Attribute-Based Encryption for Circuits of Unbounded Depth from Lattices:

Garbled Circuits of Optimal Size, Laconic Function Evaluation, and More

謝耀慶 (Yao-Ching Hsieh) Huijia Lin 罗辑 (Ji Luo)

Paul G. Allen School of Computer Science & Engineering,
University of Washington, Seattle
{ychsieh,rachel,luoji}@cs.washington.edu

5 November 2023

Abstract

Although we have known about fully homomorphic encryption (FHE) from circular security assumptions for over a decade [Gentry, STOC '09; Brakerski–Vaikuntanathan, FOCS '11], there is still a significant gap in understanding related homomorphic primitives supporting all *unrestricted* polynomial-size computations. One prominent example is attribute-based encryption (ABE). The state-of-the-art constructions, relying on the hardness of learning with errors (LWE) [Gorbunov–Vaikuntanathan–Wee, STOC '13; Boneh *et al.*, Eurocrypt '14], only accommodate circuits up to a *predetermined* depth, akin to leveled homomorphic encryption. In addition, their components (master public key, secret keys, and ciphertexts) have sizes polynomial in the maximum circuit depth. Even in the simpler setting where a single key is published (or a single circuit is involved), the depth dependency persists, showing up in constructions of 1-key ABE and related primitives, including laconic function evaluation (LFE), 1-key functional encryption (FE), and reusable garbling schemes. So far, the only approach of eliminating depth dependency relies on indistinguishability obfuscation. An interesting question that has remained open for over a decade is whether the circular security assumptions enabling FHE can similarly benefit ABE.

In this work, we introduce new lattice-based techniques to overcome the depth-dependency limitations:

- Relying on a circular security assumption, we construct LFE, 1-key FE, 1-key ABE, and reusable garbling schemes capable of evaluating circuits of unbounded depth and size.
- Based on the *evasive circular* LWE assumption, a stronger variant of the recently proposed *evasive* LWE assumption [Wee, Eurocrypt '22; Tsabary, Crypto '22], we construct a full-fledged ABE scheme for circuits of unbounded depth and size.

Our LFE, 1-key FE, and reusable garbling schemes achieve optimal succinctness (up to polynomial factors in the security parameter). Their ciphertexts and input encodings have sizes linear in the input length, while function digest, secret keys, and garbled circuits have constant sizes independent of circuit parameters (for Boolean outputs). In fact, this gives the first constant-size garbled circuits without relying on indistinguishability obfuscation. Our ABE schemes offer short components, with master public key and ciphertext sizes linear in the attribute length and secret key being constant-size.

Keywords. attribute-based encryption, laconic function evaluation, functional encryption, garbled circuits, lattice, unbounded.

This is the full version (a major revision) of a paper to appear in the proceedings of FOCS 2023.

Contents

1 Introduction

Over the past few decades, there has been remarkable progress in developing a diverse array of homomorphic primitives. They enable computations involving secrets in a non-interactive and reusable manner, all while ensuring data privacy and/or integrity. This advancement has realized long-sought-after primitives such as fully homomorphic encryption (FHE) [RAD78,Gen09], attribute-based encryption (ABE) [SW05,GPSW06], functional encryption (FE) [SW05,GPSW06, O'N10,BSW11], and indistinguishability obfuscation ($i\mathcal{O}$) [DH76,BGI+01]. Additionally, a plethora of intriguing concepts have emerged, including laconic function evaluation (LFE) [QWW18a], reusable garbling [GKP+13b], homomorphic commitments and signatures [BF11,GVW15b], constrained pseudorandom functions [BW13], among others.

A central research objective is to develop homomorphic primitives supporting all *unrestricted* polynomial-size computations, with no predetermined bound on the parameters such as description size, input length, and depth. Notable successes in this pursuit include the development of FHE from circular security assumptions [Gen09,BV11], as well as FE and $i\mathcal{O}$ from well-studied assumptions [JLS21,JLS22]. These constructions accommodate unrestricted polynomial-size circuits. Since $i\mathcal{O}$ serves as a powerful tool for achieving other cryptographic goals, the latter results imply the feasibility of FHE [CLTV15] and ABE [GGH+13a,JLL23] for unrestricted polynomial-size computations, and potentially other homomorphic primitives.

However, beyond these two successes, progress has stagnated, particularly concerning ABE and related primitives. ABE provides fined-grained access control for encrypted data, allowing data owners to encrypt data tied to public attributes x. Users hold partial decryption keys linked to various access policies f, ensuring that a ciphertext can only be decrypted by keys with matching policy, i.e., $f(x) = 1$. The state-of-the-art ABE construction based on the hardness of learning with errors (LWE) by Boneh *et al.* [BGG+14], following the initial work by Gorbunov, Vaikuntanathan, and Wee [GVW13], only supports circuits up to *predetermined* (polynomially bounded) depth, akin to leveled homomorphic encryption. This depth dependency presents itself in two ways. Functionally, it requires fixing a bound d on the maximum computation depth when generating the master public key, limiting subsequent computations to depths below d. In terms of efficiency, the scheme's components — master public keys, secret keys, and ciphertexts — grow in length polynomial in d. Such dependency has been inherited by other homomorphic primitives that employ techniques developed for ABE, including predicate encryption [GVW15a], homomorphic signatures [GVW15b], and constrained pseudorandom functions [BV15].

Even in the simpler scenario with just one secret key published, there is no known solution to overcome the depth limitation. Hence, we only have 1-key ABE for bounded-depth circuits, and similarly for LFE [QWW18a], 1-key FE [SS10,GVW12,GKP+13b], and reusable garbling [GKP+13b].

In summary, without using $i\mathcal{O}$, we are limited to *leveled* versions of ABE and related homomorphic primitives. However, these primitives are intuitively closer to homomorphic encryption in terms of capabilities and techniques than to $i\mathcal{O}$. Gentry [Gen09] introduced the bootstrapping technique based on circular security assumptions to remove depth dependency in FHE. However, after nearly *fifteen* years, there is no equivalent of bootstrapping for ABE, even when considering circular security assumptions and other lattice-based assumptions. In this work, we aim to fill the gap and give direct lattice-based constructions of unbounded depth ABE, LFE, 1-key FE, and reusable garbling. Such direct constructions without relying on $i\mathcal{O}$ yield simpler, more efficient, and post-quantum secure schemes.

1.1 Our Results

We present the first lattice-based constructions of ABE and several related primitives supporting circuits of unbounded depth (and size). Our constructions come in two versions. In the single-key setting, we construct 1-key ABE, LFE, 1-key FE, and compact reusable garbling schemes, based on a circular security assumption. In the multi-key setting, we achieve full-fledged ABE (secure against unbounded collusion) by leveraging a new *evasive circular* LWE assumption, which is a variant of the recently proposed *evasive* LWE assumptions [Wee22,Tsa22]. Notably, our LFE, 1-key FE, and reusable garbling schemes enjoy (asymptotically) optimal succinctness (up to polynomial factors in the security parameter). Their input encoding/ciphertexts scale in length linear in the input length, while function encoding/secret keys are of constant size (for Boolean-output circuits), independent of circuit size or depth. In fact, we obtain the first constant-size garbled circuits without using $i\mathcal{O}$, irrespective of the reusability property. Our ABE schemes have the same level of succinctness. Compared to prior constructions, our schemes eliminate the multiplicative overheads that grow polynomially with the maximum depth of the computations.

Below, we describe our results in more detail.

Depth-Unbounded LFE, 1-Key FE/ABE, and Reusable Garbling from Circular LWE. Circular security assumptions postulate that LWE samples are pseudorandom even when they are used to encrypt the underlying secret vector. We rely on the following specific circular LWE assumption (Assumption 1):

$$\text{with} \qquad \mathbf{A}_{\text{fhe}} = \begin{pmatrix} \overline{\mathbf{A}}_{\text{fhe}} \\ \mathbf{r}^\intercal \overline{\mathbf{A}}_{\text{fhe}} \end{pmatrix}, \ \mathbf{S} = \mathbf{A}_{\text{fhe}}\mathbf{R} - \mathsf{bits}(\mathbf{r}^\intercal, -1) \otimes \mathbf{G},$$

$$\text{it holds that} \qquad \mathbf{A}_{\text{fhe}}, \ \mathbf{S}, \ \overline{\mathbf{A}}', \ \mathbf{r}^\intercal \overline{\mathbf{A}}' \approx \$, \ \$, \ \$, \ \$,$$

where $\mathbf{r}$ consists of small Gaussian entries, $\mathbf{G}$ is the gadget matrix, $\overline{\mathbf{A}}_{\text{fhe}} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\overline{\mathbf{A}}' \xleftarrow{\$} \mathbb{Z}_q^{n \times m'}$, $\mathbf{R} \xleftarrow{\$} \{0,1\}^{m \times m''}$ for some appropriate m, m'' and any polynomial m', and independent Gaussian noises are added to the terms with wavy underlines. Here, $\mathbf{A}_{\text{fhe}}$ is a public key with the corresponding secret key $\mathbf{s} = (\mathbf{r}^\intercal, -1)^\intercal$ satisfying $\mathbf{s}^\intercal \mathbf{A}_{\text{fhe}} = \underset{\sim}{\mathbf{0}}$, and $\mathbf{S}$ is a circular encryption of (the bits of) $\mathbf{s}$ under the secret key $\mathbf{s}$, both generated honestly using the Gentry–Sahai–Waters FHE scheme [GSW13]. The assumption states that this circular encryption together with the other LWE samples are jointly pseudorandom. It is almost identical to the circular security assumption needed for bootstrapping the [GSW13] FHE scheme, except here the secret key is small Gaussian instead of uniformly random. In the literature, circular security assumptions with small secrets have already been used, e.g., for bootstrapping the Brakerski–Gentry–Vaikuntanathan FHE scheme [BGV12].

