Circuit-ABE from LWE: Unbounded Attributes and Semi-adaptive Security

Zvika Brakerski[1,2](✉) and Vinod Vaikuntanathan[1,2]

[1] Weizmann Institute of Science, Rehovot, Israel
zvika.brakerski@weizmann.ac.il
[2] MIT, Cambridge, USA

Abstract. We construct an LWE-based key-policy attribute-based encryption (ABE) scheme that supports attributes of *unbounded polynomial length*. Namely, the size of the public parameters is a fixed polynomial in the security parameter and a depth bound, and with these fixed length parameters, one can encrypt attributes of arbitrary length. Similarly, any polynomial size circuit that adheres to the depth bound can be used as the policy circuit regardless of its input length (recall that a depth d circuit can have as many as 2^d inputs). This is in contrast to previous LWE-based schemes where the length of the public parameters has to grow linearly with the maximal attribute length.

We prove that our scheme is *semi-adaptively secure*, namely, the adversary can choose the challenge attribute after seeing the public parameters (but before any decryption keys). Previous LWE-based constructions were only able to achieve selective security. (We stress that the "complexity leveraging" technique is not applicable for unbounded attributes).

We believe that our techniques are of interest at least as much as our end result. Fundamentally, selective security and bounded attributes are both shortcomings that arise out of the current LWE proof techniques that *program the challenge attributes into the public parameters*. The LWE toolbox we develop in this work allows us to *delay this programming*. In a nutshell, the new tools include a way to generate an a-priori *unbounded* sequence of LWE matrices, and have fine-grained control over which trapdoor is embedded in each and every one of them, all with succinct representation.

1 Introduction

Key-policy attribute-based encryption [22,34] is a special type of public-key encryption scheme where a (master) public key mpk is used for encryption,

Z. Brakerski—Supported by the Israel Science Foundation (Grant No. 468/14), the Alon Young Faculty Fellowship, Binational Science Foundation (Grant No. 712307) and Google Faculty Research Award.

V. Vaikuntanathan—Research supported in part by DARPA Safeware Grant, NSF CAREER Award CNS-1350619, NSF Grant CNS-1413964 (MACS: A Modular Approach to Computer Security), US-Israel Binational Science Foundation Grant No. 712307, Alfred P. Sloan Research Fellowship, Microsoft Faculty Fellowship, NEC Corporation and a Steven and Renee Finn Career Development Chair from MIT.

M. Robshaw and J. Katz (Eds.): CRYPTO 2016, Part III, LNCS 9816, pp. 363–384, 2016.
DOI: 10.1007/978-3-662-53015-3_13

and users are associated to secret keys sk_f corresponding to (policy) functions $f : \mathcal{X} \to \{0, 1\}$. The encryption of a message μ is labeled with a public attribute $x \in \mathcal{X}$, and can be decrypted using sk_f if and only if $f(x) = 0$.[1]

Intuitively, the security requirement is collusion resistance: a coalition of users learns nothing about the plaintext message μ if none of their individual keys are authorized to decrypt the ciphertext.

The past few years have seen much progress in constructing secure and efficient attribute-based encryption (ABE) schemes from different assumptions and for different settings. The first constructions [10,22,23,25,27,30,36] apply to predicates computable by Boolean formulas (which are equivalent to log-depth computations). More recently, important progress has been made on constructions for the set of all polynomial-size circuits (of a-priori bounded polynomial depth): Gorbunov et al. [19] gave a construction from the Learning With Errors (LWE) assumption, and Garg et al. [14] gave a construction using multilinear maps. In both constructions the policy functions are represented as Boolean circuits composed of fan-in 2 gates, and the secret key size is proportional to the *size* of the circuit. Boneh et al. [9] constructed an "arithmetic" ABE scheme where the secret key size is independent of the circuit-size of the function f, but rather depends only on the circuit-depth. This in turn gave the first construction of compact reusable garbled circuits [9], and led to constructions of predicate encryption [20], homomorphic signatures [21] and constrained pseudo-random functions [11].

However, despite all this progress, there are several deficiencies in these constructions. The first is that in all of them, the length of the attribute, represented as a binary string, has to be determined during the initial setup. This is a problem not just for ABE, but also for all downstream constructions (of succinct single-use functional encryption, homomorphic signatures, predicate encryption, and so on) where the size of the input to be encrypted (or signed) is limited by the initial setup.[2] We know of three exceptions to this: the first is the (selectively secure) ABE construction of Lewko et al. [26] that handles *Boolean formulas*, under assumptions on bilinear maps and the second is the (fully secure) inner product encryption and ABE construction of Okamoto and Takashima [31] that again only handles Boolean formulas. Finally, there is the recent work of Ananth and Sahai [7] who show a functional encryption scheme for Turing machines that can take arbitrarily long inputs. In particular, this gives rise to an ABE scheme with the same properties, however this construction uses the huge hammer of indistinguishability obfuscation (IO) unlike the ones in the previous paragraph.

> Q1: *Is there an ABE scheme for general circuits with* unbounded attribute length *under standard complexity assumptions?*

[1] We follow, here and after, the convention that $f(x) = 0$ signifies the ability to decrypt. This is the opposite of the standard convention, and is done purely for our convenience in the technical sections.

[2] One can modify the circuit-ABE constructions of [9,20] to support unbounded attributes in the (programmable) random oracle model. Our focus in this paper is on constructions in the standard model.

The second shortcoming of the circuit-ABE constructions based on lattices and LWE is that they are only selectively secure. Selective security means that the attacker needs to decide which challenge attribute to attack before seeing the public parameters of the scheme or any of the keys. In adaptive security (also known as full security), the challenge attribute x^* can be chosen at any point, even depending on the public parameters and decryption keys obtained by the attacker.

While we do know of adaptively secure ABE for *formulas* [25] based on bilinear maps, and for circuits based on multilinear maps [15] and on indistinguishability obfuscation [37], achieving adaptive security in LWE-based constructions seems to require fundamentally new ideas. Recently, Ananth et al. [6] came up with a generic way to go from selective to adaptive security for (collusion-resistant) FE schemes, but their transformation does not work for ABE schemes.

A well known "hack" for getting around the selectiveness issue is to use "complexity leveraging". This technique is based on the observation that an adaptive adversary can be made selective at the cost of a factor 2^ℓ increase in the running time (or loss of $2^{-\ell}$ in the success probability), where ℓ is the maximum attribute length, just by guessing the challenge attribute ahead of time. Therefore, if we start with a selective scheme that is secure against $2^\ell \cdot \text{poly}(\lambda)$ adversaries, then it is also adaptively secure against $\text{poly}(\lambda)$ time adversaries. Since usually $\ell = \text{poly}(\lambda)$, this method leads to a considerable increase in security parameter. More importantly in our situation, if the attribute space is a-priori unbounded, then complexity leveraging cannot work at all.

An intermediate milestone to adaptively secure ABE is the weaker notion of semi-adaptive security, introduced by Chen and Wee [13]. Semi-adaptive security permits an adversary to choose the challenge attributes after it sees the public parameters, but before it sees the answers to any of its secret-key queries. Chen and Wee show a simpler construction of adaptively secure ABE for *formulas.* Note that for unbounded attributes, complexity leveraging is of no use for this notion as well.

> Q2: *Is there an* adaptively (or even semi-adaptively) secure *ABE for general circuits under standard complexity assumptions?*

We resolve the first question and (semi-)resolve the second, as follows.

Theorem 1 (Informal). *Assuming the (polynomial) hardness of approximating worst-case lattice problems to within sub-exponential factor, there is a semi-adaptively secure ABE scheme for circuits of a-priori bounded (polynomial) depth which supports attributes of unbounded length.*

In particular, the setup procedure of our scheme does not require an upper bound on the length of the attributes that will be encrypted. Quite curiously, semi-adaptivity in our result seems to come *for free* from our techniques to achieve unbounded attribute ABE. We elaborate more on our techniques below.

1.1 Overview of Our Techniques

We start with an interpretation of the ABE scheme of Boneh et al. [9] (itself based on the homomorphic encryption scheme of Gentry et al. [18]) which will be instrumental for our presentation.

Given matrices $\mathbf{C}_1, \ldots, \mathbf{C}_\ell$ of appropriate dimension, and a function $f : \{0,1\}^\ell \to \{0,1\}$, represented as a Boolean circuit, one can compute a matrix $\mathbf{C}_f$ which is the "homomorphic evaluation" of f on $\{\mathbf{C}_i\}$. The property of $\mathbf{C}_f$ is that for all $x \in \{0,1\}^\ell$ there exists a low-norm matrix $\mathbf{H} = \mathbf{H}_{\vec{\mathbf{C}},f,x}$ (that is, one with "fairly small" entries, the exact amplitude depends on the depth of f and does not matter for this high level description) for which

$$\mathbf{C}_f - f(x)\mathbf{G} = \left(\underbrace{[\mathbf{C}_1 \| \cdots \| \mathbf{C}_\ell]}_{\text{denote } \vec{\mathbf{C}}} - \underbrace{[x_1\mathbf{G} \| \cdots \| x_\ell \mathbf{G}]}_{\text{denote } x\vec{\mathbf{G}}} \right) \cdot \mathbf{H}.$$

The matrix $\mathbf{G}$ is a special "gadget matrix". This means that if $\mathbf{C}_i = \mathbf{A}\mathbf{R}_i + x_i\mathbf{G}$ for some low-norm matrix $\mathbf{R}_i$, then $\mathbf{C}_f$ can be expressed as $\mathbf{A}\mathbf{R}_f + f(x)\mathbf{G}$ for a somewhat low-norm matrix $\mathbf{R}_f$.

