'_i$ are random small matrices and $h_i \in \mathbb{Z}_q$ are hidden scalars. Hence, for every $y \in \{0, 1\}^\ell$,

$$\sum_{i=1}^{\ell} y[i]\mathbf{B}_i + \mathbf{B}_0 = \mathbf{A}_0 \left(\sum_{i=1}^{\ell} y[i]\mathbf{R}'_i \right) + \left(1 + \sum_{i=1}^{\ell} h_i y[i] \right) \mathbf{B}_0.$$

Let $H(y) := 1 + \sum_{i=1}^{\ell} h_i y[i]$, and (f^*, y^*) be the attacker's challenge where y^* is adaptively selected. Our idea is to design a new key generation process such that secret keys can be simulated for any

attribute x such that $H(f^*(x)) \neq 0$. In order to dive into our key generation process, let us first explain how we expect a decryptor to homomorphically operate on our new ciphertexts.

Melding BGG+ and ABB using matrix evaluation. Essentially, our idea is to perform the BGG+ homomorphism on the BGG+ portion of the ciphertext and algebraically combine it with the ABB portion of the ciphertext. In more detail, for attribute x , define $\mathbf{Y}_{f(x)} := \sum_{i=1}^{\ell} f(x)[i] \cdot \mathbf{B}_i$. Suppose the decryptor can compute a low-norm matrix $\mathbf{H}_{\mathcal{U}_x, f}$ such that

$$\begin{aligned} [\mathbf{A}_1 - f[1]\mathbf{G} \mid \cdots \mid \mathbf{A}_L - f[L]\mathbf{G}] \mathbf{H}_{\mathcal{U}_x, f} &= [\mathbf{A}_1 \mid \cdots \mid \mathbf{A}_L] \mathbf{H}_{\mathcal{U}_x} - \mathbf{Y}_{f(x)} \\ &= \mathbf{A}_{\mathcal{U}_x} - \mathbf{Y}_{f(x)}, \end{aligned}$$

where $\mathbf{H}_{\mathcal{U}_x}$ is also a short matrix that depends only on x . It turns out that such $\mathbf{H}_{\mathcal{U}_x, f}$ (and $\mathbf{H}_{\mathcal{U}_x}$) can be computed by relying on matrix-value homomorphic evaluation algorithms proposed by Brakerski et al. [BTVW17] in the context of constrained pseudo-random functions. Using the above, the decryptor can compute

$$\begin{aligned} \mathbf{c}^T &= \mathbf{c}_0^T \cdot \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{\mathcal{U}_x, f} \end{bmatrix} + [\mathbf{0} \mid \mathbf{c}_1^T] \\ &= \mathbf{s}^T \cdot \left[\mathbf{A}_0 \mid \left(\underbrace{\mathbf{A}_{\mathcal{U}_x} - \sum y[i] \cdot \mathbf{B}_i}_{\text{Matrix Evaluation}} + \underbrace{\sum y[i] \cdot \mathbf{B}_i + \mathbf{B}_0}_{\text{ABB}(y)} \right) \right] + \text{noise}. \end{aligned}$$

Therefore, when $f(x) = y$, the component which depends on y *vanishes*. This “vanishing property” is extremely crucial for our key generation process as the key generator only gets x and does not have any information about f or y (since that is encoded in the ciphertext).

Basically, our plan is to sample a secret key as a short vector $\mathbf{sk}_x$ such that $c_2 - \mathbf{c}'^T \mathbf{sk}_x$ gives us the plaintext whenever x satisfies the policy $\phi = (f, y)$. Now this can be guaranteed if $\mathbf{sk}_x$ is sampled such that $[\mathbf{A}_0 \mid \mathbf{A}_{\mathcal{U}_x} + \mathbf{B}_0] \mathbf{sk}_x = \mathbf{u}$. Clearly, this can be done given the trapdoor of $\mathbf{A}_0$ since $\mathbf{A}_{\mathcal{U}_x}$ only depends on x and can be computed during key generation (i.e., without knowledge of f or y). This completes the overview of our design and we describe the high level proof strategy next.

Proving output-adaptivity. Our security analysis relies on the trapdoor switching technique to resolve the basic simulation problem: The simulator must generate valid secret keys for arbitrary queries. At the same time, it must remain independent of the challenge ciphertext in order to embed the LWE challenge. We address this issue by moving the security dependence from the master input trapdoor $\mathbf{T}_{\mathbf{A}_0}$ to an auxiliary output trapdoor $\mathbf{T}_{\mathbf{B}_0}$ through a sequence of hybrid games. To make this transition indistinguishable, we modify the public parameters. We embed an admissible hash function and mask it with high-entropy randomness. Specifically, we will replace the uniform $\mathbf{A}_i$ and $\mathbf{B}_i$ with matrices of the form $\mathbf{A}_i = \mathbf{A}_0 \cdot \mathbf{R}_i + f[i]\mathbf{G}$ and $\mathbf{B}_i = \mathbf{A}_0 \cdot \mathbf{R}'_i + h_i \cdot \mathbf{B}_0$.

During the matrix-valued homomorphic evaluation, the simulator computes a linear combination of the public matrices according to the input x . Because the evaluation is linear, all the randomness and hash coefficients embedded in the public matrices are aggregated into the resulting matrix.

Therefore,

$$\begin{aligned} & [\mathbf{A}_0 \mid \mathbf{A}_1 - f[1] \cdot \mathbf{G} \mid \dots \mid \mathbf{A}_L - f[L] \cdot \mathbf{G}] \cdot \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{\mathcal{U}_x} \end{bmatrix} \\ &= [\mathbf{A}_0 \mid \mathbf{A}_0 \cdot \mathbf{R}_{\text{total}} + H(y) \cdot \mathbf{B}_0]. \end{aligned}$$

This algebraic form is crucial for our security proof. Let (f^*, y^*) be the challenge policy. Now by the partitioning property of the admissible hash, we have the guarantee that $H(f^*(x)) \neq 0$ for each queried attribute x , while $H(y^*) = 0$. Thus, for any secret-key query on attribute x , since the hash value $H(f^*(x))$ is non-zero, then the simulator can then use the auxiliary trapdoor $\mathbf{T}_{\mathbf{B}_0}$ together with the `SampleRight` algorithm to generate a valid secret key. While during the challenge phase, whenever $H(y^*) = 0$, the entire ciphertext can be simulated given $\mathbf{s}^T \cdot \mathbf{A}_0 + \text{noise}$. Thus, by LWE, we can prove output-adaptivity of our construction.

Extending to key-policy ABE. The same approach extends to key-policy ABE, where secret keys are associated with functions f and ciphertexts with pairs (x, y) , with decryption enabled when $f(x) = y$. Here, output adaptivity allows the adversary to choose y^* adaptively while committing to x^* before setup.

1.4 Technical Highlights and Related Work

Our construction departs from the “tagging” paradigm common in recent lattice literature for attaining adaptive security [Tsa19,KNYY20,GLW21]. Instead, we revisit the classical *partitioning techniques* introduced by Boneh and Boyen [BB04]. While partitioning has largely been viewed as a technique limited to Identity-Based Encryption, we demonstrate that it is surprisingly potent in the lattice setting when applied to the output space of a circuit.

Our core intuition is to combine the circuit-homomorphic evaluation properties of [BGG⁺14] with an admissibility-based partitioning of the output space. And, by partitioning the space of circuit output rather than the input wires of the circuit, we construct a scheme for general circuits that is selectively secure in the function f^* but adaptively secure in target value y^* . In particular, we embed a partitioning function into the evaluation of the ciphertext. Specifically, we generate public parameters such that for the declared function f^* , the output space is partitioned into “simulatable” and “challenge” regions. The challenge ciphertext is programmed to evaluate to a target y^* that falls into the “challenge” partition, allowing us to embed the LWE challenge. Crucially, because the partitioning occurs on the *output* of the circuit rather than the input wires, we avoid the exponential blowup typically associated with partitioning complex predicates. Our partitioning technique applies to the output space in a bit-wise manner (similar to [ABB10,Wat05]), allowing our scheme to support arbitrary polynomial-length outputs. This stands in contrast to prior works that required complexity leveraging to handle large output spaces.

Related Work. The construction of ABE schemes from lattice assumptions has been a central focus in post-quantum cryptography. The seminal work of Gentry, Peikert, and Vaikuntanathan [GPV08] provided the trapdoor mechanisms necessary for lattice-based IBE. This was followed by a line of adaptively-secure IBE constructions [ABB10,CHKP12] using partitioning and puncturing techniques. Moving beyond identity-based policies, Gorbunov et al. [GVW13] and Boneh et al. [BGG⁺14] provided the first construction for circuit-based ABE. The [BGG⁺14] construction, in particular, leveraged the key-homomorphic properties of LWE to support policies

represented by polynomial-depth circuits. However, these constructions and their subsequent optimizations [GVW15, BV16, Agr17] are inherently limited to selective security. This limitation stems from the need to embed the challenge attribute into the public matrices during the setup or key generation phases in a static manner.