Assuming the above circular LWE assumption, we construct depth-unbounded LFE, 1-key FE (implying 1-key ABE), and compact reusable garbling schemes, all achieving optimal succinctness.

Laconic Function Evaluation. Introduced by Quach, Wee, and Wichs [QWW18a], LFE enables one party, Alice, to compress a large circuit C into a short digest_C. Using this digest, Bob can encrypt any input $\mathbf{x}$ in a way that allows Alice to recover $C(\mathbf{x})$ without learning anything else about $\mathbf{x}$. In LFE, the digest size and the encryption time (consequently, the ciphertext) are small — much smaller than the circuit size. It is a useful primitive in secure computation, implying Bob-optimized 2-party function evaluation and MPC protocols with laconic online computation. The prior construction of LFE [QWW18a] uses techniques developed in the constructions of ABE by [BGG+14] and 1-key FE of [GKP+13b], and inherits their depth dependency. The common reference string, circuit digests, and ciphertexts all grow in length polynomially in a bound on

the computation depth (specified when generating the common reference string). Assuming the circular LWE assumption, we remove such dependency.

Corollary 17 (LFE)**.** *Under the circular LWE assumption, there exists a very selectively secure LFE scheme for circuits of unbounded depth and bounded input/output lengths L, L' with*

$$|\mathsf{crs}| = \mathrm{O}(L), \qquad |\mathsf{digest}_C| = \mathrm{O}(L'), \qquad T_{\mathsf{Enc}} = \mathrm{O}(L + L'), \qquad |\mathsf{ct}| = \mathrm{O}(L + L').$$

In the above statement and rest of this introduction, $\mathrm{O}(\cdot)$ hides $\mathrm{poly}(\lambda)$ factors.

1-Key Functional Encryption. Sahai and Seyalioglu [SS10] introduced the idea of single-key FE and gave the first construction based on public-key encryption, followed by [GVW12] extending it to the setting with a bounded number of keys. However, these schemes only support circuits of bounded size, with components scaling with the maximum circuit size. Goldwasser, Kalai, Popa, Vaikuntanathan and Zeldovich [GKP+13b] presented the first single-key succinct FE, handling unbounded-size but bounded-depth circuits, where the components grow with the maximum depth of the computations instead of maximum size. Later, in the same work that introduced LFE [QWW18a], it was shown that LFE can be transformed into a succinct single-key FE scheme, using just a non-succinct single-key FE scheme such as the ones based on public-key encryption. Applying the same transformation to our LFE scheme for circuits of unbounded depth and size immediately yields a single-key FE for the same class of circuits.

Corollary 18 (1-key FE, implying 1-key ABE)**.** *Under the circular LWE assumption, there exists a very selectively 1-key simulation-secure FE scheme for circuits of unbounded depth and bounded input/output lengths L, L' with*

$$|\mathsf{mpk}| = \mathrm{O}(L + L'), \qquad |\mathsf{sk}_C| = \mathrm{O}(L'), \qquad |\mathsf{ct}(\mathbf{x})| = \mathrm{O}(L + L').$$

Reusable Garbled Circuits. An important application of single-key FE is reusable garbling introduced by [GKP+13b]. Here we consider reusable garbling that guarantees input privacy but not circuit privacy (i.e., the circuit C being garbled is public and does not need to be included in the garbled circuit). It enables converting a circuit C into a garbled form $\widehat{C}$ together with a public key pk. Using pk, one can encode an unbounded number of inputs $\mathbf{x}_i$ into encodings $\widehat{\mathbf{x}}_i$'s, which, together with $\widehat{C}$ and pk, reveals only the outputs $C(\mathbf{x}_i)$. As noted in [GKP+13b], 1-key FE scheme with succinct components implies succinct reusable garbled circuits — the garbled circuit is an FE secret key and the input encoding is an FE ciphertext. Therefore, our 1-key FE for unbounded-depth circuits immediately implies reusable garbled circuits with optimally succinct garbled circuit and input encodings.

Corollary 20 (reusable garbled circuits)**.** *Under the circular LWE assumption, there exists a selectively secure reusable garbling scheme for circuits with*

$$|\widehat{C}| = \mathrm{O}(L'), \qquad |\mathsf{pk}| = \mathrm{O}(L + L'), \qquad |\widehat{\mathbf{x}}| = \mathrm{O}(L + L'),$$

where L, L' are the input/output lengths of C.

Optimal Succinctness. We remark that our LFE, 1-key FE, and reusable garbling schemes for Boolean-output circuits have optimally succinct components. The digests/secret keys/garbled circuits have constant size independent of any aspect of circuit complexity. The sizes of ciphertexts/input encodings are linear in the length of the inputs encoded, which is necessary in order to hide the inputs. We remark that even for standard (non-reusable) garbling schemes,

constant-size garbled circuits were not known before without using $i\mathcal{O}$. While our notion of garbling does not hide the circuit, it is easy to generically transform such a garbling scheme to also hide the circuit — encrypt the circuit C using a one-time rate-1 secret-key encryption scheme to obtain a ciphertext $\widetilde{C}$ and garble the augmented circuit C' that on input the right secret key k and the original input $\mathbf{x}$, decrypts $\widetilde{C}$ using k to obtain C and computes $C(\mathbf{x})$. The new garbled circuit consist of $\widetilde{C}$, garbling of C', and labels for k, and has rate-1 size $(|C| + \mathrm{O}(1))$.

Full-Fledged Depth-Unbounded ABE from Evasive Circular LWE. We also construct full-fledged depth-unbounded ABE assuming a stronger assumption, called the *evasive circular LWE assumption*, which incorporates circularity into the evasive LWE assumption of [Wee22,Tsa22] (more on that later).

Construction 4 (depth-unbounded ABE)**.** *Under the evasive circular LWE assumption and the circular LWE assumption, there exists a very selectively secure ABE scheme for circuits of unbounded depth and bounded input length L with*

$$|\mathsf{mpk}| = \mathrm{O}(L), \qquad |\mathsf{sk}_C| = \mathrm{O}(1), \qquad |\mathsf{ct}_{\mathbf{x}}| = \mathrm{O}(L).$$

The secret key size of our ABE scheme is constant, while the master public key and the ciphertexts are compact, of size linear in the maximum attribute length. We can further remove the predetermined bound L on attribute length by applying the generic transformation of [GKW16] to obtain an ABE scheme for, truly, all polynomial-size computations, at the price of increasing the secret key size to be linear in the input length of the function encoded.

Corollary 24 (attribute-unbounded depth-unbounded ABE)**.** *Under the evasive circular LWE assumption and the circular LWE assumption, there also exists a very selectively secure ABE scheme for circuits of unbounded depth and input length with*

$$|\mathsf{mpk}| = \mathrm{O}(1), \qquad |\mathsf{sk}_C| = \mathrm{O}(L), \qquad |\mathsf{ct}_{\mathbf{x}}| = \mathrm{O}(|\mathbf{x}|),$$

where L is the input length of C in sk_C.