In the ABE scheme of Boneh et al. [9], the public parameters contain a matrix $\mathbf{A}$ and a set $\vec{\mathbf{C}} = (\mathbf{C}_1, \ldots, \mathbf{C}_\ell)$ so that ℓ is the length of supported attributes. The parameters are chosen so that a secret trapdoor can always find a low norm solution $\mathbf{R}$ to any equation of the form $\mathbf{C} = \mathbf{A}\mathbf{R} + y\mathbf{G}$, for all $\mathbf{C}, y$. Encrypting a message to an attribute x is done (at a high level) by considering $[\mathbf{A} \| \vec{\mathbf{C}} - x\vec{\mathbf{G}}]$ as a public key to a dual-Regev encryption scheme [17] and encrypting relative to this key. An important feature of dual-Regev is that it is possible to modify a ciphertext which was encrypted with respect to a certain public key into one that is encrypted with respect to a related key, so long as the new key is obtained by multiplying the old key by a low-norm matrix. Therefore, given some function f, the ciphertext can be converted into one that corresponds to the public key $[\mathbf{A} \| \mathbf{C}_f - f(x)\mathbf{G}]$. Indeed, ABE secret-keys sk_f are generated as dual-Regev keys for the public key $[\mathbf{A} \| \mathbf{C}_f]$, and indeed they can decrypt whenever $f(x) = 0$.[3]

In the proof of security, $\mathbf{A}$ is generated without a trapdoor, but $\mathbf{C}_i$ are generated as $\mathbf{A}\mathbf{R}_i + x_i^*\mathbf{G}$ (which is indistinguishable from their honest distribution). This means that whenever $f(x^*) = 1$, the matrix $[\mathbf{A} \| \mathbf{C}_f]$ equals to $[\mathbf{A} \| \mathbf{A}\mathbf{R}_f + \mathbf{G}]$. It had been shown by [3,29] that if $\mathbf{R}_f$ is known, then dual-Regev keys can be generated *even without a trapdoor*. Finally, the challenge ciphertext is encrypted relative to $[\mathbf{A} \| \vec{\mathbf{C}} - x^*\vec{\mathbf{G}}] = \mathbf{A} \cdot [I \| \vec{\mathbf{R}}]$, which can be shown to be LWE-hard to break if a trapdoor for $\mathbf{A}$ is not known (which indeed it isn't).

The absolutely vital technique that makes the proof of [9] work[4] is the ability to *embed the challenge attributes* into the public parameters. It is apparent from

[3] Note that this "negated policy" formulation is obviously equivalent to the standard formulation in the literature wherein decryption succeeds if $f(x) = 1$. From this point and on, purely for our convenience in the technical sections, we will assume that a ciphertext should be decryptable if $f(x) = 0$ and not decryptable otherwise.

[4] The proofs of the other circuit-ABE schemes from standard assumptions, namely [14,19], follow along similar lines.

this description that the [9] scheme is inherently selectively secure and attribute length bounded. It is important that in the security proof, the values of $\mathbf{C}_i$ are set ahead of time to the right values according to the challenge attributes x^*, making the proof inherently selectively secure. In fact, the entire paradigm of embedding the challenge ciphertext in the public parameters necessitates, for pure information-theoretic reasons, that the public parameters grow with the length of the challenge attribute.

The first thing that we should do if we want to stretch the [9] scheme to support unbounded length attributes, is to find a way to generate an unbounded number of $\mathbf{C}_i$ matrices out of a-priori bounded public parameters. Our first observation is that the scheme already exhibits a similar feature in a different context. Namely, the generation of many $\mathbf{C}_f$ out of a bounded number of $\mathbf{C}_i$. Indeed, in our scheme, the public parameters will contain $\mathbf{A}$ and a sequence of matrices $\vec{\mathbf{B}}$. We will consider a predefined and public sequence of functions ϕ_i, where $i = 1, 2, \ldots$, and let $\mathbf{C}_i$ be the output of homomorphic evaluation of ϕ_i on $\vec{\mathbf{B}}$. Thus, the scheme already allows us to generate exponentially many matrices out of a few.

This allows us to extend the functionality of the scheme to unbounded attribute length, but only syntactically, since the proof does not extend to this setting. In particular, if we try to program the matrices $\vec{\mathbf{B}}$ in the proof similarly to $\vec{\mathbf{C}}$ from previous works, we can set $\mathbf{B}_i = \mathbf{A}\mathbf{R}_i + \sigma_i \mathbf{G}$ for some string σ. If we do so, we will get that $\mathbf{C}_i = \mathbf{A}\mathbf{R}_{\phi_i} + \phi_i(\sigma)\mathbf{G}$, where $\mathbf{R}_{\phi_i}$ is low-norm and can be computed out of the $\mathbf{R}_i$ matrices. On the one hand, this is quite encouraging since it is not too far from what we need, if only there was a way to define ϕ_i and σ so that $\phi_i(\sigma) = x^*_i$ (the ith bit of the challenge attribute) we would be in business. On the other hand, this is of course impossible for mere information theoretic reasons, since the ϕ_i are public functions and σ has bounded length, so they cannot encode an x^* of arbitrary length.

Let us therefore take a step back and think, as an intermediate step, about a restricted security model where x^* is chosen randomly and not adversarially (except its length, which is still under the adversary's control). Indeed, a random x^* cannot be compressed, but in the proof of security we can swap x^* for a *pseudorandom* value that can be easily expressible as the output of a pseudorandom function. In particular, we define $\phi_i(\sigma) = \mathsf{PRF}_\sigma(i)$ for some pseudorandom function family. For a random seed σ, letting $x^*_i = \mathsf{PRF}_\sigma(i)$ will be indistinguishable from a random value, and will allow us to support random unbounded length attributes using the proof methods from above.

Indeed, we managed to hack the framework into producing an arbitrarily long sequence of $\mathbf{C}_i$ in such a way that each $\mathbf{C}_i$ encodes a trapdoor that corresponds to x^*_i. We view this as an interesting contribution by itself. However, we would like to support adversarially chosen attributes, and not just random ones. To do this, we will show how to "program" the challenge attribute into the PRF values *after the fact*. In particular, consider, as a mental experiment, an infinite string Δ which is defined such that $\Delta_i = x^*_i \oplus \mathsf{PRF}_\sigma(i)$. This string is pseudorandom to the adversary, but combining it with the PRF key σ, it contains the information about x^*. What we do in the proof, is generate decryption keys for functions

$f_\Delta(x) = f(x \oplus \Delta)$, instead of for f itself. This needs to be offset by changing the encryption algorithm to encrypt to $x \oplus \Delta$ rather than to x itself (which might seem impossible at this point, however see below). If we are able to offset our ciphertext, then the challenge ciphertext will now be encrypted respective to $x^* \oplus \Delta$ which is just our PRF value. All of this is done without the adversary noticing anything, because Δ just seems to him as a completely random string that does not depend on x^*.

We are left with two problems. The first and easier one is that Δ needs to be publicly known, but it has unlimited size and in the proof, we need to know x^* in order to generate it. This is easily managed by noticing that only the ℓ-prefix of Δ is needed in order to use a secret key for a function with ℓ-bit input. We will therefore append the appropriate prefix of Δ to any key that we release. This means that we only need to know the value of Δ when we answer key queries and not when we generate the public parameters. This very fact allows us to achieve *semi-adaptive security*, where x^* can be specified after the setup phase but before key generation. We note that of course setting Δ respective to x^* is only done in the proof. In the real scheme Δ is a random (or pseudorandom) string that is maintained by the key authority and whose prefixes are released as needed (it is important that the same Δ is used for all keys). A savvy reader would have noticed that this "delayed" definition of Δ is similar to non-committing proof techniques which, looking back, is not too surprising. It is also not hard to observe why this technique stops at *semi*-adaptive security: we managed to postpone defining Δ to the time when we generate the first secret key. Since Δ depends on x^* in the proof, we are restricted to the semi-adaptive world where all secret-key queries come after the challenge attributes have been declared.

The second and harder problem is how to encrypt in this brave new scheme. The encryption attribute needs to offset for the effect of Δ on the key, but Δ itself is not (and must not be) a part of the public parameters and is thus unknown to the encryption algorithm. This problem is solved by showing that we can encrypt for *all* possible values of Δ at the same time. Recall that in the encryption, we consider the matrices $\mathbf{C}_i - x_i\mathbf{G}$, for all i. In fact, the encryption process generates a piece of the ciphertext out of each of these matrices, and the collection of pieces constitutes the entire ciphertext. In order to allow for any possible value of Δ, we will generate a ciphertext piece $c_{i,0}$ for $\mathbf{C}_i - x_i\mathbf{G}$ (accounting for $\Delta_i = 0$) and a piece $c_{i,1}$ for $\mathbf{C}_i - (x_i \oplus 1)\mathbf{G}$ (accounting for $\Delta_i = 1$). This would allow us to take the relevant pieces and use them in the decryption process. Alas, the security of the [9] scheme shatters completely if the adversary is allowed to see encryption pieces relative to both $\mathbf{C}_i$ and $\mathbf{C}_i - \mathbf{G}$. It appears that we fixed functionality at the expense of security.

Our last technical contribution is to solve this problem by using ... attribute based encryption! (in fact, even identity based encryption would suffice, but with slightly worse parameters). As a part of our public parameters, we include parameters for a "small" ABE scheme that only needs to support bounded short attributes and low depth circuits. We will encrypt the ciphertext piece $c_{i,b}$ with respect to attribute (i, b) using the "small" scheme. Then, as a part of the functional key, we will also produce a "small" key that will allow to decrypt only attributes (i, b) for which $b = \Delta_i$. This means that an adversary can only see those ciphertext pieces that are needed for decryption. Furthermore, since the offset Δ is fixed, the adversary will only ever see $c_{i,0}$ or $c_{i,1}$ but not both, thus keeping security in tact. This completes the description of our scheme.

2 Preliminaries

2.1 Bounded Distributions and Swallowing

As in many previous works based on LWE, we will rely heavily on distributions that are supported over a bounded domain (with high probability). We will also rely on the fact that some distributions (e.g. sufficiently wide Gaussians) remain almost unchanged under small shifts. Formal definitions follow.