To achieve adaptive security in the lattice setting, recent works have focused on “tagging” techniques, often generalizing the approach of Gentry [Gen06]. Tsabary [Tsa19] extended this paradigm to design adaptive ABE for $O(1)$ -CNF formulas. This was later refined by Goyal et al. [GLW21] and Katsumata et al. [KNYY20] to subset and inner-product predicates. However, a fundamental barrier remains: these “tagging” techniques cannot support complex predicates and suffer from exponential parameter growth when applied to deeper circuits. Consequently, there are no known lattice-based constructions achieving adaptive security for general circuits ($P/poly$) or even NC^1 without resorting to complexity leveraging or non-standard assumptions.

In the setting of bilinear pairings, adaptive security for expressive policies is largely solved via the Dual System Encryption framework introduced by Waters [Wat09]. This methodology uses semi-functional keys and ciphertexts to transition the security game through a sequence of hybrids. This has enabled adaptively secure ABE for boolean formulas [LOS⁺10, AC17], regular languages [Wat12, Att16], and eventually deterministic finite automata [Att14]. Translating the Dual System framework to lattices has proven notoriously difficult due to the lack of efficient subspace-hiding mechanisms compatible with the noise growth in lattice circuit evaluation.

Our definition of “Output-Adaptive Security” (where the function f is selective, but the target y is adaptive) bears a semantic resemblance to “Compute-and-Compare” (CC) obfuscation [GKW17, WZ17], but there does not appear to any meaningful overlap in the underlying techniques or research goals. In a very recent work, Cheng and Goyal [CG26] also studied such compute-and-compare predicate classes in the context of constrained pseudorandom functions (CPRFs). While there are some technical similarities in the underlying algebraic designs, the scope and goals of the two works are entirely different. [CG26] designed the first collusion-resistant CPRF for a non-trivial class of constraints from standard lattice assumptions; while in this work, our focus is to achieve adaptive security in the context of ABE. Recent concurrent works of Goyal and Yadugiri [GY26b, GY26a] construct adaptively secure flexible and identity-based broadcast encryption, as well as bounded-user distributed broadcast encryption, using equivocal matrix-commitment techniques. These results achieve adaptive security for subset policies from non-standard falsifiable lattice assumptions.

2 Preliminaries

2.1 Lattice Preliminaries

Notation. Let PPT denote probabilistic polynomial-time. We will use lowercase bold letters for vectors (e.g. $\mathbf{v}$), uppercase bold letters for matrices (e.g. $\mathbf{A}$). For any integer $q \geq 2$, we let $\mathbb{Z}_q$ denote the ring of integers modulo q . We represent $\mathbb{Z}_q$ as integers in the range $(-q/2, q/2]$. For a vector $\mathbf{v}$, we let $\|\mathbf{v}\|$ denote its ℓ_2 norm and $\|\mathbf{v}\|_\infty$ denote its infinity norm. Similarly, for matrices $\|\cdot\|$ and $\|\cdot\|_\infty$ denote their ℓ_2 and infinity norms (respectively).

For two distributions X, Y , over a finite domain Ω , the statistical distance between X and Y is defined as $SD(X, Y) \stackrel{\text{def}}{=} \frac{1}{2} \sum_{\omega \in \Omega} |X(\omega) - Y(\omega)|$. A family of distributions $\mathcal{D}_1 = \{\mathcal{D}_1(\lambda)\}_\lambda$ and $\mathcal{D}_2 = \{\mathcal{D}_2(\lambda)\}_\lambda$, parameterized by security parameter λ , are said to be statistically indistinguishable,

represented by $\mathcal{D}_1 \approx_s \mathcal{D}_2$, if there exists a negligible function $\text{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$, $\text{SD}(\mathcal{D}_1(\lambda), \mathcal{D}_2(\lambda)) \leq \text{negl}(\lambda)$. For a family of distributions $\mathcal{D} = \{\mathcal{D}(\lambda)\}_\lambda$ over the integers, and integers bounds $B = \{B(\lambda)\}_\lambda$, we say that $\mathcal{D}$ is B -bounded if $\Pr[|x| \leq B(\lambda) : x \leftarrow \mathcal{D}(\lambda)] = 1$.

Learning with Errors For our constructions, we rely on the learning with errors assumption.

Assumption 2.1 (Learning with Errors). Let n, m, q be positive integers and let χ be a noise distribution over $\mathbb{Z}_q$. The Learning with Errors assumption $\text{LWE}_{n,m,q,\chi}$ states that the following two distributions are computationally indistinguishable:

$$\begin{aligned} & \{(\mathbf{A}, \mathbf{s}^T \mathbf{A} + \mathbf{e}^T) : \mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{s} \leftarrow \mathbb{Z}_q^n, \mathbf{e} \leftarrow \chi^m\} \\ & \approx_c \{(\mathbf{A}, \mathbf{u}^T) : \mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{u} \leftarrow \mathbb{Z}_q^m\}. \end{aligned}$$

Leftover Hash Lemma. The min-entropy of a random variable X is defined as $\mathbf{H}_\infty(X) \stackrel{\text{def}}{=} -\log_2(\max_x \Pr[X = x])$. Let $\text{SD}(X, Y)$ denote the statistical distance between two random variables X and Y . Below we state the Leftover Hash Lemma (LHL) from [HILL99, DRS04, DORS08, ABB10].

Theorem 2.2. Let $\mathcal{H} = \{h : X \rightarrow Y\}_{h \in \mathcal{H}}$ be a universal hash family, then for any random variable W taking values in X , the following holds

$$\text{SD}((h, h(W)), (h, U_Y)) \leq \frac{1}{2} \sqrt{2^{-\mathbf{H}_\infty(W)} \cdot |Y|}.$$

We will consider the following corollary, which follows from the Leftover Hash Lemma.

Corollary 2.3. Let n, m, q be positive integers such that q is an odd prime.

1. $\{(\mathbf{B}, \mathbf{B}\mathbf{R}, \mathbf{e}^T \mathbf{R})\} \approx_s \{(\mathbf{B}, \mathbf{A}, \mathbf{e}^T \mathbf{R})\}$ where $m = \Omega(n \log q)$, $\mathbf{B} \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{R} \leftarrow \{-1, 1\}^{m \times \ell}$ (or $\mathbf{R} \leftarrow \{0, 1\}^{m \times \ell}$), $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times \ell}$, $\ell = \text{poly}(n)$ and $\mathbf{e}^T \in \mathbb{Z}_q^m$ is any fixed vector.
2. $\{(\mathbf{B}, \mathbf{B}\mathbf{R})\} \approx_s \{(\mathbf{B}, \mathbf{A})\}$ where $m = \Omega(n \log q)$, $\mathbf{B} \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{R} \leftarrow \mathcal{D}_{\mathbb{Z}, \sigma}^{m \times \ell}$, $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times \ell}$, $\ell = \text{poly}(n)$ and $\sigma = \omega(\sqrt{\log m})$.

Truncated Discrete Gaussians. The truncated discrete Gaussian distribution over $\mathbb{Z}^m$ with parameter σ , denoted by $\tilde{\mathcal{D}}_{\mathbb{Z}^m, \sigma}$, is same as the discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z}^m, \sigma}$ except it outputs 0 whenever the ℓ_∞ norm exceeds $\sqrt{m} \cdot \sigma$. Note that, by definition, $\tilde{\mathcal{D}}_{\mathbb{Z}^m, \sigma}$ is $\sqrt{m} \cdot \sigma$ -bounded. Also, by the above lemma we get that $\tilde{\mathcal{D}}_{\mathbb{Z}^m, \sigma} \approx_s \mathcal{D}_{\mathbb{Z}^m, \sigma}$.

Lattice Trapdoors and Sampling Let $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ be the fixed *gadget matrix* as defined in [MP12].

Definition 2.4 (Trapdoor Lattice Sampler [Ajt99, GPV08, MP12]). A trapdoor lattice sampler consists of the following PPT algorithms:

- $\text{TrapGen}(1^n, 1^m, q) \rightarrow (\mathbf{A}, \mathbf{T}_\mathbf{A})$: The lattice generation algorithm is a randomized algorithm that takes as input the matrix dimensions n, m , modulus q , and outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with a trapdoor $\mathbf{T}_\mathbf{A}$.

- **SamplePre**($\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{u}, \sigma$) $\rightarrow \mathbf{s}$: Given matrix $\mathbf{A}$, trapdoor $\mathbf{T}_\mathbf{A}$, vector $\mathbf{u} \in \mathbb{Z}_q^n$, and Gaussian parameter σ , it outputs $\mathbf{s} \in \mathbb{Z}_q^m$.