The Evasive Circular LWE Assumption. We explain the evasive circular LWE assumption at a high level. The evasive LWE assumption [Wee22,Tsa22] asserts that LWE samples $(\mathbf{s}^\intercal\mathbf{B} + \mathbf{e}_{\mathrm{B}}^\intercal)$ remain secure even in the presence of low-norm trapdoors $\mathbf{B}^{-1}(\mathbf{P})$ mapping $\mathbf{B}$ to another (not necessarily random) matrix $\mathbf{P}$, provided that $(\mathbf{r}^\intercal\mathbf{B} + \mathbf{e}_{\mathrm{B}}^\intercal, \mathbf{r}^\intercal\mathbf{P} + \mathbf{e}_{\mathrm{P}}^\intercal)$ (with fresh random noises $\mathbf{e}_{\mathrm{P}}$) are jointly pseudorandom. A simple formal version is the following. Fix an efficiently sampleable joint distribution $\mathcal{S}$ of matrices $\overline{\mathbf{A}}', \mathbf{P}$ and auxiliary information $\mathsf{aux} \in \{0,1\}^*$, the evasive LWE assumption postulates that

$$\begin{aligned} \text{if} \quad & ①: \mathbf{B}, \overline{\mathbf{A}}', \mathbf{P}, \mathbf{r}^\intercal\mathbf{B}, \mathbf{r}^\intercal\overline{\mathbf{A}}', \mathbf{r}^\intercal\mathbf{P}, \mathsf{aux} \approx ②: \mathbf{B}, \overline{\mathbf{A}}', \mathbf{P}, \$, \$, \$, \mathsf{aux}, \\ \text{then} \quad & ③: \mathbf{B}, \overline{\mathbf{A}}', \mathbf{P}, \mathbf{r}^\intercal\mathbf{B}, \mathbf{r}^\intercal\overline{\mathbf{A}}', \mathbf{K}, \mathsf{aux} \approx ④: \mathbf{B}, \overline{\mathbf{A}}', \mathbf{P}, \$, \$, \mathbf{K}, \mathsf{aux}. \end{aligned}$$

Here, $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ for $m = \Theta(n \log p)$, and $\mathbf{K} \xleftarrow{\$} \mathbf{B}^{-1}(\mathbf{P})$ is a low-norm matrix satisfying $\mathbf{BK} = \mathbf{P}$.[1] The public-coin version of this assumption requires that aux contain the random coins used for sampling from $\mathcal{S}$.

Our multi-key depth-unbounded ABE scheme relies on a stronger variant of the evasive LWE assumption, where both the precondition and the postcondition additionally include a public key

[1] $\mathbf{K}$ can be efficiently sampled using a trapdoor [MP12] of $\mathbf{B}$.

$\mathbf{A}_{\mathsf{fhe}}$ of [GSW13] and a circular *encoding* of the secret key. The circular encoding consists of two parts — a circular encryption $\mathbf{S}$ of the secret key under itself, and the attribute encoding [BGG+14] of $\mathbf{S}$ using the same secret and a matrix $\mathbf{A}_{\mathsf{circ}}$. More formally, fix an efficiently sampleable joint distribution of $\mathbf{A}_{\mathsf{circ}}, \overline{\mathbf{A}}', \mathbf{P}, \mathsf{aux}$, the evasive circular LWE assumption (Assumption 2) stipulates that

$$\begin{aligned} \text{if} \quad & ①,\ \mathbf{A}_{\mathsf{fhe}},\ \mathbf{S},\ \mathbf{A}_{\mathsf{circ}},\ \mathbf{s}^\intercal(\mathbf{A}_{\mathsf{circ}} - \mathbf{S} \otimes \mathbf{G}) \approx ②,\ \$,\ \$,\ \mathbf{A}_{\mathsf{circ}},\ \$, \\ \text{then} \quad & ③,\ \mathbf{A}_{\mathsf{fhe}},\ \mathbf{S},\ \mathbf{A}_{\mathsf{circ}},\ \mathbf{s}^\intercal(\mathbf{A}_{\mathsf{circ}} - \mathbf{S} \otimes \mathbf{G}) \approx ④,\ \$,\ \$,\ \mathbf{A}_{\mathsf{circ}},\ \$. \end{aligned}$$

Recall that $\mathbf{s} = (\mathbf{r}^\intercal, -1)^\intercal$ is essentially the same as $\mathbf{r}$. Note also that $\mathbf{A}_{\mathsf{circ}}, \overline{\mathbf{A}}', \mathbf{P}, \mathsf{aux}$ are sampled independent of the public key $\mathbf{A}_{\mathsf{fhe}}$ and the circular ciphertext $\mathbf{S}$, which are honestly generated.

The works of [Wee22,VWW22] argue that for evasive LWE, if the precondition holds, then no attacks are known against the postcondition. In particular, the family of zeroizing attacks (e.g., [CHL+15,CVW18,HJL21,JLLS23]) that have successfully ruled out many post-quantum obfuscation candidates and several recently proposed LWE with leakage assumptions (e.g., [GP21, WW21,DQV+21]) do not work. These attacks crucially rely on collecting equations of LWE secrets over the integers from correlated LWE samples (provided by or derived from the assumption or construction being analyzed). For evasive LWE, this strategy fails, since the precondition ensures that all LWE samples that can be obtained are jointly pseudorandom, and hence one cannot collect any useful equation over the integers. Our circular variant adds an FHE public key and a circular encoding to the precondition and the postcondition, while maintaining the same justification.

Lastly, we remark that it is possible to construct contrived auxiliary information with respect to which the (circular or non-circular) evasive LWE assumption becomes false (e.g., aux contains an obfuscation [VWW22]). However, no such counterexamples are known in the public-coin case, when aux contains the randomness used for sampling the matrices. Our KP-ABE only relies on the public-coin version of the evasive circular LWE assumption.

Our Techniques in a Nutshell. Our key technical contribution is a new bootstrapping method for the ABE schemes of [BGG+14]. At a high level, our bootstrapping technique allows transforming an attribute encoding $(\mathbf{s}^\intercal(\mathbf{A} - v\mathbf{G}) + \mathbf{e}_+^\intercal)$ of a bit v with large noises $\mathbf{e}_+$ into another encoding $(\mathbf{s}^\intercal(\mathbf{A}' - v\mathbf{G}) + (\mathbf{e}')^\intercal)$ of the same bit with smaller noises $\mathbf{e}'$ of some fixed magnitude. Moreover, the new matrix $\mathbf{A}'$ can be derived efficiently from $\mathbf{A}$ and other public matrices, independent of $\mathbf{s}$, v, and the noises. The reason that the scheme of [BGG+14] only supports evaluating circuits with *a priori* bounded depth d is the follows. Starting from an input encoding $(\mathbf{s}^\intercal(\mathbf{A} - \mathbf{x} \otimes \mathbf{G}) + \mathbf{e}^\intercal)$, for any circuit C, the homomorphic evaluation procedure produces an output encoding $(\mathbf{s}^\intercal(\mathbf{A}_C - v\mathbf{G}) + \mathbf{e}_+^\intercal)$, where the noise is exponential in d. To allow decryption, the modulus q (chosen when the scheme is set up) must be larger than the maximum output noise, upper-bounding the maximum depth of circuits that can be handled. Now, using our bootstrapping technique, when the noise becomes too large, we can simply reduce the noise to obtain a *refreshed* encoding $(\mathbf{s}^\intercal(\mathbf{A}' - v\mathbf{G}) + (\mathbf{e}')^\intercal)$, and perform further homomorphic evaluation on it. Since bootstrapping can be applied for an unbounded number of times, we can handle circuits of unbounded depth.

Our ABE bootstrapping is inspired by the FHE bootstrapping [Gen09], but differs significantly. The idea of FHE bootstrapping is publishing a circular encryption $\mathbf{S}$ of the secret key $\mathbf{s}$, and whenever a ciphertext $\mathbf{C}$ becomes too noisy, one can homomorphically evaluate the decryption function $\mathsf{Dec}(\cdot, \mathbf{C})$ with the ciphertext hardcoded inside, over the circular ciphertext $\mathbf{S}$ to obtain a new, less noisy, ciphertext $\mathbf{C}'$ of the same plaintext. Unfortunately, throughout the past decade, it remained unknown how to adapt FHE bootstrapping to the context of ABE.

Our ABE bootstrapping proceeds roughly in two steps. The first step adapts the rounding (or modulus reduction) technique used in the FHE scheme of [BGV12] to the ABE setting. Rounding an attribute encoding $(\mathbf{s}^\intercal(\mathbf{A} - v\mathbf{G}) + \mathbf{e}_+^\intercal) \bmod p$ naïvely would cause the modulus to decrease, leading again to depth-bounded evaluation. Instead, our new technique first rounds and then recovers the modulus. But after recovery, the result is not a well-formed attribute encoding, rather, it should be regarded as a "*noiseless* ciphertext" of form $(\mathsf{RndPad}(\mathbf{s}) - v\mathbf{s}^\intercal\mathbf{G})$, where $v\mathbf{s}^\intercal\mathbf{G}$ is the payload and $\mathsf{RndPad}(\mathbf{s})$ is a "blinding factor" that depends only on $\mathbf{s}$ and public matrices. We complement round-then-recover with another procedure generating an encoding of form $(\mathbf{s}^\intercal\mathbf{A}' - \mathsf{RndPad}(\mathbf{s}) + (\mathbf{e}')^\intercal)$ with small noises, which then gives a well-formed attribute encoding $(\mathbf{s}^\intercal(\mathbf{A}' - v\mathbf{G}) + (\mathbf{e}')^\intercal)$ for the same bit v with small noises. The second step crucially relies on an attribute encoding of the circular ciphertext, $\mathbf{s}^\intercal(\mathbf{A}_{\mathrm{circ}} - \mathbf{S} \otimes \mathbf{G}) + \mathbf{e}_{\mathrm{circ}}^\intercal$, as described in our evasive circular LWE assumption. More details in the technical overview (Section 1.3).