Definition 1. *A distribution χ supported over $\mathbb{Z}$ is (B, ϵ)-bounded if $\Pr_{x \overset{\$}{\leftarrow} \chi}[|x| > B] < \epsilon$.*

Definition 2. *A distribution $\tilde{\chi}$ supported over $\mathbb{Z}$ is (B, ϵ)-swallowing if for all $y \in [-B, B] \cap \mathbb{Z}$ it holds that $\tilde{\chi}$ and $y + \tilde{\chi}$ are within ϵ statistical distance.*

The following is a straightforward application of the properties of rounded/discrete Gaussians.

Fact 1. *For every B, ϵ, δ there exists an efficiently sampleable distribution that is both (B, ϵ)-swallowing and $(B \cdot \sqrt{\log(1/\delta)}/\epsilon, O(\delta))$-bounded.*

Finally, we will define the notion of a distribution that is swallowing with respect to another.

Definition 3. *A distribution $\tilde{\chi}$ supported over $\mathbb{Z}$ is (χ, ϵ)-swallowing, for a distribution χ, if it holds that $\tilde{\chi}$ and $\chi + \tilde{\chi}$ are within ϵ statistical distance. We omit the ϵ when it indicates a negligible function in a security parameter that is clear from the context.*

The following corollary summarizes the swallowing properties required for our scheme.

Corollary 1. *Let $B(\lambda)$ be some function and let $\tilde{B}(\lambda) = B(\lambda) \cdot \lambda^{\omega(1)}$, then there exists an efficiently sampleable ensemble $\{\tilde{\chi}_\lambda\}_\lambda$ such that $\tilde{\chi}$ is χ-swallowing for any $B(\lambda)$-bounded $\{\chi_\lambda\}_\lambda$, and also $\tilde{B}(\lambda)$-bounded.*

2.2 Pseudorandom Functions

A pseudorandom function family is a pair of PPT algorithms $\mathsf{PRF} = (\mathsf{PRF.Gen}, \mathsf{PRF.Eval})$, such that the key generation $\mathsf{PRF.Gen}(1^\lambda)$ takes as input the security parameter, and outputs a seed $\sigma \in \{0,1\}^\eta$ (where $\eta = \eta_\lambda$ is the key length). The evaluation algorithm $\mathsf{PRF.Eval}(\sigma, x)$ takes a seed $\sigma \in \{0,1\}^\eta$ and in input $x \in \{0,1\}^*$ and returns a bit $y \in \{0,1\}$.

Definition 4. *A family* PRF *as above is secure if for every polynomial time adversary* $\mathcal{A}$ *it holds that*

$$\left|\Pr[\mathcal{A}^{\mathsf{PRF.Eval}(\sigma,\cdot)}(1^\lambda) = 1] - \Pr[\mathcal{A}^{\mathcal{O}(\cdot)}(1^\lambda) = 1]\right| = \operatorname{negl}(\lambda),$$

where $\sigma = \mathsf{PRF.Gen}(1^\lambda)$ *and* $\mathcal{O}$ *is a random oracle. The probabilities are taken over all of the randomness of the experiment.*

2.3 KP-ABE with Unbounded Attribute Length

Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_\lambda$ be an ensemble of function classes such that $\mathcal{F}_\lambda \subseteq \{0,1\}^* \to \{0,1\}$. We assume that the functions are represented as boolean circuits. A key-policy attribute based encryption (KP-ABE) scheme is defined by a tuple of PPT algorithms $\mathsf{ABE} = (\mathsf{ABE.Params}, \mathsf{ABE.Enc}, \mathsf{ABE.Keygen}, \mathsf{ABE.Dec})$ such that:

- The setup algorithm $\mathsf{ABE.Params}(1^\lambda)$ takes the security parameter as input and outputs a master secret key msk and a set of public parameters pp.
- The encryption algorithm $\mathsf{ABE.Enc}_{\mathsf{pp}}(\mu, x)$ uses the public parameters pp and takes as input a message μ from a message space $\mathcal{M} = \mathcal{M}_\lambda$ and an attribute $x \in \{0,1\}^*$. It outputs a ciphertext $\mathsf{ct} \in \{0,1\}^*$.
- The key generation algorithm $\mathsf{ABE.Keygen}_{\mathsf{msk}}(f)$ uses the master secret key msk and takes as input a function $f \in \mathcal{F}_\lambda$. It outputs a secret key sk_f.
- The decryption algorithm $\mathsf{ABE.Dec}_{\mathsf{pp}}(\mathsf{sk}_f, x, \mathsf{ct})$ takes as input a function secret key sk_f, an attribute $x \in \{0,1\}^*$ and a ciphertext ct, and outputs a message $\mu' \in \mathcal{M}$.

Definition 5 (Correctness of KP-ABE). *A scheme* ABE *is* correct *if the following holds. Consider a sequence of functions* $\{f_\lambda \in \mathcal{F}_\lambda\}_\lambda$ *and a sequence of attributes* $\{x_\lambda \in \{0,1\}^*\}_\lambda$*, such that for all* λ*, the input size of* f *is* exactly $|x_\lambda|$ *and* $f_\lambda(x_\lambda) = 0$.[5] *For all such sequences and for any sequence* $\{m_\lambda \in \mathcal{M}_\lambda\}_\lambda$*, it holds that*

$$\Pr[\mathsf{ABE.Dec}_{\mathsf{pp}}(\mathsf{sk}_f, x, \mathsf{ct}) \neq \mu] = \operatorname{negl}(\lambda),$$

where $(\mathsf{msk}, \mathsf{pp}) = \mathsf{ABE.Params}(1^\lambda)$*,* $\mathsf{ct} = \mathsf{ABE.Enc}_{\mathsf{pp}}(\mu, x)$*,* $\mathsf{sk}_f = \mathsf{ABE.Keygen}_{\mathsf{msk}}(f)$*.*

Definition 6 (Security for KP-ABE). *Let* ABE *be a KP-ABE encryption scheme as above, and consider the following game between the challenger and adversary.*

1. *The challenger generates* $(\mathsf{msk}, \mathsf{pp}) = \mathsf{ABE.Params}(1^\lambda)$*, and sends* pp *to the adversary.*
2. *The adversary makes arbitrarily many key queries by sending functions* f_i *(represented as circuits) to the challenger. Upon receiving such function, the challenger creates* $\mathsf{sk}_i = \mathsf{ABE.Keygen}_{\mathsf{msk}}(f_i)$ *and sends* sk_i *to the adversary.*

[5] Recall our convention that $f(x) = 0$ is the event when decryption succeeds.

3. *The adversary sends an attribute x^* and a pair of messages m_0, m_1 to the challenger. The challenger samples $b \in \{0,1\}$ and computes the challenge ciphertext $\mathsf{ct}^* = \mathsf{ABE.Enc}_{\mathsf{pp}}(m_b, x^*)$. It sends ct^* to the adversary.*
4. *The adversary makes arbitrarily many key queries as in Step 2 above.*
5. *The adversary outputs $\tilde{b} \in \{0,1\}$.*
6. *Let legal denote the event where all key queries of the adversary are such that $f_i(x^*) = 1$. If legal, the output of the game is $b' = \tilde{b}$, otherwise the output b' is a uniformly random bit.*

The advantage *of an adversary $\mathcal{A}$ is $|\Pr[b' = b] - 1/2|$, where b, b' are generated in the game played between the challenger and the adversary $\mathcal{A}(1^\lambda)$. If x^* is too short or too long compared to the prescribed input size of f_i then it is truncated or padded with zeros appropriately (see discussion below).*

The game above is called the adaptive security game *for ABE, and it has relaxed variants. In the* selective security game*, the adversary sends x^* before Step 1. In the* semi-adaptive security game*, the adversary sends x^* before Step 2.*

The scheme ABE *is* adaptively/selectively/semi-adaptively secure *if any* PPT *adversary $\mathcal{A}$ only has negligible advantage in the adaptive/selective/semi-adaptive security game (respectively).*

Negated Policies. We allow decryption when $f(x) = 0$ and require that in the security game all queries are such that $f(x^*) = 1$. In LWE-based constructions it is often much more convenient to work with this negated version of the policy, which explains the apparent strangeness. This variant is obviously equivalent.

Discussion. Our definition does not place any restrictions on the attribute length so the only restriction comes from limiting the adversary to run in polynomial time (so it can only output x^* and f_i that are polynomially bounded). It is important to notice that in this regime, there are no known generic transformations from selective to semi-adaptive to adaptive security, even if we strengthen the hardness assumption. In particular, the complexity leveraging technique, in which the challenger "guesses" x^* in the beginning of the experiment, and a sub-exponential hardness assumption is made to account for the success probability of this guess, is no longer applicable. In this light, we view our semi-adaptive security improvement as *qualitative rather than quantitative.*

Lastly, we note that in the security definition (but not in the correctness requirement!) we chose to allow $f(x^*)$ to be well defined even if there is a mismatch between the input length of f and the length of x^* (by truncating x^* or padding with zeros). A different valid approach would be to consider an alternate, stronger, definition that if there is a mismatch then $f(x^*) = 1$ (and thus it is legal for the adversary to query any function that does not have the same input length as $|x^*|$). We notice that this notion of security is derived from ours by adding the length itself to the attribute. More explicitly, when you want to encrypt with attribute x of length ℓ, use the ABE scheme with attribute (ℓ, x), and in the key generation process, when you want to generate a key for function f, generate a key for $f'(\ell, x)$ that first checks that ℓ is indeed the intended input length. Therefore, using our definition does not limit generality in this aspect.

3 LWE, Trapdoors, Homomorphism

This section summarizes tools from previous works that are used in our construction. This includes the definition of the LWE problem and its relation to worst case lattice problems, the notion of trapdoors for lattices and operations on trapdoors, and homomorphic evaluation of matrices with special properties.

Learning with Errors (LWE). The Learning with Errors (LWE) problem was introduced by Regev [33] as a generalization of "learning parity with noise" [5,8]. We now define the decisional version of LWE. (Unless otherwise stated, we will treat all vectors as column vectors in this paper).