These algorithms satisfy the following properties:

1. **Correctness**: For all $\mathbf{u}, \sigma$, if $(\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \text{TrapGen}(1^n, 1^m, q)$ and $\mathbf{s} \leftarrow \text{SamplePre}(\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{u}, \sigma)$, then $\mathbf{A}\mathbf{s} = \mathbf{u}$ and $\|\mathbf{s}\|_\infty \leq \sqrt{m} \cdot \sigma$.
2. **Well Distributedness of Matrix**: The following distributions are statistically indistinguishable:

$$\{\mathbf{A} : (\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \text{TrapGen}(1^n, 1^m, q)\} \approx_s \{\mathbf{A} : \mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}\}.$$

3. **Well Distributedness of Preimage**: For all $(\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \text{TrapGen}(1^n, 1^m, q)$, if $\sigma = \omega(\sqrt{n \cdot \log q \cdot \log m})$, then the following distributions are statistically indistinguishable:

$$\{\mathbf{s} : \mathbf{u} \leftarrow \mathbb{Z}_q^n, \mathbf{s} \leftarrow \text{SamplePre}(\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{u}, \sigma)\} \approx_s \mathcal{D}_{\mathbb{Z}^m, \sigma}.$$

Definition 2.5 (Sampling on Augmented Matrices [ABB10, MP12]). As an extension to Theorem 2.4, the following algorithms are used for sampling preimages on augmented matrices:

- **SampleRight**($\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{B}, \mathbf{u}, \sigma$) $\rightarrow \mathbf{s}$: Given matrices $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_q^{n \times m}$, a trapdoor $\mathbf{T}_\mathbf{A}$ for $\mathbf{A}$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and parameter σ , it outputs a vector $\mathbf{s} \in \mathbb{Z}_q^{2m}$.
- **SampleLeft**($\mathbf{A}, \mathbf{B}, \mathbf{T}_\mathbf{B}, \mathbf{R}, y, \mathbf{u}, \sigma$) $\rightarrow \mathbf{s}$: Given matrix $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_q^{n \times m}$, the trapdoor $\mathbf{T}_\mathbf{B}$ of $\mathbf{B}$, a small-norm matrix $\mathbf{R} \in \mathbb{Z}^{m \times m}$, a scalar $y \neq 0 \in \mathbb{Z}_q$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and parameter σ , it outputs a vector $\mathbf{s} \in \mathbb{Z}_q^{2m}$.

These algorithms satisfy the following properties:

1. **Correctness**: For all $\mathbf{u}, \sigma$, if $(\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \text{TrapGen}(1^n, 1^m, q)$, if $\mathbf{s} \leftarrow \text{SampleRight}(\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{B}, \mathbf{u}, \sigma)$ and $\|\mathbf{T}_\mathbf{A}\| \cdot \omega(m\sqrt{\log m}) \leq \sigma \leq q$, then $[\mathbf{A} \mid \mathbf{B}] \cdot \mathbf{s} = \mathbf{u}$ and $\|\mathbf{s}\|_\infty \leq \sqrt{2m} \cdot \sigma$. Similarly, if $(\mathbf{B}, \mathbf{T}_\mathbf{B}) \leftarrow \text{TrapGen}(1^n, 1^m, q)$, $\mathbf{s} \leftarrow \text{SampleLeft}(\mathbf{A}, \mathbf{B}, \mathbf{T}_\mathbf{B}, \mathbf{R}, y, \mathbf{u}, \sigma)$, then $[\mathbf{A} \mid \mathbf{A}\mathbf{R} + y\mathbf{B}] \cdot \mathbf{s} = \mathbf{u}$ and $\|\mathbf{s}\|_\infty \leq \sqrt{2m} \cdot \sigma$.
2. **Indistinguishability of Preimage**: For any correctly generated matrices $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_q^{n \times m}$ with trapdoors $\mathbf{T}_\mathbf{A}, \mathbf{T}_\mathbf{B}$, and any target vector $\mathbf{u} \in \mathbb{Z}_q^n$, the distributions of the outputs from **SampleRight** and **SampleLeft** are statistically indistinguishable from each other:

$$\begin{aligned} & \{\mathbf{s} : \mathbf{s} \leftarrow \text{SampleRight}(\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{B}, \mathbf{u}, \sigma)\} \\ & \approx_s \{\mathbf{s} : \mathbf{s} \leftarrow \text{SampleLeft}(\mathbf{A}, \mathbf{B}, \mathbf{T}_\mathbf{B}, \mathbf{R}, y, \mathbf{u}, \sigma)\}. \end{aligned}$$

Lattice Evaluation of Matrix Functions We start by introducing the lattice evaluation that was developed by the FHE and ABE schemes [GSW13, BGG⁺14].

Theorem 2.6. Given n, q, ℓ and $m = \Omega(n \log q)$, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times \ell m}$, and a binary function $f : \{0, 1\}^\ell \rightarrow \{0, 1\}$ of depth d , there exists efficient deterministic algorithms **EvalF** and **EvalFX** such that for all $x \in \{0, 1\}^\ell$, $\mathbf{H}_f = \text{EvalF}(f, \mathbf{A})$, and $\mathbf{H}_{f,x} = \text{EvalFX}(f, x, \mathbf{A})$, it satisfies that

$$(\mathbf{A} - x \otimes \mathbf{G}) \cdot \mathbf{H}_{f,x} = \mathbf{A}_f - f(x) \otimes \mathbf{G} \quad (1)$$

where $\mathbf{A}_f = \mathbf{A} \cdot \mathbf{H}_f$, $\mathbf{H}_f \in \mathbb{Z}_q^{\ell m \times m}$, $\mathbf{A}_f \in \mathbb{Z}_q^{n \times m}$, $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ is the gadget matrix, and $\|\mathbf{H}_f\|_\infty, \|\mathbf{H}_{f,x}\|_\infty \leq m^{O(d)}$.

Extending to the evaluation of matrix functions In [BTVW17], Brakerski et al. considers the following variant of the above lattice evaluation: Given n, m, q , then for $i \in [n], j \in [m]$ and $t = (i, j)$, we define matrix $\mathbf{E}_{t,\ell} \in \mathbb{Z}_q^{n \times m}$ as the matrix where all the entries of $\mathbf{E}_{t,\ell}$ are zeros except for the entry (i, j) , and such entry has value $2^{\ell-1}$. Then, given matrix $\mathbf{X}$ such that $x_{t,\ell}$ is the ℓ -th bit of entry t of $\mathbf{X}$, we have:

$$\mathbf{X} = \sum_{t,\ell} x_{t,\ell} \cdot \mathbf{E}_{t,\ell}.$$

Next, given a function $f : \{0, 1\}^\ell \rightarrow \mathbb{Z}_q^{n \times m}$ and $x \in \{0, 1\}^\ell$, let $\mathbf{X}_f = f(x)$, and let function $f_{t,\ell} : x \rightarrow \{0, 1\}$ be the binary function that outputs the ℓ -th bit of entry t of $\mathbf{X}_f$. Then for $\mathbf{A} \in \mathbb{Z}_q^{\ell m}$, let $\mathbf{H}_{f,t,\ell} = \text{EvalF}(f_{t,\ell}, \mathbf{A})$ and let $\mathbf{H}_{f,t,\ell,x} = \text{EvalFX}(f_{t,\ell}, x, \mathbf{A})$. Let

$$\mathbf{H}_f = \sum_{t,\ell} \mathbf{H}_{f,t,\ell} \cdot \mathbf{G}^{-1}(\mathbf{E}_{t,\ell}), \mathbf{H}_{f,x} = \sum_{t,\ell} \mathbf{H}_{f,t,\ell,x} \cdot \mathbf{G}^{-1}(\mathbf{E}_{t,\ell}).$$

Then, one could extend Eq. (1) as the following:

$$(\mathbf{A} - x \otimes \mathbf{G}) \cdot \mathbf{H}_{f,x} = \sum_{t,\ell} (\mathbf{A} - x \otimes \mathbf{G}) \cdot \mathbf{H}_{f,t,\ell,x} \cdot \mathbf{G}^{-1}(\mathbf{E}_{t,\ell}) = \sum_{t,\ell} (\mathbf{A} \cdot \mathbf{H}_{f,t,\ell} - f_{t,\ell}(x) \cdot \mathbf{G}) \cdot \mathbf{G}^{-1}(\mathbf{E}_{t,\ell}) = \mathbf{A} \cdot \mathbf{H}_f - \mathbf{X}_f.$$

Assuming that function f is of depth d , then by Theorem 2.6, $\|\mathbf{H}_{f,t,\ell}\|_\infty, \|\mathbf{H}_{f,t,\ell,x}\|_\infty \leq m^{O(d)}$. Thus, it also holds that $\|\mathbf{H}_f\|_\infty, \|\mathbf{H}_{f,x}\|_\infty \leq m^{O(d)}$. Formally, we conclude the above using the following theorem:

Theorem 2.7 ([BTVW17], Section 4.1). Given n, q, ℓ and $m = \Omega(n \log q)$, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times \ell m}$, and a function $f : \{0, 1\}^\ell \rightarrow \mathbb{Z}_q^{n \times m}$ of depth d , there exists efficient deterministic algorithms EvalF and EvalFX such that for all $x \in \{0, 1\}^\ell$, letting $\mathbf{X}_f = f(x)$, $\mathbf{H}_f = \text{EvalF}(f, \mathbf{A})$, and $\mathbf{H}_{f,x} = \text{EvalFX}(f, x, \mathbf{A})$, it satisfies that

$$(\mathbf{A} - x \otimes \mathbf{G}) \cdot \mathbf{H}_{f,x} = \mathbf{A}_f - \mathbf{X}_f,$$

where $\mathbf{A}_f = \mathbf{A} \cdot \mathbf{H}_f$, $\mathbf{H}_f \in \mathbb{Z}_q^{\ell m \times m}$, $\mathbf{A}_f \in \mathbb{Z}_q^{n \times m}$, $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ is the gadget matrix, and $\|\mathbf{H}_f\|_\infty, \|\mathbf{H}_{f,x}\|_\infty \leq m^{O(d)}$.