Lastly, we mention that it is not clear how to combine our bootstrapping techniques with the lattice trapdoor techniques employed in prior ABE constructions [GVW13,BGG+14]. As a result, our security proof does not rely on trapdoor simulation, more similar to recent works [QWW18a, LLL22,Wee22]. This contributed to our full-fledged ABE schemes relying on the evasive circular LWE assumption. We leave the question of removing the new assumption as an interesting future direction.

1.2 Related Works

Table 1 summarizes the current state of KP-ABE for circuits.

ABE Constructions Using $i\mathcal{O}$ and Related Primitives. Prior to our work, depth-unbounded ABE and (1-key or multi-key) ABE with depth-independent succinctness were only known using the strong tools of $i\mathcal{O}$ or $i\mathcal{O}$-related primitives. The work of [GKP+13a] builds ABE for Turing machines from extractable witness encryption and SNARK. Both the existence of extractable witness encryption and that of SNARK are knowledge assumptions in nature, and the only candidate of extractable witness encryption relies on differing-input obfuscation [ABG+13], which is even stronger than $i\mathcal{O}$. It is well known that $i\mathcal{O}$ itself implies FE for unbounded-depth circuits [GGH+13a], which implies ABE for unbounded-depth circuits. However, the direct construction yields an FE with non-succinct secret keys, of size polynomial in the circuit size. The recent work of [JLL23] uses such FE with non-succinct keys to construct FE and ABE for random-access machines (RAM) with succinct components. All the components of their scheme, the master public key, secret keys, and ciphertexts, are of constant size, whereas the master public key and ciphertexts of our ABE schemes still scales with input length.

ABE Constructions Without $i\mathcal{O}$. As mentioned earlier, the lattice-based ABE schemes of [GVW13,BGG+14] support circuits of *a priori* bounded depth and input length, and the scheme of [BGG+14] has components of size polynomial in the maximum depth. Several follow-up works improve their construction on the fronts considered in this work. The work of [LLL22] improves the secret key size from $\mathrm{poly}(d, \lambda)$ to $\mathrm{poly}(\lambda)$, but unfortunately still suffer the constraint of being depth-bounded and have the master public key and ciphertexts of size polynomially dependent on d. In addition, their scheme is designed in the generic pairing group model, thus not post-quantum secure. Observe that if using the [LLL22] scheme to construct LFE, 1-key FE, or resuable garbling, the resulting schemes would still have ciphertexts/input encodings scaling with computation depth, hence not optimally succinct.

Brakerski and Vaikuntanathan [BV16] presented an ABE scheme supporting unbounded attributes and satisfying semi-adaptive security, based on and modified from the scheme of [BGG+14].

Table 1. Comparison among select KP-ABE schemes for circuits.

reference	depth-unbounded	$\lvert\mathsf{mpk}\rvert$	$\lvert\mathsf{sk}_C\rvert$	$\lvert\mathsf{ct}_{\mathbf{x}}\rvert$	assumptions
[GGH+13a]	✓	$\mathrm{poly}(L)$	$\mathrm{poly}(\lvert C\rvert)$	$\mathrm{poly}(L)$	$i\mathcal{O}$
[GKP+13a]	✓	$\mathrm{O}(1)$	$\mathrm{O}(1)$	$\mathrm{poly}(L)$	exWE & SNARK
[AS16]	✓	$\mathrm{O}(1)$	$\mathrm{poly}(\lvert C\rvert)$	$\mathrm{poly}(L)$	$i\mathcal{O}$
[AJS17]	✓	$\mathrm{O}(1)$	$\mathrm{O}(\lvert C\rvert)$	$\mathrm{O}(L)$	$i\mathcal{O}$ (subexp)
[AM18]	✓	$\mathrm{O}(1)$	$\mathrm{poly}(\lvert C\rvert)$	$\mathrm{O}(L)$	FE
[KNTY19]	✓	$\mathrm{O}(1)$	$\mathrm{poly}(\lvert C\rvert)$	$\mathrm{poly}(L)$	FE
[GWZ22]	✓	$\mathrm{poly}(L)$	$\mathrm{poly}(\lvert C\rvert)$	$\mathrm{O}(L)$	$i\mathcal{O}$
[ACFQ22]	✓	$\mathrm{O}(1)$	$\mathrm{poly}(\lvert C\rvert)$	$\mathrm{poly}(L)$	FE & DE-PIR
[JLL23]	✓	$\mathrm{O}(1)$	$\mathrm{O}(1)$	$\mathrm{O}(1)$	FE
[GGH+13b]		$L\,\mathrm{poly}(d)$	$\lvert C\rvert\,\mathrm{poly}(d)$	$L\,\mathrm{poly}(d)$	MMaps
[GVW13]		$L\,\mathrm{poly}(d)$	$\lvert C\rvert\,\mathrm{poly}(d)$	$L\,\mathrm{poly}(d)$	LWE
[BGG+14]		$L\,\mathrm{poly}(d)$	$\mathrm{poly}(d)$	$L\,\mathrm{poly}(d)$	LWE
[BV16]		$\mathrm{O}(1)$	$\mathrm{O}(L)+\mathrm{poly}(d)$	$\lvert\mathbf{x}\rvert\,\mathrm{poly}(d)$	LWE
this work, 1-key	✓	$\mathrm{O}(L)$	$\mathrm{O}(1)$	$\mathrm{O}(L)$	csLWE
this work	✓	$\mathrm{O}(L)$	$\mathrm{O}(1)$	$\mathrm{O}(L)$	evcsLWE
this work	✓	$\mathrm{O}(1)$	$\mathrm{O}(L)$	$\mathrm{O}(\lvert\mathbf{x}\rvert)$	evcsLWE

$L, d, |C|$ are the input length, the depth, and the size of C, and $|\mathbf{x}|$ is the length of $\mathbf{x}$ in attribute-unbounded schemes. For schemes supporting Turing machines or random-access machines, the shown efficiency is that when the scheme is used for circuits. In component sizes, $\mathrm{poly}(\lambda)$ factors are ignored. For assumptions: exWE is extractable witness encryption; subexp means subexponential security; FE is for circuits; DE-PIR is doubly efficient private information retrieval; MMaps is multilinear maps; csLWE is circular small-secret LWE; evcsLWE is evasive small-secret LWE; only the heaviest assumptions are listed.

Goyal, Koppula, and Waters [GKW16] presented a generic transformation converting any selectively secure attribute-bounded ABE into a semi-adaptively secure attribute-unbounded one. As mentioned above, the transformation of [GKW16] can also be applied to our depth-unbounded ABE scheme to further remove the predetermined bound on attribute length, yielding a scheme that handles truly all polynomial-size computations.

1.3 Technical Overview

In this section, we present the core techniques of our unbounded homomorphic evaluation before exemplifying its usage with attribute-based laconic function evaluation (AB-LFE).

[BGG+14] Bounded Homomorphism. Our starting point is the attribute encoding and its homomorphic evaluation due to [BGG+14]. Let $\mathbf{G}$ be the gadget matrix. Given public matrix $\mathbf{A}$ and LWE secret $\mathbf{s}$, the encoding of $\mathbf{x}$ (bit-string) is $\underset{\sim}{\mathbf{s}^\intercal(\mathbf{A}-\mathbf{x}\otimes\mathbf{G})}$, where the wavy underline indicates the presence of noise. Given a circuit C with Boolean output, one can compute a low-norm matrix $\mathbf{H}_C$, and both C and $\mathbf{x}$, a low-norm matrix $\mathbf{H}_{C,\mathbf{x}}$, satisfying

$$(\mathbf{A}-\mathbf{x}\otimes\mathbf{G})\mathbf{H}_{C,\mathbf{x}} = \mathbf{A}\mathbf{H}_C - C(\mathbf{x})\mathbf{G} \qquad \Longrightarrow \qquad \underset{\sim}{\mathbf{s}^\intercal(\mathbf{A}-\mathbf{x}\otimes\mathbf{G})}\mathbf{H}_{C,\mathbf{x}} = \underset{\sim}{\mathbf{s}^\intercal(\mathbf{A}\mathbf{H}_C - C(\mathbf{x})\mathbf{G})}.$$

The procedure can be regarded as evaluation on a matrix-valued circuit $\mathbf{x}\mapsto C(\mathbf{x})\mathbf{G}$, and it can be extended to such circuits having arbitrary matrix output (not just multiples of $\mathbf{G}$).