Definition 7 (Decisional LWE (DLWE) [33]**).** *Let λ be the security parameter, $n = n(\lambda)$, $m = m(\lambda)$, and $q = q(\lambda)$ be integers and $\chi = \chi(\lambda)$ be a probability distribution over $\mathbb{Z}$. The* $\mathrm{DLWE}_{n,q,\chi}$ *problem states that for all $m = \mathrm{poly}(n)$, letting $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \chi^m$, and $\mathbf{u} \leftarrow \mathbb{Z}_q^m$, the following distributions are computationally indistinguishable:*

$$\left(\mathbf{A}, \mathbf{s}^T\mathbf{A} + \mathbf{e}^T\right) \overset{c}{\approx} \left(\mathbf{A}, \mathbf{u}^T\right)$$

There are known quantum (Regev [33]) and classical (Peikert [32]) reductions between $\mathrm{DLWE}_{n,q,\chi}$ and approximating short vector problems in lattices. Specifically, these reductions take χ to be a discrete Gaussian distribution $D_{\mathbb{Z},\alpha q}$ for some $\alpha < 1$. We write $\mathrm{DLWE}_{n,q,\alpha}$ to indicate this instantiation. We now state a corollary of the results of [28,29,32,33]. These results also extend to additional forms of q (see [28,29]).

Corollary 2 [28,29,32,33]**.** *Let $q = q(n) \in \mathbb{N}$ be either a prime power $q = p^r$, or a product of co-prime numbers $q = \prod q_i$ such that for all i, $q_i = \mathrm{poly}(n)$, and let $\alpha \geq \sqrt{n}/q$. If there is an efficient algorithm that solves the (average-case)* $\mathrm{DLWE}_{n,q,\alpha}$ *problem, then:*

- *There is an efficient quantum algorithm that solves* $\mathsf{GapSVP}_{\tilde{O}(n/\alpha)}$ *(and* $\mathsf{SIVP}_{\tilde{O}(n/\alpha)}$*) on any n-dimensional lattice.*
- *If in addition $q \geq \tilde{O}(2^{n/2})$, there is an efficient classical algorithm for* $\mathsf{GapSVP}_{\tilde{O}(n/\alpha)}$ *on any n-dimensional lattice.*

Recall that GapSVP_γ is the (promise) problem of distinguishing, given a basis for a lattice and a parameter d, between the case where the lattice has a vector shorter than d, and the case where the lattice doesn't have any vector shorter than $\gamma \cdot d$. SIVP is the search problem of finding a set of "short" vectors. The best known algorithms for GapSVP_γ [35] require at least $2^{\tilde{\Omega}(n/\log\gamma)}$ time. We refer the reader to [32,33] for more information.

In this work, we will only consider the case where $q \leq 2^n$. Furthermore, the underlying security parameter λ is assumed to be polynomially related to the dimension n.

Lastly, we derive the following corollary which will allow us to choose the LWE parameters for our scheme. The corollary follows immediately from the fact that the discrete Gaussian $D_{\mathbb{Z},\alpha q}$ is $(\alpha q \cdot t, 2^{-\Omega(t^2)})$-bounded for all t.

Corollary 3. *For all $\epsilon > 0$ there exist functions $q = q(n) \leq 2^n, \chi = \chi(n)$ such that χ is B-bounded for some $B = B(n)$, $q/B \geq 2^{n^\epsilon}$ and such that $\mathrm{DLWE}_{n,q,\chi}$ is at least as hard as the classical hardness of GapSVP_γ and the quantum hardness of SIVP_γ for $\gamma = 2^{\Omega(n^\epsilon)}$.*

The Gadget Matrix. Let $N = n \cdot \lceil \log q \rceil$ and define the "gadget matrix" $\mathbf{G} = \mathbf{g} \otimes \mathbf{I}_n \in \mathbb{Z}_q^{n \times N}$ where $\mathbf{g} = (1, 2, 4, \ldots, 2^{\lceil \log q \rceil - 1}) \in \mathbb{Z}_q^{\lceil \log q \rceil}$. We will also refer to this gadget matrix as the "powers-of-two" matrix. We define the inverse function $\mathbf{G}^{-1} : \mathbb{Z}_q^{n \times m} \to \{0,1\}^{N \times m}$ which expands each entry $a \in \mathbb{Z}_q$ of the input matrix into a column of size $\lceil \log q \rceil$ consisting of the bits of the binary representation of a. We have the property that for any matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, it holds that $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{A}) = \mathbf{A}$.

Trapdoors. Let $n, m, q \in \mathbb{N}$ and consider a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. For all $\mathbf{V} \in \mathbb{Z}_q^{n \times m'}$, we let $\mathbf{A}_\tau^{-1}(\mathbf{V})$ denote the random variable whose distribution is a Gaussian $D_{\mathbb{Z}^m,\tau}^{m'}$ conditioned on $\mathbf{A} \cdot \mathbf{A}_\tau^{-1}(\mathbf{V}) = \mathbf{V}$. A τ-trapdoor for $\mathbf{A}$ is a procedure that can sample from the distribution $\mathbf{A}_\tau^{-1}(\mathbf{V})$ in time $\mathrm{poly}(n, m, m', \log q)$, for any $\mathbf{V}$. We slightly overload notation and denote a τ-trapdoor for $\mathbf{A}$ by $\mathbf{A}_\tau^{-1}$.

The following properties had been established in a long sequence of works.

Corollary 4 (Properties of Trapdoors [2–4,12,17,29]**).** *Lattice trapdoors exhibit the following properties.*

1. *Given $\mathbf{A}_\tau^{-1}$, one can obtain $\mathbf{A}_{\tau'}^{-1}$ for any $\tau' \geq \tau$.*
2. *Given $\mathbf{A}_\tau^{-1}$, one can obtain $[\mathbf{A} \| \mathbf{B}]_\tau^{-1}$ and $[\mathbf{B} \| \mathbf{A}]_\tau^{-1}$ for any $\mathbf{B}$.*
3. *For all $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{R} \in \mathbb{Z}^{m \times N}$, with $N = n \lceil \log q \rceil$, one can obtain $[\mathbf{AR} + \mathbf{G} \| \mathbf{A}]_\tau^{-1}$ for $\tau = O(m \cdot \|\mathbf{R}\|_\infty)$.*
4. *There exists an efficient procedure $\mathsf{TrapGen}(1^n, q)$ that outputs $(\mathbf{A}, \mathbf{A}_{\tau_0}^{-1})$ where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ for some $m = O(n \log q)$ and is 2^{-n}-uniform, where $\tau_0 = O(\sqrt{n \log q \log n})$.*

Homomorphic Evaluation. Consider some $n, q \in \mathbb{N}$. Consider $\mathbf{C}_1, \ldots, \mathbf{C}_\ell \in \mathbb{Z}_q^{n \times N}$ where $N = n \lceil \log q \rceil$, and denote $\vec{\mathbf{C}} = [\mathbf{C}_1 \| \cdots \| \mathbf{C}_\ell]$. Let f be a boolean circuit of depth d computing a function $\{0,1\}^\ell \to \{0,1\}$, and assume that f contains only NAND gates. We define $\mathbf{C}_f = \mathsf{Eval}(f, \vec{\mathbf{C}})$ recursively: associate $\mathbf{C}_1, \ldots, \mathbf{C}_\ell$ with the input wires of the circuit. For every wire w in f, letting u, v be its predecessors and define $\mathbf{C}_w = \mathbf{G} - \mathbf{C}_u \cdot \mathbf{G}^{-1}(\mathbf{C}_v)$. Finally $\mathbf{C}_f$ is the matrix associated with the output wire.

Denoting $x\vec{\mathbf{G}} = [x_1\mathbf{G}\|\cdots\|x_\ell\mathbf{G}]$, it holds that if $\mathbf{C}_f = \mathsf{Eval}(f, \vec{\mathbf{C}})$, then $\mathbf{C}_f - f(x)\mathbf{G} = (\vec{\mathbf{C}} - x\vec{\mathbf{G}}) \cdot \mathbf{H}_{f,x,\vec{\mathbf{C}}}$, for a matrix $\mathbf{H}_{f,x,\vec{\mathbf{C}}}$ with $\left\|\mathbf{H}_{f,x,\vec{\mathbf{C}}}\right\|_\infty \leq (N+1)^d$. In particular, if $\mathbf{C}_i = \mathbf{A}\mathbf{R}_i + x_i\mathbf{G}$, i.e. $\vec{\mathbf{C}} = \mathbf{A}\vec{\mathbf{R}} + x\vec{\mathbf{G}}$ for $\vec{\mathbf{R}} = [\mathbf{R}_1\|\cdots\|\mathbf{R}_\ell]$, then $\mathbf{C}_f = \mathbf{A}\mathbf{R}_f + f(x)\mathbf{G}$ for $\mathbf{R}_f = \vec{\mathbf{R}} \cdot \mathbf{H}_{f,x,\vec{\mathbf{C}}}$ (where $\mathbf{H}$ is independent of $\vec{\mathbf{R}}$).

4 Our Scheme

We now present our scheme and prove its correctness and security. As in previous works on LWE-based ABE schemes [9,19], it would be easier for us to work with "negated policies", so that sk_f can decrypt ciphertexts with attribute x if $f(x) = 0$. We start by defining the class of depth bounded circuits, to which our construction is targeted.

Definition 8 (Depth-bounded Circuits). *The class of d-bounded circuits, denoted $\mathcal{P}_d$, for some function $d = d(\lambda)$ is the ensemble of functions $\{\mathcal{P}_{d,\lambda}\}_\lambda$ such that $\mathcal{P}_{d,\lambda}$ is the set of boolean circuits of depth at most $d(\lambda)$ and input length at most 2^ν for some $\nu(\lambda) = \omega(\log\lambda)$ which will be clear from the context.*

Next, we define another class of circuits. These are very simple circuits that contain a hardcoded string, and upon receiving an index and bit as input, they check whether the relevant location in the string is indeed the supplied value.