3 Ciphertext Policy ABE

We provide the formal syntax and the output-adaptive security definitions for the ciphertext policy ABE. In these schemes, decryption succeeds if and only if the evaluated result of a function matches a specific target value.

3.1 Syntax and Security Definition

Syntax. A CP-ABE scheme for the compute and compare functionality consists of the following four polynomial-time algorithms. For simplicity, mpk is an implicit input to key generation.

Setup($1^\lambda, 1^L, 1^\ell$) $\rightarrow$ (mpk, msk): The setup algorithm takes as input the security parameter λ , the maximum function description length L , and the output dimension ℓ . It outputs the master public key mpk and the master secret key msk .

$\text{KeyGen}(\text{msk}, x) \rightarrow \text{sk}_x$: The extraction algorithm takes as input the master secret key msk and an attribute x , and outputs a secret key sk_x .

$\text{Enc}(\text{mpk}, f, y, \mu) \rightarrow \text{ct}$: The encryption algorithm takes as input the public key mpk , a policy function f , a target output $y \in \{0, 1\}^\ell$, and a message μ . It outputs a ciphertext ct .

$\text{Dec}(\text{sk}_x, \text{ct}) \rightarrow \mu$: The decryption algorithm takes as input a private key sk_x and a ciphertext ct . It outputs the decrypted message μ .

Correctness. A ciphertext policy compute and compare ABE is correct if for every message μ , function f , output y , and attribute x such that $f(x) = y$, there exists a negligible function $\text{negl}(\cdot)$ such that for all $\lambda, L, \ell \in \mathbb{N}$:

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

Output-Adaptive Security. A CP-ABE scheme satisfies output-adaptive security if for every stateful PPT adversary $\mathcal{A}$, there exists a negligible function $\text{negl}(\cdot)$ such that the advantage of $\mathcal{A}$ in the following experiment is negligible:

$$\Pr \left[\begin{array}{l} f \leftarrow \mathcal{A}(1^\lambda, 1^L, 1^\ell) \\ (\text{mpk}, \text{msk}) \leftarrow \text{Setup}(1^\lambda, 1^L, 1^\ell) \\ (y^*, \mu_0, \mu_1) \leftarrow \mathcal{A}^{\text{KeyGen}(\text{msk}, \cdot)}(\text{mpk}) \\ b \leftarrow \{0, 1\} \\ \text{ct} \leftarrow \text{Enc}(\text{mpk}, f, y^*, \mu_b) \\ b' \leftarrow \mathcal{A}^{\text{KeyGen}(\text{msk}, \cdot)}(\text{ct}) \end{array} \right] \leq \frac{1}{2} + \text{negl}(\lambda),$$

where $\mathcal{A}$ is admissible if for all queried attribute vectors x_i to the KeyGen oracle, it holds that $f(x_i) \neq y^*$.

3.2 Construction

Parameters. Let χ be the truncated discrete gaussian distribution set as $\tilde{D}_{\mathbb{Z}, \sigma}$. Fix any $\epsilon < 1/2$. Let n, m, q, p, Q, σ be parameters with the following constraints:

- $n = \text{poly}(\lambda)$, $q \leq 2^{n^\epsilon}$, $m = \Omega(n \log q)$ (for LWE security)
- $\sigma \geq L \cdot m^{\Omega(d)} + \ell \cdot m$ (for correctness)
- $p \geq \frac{2Q}{\epsilon_A} = \text{poly}(\lambda)$ (for admissible hash)

One possible parameter setting that satisfies the above requirement is as follows:

$$n = 2\lambda/\epsilon, m = n \log q, p = 2Q/\epsilon_A, q = 2^{n^\epsilon}, \sigma = L \cdot m^{\Theta(d)} + \ell \cdot m.$$

Notation. The matrix evaluation algorithms follow from Theorem 2.7. The parameter d denotes the circuit depth of the ABE policy function that specifies the access structure. $\epsilon_{\mathcal{A}}$ denotes the advantage of PPT adversary $\mathcal{A}$ against our scheme.

Setup($1^\lambda, 1^L, 1^\ell, 1^d$): It takes as input the security parameter λ , the description length L of function f , the function output length ℓ . It samples $(\mathbf{A}_0, \mathbf{T}_{\mathbf{A}_0}) \leftarrow \text{TrapGen}(1^n, 1^m, q)$, and $\mathbf{B}_0 \leftarrow \mathbb{Z}_q^{n \times m}$. It also samples $\mathbf{A}_i \leftarrow \mathbb{Z}_q^{n \times m}$ for $i \in [L]$, $\mathbf{B}_i \leftarrow \mathbb{Z}_q^{n \times m}$ for $i \in [\ell]$. Finally, it samples a uniform target vector $\mathbf{u} \leftarrow \mathbb{Z}_q^n$. It outputs $\text{mpk} := (\mathbf{A}_0, \mathbf{B}_0, \{\mathbf{A}_i\}_{i \in [L]}, \{\mathbf{B}_i\}_{i \in [\ell]}, \mathbf{u})$ and $\text{msk} := \mathbf{T}_{\mathbf{A}_0}$.

KeyGen(msk, x): It takes as input the master secret key msk and a function input x . Let $\mathcal{U}_x$ be the universal circuit with x hardwired. On input a circuit description f , let $y := f(x)$. It outputs the matrix $\mathcal{U}_x(f) = \sum_{i=1}^L y[i] \cdot \mathbf{B}_i$. It sets

$$\mathbf{H}_{\mathcal{U}_x} := \text{EvalF}(\mathcal{U}_x, [\mathbf{A}_1 \mid \dots \mid \mathbf{A}_L]), \quad \mathbf{A}_{\mathcal{U}_x} := [\mathbf{A}_1 \mid \dots \mid \mathbf{A}_L] \cdot \mathbf{H}_{\mathcal{U}_x}.$$

It then generates a short vector $\mathbf{e}_x \leftarrow \text{SampleRight}(\mathbf{A}_0, \mathbf{T}_{\mathbf{A}_0}, \mathbf{A}_{\mathcal{U}_x} + \mathbf{B}_0, \mathbf{u}, \sigma)$, such that $[\mathbf{A}_0 \mid \mathbf{A}_{\mathcal{U}_x} + \mathbf{B}_0] \cdot \mathbf{e}_x = \mathbf{u}$. It outputs $\text{sk}_x := \mathbf{e}_x$.

Enc(mpk, f, y, μ): It takes as input the public parameters mpk , a function f , function output y , and a message bit $\mu \in \{0, 1\}$. It constructs $\mathbf{A}_f$:

$$\mathbf{A}_f := [\mathbf{A}_0 \mid \mathbf{A}_1 - f[1]\mathbf{G} \mid \dots \mid \mathbf{A}_L - f[L]\mathbf{G}].$$

It samples $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, error vectors $\mathbf{e}_0 \leftarrow \chi^m, e_1 \leftarrow \chi$, $\mathbf{R}_i \leftarrow \{-1, 1\}^{m \times m}$ for $i \in [L]$, and $\mathbf{R}'_i \leftarrow \{-1, 1\}^{m \times m}$ for $i \in [\ell]$. It computes the ciphertext $(\mathbf{c}_0, \mathbf{c}_1, c_2)$ such that

$$\begin{aligned} \mathbf{c}_0^T &= \mathbf{s}^T \mathbf{A}_f + \mathbf{e}_0^T \cdot [\mathbf{I} \mid \mathbf{R}_1 \mid \dots \mid \mathbf{R}_L], \\ \mathbf{c}_1^T &= \mathbf{s}^T \left(\sum_{i=1}^{\ell} y[i] \cdot \mathbf{B}_i + \mathbf{B}_0 \right) + \mathbf{e}_0^T \left(\sum_{i=1}^{\ell} y[i] \cdot \mathbf{R}'_i \right), \\ c_2 &= \mathbf{s}^T \mathbf{u} + \mu \cdot \lfloor q/2 \rfloor + e_1. \end{aligned}$$

It outputs $\mathbf{c} := (f, \mathbf{c}_0, \mathbf{c}_1, c_2)$.