The norms of $\mathbf{H}_C$ and $\mathbf{H}_{C,\mathbf{x}}$ grow exponentially with the depth of C, which translates to the noise growth. Once the modulus q is fixed, it puts a polynomial bound on the depth — $\mathrm{O}(\log q)$ at

the very most. After some *a priori* fixed polynomial depth, the noise will grow beyond tolerance. Beyond correctness, the security proof of [BGG+14] also relies on $\mathbf{H}$ being low-norm. Clearly, the crux of the matter is controlling noise growth during homomorphic evaluation.

Inspirations from FHE. In fully homomorphic encryption literature, there are two major ways of dealing with noise, *rounding* and *bootstrapping*.

Let M (factor of q) be a rounding resolution, C a low-depth subcircuit, $\mathbf{x}$ an input, and $\mathbf{A}_C = \mathbf{A}\mathbf{H}_C$ the public matrix for C. As a first attempt, after evaluation of C with noise about to overflow, we might try

$$\left\lfloor \frac{(\mathbf{s}^\intercal(\mathbf{A}_C - C(\mathbf{x})\mathbf{G}) + \mathbf{e}_{\text{large}}^\intercal) \bmod q}{M} \right\rceil = (\mathbf{s}^\intercal \mathbf{A}_{C,\text{small}} - C(\mathbf{x})\mathbf{s}^\intercal \mathbf{G}_{\text{small}} + \mathbf{e}_{\text{small}}^\intercal) \bmod \frac{q}{M}. \quad (1)$$

The secret $\mathbf{s}$ is now assumed to be low-norm. Here, $\mathbf{e}_{\text{small}}$ contains both the rounded $\mathbf{e}_{\text{large}}$ and the rounding error. Rounding reduces the absolute magnitude of noise, but it also shrinks the modulus, and the modulus-to-noise ratio remains large. Therefore, rounding does not enable unbounded homomorphic evaluation *by itself*.

Bootstrapping reduces the noise without diminishing the modulus. Let $\mathsf{hct}(\mathbf{x})$ represent an FHE ciphertext of $\mathbf{x}$. Suppose HEval is the FHE evaluation procedure such that

$$\mathsf{HEval}_C(\mathsf{hct}(\mathbf{x})) = \mathsf{hct}(C(\mathbf{x})) \qquad \text{for any circuit } C : \mathbf{x} \mapsto C(\mathbf{x}),$$

where the noise magnitude of the output $\mathsf{hct}(C(\mathbf{x}))$ depends on that of the input $\mathsf{hct}(\mathbf{x})$ and the depth of C. To bootstrap, we publish $\mathsf{hct}(\mathsf{hsk})$, an FHE ciphertext of the FHE secret key hsk (encrypted under itself). Given $\mathsf{hct}_{\text{large}} = \mathsf{hct}(x)$, which, though still decryptable to x, might contain large noise, we let $C_u(v)$ be a circuit with u hardwired that performs FHE decryption on u (ciphertext) using input v (key), and run

$$\mathsf{hct}_{\text{small}} = \mathsf{HEval}_{C_{\mathsf{hct}_{\text{large}}}}(\mathsf{hct}(\mathsf{hsk})) = \mathsf{hct}(C_{\mathsf{hct}_{\text{large}}}(\mathsf{hsk})) = \mathsf{hct}(x).$$

The output $\mathsf{hct}_{\text{small}}$ is again a ciphertext of x, but its noise magnitude only depends on that of $\mathsf{hct}(\mathsf{hsk})$ and the depth of FHE decryption circuit, not that of $\mathsf{hct}_{\text{large}}$. This procedure restores the noise to a fixed amount and helps achieving (non-leveled) FHE. However, it is not clear how bootstrapping can be applied to attribute encoding [BGG+14] (or more generally, ABE).

Difficulties of Bootstrapping ABE. A naïve envision of reducing the noise in $\mathbf{c}^\intercal = \mathbf{s}^\intercal(\mathbf{A}_C - C(\mathbf{x})\mathbf{G})$ by bootstrapping is to homomorphically evaluate, on an attribute encoding of $\mathbf{s}$, a circuit $C_u(v)$ with $u = \mathbf{c}$ hardwired that outputs the value (i.e., $C(\mathbf{x})$) encoded using the input $v = \mathbf{s}$ so that[2]

$$\text{(wishful thinking)} \qquad \mathbf{s}^\intercal(\mathbf{A} - \mathbf{s} \otimes \mathbf{G})\mathbf{H}_{C_\mathbf{c}} = \mathbf{s}^\intercal(\mathbf{A}\mathbf{H}_{C_\mathbf{c}} - C_\mathbf{c}(\mathbf{s})\mathbf{G}) = \mathbf{s}^\intercal(\mathbf{A}\mathbf{H}_{C_\mathbf{c}} - C(\mathbf{x})\mathbf{G}),$$

where the output noise only depends on that in $\mathbf{s}^\intercal(\mathbf{A} - \mathbf{s} \otimes \mathbf{G})$ and the depth of $C_\mathbf{c}$, but not that in $\mathbf{c}$. There are two issues with this approach.

One is that the circuit $C_\mathbf{c}$ is ciphertext-dependent, thus unknown at key generation time, making it difficult, if possible at all, to generate the key corresponding to the correct circuit. This is also highlighted by the difference between the security of FHE and ABE — in FHE, decryption works regardless of the homomorphic computation applied to the ciphertext, whereas in ABE, since the

[2]Precisely speaking, the attribute encoding (and FHE ciphertexts of $\mathbf{s}$ later) is bit by bit for $\mathbf{s}$ (and $\mathbf{S}$ later), and needs to include a helper encoding of 1. We let go of those details for a simplified exposition in this overview.

key is bound to a specific circuit, the homomorphic evaluation must be *authenticated* and decryption must implicitly verify that the correct computation is performed.

The other difficulty is that in [BGG+14] homomorphism, it is necessary to know the value being encoded, so evaluating a circuit on $\mathbf{s}$ requires knowing $\mathbf{s}$, which we cannot afford as revealing it would destroy security.

Our Unbounded Homomorphic Evaluation. We achieve unbounded homomorphism for [BGG+14] attribute encoding by combining rounding and (circular-FHE-style) bootstrapping.

Noise Removal. We make rounding *noiseless*. Recall that $\mathbf{e}_{\text{small}}$ in Equation (1) contains both the rounded $\mathbf{e}_{\text{large}}$ and the rounding error. To remove the former, we round when $\|\mathbf{e}_{\text{large}}\|$ is much less than M. To get rid of the latter, we draw inspiration from the learning with rounding (LWR) assumption — instead of taking $\mathbf{s}$ out from rounding, we keep it inside:

$$\begin{aligned}\left\lfloor \frac{(\mathbf{s}^\intercal(\mathbf{A}_C - C(\mathbf{x})\mathbf{G}) + \mathbf{e}_{\text{large}}^\intercal) \bmod q}{M} \right\rceil &= \left(\left\lfloor \frac{(\mathbf{s}^\intercal \mathbf{A}_C + \mathbf{e}_{\text{large}}^\intercal) \bmod q}{M} \right\rceil - C(\mathbf{x})\mathbf{s}^\intercal \mathbf{G}_{\text{small}} \right) \bmod \frac{q}{M} \\ \text{(with high probability)} \quad &= \left(\left\lfloor \frac{\mathbf{s}^\intercal \mathbf{A}_C \bmod q}{M} \right\rceil - C(\mathbf{x})\mathbf{s}^\intercal \mathbf{G}_{\text{small}} \right) \bmod \frac{q}{M}.\end{aligned}$$

For the first equality to hold, we need M to be a power of two so that $M \mid \mathbf{G}$ (ignoring the small entries in $\mathbf{G}$ for now) and $C(\mathbf{x}), \mathbf{s}, \frac{\mathbf{G}}{M}$ are integral hence can be freely taken out of rounding. The second equality holds when $\mathbf{e}_{\text{large}}$ does not introduce carrying/borrowing. Intuitively, $\mathbf{s}^\intercal \mathbf{A}_C$ for $\mathbf{A}_C$ arising from homomorphic evaluation should just be random, hence is likely to be far away from the boundary of rounding jumps.