Definition 9. *Consider the family of circuits $\{\mathsf{BitCheck}_{\nu,x}\}$ s.t. for all $\nu \in \mathbb{N}$ and $x \in \{0,1\}^*$, $|x| \leq 2^\nu$, we define $\mathsf{BitCheck}_{\nu,x} : [2^\nu] \times [2^\nu] \times \{0,1\} \to \{0,1\}$ such that $\mathsf{BitCheck}_{\nu,x}(\ell, i, b) = 0$ if and only if $|x| = \ell$ and also $x_i = b$. Note that $\mathsf{BitCheck}_{\nu,x}$ can always be computed by a boolean circuit of depth $O(\log|x|) = O(\nu)$ (we assume that ℓ, i are in standard ν-bit binary representation).*

The Scheme. Let $\nu = \nu(\lambda)$ be any super-logarithmic function (so that 2^ν is super-polynomial). Let $\mathsf{oldABE} = (\mathsf{oldABE.Params}, \mathsf{oldABE.Enc}, \mathsf{oldABE.Keygen}, \mathsf{oldABE.Dec})$ be a selectively-secure key-policy ABE scheme for the function class $\{\{\mathsf{BitCheck}_{\nu(\lambda),x} : |x| \leq 2^\nu\}\}_\lambda$ where ν is as above (i.e. oldABE only need to support bounded length attributes, and furthermore this length can be any super-logarithmic function). Let PRF be a family of pseudorandom functions and let $\eta = \eta_\lambda$ be the seed length (for security parameter λ). Let d_{prf} be the depth of $\mathsf{PRF.Eval}(\sigma, x)$ for $|x| = \nu$ (by definition $d_{\mathrm{prf}} = \mathrm{poly}(\lambda)$).

We now present our ABE scheme for any class of circuits of a-priori polynomial depth bound. We note that as in previous works, we submit the depth bound as an additional parameter to the setup procedure. In order to support the class $\mathcal{P}_d$, the setup procedure is to be executed on input $(1^\lambda, 1^{d(\lambda)})$. Finally, the scheme is parameterized by a constant $\epsilon \in (0,1)$ that determines the tradeoff between the lattice approximation factor on which security is based, and the efficiency of the scheme.

- $\mathsf{ABE.Params}(1^\lambda, 1^d)$. We start by setting DLWE parameters based on Corollary 3. Let n be s.t. $(n^2+1)^{2(d_{\mathrm{prf}}+d)} \cdot 2^{3\nu} \le 2^{n^\epsilon}$. The solution to the equation is of the form $n \le (\lambda d)^{O(1/\epsilon)}$, which is polynomial in the security parameter for any constant ϵ. We choose q, χ, B accordingly based on Corollary 3, and note that by definition $q/B \ge (N+1)^{2(d_{\mathrm{prf}}+d)} \cdot 2^{3\nu}$ (recall that $N = n\lceil \log q \rceil$). We further let $\tilde{\chi}$ be a B'-swallowing and $\tilde{B}$-bounded distribution, for $B' = B \cdot m\eta N(N+1)^{d_{\mathrm{prf}}}$ and $\tilde{B} = 2^\nu \cdot B'$, whose existence is guaranteed by Corollary 1.

  Generate a matrix-trapdoor pair $(\mathbf{A}, \mathbf{A}_{\tau_0}^{-1}) = \mathsf{TrapGen}(1^n, q)$ (see Corollary 4), vector $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^n$, and matrices $\mathbf{B}_1, \ldots, \mathbf{B}_\eta \xleftarrow{\$} \mathbb{Z}_q^{n \times N}$, and denote $\vec{\mathbf{B}} = [\mathbf{B}_1 \| \ldots \| \mathbf{B}_\eta]$. We assume w.l.o.g that $m \ge n\lceil \log q \rceil + 2\lambda$ (otherwise random padding can be applied). Generate a key pair for oldABE: $(\mathsf{oldabemsk}, \mathsf{oldabepp}) = \mathsf{oldABE.Params}(1^\lambda)$. Generate a seed for a PRF $\sigma = \mathsf{PRF.Gen}(1^\lambda)$.

  We set $\mathsf{msk} = (\mathbf{A}_\tau^{-1}, \mathsf{oldabemsk}, \sigma)$ and $\mathsf{pp} = (\mathbf{A}, \vec{\mathbf{B}}, \mathsf{oldabepp})$.
- $\mathsf{ABE.Enc}_{\mathsf{pp}}(\mu, x)$, where $\mathsf{pp} = (\mathbf{A}, \vec{\mathbf{B}}, \mathsf{oldabepp})$, $\mu \in \{0,1\}$ and $x \in \{0,1\}^*$. We let $\ell = |x|$ denote the length of the attribute string. For all $i \in [\ell]$, generate $\mathbf{C}_i = \mathsf{Eval}(\mathsf{PRF.Eval}(\cdot, i), \vec{\mathbf{B}})$. (Where $\mathsf{PRF.Eval}(\cdot, i)$ is the circuit that takes a seed σ and outputs $\mathsf{PRF.Eval}(\sigma, i)$.)

  Sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \xleftarrow{\$} \chi^m$, $e' \xleftarrow{\$} \chi$, let

$$\mathbf{c}_0^T = \mathbf{s}^T[\mathbf{A} \| \mathbf{v}] + [\mathbf{e}^T \| e'] + \mu \lfloor q/2 \rceil \cdot [\mathbf{0}^T \| 1].$$

  This is essentially a dual-Regev encryption of μ under public key $\mathbf{A}, \mathbf{v}$. The rest of the ciphertext will contain auxiliary information that will allow to decrypt given a proper functional secret key. Specifically, we sample for all $i \in [\ell]$ a noise vector $\tilde{\mathbf{e}}_i \xleftarrow{\$} \tilde{\chi}^N$, and compute

$$\mathbf{c}_{i, x_i \oplus \beta}^T = \mathbf{s}^T(\mathbf{C}_i - (x_i \oplus \beta)\mathbf{G}) + \tilde{\mathbf{e}}_i^T, \tag{1}$$

  Finally, the vectors $\mathbf{c}_{i, x_i \oplus \beta}$ are encrypted again using the old ABE scheme:

$$\psi_{i,\beta} = \mathsf{oldABE.Enc}_{\mathsf{oldabepp}}(\mathbf{c}_{i, x_i \oplus \beta}, (\ell, i, \beta)).$$

  The final ciphertext is

$$\mathsf{ct} = \left(\mathbf{c}_0, (\psi_{i,\beta})_{i \in [\ell], \beta \in \{0,1\}}\right).$$

- $\mathsf{ABE.Keygen}_{\mathsf{msk}}(f)$. Given a circuit f computing a function $\{0,1\}^\ell \to \{0,1\}$, the key is generated as follows. We recall that we work with negated policies so sk_f should decrypt only when $f(x) = 0$.

  For all i, define $\varDelta_i = \mathsf{PRF.Eval}(\sigma, i)$. Further let $\varDelta_{\le \ell} = \varDelta_1 \cdots \varDelta_\ell$ be the ℓ-prefix of the infinite string $\varDelta$ (in fact, we can think of $\varDelta$ as having length 2^ν, which is finite but super-polynomial).

  Generate a key for the old scheme $\mathsf{oldabesk}_\ell = \mathsf{oldABE.Keygen}_{\mathsf{oldabemsk}}$ $(\mathsf{BitCheck}_{\nu, \varDelta_{\le \ell}})$.

Note that $\Delta_{\leq \ell}$ and $\mathsf{oldabesk}_\ell$ depend only on msk and ℓ, and not on f, and therefore they can be generated and published once and for all for each value of ℓ (however, since ℓ is a-priori unbounded, it is impossible to publish this information for "all possible ℓ" at the same time). Define $f_\Delta : \{0,1\}^\ell \rightarrow \{0,1\}$ as $f_\Delta(x) = f(x \oplus \Delta_{\leq \ell})$.

For all $i \in [\ell]$, generate $\mathbf{C}_i = \mathsf{Eval}(\mathsf{PRF.Eval}(\cdot, i), \vec{\mathbf{B}})$ (as in the encryption algorithm). Let $\vec{\mathbf{C}} = [\mathbf{C}_1 \| \cdots \| \mathbf{C}_\ell]$ and set $\mathbf{C}_f = \mathsf{Eval}(f_\Delta, \vec{\mathbf{C}})$. Let

$$\mathbf{r}_f = [\mathbf{C}_f \| \mathbf{A}]_\tau^{-1}(\mathbf{v}),$$

where $\tau = 2^\nu \cdot mN^2(N+1)^{d+d_{\mathrm{prf}}} \geq \tau_0$ and $\mathbf{t}_f = [-\mathbf{r}_f^T \| 1]^T$. Note that $[\mathbf{C}_f \| \mathbf{A} \| \mathbf{v}] \cdot \mathbf{t}_f = 0$.

Output $\mathsf{sk}_f = (f, \Delta_{\leq \ell}, \mathsf{oldabesk}_\ell, \mathbf{t}_f)$.

– $\mathsf{ABE.Dec}(\mathsf{sk}_f, x, \mathsf{ct})$. Given $\mathsf{sk}_f = (f, \Delta_{\leq \ell}, \mathsf{oldabesk}_\ell, \mathbf{t}_f)$, $x \in \{0,1\}^\ell$ such that $f(x) = 0$, and $\mathsf{ct} = \left(\mathbf{c}_0, (\psi_{i,\beta})_{i \in [\ell], \beta \in \{0,1\}}\right)$, the decryption process runs as follows.

Use $\mathsf{oldabesk}_\ell$ to compute

$$\mathbf{c}_{i, x_i \oplus \Delta_i} = \mathsf{oldABE.Dec}(\mathsf{oldabesk}_\ell, \psi_{i,\Delta_i}, (\ell, i, \Delta_i)), \tag{2}$$

and recompose

$$\mathbf{c}^T_{x \oplus \Delta_{\leq \ell}} = [\mathbf{c}^T_{1, x_1 \oplus \Delta_1} \| \cdots \| \mathbf{c}^T_{\ell, x_\ell \oplus \Delta_\ell}].$$

We again compute $\mathbf{C}_i = \mathsf{Eval}(\mathsf{PRF.Eval}(\cdot, i), \vec{\mathbf{B}})$, $\vec{\mathbf{C}} = [\mathbf{C}_1 \| \cdots \| \mathbf{C}_\ell]$ and $\mathbf{C}_f = \mathsf{Eval}(f_\Delta, \vec{\mathbf{C}})$. We also compute $\mathbf{H} = \mathbf{H}_{f_\Delta, x \oplus \Delta_{\leq \ell}, \vec{\mathbf{C}}}$. Note that by the properties stated above, it holds that

$$(\vec{\mathbf{C}} - (x \oplus \Delta_{\leq \ell})\vec{\mathbf{G}}) \cdot \mathbf{H} = \mathbf{C}_f - f_\Delta(x \oplus \Delta_{\leq \ell})\mathbf{G} = \mathbf{C}_f,$$

since $f_\Delta(x \oplus \Delta_{\leq \ell}) = f(x) = 0$.