Dec($\text{sk}, x, \mathbf{c}$): It takes as input the secret key sk , function input x , and a ciphertext $\mathbf{c} = (f, \mathbf{c}_0, \mathbf{c}_1, c_2)$. It sets $\mathbf{H}_{\mathcal{U}_x, f}, \mathbf{c}'$ such that:

$$\mathbf{H}_{\mathcal{U}_x, f} := \text{EvalFX}(\mathcal{U}_x, [\mathbf{A}_1 \mid \dots \mid \mathbf{A}_L], f), \quad \mathbf{c}'^T := \mathbf{c}_0^T \cdot \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{\mathcal{U}_x, f} \end{bmatrix} + [\mathbf{0} \mid \mathbf{c}_1^T]$$

Next, it computes w where

$$w := c_2 - \mathbf{c}'^T \cdot \text{sk} \pmod{q}.$$

The algorithm outputs 0 if $|w| < q/4$, and 1 otherwise.

3.3 Correctness

For the correctness property to hold, we need to prove that if $f(x) = y$, then the decryption algorithm recovers the message bit μ with high probability. By the properties of the matrix evaluation algorithms in Theorem 2.7, there exists a low-norm matrix $\mathbf{H}_{\mathcal{U}_x, f}$ such that:

$$\begin{aligned} \mathbf{A}_f \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{\mathcal{U}_x, f} \end{bmatrix} &= [\mathbf{A}_0 \mid \mathbf{A}_1 - f[1]\mathbf{G} \mid \cdots \mid \mathbf{A}_L - f[L]\mathbf{G}] \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{\mathcal{U}_x, f} \end{bmatrix} \\ &= [\mathbf{A}_0 \mid \mathbf{A}_{\mathcal{U}_x} - \mathcal{U}_x(f)]. \end{aligned}$$

In this construction, when $f(x) = y$, the homomorphic evaluation ensures that:

$$\mathbf{A}_f \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{\mathcal{U}_x, f} \end{bmatrix} = \left[\mathbf{A}_0 \mid \mathbf{A}_{\mathcal{U}_x} - \sum_{i=1}^{\ell} y[i]\mathbf{B}_i \right].$$

We now expand the term $\mathbf{c}'^T$ computed during decryption:

$$\begin{aligned} \mathbf{c}'^T &= \mathbf{c}_0^T \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{\mathcal{U}_x, f} \end{bmatrix} + [\mathbf{0} \mid \mathbf{c}_1^T] \\ &= (\mathbf{s}^T \mathbf{A}_f + \mathbf{e}_0^T \cdot [\mathbf{I} \mid \mathbf{R}_1 \mid \cdots \mid \mathbf{R}_L]) \cdot \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{\mathcal{U}_x, f} \end{bmatrix} \\ &\quad + \left[\mathbf{0} \mid \mathbf{s}^T \left(\sum_{i=1}^{\ell} y[i] \cdot \mathbf{B}_i + \mathbf{B}_0 \right) + \mathbf{e}_0^T \left(\sum_{i=1}^{\ell} y^*[i] \cdot \mathbf{R}'_i \right) \right] \\ &= \mathbf{s}^T \left(\mathbf{A}_f \cdot \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{\mathcal{U}_x, f} \end{bmatrix} + \left[\mathbf{0} \mid \sum_{i=1}^{\ell} y[i] \cdot \mathbf{B}_i + \mathbf{B}_0 \right] \right) \\ &\quad + \mathbf{e}_0^T \cdot [\mathbf{I} \mid \mathbf{R}_1 \mid \cdots \mid \mathbf{R}_L] \cdot \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{\mathcal{U}_x, f} \end{bmatrix} \\ &\quad + \left[\mathbf{0} \mid \mathbf{e}_0^T \left(\sum_{i=1}^{\ell} y^*[i] \cdot \mathbf{R}'_i \right) \right] \\ &= \mathbf{s}^T \cdot [\mathbf{A}_0 \mid \mathbf{A}_{\mathcal{U}_x} + \mathbf{B}_0] \\ &\quad + \mathbf{e}_0^T \cdot [\mathbf{I} \mid \mathbf{R}_1 \mid \cdots \mid \mathbf{R}_L] \cdot \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{\mathcal{U}_x, f} \end{bmatrix} + \left[\mathbf{0} \mid \mathbf{e}_0^T \left(\sum_{i=1}^{\ell} y[i] \cdot \mathbf{R}'_i \right) \right]. \end{aligned}$$

Denote $\mathbf{e}'$ as the highlighted noise in the above equation, such that $\mathbf{c}'^T = \mathbf{s}^T \cdot [\mathbf{A}_0 \mid \mathbf{A}_{\mathcal{U}_x} + \mathbf{B}_0] + \mathbf{e}'^T$. Based on the parameter settings and the truncation properties of the discrete Gaussian distribution, $\|\mathbf{e}_0\|_{\infty}, \|e_1\|_{\infty} \leq \sigma m$, and thus $\|\mathbf{e}'\|_{\infty}$ is upper bounded by $\sigma \cdot L \cdot m^{O(d)} + \sigma \cdot \ell \cdot m^2$. Next, we analyze the expression for w . Recalling that $\mathbf{sk}_x = \mathbf{e}_x$ satisfies $[\mathbf{A}_0 \mid \mathbf{A}_{\mathcal{U}_x} + \mathbf{B}_0] \cdot \mathbf{e}_x = \mathbf{u}$:

$$\begin{aligned} w &= c_2 - \mathbf{c}'^T \cdot \mathbf{sk}_x \\ &= (\mathbf{s}^T \cdot \mathbf{u} + \mu \cdot \lfloor q/2 \rfloor + e_1) - (\mathbf{s}^T \cdot [\mathbf{A}_0 \mid \mathbf{A}_{\mathcal{U}_x} + \mathbf{B}_0] + \mathbf{e}'^T) \cdot \mathbf{e}_x \\ &= \mu \cdot \lfloor q/2 \rfloor + e_1 - \mathbf{e}'^T \cdot \mathbf{e}_x. \end{aligned}$$

Finally, we analyze the norm bound of the highlighted error. Given that $\|\mathbf{e}_x\|_\infty \leq 2\sigma m$ and $\|\mathbf{H}_{\mathcal{U}_x, f}\|_\infty \leq m^{O(d)}$, the following holds:

$$\left\| \mathbf{e}_1 - \mathbf{e}'^T \cdot \mathbf{e}_x \right\|_\infty \leq \sigma m + \sigma(Lm^{O(d)} + \ell m^2) \cdot 2\sigma m$$

Note that q is sufficiently large such that the error bound is much smaller than $q/4$, the algorithm outputs $\mu = 0$ if $|w| < q/4$, and $\mu = 1$ if $|w| \approx q/2$. Thus, the decryption algorithm succeeds with high probability.

3.4 Output-Adaptive Security

We prove the security of the construction through a sequence of hybrid games. Let $\text{Adv}_{\mathcal{A}}^i$ denote the advantage of a PPT adversary $\mathcal{A}$ in Hybrid i .

Hybrid 0: This is the original IND-CPA security game. The challenger generates the public parameters honestly and answers all queries using the master secret key $\text{msk} := \mathbf{T}_{\mathbf{A}}$. The attacker makes a number of Q queries.

1. The attacker $\mathcal{A}$ outputs the target function f .
2. The challenger samples $(\mathbf{A}_0, \mathbf{T}_{\mathbf{A}_0}) \leftarrow \text{TrapGen}(1^n, 1^m, q)$, and $\mathbf{B}_0 \leftarrow \mathbb{Z}_q^{n \times m}$. It also samples $\mathbf{A}_1, \dots, \mathbf{A}_L \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{R}_1, \dots, \mathbf{R}_L \leftarrow \{-1, 1\}^{m \times m}$, $\mathbf{B}_1, \dots, \mathbf{B}_\ell \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{R}'_1, \dots, \mathbf{R}'_\ell \leftarrow \{-1, 1\}^{m \times m}$. It then samples the target vector $\mathbf{u} \leftarrow \mathbb{Z}_q^n$. It outputs $\text{mpk} := (\mathbf{A}_0, \mathbf{B}_0, \{\mathbf{A}_i\}_{i \in [L]}, \{\mathbf{B}_i\}_{i \in [\ell]}, \mathbf{u})$, and samples a bit $b \leftarrow \{0, 1\}$.
3. During each key query, the attacker outputs x . The challenger sets $\mathcal{U}_x$ as in the construction. It computes $\mathbf{H}_{\mathcal{U}_x} := \text{EvalF}(\mathcal{U}_x, [\mathbf{A}_1 \mid \dots \mid \mathbf{A}_L])$, and $\mathbf{A}_{\mathcal{U}_x} := [\mathbf{A}_1 \mid \dots \mid \mathbf{A}_L] \cdot \mathbf{H}_{\mathcal{U}_x}$. Next, the challenger generates a short vector $\mathbf{e}_x \leftarrow \text{SampleRight}(\mathbf{A}_0, \mathbf{T}_{\mathbf{A}_0}, \mathbf{A}_{\mathcal{U}_x} + \mathbf{B}_0, \mathbf{u}, \sigma)$, such that $[\mathbf{A}_0 \mid \mathbf{A}_{\mathcal{U}_x} + \mathbf{B}_0] \cdot \mathbf{e}_x = \mathbf{u}$. It outputs $\mathbf{e}_x$ and adds x to $\mathcal{X}$.
4. During the challenge query, $\mathcal{A}$ outputs y^*, μ_0, μ_1 . The challenger samples $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, error vectors $\mathbf{e}_0 \leftarrow \chi^m$, $e_1 \leftarrow \chi$. It sets