Rounding transfers the encoding to a smaller modulus $\frac{q}{M}$. We restore the large modulus q by multiplication:

$$M \left\lfloor \frac{(\mathbf{s}^\intercal(\mathbf{A}_C - C(\mathbf{x})\mathbf{G}) + \mathbf{e}_{\text{large}}^\intercal) \bmod q}{M} \right\rceil = \left(M \left\lfloor \frac{\mathbf{s}^\intercal \mathbf{A}_C \bmod q}{M} \right\rceil - C(\mathbf{x})\mathbf{s}^\intercal \cdot M\mathbf{G}_{\text{small}} \right) \bmod q.$$

We want to recover $\mathbf{G}$, but $M\mathbf{G}_{\text{small}}$ only contains the large powers of two in $\mathbf{G}$. We also glossed over the issue of entries in $\mathbf{G}$ less than M when rounding. The fix is to rearrange $\mathbf{G}$ by small and large portions $\mathbf{G}_{\text{L}}, \mathbf{G}_{\text{R}}$ and amplify $\mathbf{G}_{\text{L}}$ by M before rounding. Let $\mathbf{Q}$ be the permutation matrix such that $(\mathbf{G}_{\text{L}}, \mathbf{G}_{\text{R}})\mathbf{Q} = \mathbf{G}$, then the rounding procedure is

$$\left\lfloor \frac{\mathbf{s}^\intercal(\mathbf{A}_C - C(\mathbf{x})\mathbf{G})\mathbf{G}^{-1}(M\mathbf{G}_{\text{L}}, \mathbf{G}_{\text{R}}) \bmod q}{M} \right\rceil \begin{pmatrix} \mathbf{I} & \\ & M\mathbf{I} \end{pmatrix} \mathbf{Q}.$$

The matrix multiplying $C(\mathbf{x})\mathbf{s}^\intercal$ is (note that the quantity being rounded is integral)

$$\left\lfloor \frac{\mathbf{G}\mathbf{G}^{-1}(M\mathbf{G}_{\text{L}}, \mathbf{G}_{\text{R}})}{M} \right\rceil \begin{pmatrix} \mathbf{I} & \\ & M\mathbf{I} \end{pmatrix} \mathbf{Q} = \left(\mathbf{G}_{\text{L}}, \frac{\mathbf{G}_{\text{R}}}{M} \right) \begin{pmatrix} \mathbf{I} & \\ & M\mathbf{I} \end{pmatrix} \mathbf{Q} = (\mathbf{G}_{\text{L}}, \mathbf{G}_{\text{R}})\mathbf{Q} = \mathbf{G},$$

and the other part is a *rounded pad* dependent on $\mathbf{A}_C$ and $\mathbf{s}$,

$$\mathsf{RndPad}_{\mathbf{A}_C}(\mathbf{s}) = \left\lfloor \frac{\mathbf{s}^\intercal \mathbf{A}_C \mathbf{G}^{-1}(M\mathbf{G}_{\text{L}}, \mathbf{G}_{\text{R}}) \bmod q}{M} \right\rceil \begin{pmatrix} \mathbf{I} & \\ & M\mathbf{I} \end{pmatrix} \mathbf{Q}.$$

Bootstrapping. After noise removal (rounding and modulus restoration), the encoding is no longer amenable to [BGG+14] homomorphism. If we can compute $\mathbf{s}^\intercal \mathbf{A}'_C - \mathsf{RndPad}_{\mathbf{A}_C}(\mathbf{s})$ for some (other) public matrix $\mathbf{A}'_C$ related to C, we will be able to continue homomorphic evaluation using

$$\mathbf{s}^\intercal \mathbf{A}'_C - \mathsf{RndPad}_{\mathbf{A}_C}(\mathbf{s}) + (\mathsf{RndPad}_{\mathbf{A}_C}(\mathbf{s}) - C(\mathbf{x})\mathbf{s}^\intercal \mathbf{G}) = \mathbf{s}^\intercal(\mathbf{A}'_C - C(\mathbf{x})\mathbf{G}).$$

This naturally calls for homomorphic evaluation on $\mathbf{s}$. But again, we cannot perform [BGG+14] evaluation on $\mathbf{s}$ as it must not be known to the evaluator for security. Instead, we circularly encrypt $\mathbf{s}$, perform [BGG+14] evaluation on its ciphertext $\mathbf{S}$ under itself, and employ the dual-use technique of [BTVW17] for *automatic decryption*.

In the FHE scheme of [GSW13], the secret key is $\mathbf{s}^\intercal$, a ciphertext is a matrix $\mathbf{C}$, and decryption is noisy linear. Suppose $\mathbf{C}$ encrypts $\mathbf{x}^\intercal$, then $\mathbf{s}^\intercal \mathbf{C} = \mathbf{x}^\intercal + \text{noise}$.[3] The technique of [BTVW17] is to use the same $\mathbf{s}$ for attribute encoding and FHE secret key. We additionally publish

$$\text{(circular encryption)} \quad \mathbf{S} = \mathsf{hct}(\mathbf{s}), \qquad \text{(circular encoding)} \quad \mathbf{s}^\intercal(\mathbf{A}_{\mathrm{circ}} - \mathbf{S} \otimes \mathbf{G}).$$

Given $\mathbf{A}_C$, we evaluate $\mathsf{HEval}_{\mathsf{RndPad}_{\mathbf{A}_C}}$ on attribute $\mathbf{S}$, which yields the desired

$$\begin{aligned}
& \mathbf{s}^\intercal(\mathbf{A}_{\mathrm{circ}} - \mathbf{S} \otimes \mathbf{G})\mathbf{H}_{\mathsf{HEval}_{\mathsf{RndPad}_{\mathbf{A}_C}},\mathbf{S}} && \mathbf{A}_{\mathrm{circ}}\mathbf{H}_{\mathsf{HEval}_{\mathsf{RndPad}_{\mathbf{A}_C}}} \uparrow \\
\text{([BGG+14])} \quad &= \mathbf{s}^\intercal \mathbf{A}_{\mathrm{circ}} \mathbf{H}_{\mathsf{HEval}_{\mathsf{RndPad}_{\mathbf{A}_C}}} - \mathbf{s}^\intercal \mathsf{HEval}_{\mathsf{RndPad}_{\mathbf{A}_C}}(\mathbf{S}) = \mathbf{s}^\intercal \mathbf{A}'_C - \mathbf{s}^\intercal \mathsf{HEval}_{\mathsf{RndPad}_{\mathbf{A}_C}}(\mathsf{hct}(\mathbf{s})) \\
\text{([GSW13])} \quad &= \mathbf{s}^\intercal \mathbf{A}'_C - \mathbf{s}^\intercal \mathsf{hct}(\mathsf{RndPad}_{\mathbf{A}_C}(\mathbf{s})) = \mathbf{s}^\intercal \mathbf{A}'_C - \mathsf{RndPad}_{\mathbf{A}_C}(\mathbf{s}).
\end{aligned}$$

Note that FHE decryption happens automatically when an FHE ciphertext is evaluated on the attribute. The noise in $\mathbf{s}^\intercal \mathbf{A}'_C - \mathsf{RndPad}_{\mathbf{A}_C}(\mathbf{s})$, thus that in $\mathbf{s}^\intercal(\mathbf{A}'_C - C(\mathbf{x})\mathbf{G})$, only grows with the depth of $\mathsf{HEval}_{\mathsf{RndPad}_{\mathbf{A}_C}}$, which is fixed and does not grow with that of C.

Summary. By removing noise and bootstrapping after every gate, we keep the noise at a fixed level hence achieve unbounded homomorphic evaluation. Abstractly, the procedure gives rise to two efficient algorithms (corresponding to $\mathbf{H}_C, \mathbf{H}_{C,\mathbf{x}}$ of [BGG+14])

$$\mathsf{UEvalC}(\mathbf{A}_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, C) = \mathbf{A}_C, \qquad \mathsf{UEvalCX}(\mathbf{A}_{\mathrm{attr}}, \mathbf{c}^\intercal_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, \mathbf{c}^\intercal_{\mathrm{attr}}, C, \mathbf{x}, \mathbf{S}) = \mathbf{c}^\intercal_C,$$

satisfying (with high probability)

$$\mathsf{UEvalCX}(\mathbf{A}_{\mathrm{attr}}, \mathbf{s}^\intercal(\mathbf{A}_{\mathrm{attr}} - \mathbf{x} \otimes \mathbf{G}), \mathbf{A}_{\mathrm{circ}}, \mathbf{s}^\intercal(\mathbf{A}_{\mathrm{attr}} - \mathbf{S} \otimes \mathbf{G}), C, \mathbf{x}, \mathbf{S}) = \mathbf{s}^\intercal(\overset{\mathsf{UEvalC}(\mathbf{A}_{\mathrm{attr}},\mathbf{A}_{\mathrm{circ}},C)}{\uparrow \mathbf{A}_C} - C(\mathbf{x})\mathbf{G})$$

if $\mathbf{S}$ is a circular [GSW13] encryption of $\mathbf{s}$.