Recalling that $\mathbf{c}^T_{x \oplus \Delta_{\leq \ell}}$ is linear (up to noise) in $\vec{\mathbf{C}} - (x \oplus \Delta_{\leq \ell})\vec{\mathbf{G}}$, we will set $\mathbf{c}_f^T = \mathbf{c}^T_{x \oplus \Delta_{\leq \ell}} \cdot \mathbf{H}_{f_\Delta, x \oplus \Delta_{\leq \ell}, \vec{\mathbf{C}}}$, with intent to show that $\mathbf{c}_f^T$ is linear (up to noise) in $\mathbf{C}_f$.

Finally, we compute $\tilde{\mu} = [\mathbf{c}_f^T \| \mathbf{c}_0^T] \cdot \mathbf{t}_f$, and output $\mu' = 0$ if $|\tilde{\mu}| < q/4$ and $\mu' = 1$ if $|\tilde{\mu}| \geq q/4$.

4.1 Correctness

Let $\{(f_\lambda, x_\lambda)\}_\lambda$ be an arbitrary sequence of function-message pairs s.t. f_λ has size $\mathsf{poly}(\lambda)$, depth at most $d(\lambda)$, and $|x| = \ell(\lambda)$ for some $\ell(\lambda) \leq 2^{\nu(\lambda)}$. Consider properly generated $(\mathsf{pp}, \mathsf{msk}) = \mathsf{ABE.Params}(1^\lambda)$, a properly encrypted ciphertext $\mathsf{ct} = \mathsf{Enc}_{\mathsf{pp}}(\mu, x)$ for some value $\mu \in \{0,1\}$ and a properly generated functional key $\mathsf{sk}_f = \mathsf{ABE.Keygen}_{\mathsf{msk}}(f)$.

Consider the execution of $\mathsf{ABE.Dec}(\mathsf{sk}_f, x, \mathsf{ct})$. The correctness of oldABE implies that with all but negligible probability, the vectors $\mathbf{c}_{i, x_i \oplus \Delta_i}$ computed in Eq. (2) are indeed equal to the ones encrypted in Eq. (1). Namely, that

$$\mathbf{c}^T_{i, x_i \oplus \Delta_i} = \mathbf{s}^T(\mathbf{C}_i - (x_i \oplus \Delta_i)\mathbf{G}) + \mathbf{e}^T \mathbf{R}_i,$$

and therefore

$$\mathbf{c}^T_{x \oplus \Delta_{\leq \ell}} = \mathbf{s}^T(\vec{\mathbf{C}} - (x \oplus \Delta_{\leq \ell})\vec{\mathbf{G}}) + \tilde{\mathbf{e}}^T,$$

which, recalling that $f(x) = 0$ and denoting $\mathbf{H} = \mathbf{H}_{f_\Delta, x \oplus \Delta_{\leq \ell}, \vec{\mathbf{C}}}$, implies that

$$\mathbf{c}^T_f = \mathbf{c}^T_{x \oplus \Delta_{\leq \ell}} \cdot \mathbf{H} = \mathbf{s}^T \mathbf{C}_f + \tilde{\mathbf{e}}^T \mathbf{H}.$$

Finally, we get that

$$[\mathbf{c}^T_f \| \mathbf{c}^T_0] = \mathbf{s}^T [\mathbf{C}_f \| \mathbf{A} \| \mathbf{v}] + \left[\tilde{\mathbf{e}}^T \mathbf{H} \| \mathbf{e}^T \| e'\right] + \mu \lfloor q/2 \rceil \cdot [\mathbf{0}^T \| 1],$$

and therefore that

$$[\mathbf{c}^T_f \| \mathbf{c}^T_0] \cdot \mathbf{t}_f = \left[\tilde{\mathbf{e}}^T \mathbf{H} \| \mathbf{e}^T \| e'\right] \cdot \mathbf{t}_f + \mu \lfloor q/2 \rceil .$$

We conclude that we have correct decryption so long as $\left| \left[\tilde{\mathbf{e}}^T \mathbf{H} \| \mathbf{e}^T \| e'\right] \cdot \mathbf{t}_f \right|$ is bounded away from $q/4$. We will produce a fairly loose bound, since the asymptotic parameters will only be effected marginally. A precise analysis could be obtained using standard techniques. We recall that by the properties of discrete Gaussians, it holds that $\|\mathbf{t}_f\|_\infty \leq \tau\sqrt{m+N}$ with all but $2^{-(m+N)} = \text{negl}(\lambda)$ probability, and also that asymptotically $\ell \leq 2^\nu$. Therefore, with all but negligible probability

$$\begin{aligned}
\left[\tilde{\mathbf{e}}^T \mathbf{H} \| \mathbf{e}^T \| e'\right] \cdot \mathbf{t}_f &\leq \left\| \left[\tilde{\mathbf{e}}^T \mathbf{H} \| \mathbf{e}^T \| e'\right] \right\|_\infty \cdot \|\mathbf{t}_f\|_\infty \cdot (N + m + 1) \\
&\leq \left(\tilde{B} \cdot (N+1)^d \cdot (\ell N) + B \cdot (m+1)\right) \cdot \|\mathbf{t}_f\|_\infty \cdot (N + m + 1) \\
&\leq \left(\tilde{B} \cdot (N+1)^d \cdot (\ell N) + B \cdot (m+1)\right) \tau\sqrt{m+N} \cdot (N + m + 1) \\
&\leq B \cdot (N+1)^{2(d_{\text{prf}}+d)} 2^{2\nu} \cdot \text{poly}(n, \log q).
\end{aligned}$$

Since we set $q/B \geq (N+1)^{2(d_{\text{prf}}+d)} 2^{3\nu}$, we get that correctness holds asymptotically for any such $\ell(\lambda), d(\lambda)$.

4.2 Security

We prove that our scheme is semi-adaptively secure as per Definition 6. Our proof heavily relies on the structure of the string Δ. Whereas Δ has a succinct representation as the output of a PRF, the view of the adversary does not depend on the seed of the PRF in any way except through the bits of Δ. Therefore, it follows from the pseudorandomness property that Δ is indistinguishable from a completely random string. It follows, therefore, that XORing x^* into Δ will go unnoticed by the adversary. However, this allows us to embed the challenge attribute in the public parameters in an indirect way, namely, now the XOR of the PRF's ith bit with Δ_i is exactly x^*_i. This means that $x^*_i \oplus \Delta_i = \mathsf{PRF}(i)$ and thus that $\mathbf{C}_i - (x^*_i \oplus \Delta_i)\mathbf{G}$ is independent of x^* itself and therefore can be known to the reduction ahead of time. This will allow us to apply similar techniques to those in [9] to prove security. A formal statement of the lemma together with a detailed sketch of the proof follows.

Lemma 1. *Let* PRF *be a family of secure pseudorandom functions as per Sect. 2.2, and let* oldABE *be a selectively secure ABE scheme for the function class* $\mathsf{BitCheck}_{\nu,x}$ *for some super-logarithmic* $\nu = \nu(\lambda)$*. Then under the* $\mathrm{DLWE}_{n,q,\chi}$ *assumption, the scheme* ABE *is a semi-adaptively secure ABE scheme for the function class* $\mathcal{P}_d$*.*

Proof (Extended sketch). We use ℓ^* to denote the length of the challenge attribute x^*. We also extend the notation x^*_i as follows: if $i \leq \ell^*$ then x^*_i denotes the ith bit of x^* as usual, however, for $i > \ell^*$ our convention is that $x^*_i = 0$.

The proof follows by a sequence of hybrids. We consider an adversary $\mathcal{A}$ for the semi-adaptive security game in Definition 6. Let $\mathrm{Adv}[\mathcal{A}]$ denote the advantage of $\mathcal{A}$ in the security game. We will denote by $\mathrm{Adv}_{\mathcal{H}}[\mathcal{A}]$ the advantage of $\mathcal{A}$ in the experiment described in hybrid $\mathcal{H}$.

Hybrid $\mathcal{H}_0$*.* This is the ABE semi-adaptive security game as per Definition 5. By definition $\mathrm{Adv}[\mathcal{A}] = \mathrm{Adv}_{\mathcal{H}_0}[\mathcal{A}]$.

Hybrid $\mathcal{H}_1$*.* In this hybrid, we change the way the (infinite) string $\varDelta$ is defined. Recall that in the previous hybrid, $\varDelta_i = \mathsf{PRF.Eval}(\sigma, i)$. However in this hybrid and throughout the proof we set

$$\varDelta_i = \begin{cases} (\mathsf{PRF.Eval}(\sigma, i) \oplus x^*_i) & \text{if } i \leq \ell^*, \\ \mathsf{PRF.Eval}(\sigma, i) & \text{otherwise.} \end{cases} \tag{3}$$

Note that now x^* needs to be known in order to compute $\varDelta$. However, $\varDelta$ is not used at all until the first key query is answered. Therefore, to execute this hybrid, the challenger only needs to know x^* before responding to the first key query, which is consistent with semi-adaptive security.