$$\begin{aligned} \mathbf{c}_0^T &= \mathbf{s}^T \cdot [\mathbf{A}_0 \mid \mathbf{A}_1 - f[1]\mathbf{G} \mid \dots \mid \mathbf{A}_L - f[L]\mathbf{G}] \\ &\quad + \mathbf{e}_0^T \cdot [\mathbf{I} \mid \mathbf{R}_1 \mid \dots \mid \mathbf{R}_L], \\ \mathbf{c}_1^T &= \mathbf{s}^T \cdot \left(\sum_{i=1}^{\ell} y^*[i] \cdot \mathbf{B}_i + \mathbf{B}_0 \right) + \mathbf{e}_0^T \cdot \left(\sum_{i=1}^{\ell} y^*[i] \cdot \mathbf{R}'_i \right). \end{aligned}$$

It sets $c_2 := \mathbf{s}^T \cdot \mathbf{u} + \mu_b \cdot \lfloor q/2 \rfloor + e_1^T$. It outputs $(f, \mathbf{c}_0, \mathbf{c}_1, c_2)$.

5. $\mathcal{A}$ outputs a guess b' . It wins if and only if $b' = b$ and $f(x) \neq y^*$ for all $x \in \mathcal{X}$.

Hybrid 1: This is identical to Hybrid 0, except that the challenger modifies the sampling of matrices $\{\mathbf{A}_i\}_{i \in [L]}$ and embeds the function f .

2. The challenger samples $(\mathbf{A}_0, \mathbf{T}_{\mathbf{A}_0}) \leftarrow \text{TrapGen}(1^n, 1^m, q)$, and $\mathbf{B}_0 \leftarrow \mathbb{Z}_q^{n \times m}$. It also samples $\mathbf{R}_1, \dots, \mathbf{R}_L \leftarrow \{-1, 1\}^{m \times m}$, $\mathbf{B}_1, \dots, \mathbf{B}_\ell \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{R}'_1, \dots, \mathbf{R}'_\ell \leftarrow \{-1, 1\}^{m \times m}$. **Instead of sampling $\mathbf{A}_i \leftarrow \mathbb{Z}_q^{n \times m}$, it sets $\mathbf{A}_i := \mathbf{A}_0 \mathbf{R}_i + f[i]\mathbf{G} \pmod{q}$ for $i \in [L]$.** It then samples the target vector $\mathbf{u} \leftarrow \mathbb{Z}_q^n$. It outputs $\text{mpk} := (\mathbf{A}_0, \mathbf{B}_0, \{\mathbf{A}_i\}_{i \in [L]}, \{\mathbf{B}_i\}_{i \in [\ell]}, \mathbf{u})$, and samples a bit $b \leftarrow \{0, 1\}$.

Hybrid 2: In this game, the challenger transitions from using $\mathbf{T}_{\mathbf{A}_0}$ to using a trapdoor $\mathbf{T}_{\mathbf{B}_0}$ for $\mathbf{B}_0$ to simulate secret keys. It also embeds an admissible hash $H(y) := 1 + \sum h_i y[i]$.

2. The challenger samples $\mathbf{A}_0 \leftarrow \mathbb{Z}_q^{n \times m}$ uniformly without a trapdoor. It samples $(\mathbf{B}_0, \mathbf{T}_{\mathbf{B}_0}) \leftarrow \text{TrapGen}(1^n, 1^m, q)$. As in Hybrid 1, the challenger sets $\mathbf{A}_i := \mathbf{A}_0 \mathbf{R}_i + f[i] \mathbf{G} \pmod{q}$ for $i \in [L]$. Next, instead of sampling $\mathbf{B}_1, \dots, \mathbf{B}_\ell \leftarrow \mathbb{Z}_q^{n \times m}$, it samples scalar $h_i \leftarrow \mathbb{Z}_p$ and sets $\mathbf{B}_i := \mathbf{A}_0 \mathbf{R}'_i + h_i \mathbf{B}_0$ for $i \in [\ell]$. It then samples the target vector $\mathbf{u} \leftarrow \mathbb{Z}_q^n$. It outputs $\text{mpk} := (\mathbf{A}_0, \mathbf{B}_0, \{\mathbf{A}_i\}_{i \in [L]}, \{\mathbf{B}_i\}_{i \in [\ell]}, \mathbf{u})$, and samples a bit $b \leftarrow \{0, 1\}$.
3. During each key query, the attacker outputs x . The challenger computes $y = f(x)$ and the admissible hash value $H(y) = 1 + \sum_{i=1}^\ell y[i] h_i$. If $H(y) = 0$, the challenger aborts. Otherwise, it computes $\mathbf{H}_{\mathcal{U}_x, f} := \text{EvalFX}(\mathcal{U}_x, [\mathbf{A}_0 \mathbf{R}_1 \mid \dots \mid \mathbf{A}_0 \mathbf{R}_L], f)$, $\mathbf{A}_{\mathcal{U}_x, f} := [\mathbf{A}_0 \mathbf{R}_1 \mid \dots \mid \mathbf{A}_0 \mathbf{R}_L] \cdot \mathbf{H}_{\mathcal{U}_x, f}$ and sets the target matrix $\mathbf{F}_x := [\mathbf{A}_0 \mid \mathbf{A}_{\mathcal{U}_x, f} + \sum_{i=1}^\ell y[i] (\mathbf{A}_0 \cdot \mathbf{R}'_i + h_i \mathbf{B}_0) + \mathbf{B}_0]$. Set $\mathbf{R}_{\text{total}}$ as $[\mathbf{R}_1 \mid \dots \mid \mathbf{R}_L] \cdot \mathbf{H}_{\mathcal{U}_x, f} + \sum_{i=1}^\ell y[i] \mathbf{R}'_i$. Note that $\mathbf{F}_x$ can be written as $[\mathbf{A}_0 \mid \mathbf{A}_0 \cdot \mathbf{R}_{\text{total}} + H(y) \mathbf{B}_0]$. The challenger samples $\text{sk}_x \leftarrow \text{SampleLeft}(\mathbf{A}_0, \mathbf{B}_0, \mathbf{T}_{\mathbf{B}_0}, \mathbf{R}_{\text{total}}, H(y), \mathbf{u}, \sigma)$.

Hybrid 3: The challenge ciphertext is replaced by a uniformly random vector.

4. During the challenge query, if $H(y^*) \neq 0$, the game aborts. Otherwise, the challenger samples $\mathbf{r}_0 \leftarrow \mathbb{Z}_q^m$, $r_1 \leftarrow \mathbb{Z}_q$, $\mathbf{e}_0 \leftarrow \chi^m$. It sets $\mathbf{c}_0$ and $\mathbf{c}_1$ such that $\mathbf{c}_0^T = \mathbf{r}_0^T \cdot [\mathbf{I} \mid \mathbf{R}_1 \mid \dots \mid \mathbf{R}_L]$, $\mathbf{c}_1^T = \mathbf{r}_0^T \cdot \sum_{i=1}^\ell y^*[i] \cdot \mathbf{R}'_i$. The challenger outputs $\mathbf{c}^* := (f, \mathbf{c}_0, \mathbf{c}_1, r_1)$.

Lemma 3.1. For any adversary $\mathcal{A}$ of unbounded running time, there exists a negligible function $\text{negl}(\cdot)$ such that $|\text{Adv}_{\mathcal{A}}^0 - \text{Adv}_{\mathcal{A}}^1| \leq \text{negl}(\lambda)$.