One More Thing. We highlight a technicality here. Recall that noise removal relies on $\mathbf{s}^\intercal \mathbf{A}_C$ being far away from the boundary of rounding jumps. The intuition was $\mathbf{s}^\intercal \mathbf{A}_C$ is (pseudo-)random, but in reality, $\mathbf{s}^\intercal \mathbf{A}_C = \mathbf{s}^\intercal \mathbf{A} \mathbf{H}_C$ for some low-norm $\mathbf{H}_C$ dependent on $\mathbf{A}$. Although $\mathbf{s}^\intercal \mathbf{A}$ is marginally random, $\mathbf{H}_C$ has complicated dependency (described by C) on it, so the distribution of $\mathbf{s}^\intercal \mathbf{A}_C$ is not easy to work with. For our scheme, the last entry of $\mathbf{s}$ is always -1, in the worst scenario without considering the exact structure of $\mathbf{H}_C$, for every $\mathbf{z}$, there exists a function $H : \mathbf{A} \mapsto \mathbf{H}$ outputting a low-norm matrix that makes $\mathbf{s}^\intercal \mathbf{A} H(\mathbf{A}) = \mathbf{z}^\intercal$ happen with overwhelming probability.[4]

[3] The formulation in [GSW13] is different. For their bit encryption, we can regard $x\mathbf{s}^\intercal \mathbf{G}$ as the plaintext $\mathbf{x}^\intercal$. It extends to any vector $\mathbf{x}^\intercal$ and to ciphertexts homomorphically evaluated for vector-valued circuits. (See Section 2.5.)

[4] For most $\mathbf{z}$, this H is not known (nor believed) to be efficiently computable.

We take advantage of the fact that the low-norm $\mathbf{H}_C$ is efficiently computable. Our technique is to introduce (another, independent) noise $\mathbf{e}$. Under the LWE assumption, $(\mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T})\mathbf{H}_C$ is indistinguishable from $\boldsymbol{\delta}^\mathsf{T}\mathbf{H}_C$ (for $\boldsymbol{\delta}$ random and independent of $\mathbf{H}_C$). The latter can be shown to be far away from the boundary of rounding jumps. (See Lemma 8.)

Alternatively, the issue can be worked around by adding a random shift to the value before rounding. The shifts can be generated using a PRF key.

Laconic Function Evaluation. Putting things together, we present our AB-LFE construction using unbounded homomorphic evaluation. Its components are

$$\mathsf{crs} = (\mathbf{A}_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}), \qquad \mathsf{digest}_C = \mathbf{A}_C, \qquad \mathsf{ct}_{\mathbf{x}}(\mu) = (\mathbf{x}, \mathbf{S}, \mathbf{c}_{\mathrm{attr}}, \mathbf{c}_{\mathrm{circ}}, \mathbf{z}, c_{\mathrm{msg}}),$$

where $\mathbf{S} = \begin{pmatrix} \overline{\mathbf{A}}_{\mathrm{fhe}} \\ \mathbf{s}^\mathsf{T}\overline{\mathbf{A}}_{\mathrm{fhe}} + \mathbf{e}^\mathsf{T}_{\mathrm{fhe}} \end{pmatrix}\mathbf{R} - \mathbf{s} \otimes \mathbf{G}$ is the circular ciphertext, and

$$\mathbf{c}^\mathsf{T}_{\mathrm{attr}} = \mathbf{s}^\mathsf{T}(\mathbf{A}_{\mathrm{attr}} - \mathbf{x} \otimes \mathbf{G}) + \mathbf{e}^\mathsf{T}_{\mathrm{attr}}, \qquad \mathbf{c}^\mathsf{T}_{\mathrm{circ}} = \mathbf{s}^\mathsf{T}(\mathbf{A}_{\mathrm{circ}} - \mathbf{S} \otimes \mathbf{G}) + \mathbf{e}^\mathsf{T}_{\mathrm{circ}},$$
$$c_{\mathrm{msg}} = \mathbf{s}^\mathsf{T}\mathbf{A}_C\mathbf{G}^{-1}(\mathbf{z}) + e_{\mathrm{msg}} + \mu \cdot \lfloor q/2 \rceil$$

are the attribute/circular/message encodings. To decrypt when $C(\mathbf{x}) = 0$, run $\mathsf{UEvalCX}$ to obtain

$$\mathbf{c}^\mathsf{T}_{C,\mathbf{x}} = \mathbf{s}^\mathsf{T}(\mathbf{A}_C - C(\mathbf{x})\mathbf{G}) = \mathbf{s}^\mathsf{T}\mathbf{A}_C,$$

which can be used to cancel $\mathbf{s}^\mathsf{T}\mathbf{A}_C\mathbf{G}^{-1}(\mathbf{z})$ in c_{msg} for message recovery.

The security proof when $C(\mathbf{x}) = 1$ involves two steps. First, simulate message encoding using

$$\mathbf{s}^\mathsf{T}\mathbf{A}_C\mathbf{G}^{-1}(\mathbf{z}) + \mu \cdot \lfloor q/2 \rceil = \mu \cdot \lfloor q/2 \rceil + \underbrace{\mathbf{c}^\mathsf{T}_{C,\mathbf{x}}\mathbf{G}^{-1}(\mathbf{z})}_{\mathbf{s}^\mathsf{T}(\mathbf{A}_C - C(\mathbf{x})\mathbf{G})\mathbf{G}^{-1}(\mathbf{z})} + \overbrace{C(\mathbf{x})}^{=1} \cdot \mathbf{s}^\mathsf{T}\mathbf{z}.$$

Then, invoke circular LWE on secret $\mathbf{s}$ to hide μ with the pseudorandom pad $\mathbf{s}^\mathsf{T}\mathbf{z}$.

2 Preliminaries

The security parameter is λ, which is omitted for brevity except in definitions. Efficient means probabilistic polynomial-time. Adversaries might also be given $\mathrm{poly}(\lambda)$-bit advice dependent on λ.[5] All circuits are Boolean and use only conjunction, disjunction, and negation gates. For conceptual reasons, the domain/codomain of circuits might be written as any finite set, but pragmatically, the input/output are always encoded as bits. We write $A \xrightarrow{\$} B$ for distributions over B indexed by A, i.e., a randomized function mapping A to B.

Vectors are denoted by boldfaced lowercase letters, and matrices, boldfaced uppercase letters. They are indexed using brackets, not subscripts, so $\mathbf{w}_1, \mathbf{w}_2$ are two vectors, and $\mathbf{W}[i, j]$ is an entry of $\mathbf{W}$. We write $\mathbf{I}_n$ (or simply $\mathbf{I}$) for the $n \times n$ identity matrix, and $\mathbf{0}_{n\times m}$ (or simply $\mathbf{0}$) for the $n \times m$ zero matrix. When the dimension is clear, the standard basis vectors are denoted by $\boldsymbol{\iota}_1, \boldsymbol{\iota}_2, \ldots$, i.e., $\boldsymbol{\iota}_i$ is the i^{th} column of $\mathbf{I}$. We consider the infinity norm and its operator norm:

$$\|\mathbf{w}\| = \max_i |\mathbf{w}[i]|, \quad \|\mathbf{W}\| = \max_i \sum_j |\mathbf{W}[i, j]|.$$

[5] The reductions in this work are advice-preserving.

We strictly follow the convention of vectors being columns. If $\mathbf{w} \in \mathbb{Z}^z$ is a vector, $\|\mathbf{w}^\intercal\|$ is an operator norm and $\|\mathbf{w}^\intercal\| \leq z\|\mathbf{w}\|$. Given an object, we write $\mathsf{bits}(\cdots)$ for its fixed-length bit representation, arrange in a row, i.e., a matrix in $\{0,1\}^{1\times L}$ for some L. For two matrices $\mathbf{A}, \mathbf{B}$ of shapes $n_1 \times m_1$ and $n_2 \times m_2$, their Kronecker product is an $n_1 n_2 \times m_1 m_2$ matrix,

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} \mathbf{A}[1,1]\mathbf{B} & \cdots & \mathbf{A}[1,m_1]\mathbf{B} \\ \vdots & \ddots & \vdots \\ \mathbf{A}[n_1,1]\mathbf{B} & \cdots & \mathbf{A}[n_1,m_1]\mathbf{B} \end{pmatrix}.$$

A useful property is $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}$ whenever all multiplications are compatible.