To see why the view of the adversary is indistinguishable in $\mathcal{H}_1$ and $\mathcal{H}_0$, consider replacing $\mathsf{PRF.Eval}(\sigma, i)$ with an oracle that returns a random bit for every i. In such case, the distributions in both hybrids are identical. Since σ itself is not used anywhere except to generate $\mathsf{PRF.Eval}(\sigma, i)$, the pseudorandomness of PRF guarantees that the views when using $\mathsf{PRF.Eval}(\sigma, i)$ are computationally indistinguishable. We conclude that

$$|\mathrm{Adv}_{\mathcal{H}_1}[\mathcal{A}] - \mathrm{Adv}_{\mathcal{H}_0}[\mathcal{A}]| = \mathrm{negl}(\lambda).$$

We remark that this is the only place where the pseudorandomness of the PRF is used, and from this hybrid and on one can think of σ as public.

Lastly, we notice that since we extended our notation so that $x^*_i = 0$ for $i > \ell^*$, we can say that from this hybrid and throughout the proof, it holds that $\varDelta_i = \mathsf{PRF.Eval}(\sigma, i) \oplus x^*_i$ for all $i \in \mathbb{N}$.

Hybrid $\mathcal{H}_2$*.* We now change the way the matrices $\vec{\mathbf{B}}$ are generated. We will now generate $\mathbf{B}_i$ as follows: Sample $\mathbf{R}_i \xleftarrow{\$} \{0,1\}^{m \times N}$ and set $\mathbf{B}_i = \mathbf{A}\mathbf{R}_i + \sigma_i \mathbf{G}$. Indistinguishability will follow from the leftover hash lemma since $m \geq n\lceil \log q \rceil + 2\lambda$. We point out that one has to be careful when applying the leftover hash lemma since $\mathbf{A}$ is only statistically close to uniform, and it is generated together with $\mathbf{A}_{\tau_0}^{-1}$. We notice, however that $\mathbf{A}_{\tau_0}^{-1} - \mathbf{A} - \mathbf{A}\mathbf{R}_i$ is a Markov chain, and therefore we can think about first sampling $\mathbf{A}$ and then sampling $\mathbf{A}_{\tau_0}^{-1}$ and $\mathbf{A}\mathbf{R}_i$ independently from the marginals. Therefore, since $(\mathbf{A}, \mathbf{A}\mathbf{R}_i)$ is statistically indistinguishable from uniform when $\mathbf{A}$ is uniform, it also holds true when $\mathbf{A}$ is only statistically close to uniform, and also holds true when $\mathbf{A}_{\tau_0}^{-1}$ is known as well.

$$|\mathrm{Adv}_{\mathcal{H}_2}[\mathcal{A}] - \mathrm{Adv}_{\mathcal{H}_1}[\mathcal{A}]| = \mathrm{negl}(\lambda).$$

We notice that in this hybrid, we now have that $\vec{\mathbf{B}} = \mathbf{A}\vec{\mathbf{R}} + \sigma\vec{\mathbf{G}}$, where $\vec{\mathbf{R}} = [\mathbf{R}_1\| \cdots \|\mathbf{R}_\eta]$. Recalling that $\mathbf{C}_i = \mathsf{Eval}(\mathsf{PRF.Eval}(\cdot, i), \vec{\mathbf{B}})$, we can define $\mathbf{H}_i^* = \mathbf{H}_{\mathsf{PRF.Eval}(\cdot,i),\sigma,\vec{\mathbf{B}}}$, and it will hold that

$$\mathbf{C}_i = \mathbf{A}\vec{\mathbf{R}}\mathbf{H}_i^* + \mathsf{PRF.Eval}(\sigma, i) \cdot \mathbf{G} = \mathbf{A}\vec{\mathbf{R}}\mathbf{H}_i^* + (x_i^* \oplus \Delta_i)\mathbf{G}. \tag{4}$$

We recall that $\mathbf{H}_i^*$ is computable given σ, and furthermore $\|\mathbf{H}_i^*\|_\infty \leq (N+1)^{d_{\mathrm{prf}}}$. If we denote $\vec{\mathbf{H}}^* = [\mathbf{H}_1^*\| \cdots \|\mathbf{H}_\ell^*]$, we conclude that

$$\vec{\mathbf{C}} - (x^* \oplus \Delta_{\leq \ell})\vec{\mathbf{G}} = \mathbf{A}\vec{\mathbf{R}}\vec{\mathbf{H}}^*. \tag{5}$$

Hybrid $\mathcal{H}_3$. In this hybrid we will switch from generating sk_f using $\mathbf{A}_{\tau_0}^{-1}$ to generating them using $\vec{\mathbf{R}}$. We recall that we are only required to generate keys for f s.t. $f(x^*) = 1$, otherwise the adversary loses in the semi-adaptive security game.

We recall that by definition, in order to derive sk_f, we need to sample from $[\mathbf{C}_f\|\mathbf{A}]_\tau^{-1}$. We recall that we defined $\mathbf{C}_f = \mathsf{Eval}(f_\Delta, \vec{\mathbf{C}})$, and therefore, denoting $\mathbf{H} = \mathbf{H}_{f_\Delta,(x^*\oplus\Delta_{\leq\ell}),\vec{\mathbf{C}}}$, it holds that

$$\mathbf{C}_f - f_\Delta(x^* \oplus \Delta_{\leq \ell}) \cdot \mathbf{G} = \left(\vec{\mathbf{C}} - (x^* \oplus \Delta_{\leq \ell})\vec{\mathbf{G}}\right) \cdot \mathbf{H}.$$

Plugging in Eq. (5), and since $f_\Delta(x^* \oplus \Delta_{\leq \ell}) = f(x^*) = 1$, we get that

$$\mathbf{C}_f = \mathbf{A}\vec{\mathbf{R}}\vec{\mathbf{H}}^*\mathbf{H} + \mathbf{G}.$$

Therefore, $[\mathbf{C}_f\|\mathbf{A}] = [\mathbf{A} \cdot (\vec{\mathbf{R}}\vec{\mathbf{H}}^*\mathbf{H}) + \mathbf{G}\|\mathbf{A}]$. This means that given $\vec{\mathbf{R}}$ and the computable matrices $\vec{\mathbf{H}}^*, \mathbf{H}$, one can sample from $[\mathbf{C}_f\|\mathbf{A}]_\tau^{-1}$ for all values of $\tau \geq \tau'$ for $\tau' = O\left(m \cdot \left\|\vec{\mathbf{R}} \cdot \vec{\mathbf{H}}^* \cdot \mathbf{H}\right\|_\infty\right)$. Plugging in the known bounds, we get that

$$\tau' = O(m \cdot N\eta \cdot (N+1)^{d_{\mathrm{prf}}} \cdot N\ell \cdot (N+1)^d) = O(\ell) \cdot (N+1)^{d+d_{\mathrm{prf}}} \cdot mN^2,$$

Recall that we need to sample with $\tau = 2^\nu \cdot mN^2(N+1)^{d+d_{\mathrm{prf}}}$ which is asymptotically greater than τ', which is enabled by our parameter setting.

It follows that changing our method of sampling $\mathbf{r}_f$ does not change the resulting distribution, and therefore

$$\mathrm{Adv}_{\mathcal{H}_3}[\mathcal{A}] = \mathrm{Adv}_{\mathcal{H}_2}[\mathcal{A}].$$

We notice that in this hybrid, the challenger does not require $\mathbf{A}_{\tau_0}^{-1}$ at all.

Hybrid $\mathcal{H}_4$*.* In this hybrid, we change the distribution of $\mathbf{A}$ and sample it uniformly from $\mathbb{Z}_q^{n\times m}$ rather than via $\mathsf{TrapGen}$. Since $\mathsf{TrapGen}$ samples $\mathbf{A}$ which is statistically indistinguishable from uniform, we conclude that the distribution produced in the two hybrids are statistically indistinguishable as well.

$$|\mathrm{Adv}_{\mathcal{H}_4}[\mathcal{A}] - \mathrm{Adv}_{\mathcal{H}_3}[\mathcal{A}]| = \mathrm{negl}(\lambda).$$

Hybrid $\mathcal{H}_5$*.* In this hybrid we change the way the challenge ciphertext is computed. Specifically we change the way we compute $\psi_{i,1-\Delta_i}$, for all i, and set

$$\psi_{i,1-\Delta_i} = \mathsf{oldABE.Enc}_{\mathsf{oldabepp}}(\mathbf{0}, (\ell^*, i, 1-\Delta_i)),$$

where the zero vector has the same length as $\mathbf{c}_{i,x_i^*\oplus\Delta_i\oplus 1}$.

Since for all ℓ, i, $\mathsf{BitCheck}_{n,\Delta_{\leq\ell}}(\ell, i, 1-\Delta_i) = 1$, and thus for all ℓ, the key $\mathsf{oldabesk}_\ell$ must not decrypt $\psi_{i,1-\Delta_i}$, we would like to use the security of oldABE to argue that $\mathcal{H}_5$ is computationally indistinguishable from $\mathcal{H}_4$. However, some care needs to be taken since we only assume that oldABE is *selectively* secure.

The formal proof will proceed via a hybrid argument going over all values of ℓ and β (note that we at this point we have an upper bound on ℓ given by the running time of $\mathcal{A}$). In the (i,β) hybrid, we change all ciphertexts $\psi_{i',\beta'}$ such that $(i',\beta') < (i,\beta)$ (lexicographically) to $\mathbf{0}$ if $\beta' \neq \Delta_{i'}$. To argue that two adjacent hybrids are indistinguishable, we rely on the selective hardness of oldABE for the fixed attribute (i,β) which can be provided in the beginning of the game as required for selective security.