Proof. It follows from Theorem 2.3 that for random $\mathbf{A}_0$ and small-norm $\mathbf{R}_i$,

$$\begin{aligned} & (\mathbf{A}_0, \mathbf{A}_0 \cdot [\mathbf{R}_1 \mid \dots \mid \mathbf{R}_L], \mathbf{e}_0^T \cdot [\mathbf{R}_1 \mid \dots \mid \mathbf{R}_L]) \\ & \approx_s (\mathbf{A}_0, [\mathbf{A}_1 \mid \dots \mid \mathbf{A}_L], \mathbf{e}_0^T \cdot [\mathbf{R}_1 \mid \dots \mid \mathbf{R}_L]). \end{aligned}$$

Thus, in the view of the attacker, the distribution of $\{\mathbf{A}_i\}_{i \in [L]}$ in Hybrid 1 is statistically close to Hybrid 0. $\square$

Lemma 3.2. For any adversary $\mathcal{A}$ of unbounded running time, conditioned on $H(f(x)) \neq 0$ for every secret-key query x , there exists a negligible function $\text{negl}(\cdot)$ such that $|\text{Adv}_{\mathcal{A}}^1 - \text{Adv}_{\mathcal{A}}^2| \leq \text{negl}(\lambda)$.

Proof. By Theorem 2.3,

$$\begin{aligned} & (\mathbf{A}_0, \mathbf{A}_0 \cdot [\mathbf{R}'_1 \mid \dots \mid \mathbf{R}'_\ell], \mathbf{e}_0^T \cdot [\mathbf{R}'_1 \mid \dots \mid \mathbf{R}'_\ell]) \\ & \approx_s (\mathbf{A}_0, [\mathbf{B}_1 \mid \dots \mid \mathbf{B}_\ell], \mathbf{e}_0^T \cdot [\mathbf{R}'_1 \mid \dots \mid \mathbf{R}'_\ell]). \end{aligned}$$

Furthermore, writing $y = f(x)$, the the matrix-evaluation relation give

$$\begin{aligned} [\mathbf{A}_0 \mid \mathbf{A}_{\mathcal{U}_x} + \mathbf{B}_0] &= \left[\mathbf{A}_0 \mid \mathbf{A}_{\mathcal{U}_x, f} + \sum_{i=1}^\ell y[i] \mathbf{B}_i + \mathbf{B}_0 \right] \\ &= [\mathbf{A}_0 \mid \mathbf{A}_0 \mathbf{R}_{\text{total}} + H(y) \mathbf{B}_0]. \end{aligned}$$

Thus, Hybrid 1 and Hybrid 2 sample a preimage for the same key-generation matrix, using $\mathbf{T}_{\mathbf{A}_0}$ via `SampleRight` and $\mathbf{T}_{\mathbf{B}_0}$ via `SampleLeft`, respectively. Since $H(y) \neq 0$, the two key distributions are statistically indistinguishable by Theorem 2.5. Therefore, Hybrid 1 and Hybrid 2 are statistically indistinguishable. $\square$

Let $x_1, \dots, x_Q$ denote the adversary's secret-key queries and set $y_i := f^*(x_i)$. Define the good partitioning event

$$\text{Good} := \bigcap_{i \in [Q]} \{H(y_i) \neq 0\} \cap \{H(y^*) = 0\}.$$

Lemma 3.3. Assuming that LWE holds, for any PPT adversary $\mathcal{A}$, conditioned on the good partitioning event `Good` defined above, there exists a negligible function $\text{negl}(\cdot)$ such that $|\text{Adv}_{\mathcal{A}}^2 - \text{Adv}_{\mathcal{A}}^3| \leq \text{negl}(\lambda)$.

Proof. Assume that the above is non-negligible, we construct a reduction algorithm $\mathcal{B}$ to solve LWE. $\mathcal{B}$ queries the LWE challenger and receives $(\mathbf{A}_0, \mathbf{u}, \mathbf{r}_0^T, r_1)$. Here, either $\mathbf{r}_0^T$ and r_1 are uniformly random over their respective domains, or $\mathbf{r}_0^T = \mathbf{s}^T \mathbf{A}_0 + \mathbf{e}_0^T$ and $r_1 = \mathbf{s}^T \mathbf{u} + e_1$, where $\mathbf{s}$ is the LWE secret and $\mathbf{e}_0, e_1$ are noise terms sampled by the challenger. $\mathcal{B}$ then sets $\mathbf{A}_0$ as the public matrix and generates $\mathbf{B}_0$ with $\mathbf{T}_{\mathbf{B}_0}$ via `TrapGen`. It simulates $\mathbf{A}_i, \mathbf{B}_i$ as in Hybrid 2 using the secret h_i . It outputs $(\mathbf{A}_0, \mathbf{B}_0, \{\mathbf{A}_i\}_{i \in [L]}, \{\mathbf{B}_i\}_{i \in [\ell]}, \mathbf{u})$. For each key query, $\mathcal{B}$ responds using $\mathbf{T}_{\mathbf{B}_0}$ via `SampleRight`. If $H(y) = 0$ for any query, it aborts. During the challenge phase, $\mathcal{B}$ receives (x^*, μ, y^*) . If $H(y^*) \neq 0$, it aborts. Otherwise, it samples $\mathbf{e}_0 \leftarrow \chi^m$ and constructs $\mathbf{c}_0$ and $\mathbf{c}_1$ such that $\mathbf{c}_0^T = \mathbf{r}_0^T \cdot [\mathbf{I} \mid \mathbf{R}_1 \mid \dots \mid \mathbf{R}_L]$, $\mathbf{c}_1^T = \mathbf{r}_0^T \cdot \sum_{i=1}^{\ell} y^*[i] \cdot \mathbf{R}'_i$. It outputs $\mathbf{c}^* := (f, \mathbf{c}_0, \mathbf{c}_1, r_1)$.

It remains to account for the abort probability in the partitioning step. Conditioned on `Good`, if $(\mathbf{A}_0, \mathbf{u}, \mathbf{r}_0^T, r_1)$ is an LWE sample, the above is identically distributed as Hybrid 2 in the view of $\mathcal{A}$. If $(\mathbf{A}_0, \mathbf{u}, \mathbf{r}_0^T, r_1)$ is uniform, then it is identically distributed as Hybrid 3.

The conditioning can be removed using the standard artificial-abort analysis for partitioning-based reductions. Following the refined analysis of the partitioning function ($H(y)$ in our reduction) presented in [HKK⁺24] (Section 5.4, Remark 2), the LWE solver $\mathcal{B}$ inherits an advantage of $O\left((\text{Adv}_{\mathcal{A}}^2 - \text{Adv}_{\mathcal{A}}^3)^2 / Q^2\right)$, which remains non-negligible as long as $\epsilon_{\mathcal{A}}$ is non-negligible and Q is polynomially bounded. $\square$

Theorem 3.4. Assuming that LWE assumption holds, then the above ciphertext policy ABE satisfies output-adaptive security.

Proof. The theorem follows from Theorems 3.1 to 3.3, and the fact that the adversary has exactly 1/2 advantage in Hybrid 3. $\square$

References

- [ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572, French Riviera, May 30 – June 3, 2010. Springer Berlin Heidelberg, Germany.
- [ABSV15] Prabhanjan Ananth, Zvika Brakerski, Gil Segev, and Vinod Vaikuntanathan. From selective to adaptive security in functional encryption. In Rosario Gennaro and Matthew

- J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 657–677, Santa Barbara, CA, USA, August 16–20, 2015. Springer Berlin Heidelberg, Germany.
- [AC17] Shashank Agrawal and Melissa Chase. Simplifying design and analysis of complex predicate encryption schemes. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part I*, volume 10210 of *Lecture Notes in Computer Science*, pages 627–656, Paris, France, April 30 – May 4, 2017. Springer, Cham, Switzerland.
- [Agr17] Shweta Agrawal. Stronger security for reusable garbled circuits, general definitions and attacks. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 3–35, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Cham, Switzerland.
- [Ajt99] Miklós Ajtai. Generating hard instances of the short basis problem. In Jiri Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *ICALP 99: 26th International Colloquium on Automata, Languages and Programming*, volume 1644 of *Lecture Notes in Computer Science*, pages 1–9, Prague, Czech Republic, July 11–15, 1999. Springer Berlin Heidelberg, Germany.
- [AMR25] Damiano Abram, Giulio Malavolta, and Lawrence Roy. Succinct oblivious tensor evaluation and applications: Adaptively-secure laconic function evaluation and trapdoor hashing for all circuits. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 1875–1886, 2025.
- [Att14] Nuttapong Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 557–577, Copenhagen, Denmark, May 11–15, 2014. Springer Berlin Heidelberg, Germany.
- [Att16] Nuttapong Attrapadung. Dual system encryption framework in prime-order groups via computational pair encodings. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016, Part II*, volume 10032 of *Lecture Notes in Computer Science*, pages 591–623, Hanoi, Vietnam, December 4–8, 2016. Springer Berlin Heidelberg, Germany.
- [BB04] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 443–459, Santa Barbara, CA, USA, August 15–19, 2004. Springer Berlin Heidelberg, Germany.
- [BGG⁺14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 533–556, Copenhagen, Denmark, May 11–15, 2014. Springer Berlin Heidelberg, Germany.