We conclude that this hybrid is computationally indistinguishable from the previous one.

$$|\mathrm{Adv}_{\mathcal{H}_5}[\mathcal{A}] - \mathrm{Adv}_{\mathcal{H}_4}[\mathcal{A}]| = \mathrm{negl}(\lambda).$$

Hybrid $\mathcal{H}_6$*.* We again change the contents of the challenge ciphertext as follows. We generate $\mathbf{s}, \mathbf{e}, e'$ as before, and set $\mathbf{b}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}^T$, and $b' = \mathbf{s}^T\mathbf{v} + e'$. The vector $\mathbf{c}_0$ is generated identically to before, but we can express it in terms of $\mathbf{b}, b'$ as

$$\mathbf{c}_0^T = [\mathbf{b}^T \| b'] + \mu \lfloor q/2 \rceil \cdot [\mathbf{0}^T \| 1].$$

We recall that as of the previous hybrid, the values $\mathbf{c}_{i,x_i^*\oplus\Delta_i\oplus 1}$ no longer appear in the challenge ciphertext, so they are not generated at all. The only change that we make is in the generation of $\mathbf{c}_{i,x_i^*\oplus\Delta_i}$. We recall that in the previous hybrid

$$\mathbf{c}_{i,x_i^*\oplus\Delta_i}^T = \mathbf{s}^T(\mathbf{C}_i - (x_i^*\oplus\Delta_i)\mathbf{G}) + \tilde{\mathbf{e}}_i^T.$$

and since at this point $(\mathbf{C}_i - (x_i^*\oplus\Delta_i)\mathbf{G}) = \mathbf{A}\vec{\mathbf{R}}\mathbf{H}_i^*$, as per Eq. (4), we had that

$$\mathbf{c}_{i,x_i^*\oplus\Delta_i}^T = \mathbf{s}^T\mathbf{A}\vec{\mathbf{R}}\mathbf{H}_i^* + \tilde{\mathbf{e}}_i^T.$$

In this hybrid, we change these values to

$$\mathbf{c}_{i,x_i^*\oplus\Delta_i}^T = \mathbf{b}^T\vec{\mathbf{R}}\mathbf{H}_i^* + \tilde{\mathbf{e}}_i^T = \mathbf{s}^T\mathbf{A}\vec{\mathbf{R}}\mathbf{H}_i^* + \mathbf{e}^T\vec{\mathbf{R}}\mathbf{H}_i^* + \tilde{\mathbf{e}}_i^T.$$

This distribution, however, is statistically close to the previous one, since the distribution $\mathbf{e}^T \vec{\mathbf{R}}\mathbf{H}_i^*$ is $\big(B \cdot m \cdot \eta N \cdot (N+1)^{d_{\mathrm{prf}}}\big)$-bounded and since we selected $\tilde{\chi}$ to be $\big(Bm\eta N(N+1)^{d_{\mathrm{prf}}}\big)$-swallowing, statistical indistinguishability follows by definition.

$$|\mathrm{Adv}_{\mathcal{H}_6}[\mathcal{A}] - \mathrm{Adv}_{\mathcal{H}_5}[\mathcal{A}]| = \mathrm{negl}(\lambda).$$

We note that in this hybrid, given $\mathbf{b}, b'$, the challenger does not need to know the values of $\mathbf{s}, \mathbf{e}, e'$ since they are not used directly.

Hybrid $\mathcal{H}_7$. In the final hybrid, we change the distribution of $\mathbf{b}, b'$ to be uniform in $\mathbb{Z}_q^m, \mathbb{Z}_q$, respectively. Indistinguishability follows by definition from the $\mathrm{DLWE}_{n,q,\chi}$ assumption. We have

$$|\mathrm{Adv}_{\mathcal{H}_7}[\mathcal{A}] - \mathrm{Adv}_{\mathcal{H}_6}[\mathcal{A}]| = \mathrm{negl}(\lambda).$$

Clearly, in this hybrid the adversary has no advantage since b' is uniform and completely masks the value of μ. It follows therefore that

$$\mathrm{Adv}_{\mathcal{H}_7}[\mathcal{A}] = 1/2,$$

and therefore

$$|\mathrm{Adv}[\mathcal{A}] - 1/2| = \mathrm{negl}(\lambda),$$

which completes the proof of security.

4.3 Conclusion

Finally we can put all the pieces together and state our result with all parameters.

Theorem 2. *Assume that* GapSVP *(respectively* SIVP*) is hard to approximate by a polynomial time classical (respectively quantum) algorithm to within a factor of* 2^{n^ϵ}*. Then for any polynomial* $d = d(\lambda)$ *there exists a correct and semi-adaptively secure ABE scheme for the policy class* $\mathcal{P}_d$*.*

Letting $k = (\lambda d)^{1/\epsilon}$*, the public parameters of the scheme are of size* $\mathrm{poly}(k)$*, ciphertexts are of length* $\ell \cdot \mathrm{poly}(k)$*, where* ℓ *is the attribute length, and the key length is* $\ell + \mathrm{poly}(k)$*, where* ℓ *is the input length of the policy function (all* $\mathrm{poly}(\cdot)$ *notations indicate a specific polynomial function).*

Proof. A secure family of pseudorandom functions can be instantiated based on the existence of any one-way function, and in particular on the hardness of lattice approximation to within $\mathrm{poly}(n) \ll 2^{n^\epsilon}$ factor.

We instantiate oldABE using the scheme from [9]. Recall that oldABE only needs to support attributes of length $O(\nu)$ and policies which can be represented by circuits of depth $O(\log(\nu))$. This means that such a scheme can be based on the hardness of DLWE with parameters that translate to the hardness of lattice approximation to within a factor of $2^{n^{o(1)}} \ll 2^{n^\epsilon}$. The keys and ciphertexts of oldABE will have overhead $\mathrm{poly}(\lambda)$ for a fixed polynomial.

Combining these primitives with the correctness analysis and with the security analysis in Lemma 1, the theorem follows.

References

1. Abdalla, M., Catalano, D., Dent, A.W., Malone-Lee, J., Neven, G., Smart, N.P.: Identity-based encryption gone wild. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 300–311. Springer, Heidelberg (2006)
2. 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, Heidelberg (2010)
3. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 98–115. Springer, Heidelberg (2010)
4. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: Miller, G.L. (ed.) Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, 22–24 May 1996, pp. 99–108. ACM (1996)
5. Alekhnovich, M.: More on average case vs. approximation complexity. In: Proceedings of the 44th Symposium on Foundations of Computer Science (FOCS 2003), Cambridge, MA, USA, 11–14 October 2003, pp. 298–307. IEEE Computer Society (2003)
6. Ananth, P., Brakerski, Z., Segev, G., Vaikuntanathan, V.: From selective to adaptive security in functional encryption. In: Gennaro and Robshaw [16], pp. 657–677 (2015)
7. Ananth, P., Sahai, A.: Functional encryption for turing machines. IACR Cryptology ePrint Archive 2015:776 (2015)
8. Blum, A., Furst, M.L., Kearns, M., Lipton, R.J.: Cryptographic primitives based on hard learning problems. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 278–291. Springer, Heidelberg (1994)
9. 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, Heidelberg (2014)
10. Boyen, X.: Attribute-based functional encryption on lattices. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 122–142. Springer, Heidelberg (2013)
11. Brakerski, Z., Vaikuntanathan, V.: Constrained key-homomorphic PRFs from standard lattice assumptions. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part II. LNCS, vol. 9015, pp. 1–30. Springer, Heidelberg (2015)
12. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. J. Cryptol. **25**(4), 601–639 (2012)
13. Chen, J., Wee, H.: Semi-adaptive attribute-based encryption and improved delegation for Boolean formula. In: Abdalla, M., De Prisco, R. (eds.) SCN 2014. LNCS, vol. 8642, pp. 277–297. Springer, Heidelberg (2014)
14. Garg, S., Gentry, C., Halevi, S., Sahai, A., Waters, B.: Attribute-based encryption for circuits from multilinear maps. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 479–499. Springer, Heidelberg (2013)
15. Garg, S., Gentry, C., Halevi, S., Zhandry, M.: Fully secure attribute based encryption from multilinear maps. IACR Cryptology ePrint Archive 2014:622 (2014)
16. Gennaro, R., Robshaw, M. (eds.): CRYPTO 2015. LNCS, vol. 9216. Springer, Heidelberg (2015)

17. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC (2008)
18. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg (2013)
19. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) Symposium on Theory of Computing Conference, STOC 2013, Palo Alto, CA, USA, 1–4 June 2013, pp. 545–554. ACM (2013)
20. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Predicate encryption for circuits from LWE. In: Gennaro and Robshaw [16], pp. 503–523 (2015)
21. Gorbunov, S., Vaikuntanathan, V., Wichs, D.: Leveled fully homomorphic signatures from standard lattices. In: Servedio, R.A., Rubinfeld, R. (eds.) Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, 14–17 June 2015, pp. 469–477. ACM (2015)
22. 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., di Vimercati, S.D.C. (eds.) Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, 30 October–3 November 2006, pp. 89–98. ACM (2006)
23. Hohenberger, S., Waters, B.: Attribute-based encryption with fast decryption. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 162–179. Springer, Heidelberg (2013)
24. Katz, J., Wang, N.: Efficiency improvements for signature schemes with tight security reductions. In: Jajodia, S., Atluri, V., Jaeger, T. (eds.) Proceedings of the 10th ACM Conference on Computer and Communications Security, CCS 2003, Washington, DC, USA, 27–30 October 2003, pp. 155–164. ACM (2003)
25. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010)
26. Lewko, A., Waters, B.: Unbounded HIBE and attribute-based encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 547–567. Springer, Heidelberg (2011)
27. Lewko, A., Waters, B.: New proof methods for attribute-based encryption: achieving full security through selective techniques. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 180–198. Springer, Heidelberg (2012)
28. Micciancio, D., Mol, P.: Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 465–484. Springer, Heidelberg (2011)
29. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012)
30. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (2010)
31. Okamoto, T., Takashima, K.: Fully secure unbounded inner-product and attribute-based encryption. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 349–366. Springer, Heidelberg (2012)

32. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, 31 May–2 June 2009, pp. 333–342 (2009)
33. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, 22–24 May 2005, pp. 84–93 (2005)
34. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
35. Schnorr, C.: A hierarchy of polynomial time lattice basis reduction algorithms. Theor. Comput. Sci. **53**, 201–224 (1987)
36. Waters, B.: Functional encryption for regular languages. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 218–235. Springer, Heidelberg (2012)
37. Waters, B.: A punctured programming approach to adaptively secure functional encryption. In: Gennaro and Robshaw [16], pp. 678–697 (2015)