- [BGI⁺01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18, August 2001.
- [BM24] Zvika Brakerski and Stav Medina. Limits on adaptive security for attribute-based encryption. In Elette Boyle and Mohammad Mahmoody, editors, *TCC 2024: 22nd Theory of Cryptography Conference, Part III*, volume 15366 of *Lecture Notes in Computer Science*, pages 91–123, Milan, Italy, December 2–6, 2024. Springer, Cham, Switzerland.
- [BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 253–273, Providence, RI, USA, March 28–30, 2011. Springer Berlin Heidelberg, Germany.
- [BTVW17] Zvika Brakerski, Rotem Tsabary, Vinod Vaikuntanathan, and Hoeteck Wee. Private constrained PRFs (and more) from LWE. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 264–302, Baltimore, MD, USA, November 12–15, 2017. Springer, Cham, Switzerland.
- [BV16] Zvika Brakerski and Vinod Vaikuntanathan. Circuit-ABE from LWE: Unbounded attributes and semi-adaptive security. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 363–384, Santa Barbara, CA, USA, August 14–18, 2016. Springer Berlin Heidelberg, Germany.
- [CG26] Jiaqi Cheng and Rishab Goyal. Collusion-resistant constrained prfs for compute-&-compare predicates from lwe. In *Annual International Cryptology Conference*, pages 508–538. Springer, 2026.
- [CGKW18] Jie Chen, Junqing Gong, Lucas Kowalczyk, and Hoeteck Wee. Unbounded ABE via bilinear entropy expansion, revisited. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 503–534, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Cham, Switzerland.
- [CHKP12] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *Journal of Cryptology*, 25(4):601–639, October 2012.
- [CW14] Jie Chen and Hoeteck Wee. Semi-adaptive attribute-based encryption and improved delegation for Boolean formula. In Michel Abdalla and Roberto De Prisco, editors, *SCN 14: 9th International Conference on Security in Communication Networks*, volume 8642 of *Lecture Notes in Computer Science*, pages 277–297, Amalfi, Italy, September 3–5, 2014. Springer, Cham, Switzerland.
- [DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam D. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

- [DRS04] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540, Interlaken, Switzerland, May 2–6, 2004. Springer Berlin Heidelberg, Germany.
- [Gen06] Craig Gentry. Practical identity-based encryption without random oracles. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 445–464, St. Petersburg, Russia, May 28 – June 1, 2006. Springer Berlin Heidelberg, Germany.
- [GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing*, pages 467–476, Palo Alto, CA, USA, June 1–4, 2013. ACM Press.
- [GKW16] Rishab Goyal, Venkata Koppula, and Brent Waters. Semi-adaptive security and bundling functionalities made generic and easy. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B: 14th Theory of Cryptography Conference, Part II*, volume 9986 of *Lecture Notes in Computer Science*, pages 361–388, Beijing, China, October 31 – November 3, 2016. Springer Berlin Heidelberg, Germany.
- [GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In Chris Umans, editor, *58th Annual Symposium on Foundations of Computer Science*, pages 612–621, Berkeley, CA, USA, October 15–17, 2017. IEEE Computer Society Press.
- [GLW21] Rishab Goyal, Jiahui Liu, and Brent Waters. Adaptive security via deletion in attribute-based encryption: Solutions from search assumptions in bilinear groups. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021, Part IV*, volume 13093 of *Lecture Notes in Computer Science*, pages 311–341, Singapore, December 6–10, 2021. Springer, Cham, Switzerland.
- [GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006: 13th Conference on Computer and Communications Security*, pages 89–98, Alexandria, Virginia, USA, October 30 – November 3, 2006. ACM Press. Available as Cryptology ePrint Archive Report 2006/309.
- [GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 197–206, Victoria, BC, Canada, May 17–20, 2008. ACM Press.
- [GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92, Santa Barbara, CA, USA, August 18–22, 2013. Springer Berlin Heidelberg, Germany.

- [GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing*, pages 545–554, Palo Alto, CA, USA, June 1–4, 2013. ACM Press.
- [GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 503–523, Santa Barbara, CA, USA, August 16–20, 2015. Springer Berlin Heidelberg, Germany.
- [GY25] Rishab Goyal and Saikumar Yadugiri. Delegatable ABE with $O(1)$ delegations from witness encryption. Cryptology ePrint Archive, Report 2025/407, 2025.
- [GY26a] Rishab Goyal and Saikumar Yadugiri. Adaptively-secure flexible and identity-based broadcast encryption from decomposed LWE. Cryptology ePrint Archive, Paper 2026/862, 2026.
- [GY26b] Rishab Goyal and Saikumar Yadugiri. Equivocal broadcast encryption: Adaptively-secure optimal distributed broadcast encryption from lattices. In *Annual International Cryptology Conference*, pages 414–445. Springer, 2026.
- [HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.
- [HKK⁺24] Goichiro Hanaoka, Shuichi Katsumata, Kei Kimura, Kaoru Takemure, and Shota Yamada. Tighter adaptive IBEs and VRFs: Revisiting waters’ artificial abort. In Elette Boyle and Mohammad Mahmoody, editors, *TCC 2024: 22nd Theory of Cryptography Conference, Part III*, volume 15366 of *Lecture Notes in Computer Science*, pages 124–155, Milan, Italy, December 2–6, 2024. Springer, Cham, Switzerland.
- [KNYY20] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Adaptively secure inner product encryption from LWE. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020, Part III*, volume 12493 of *Lecture Notes in Computer Science*, pages 375–404, Daejeon, South Korea, December 7–11, 2020. Springer, Cham, Switzerland.
- [KW20] Lucas Kowalczyk and Hoeteck Wee. Compact adaptively secure abe for $nc\ 1$ from k -lin. *Journal of Cryptology*, 33(3):954–1002, 2020.
- [LOS⁺10] Allison B. Lewko, Tatsuoaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 62–91, French Riviera, May 30 – June 3, 2010. Springer Berlin Heidelberg, Germany.
- [LW11a] Allison B. Lewko and Brent Waters. Decentralizing attribute-based encryption. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume

- 6632 of *Lecture Notes in Computer Science*, pages 568–588, Tallinn, Estonia, May 15–19, 2011. Springer Berlin Heidelberg, Germany.
- [LW11b] Allison B. Lewko and Brent Waters. Unbounded HIBE and attribute-based encryption. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 547–567, Tallinn, Estonia, May 15–19, 2011. Springer Berlin Heidelberg, Germany.
 - [LW12] Allison B. Lewko and Brent Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 180–198, Santa Barbara, CA, USA, August 19–23, 2012. Springer Berlin Heidelberg, Germany.
 - [MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718, Cambridge, UK, April 15–19, 2012. Springer Berlin Heidelberg, Germany.
 - [OT10] Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 191–208, Santa Barbara, CA, USA, August 15–19, 2010. Springer Berlin Heidelberg, Germany.
 - [QWW18] Willy Quach, Hoeteck Wee, and Daniel Wichs. Laconic function evaluation and applications. In Mikkel Thorup, editor, *59th Annual Symposium on Foundations of Computer Science*, pages 859–870, Paris, France, October 7–9, 2018. IEEE Computer Society Press.
 - [Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 84–93, Baltimore, MA, USA, May 22–24, 2005. ACM Press.
 - [RW13] Yannis Rouselakis and Brent Waters. Practical constructions and new proof methods for large universe attribute-based encryption. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 463–474, 2013.
 - [RW22] Doreen Riepel and Hoeteck Wee. FABEO: Fast attribute-based encryption with optimal security. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022: 29th Conference on Computer and Communications Security*, pages 2491–2504, Los Angeles, CA, USA, November 7–11, 2022. ACM Press.
 - [SW05] Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473, Aarhus, Denmark, May 22–26, 2005. Springer Berlin Heidelberg, Germany.

- [Tsa19] Rotem Tsabary. Fully secure attribute-based encryption for t-CNF from LWE. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 62–85, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Cham, Switzerland.
- [Wat05] Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127, Aarhus, Denmark, May 22–26, 2005. Springer Berlin Heidelberg, Germany.
- [Wat09] Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636, Santa Barbara, CA, USA, August 16–20, 2009. Springer Berlin Heidelberg, Germany.
- [Wat12] Brent Waters. Functional encryption for regular languages. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 218–235, Santa Barbara, CA, USA, August 19–23, 2012. Springer Berlin Heidelberg, Germany.
- [Wat15] Brent Waters. A punctured programming approach to adaptively secure functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 678–697, Santa Barbara, CA, USA, August 16–20, 2015. Springer Berlin Heidelberg, Germany.
- [WW24] Brent Waters and Daniel Wichs. Adaptively secure attribute-based encryption from witness encryption. In Elette Boyle and Mohammad Mahmoody, editors, *TCC 2024: 22nd Theory of Cryptography Conference, Part III*, volume 15366 of *Lecture Notes in Computer Science*, pages 65–90, Milan, Italy, December 2–6, 2024. Springer, Cham, Switzerland.
- [WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In Chris Umans, editor, *58th Annual Symposium on Foundations of Computer Science*, pages 600–611, Berkeley, CA, USA, October 15–17, 2017. IEEE Computer Society Press.