For $a, b \in \mathbb{R}$, we write $[a, b)$ for $\{\, z \in \mathbb{Z} \mid a \leq z < b \,\}$, i.e., intervals are intersected with $\mathbb{Z}$. Useful facts are $|[a,b)| = \lceil b \rceil - \lceil a \rceil$ for all $a \leq b$, and $2a - 1 < \lceil a \rceil - \lceil -a \rceil < 2a + 1$ for all a. Given a natural number n, we use $[n]$ as a shorthand of $[1, n+1)$. For natural number $q \geq 2$, we denote by $\mathbb{Z}_q$ the integers modulo q. Matrices over $\mathbb{Z}$ are naturally and implicitly mapped to those over $\mathbb{Z}_q$ so that they can be arbitrarily mixed for various operations. For $z \in \mathbb{Z}_q$, the canonical $\mathsf{bits}(z)$ is the binary representation of its smallest non-negative representative (less than q), low to high, potentially with extra trailing (high) zeros. The expression $(z \bmod q)$ denotes the representative in $\left[-\frac{q}{2}, \frac{q}{2}\right)$. We also write $(z \bmod p)$ when p divides q. For $x \in \mathbb{R}$, we define $\lfloor x \rceil$ to be $\left\lfloor x + \frac{1}{2} \right\rfloor$. Bits, remainder, and rounding extend to matrices entry-wise.

Symbols. Table 2 explains select single-letter symbols used in this work.

Table 2. Non-self-explanatory symbols in this work.

symbol	meaning
$\lambda, \beta, \mathcal{A}, \mathcal{S}$	security parameter, challenge bit, adversary, sampler
J, j	key count, index
P, X, Y, μ	ABE predicate, policy set, attribute set, message
y, C, d	policy (abstract), policy circuit (concrete), depth
$x, \mathbf{x}, L, \ell$	attribute (abstract), attribute (concrete), length, index
$\boldsymbol{\iota}, \mathbf{I}, \mathbf{g}, \mathbf{G}$	standard basis, identity matrix, gadget vector, gadget matrix
n, m	LWE dimension, sample count (shape of matrix with trapdoor)
m', q, σ	LWE sample count, modulus, Gaussian error width
θ, B	hardness exponent, error bound
$\mathbf{A}, \overline{\mathbf{A}}, \underline{\mathbf{a}}^\intercal$	matrix without trapdoor, first n rows, last row
$\mathbf{B}, \tau$	matrix with trapdoor, trapdoor of matrix
$\mathbf{p}, \mathbf{P}, \mathbf{k}, \mathbf{K}$	image, many, Gaussian preimage, many
$\mathbf{r}, \mathbf{s}$	circular LWE secret without -1, with -1 $[\mathbf{s}^\intercal = (\mathbf{r}^\intercal, -1)]$
$e, \mathbf{e}, \mathbf{c}, \delta, \boldsymbol{\delta}, \boldsymbol{\Delta}$	error, many, LWE samples, random value, many, many
$\mathbf{R}, \mathbf{X}, \mathbf{C}, \mathbf{S}$	FHE randomness, ciphertexts of $\mathbf{x}$, of $C(\mathbf{x})$, of $\mathbf{s}$
$\mathbf{H}, \mathbf{h}, h$	homomorphic evaluation matrix, column, column function
M, p	rounding resolution (dividing q), generic factor of q
$\mathbf{G}_\mathrm{L}, \mathbf{G}_\mathrm{R}, \mathbf{Q}$	"left" part of $\mathbf{G}$, "right" part, permutation $[(\mathbf{G}_\mathrm{L}, \mathbf{G}_\mathrm{R})\mathbf{Q} = \mathbf{G}]$
$\mathbf{z}$	public vector creating one-time pad

2.1 Laconic Function Evaluation

Definition 1 (LFE [QWW18a])**.** Let $F = \{F_{\lambda,\mathsf{param}^F}\}_{\lambda\in\mathbb{N},\mathsf{param}^F\in\mathsf{Params}^F_\lambda}$ be a family of sets of functions, where each $F_{\lambda,\mathsf{param}^F}$ is a set of functions $\{ f : X_{\mathsf{param}_f} \to \{0,1\}^* \}$, with each X_{param_f} being a set described by param_f (which itself is dependent on f) and $\mathsf{Params}^F = \{\mathsf{Params}^F_\lambda\}_{\lambda\in\mathbb{N}}$ a sequence of function set description sets. A *laconic function evaluation (LFE) scheme for* F consists of four efficient algorithms:

- $\mathsf{GenCRS}(1^\lambda, \mathsf{param}^F)$ takes a function set description $\mathsf{param}^F \in \mathsf{Params}^F_\lambda$ as input, and outputs a common reference string crs.
- $\mathsf{Compress}(1^\lambda, \mathsf{crs}, f)$ takes as input crs and some $f \in F_{\lambda,\mathsf{param}^F}$. It outputs a deterministic param_f (fully determined by $\lambda, \mathsf{param}^F, f$) and a potentially randomized digest_f. If the algorithm is randomized, digest_f must be a pair whose first element is the random tape prefix read by that invocation of $\mathsf{Compress}$ (i.e., the algorithm is public-coin).
- $\mathsf{Enc}(1^\lambda, \mathsf{crs}, \mathsf{param}_f, \mathsf{digest}_f, x)$ takes as input crs, param_f, digest_f, and some $x \in X_{\mathsf{param}_f}$. It outputs a ciphertext ct of x.
- $\mathsf{Dec}(1^\lambda, \mathsf{crs}, f, \mathsf{digest}_f, \mathsf{ct})$ is supposed to compute $f(x)$.

An LFE scheme for *circuits of unbounded depth* is one with

$$\mathsf{Params}^F_\lambda = \{ 1^L \mid L \in \mathbb{N} \}, \qquad F_{\lambda,1^L} = \{ \text{circuit } C \mid C : \{0,1\}^L \to \{0,1\}^* \},$$
$$\mathsf{param}_C = L \text{ for } C : \{0,1\}^L \to \{0,1\}^*, \qquad X_L = \{0,1\}^L.$$

An *AB-LFE scheme for circuits of unbounded depth* is one with[6]

$$\mathsf{Params}^F_\lambda = \{ 1^L \mid L \in \mathbb{N} \}, \qquad F_{\lambda,1^L} = \{ f_C \mid \text{circuit } C : \{0,1\}^L \to \{0,1\}^{1\times *} \},$$
$$\mathsf{param}_C = (L, L') \text{ for } C : \{0,1\}^L \to \{0,1\}^{1\times L'}, \qquad X_{L,L'} = \{0,1\}^L \times \{0,1\}^{L'},$$
$$f_C(\mathbf{x}, \boldsymbol{\mu}) = (\mathbf{x}, \boldsymbol{\mu}^\mathsf{T} \wedge \neg C(\mathbf{x})) \text{ for } C : \{0,1\}^L \to \{0,1\}^{L'}, \mathbf{x} \in \{0,1\}^L, \boldsymbol{\mu} \in \{0,1\}^{L'}.$$

Here, "$\wedge$" (resp. "$\neg$") is bitwise conjunction (resp. negation).

Offline-Online Encryption. We consider LFE whose encryption can be decomposed into two phases. The offline phase only depends on crs and x and outputs a partial ciphertext and a (hopefully small) state. The online phase only depends on param_f, digest_f, and the state (not directly crs, x, part of which can be passed in the state if needed), and it completes the ciphertext.

Definition 2 (LFE two-phase encryption)**.** Given an LFE scheme (Definition 1), consider two efficient algorithms:

- $\mathsf{EncX}(1^\lambda, \mathsf{crs}, x)$ takes crs, x as input. It outputs $\mathsf{ct}_{\mathrm{off}}$ and st.
- $\mathsf{EncD}(1^\lambda, \mathsf{st}, \mathsf{param}_f, \mathsf{digest}_f)$ takes $\mathsf{st}, \mathsf{param}_f, \mathsf{digest}_f$ as input. It outputs $\mathsf{ct}_{\mathrm{on}}$.

The scheme has *two-phase encryption* of $(\mathsf{EncX}, \mathsf{EncD})$ if Enc operates as follows:

1. Run $(\mathsf{ct}_{\mathrm{off}}, \mathsf{st}) \xleftarrow{\$} \mathsf{EncX}(1^\lambda, \mathsf{crs}, x)$.

[6] Strictly speaking, we should say param_{f_C}, but represent f_C by C, so param_C is a reasonable notation.

2. Run $\mathsf{ct}_{\mathrm{on}} \xleftarrow{\$} \mathsf{EncD}(1^\lambda, \mathsf{st}, \mathsf{param}_f, \mathsf{digest}_f)$.

3. Output $(\mathsf{ct}_{\mathrm{off}}, \mathsf{ct}_{\mathrm{on}})$.

Definition 3 (LFE correctness)**.** Given an LFE scheme (Definition 1), consider $\mathsf{Exp}^{\mathrm{adptv}}_{\mathrm{LFE}\checkmark}(1^\lambda, \mathcal{A})$:

- **Setup.** Launch $\mathcal{A}(1^\lambda)$ and receive $\mathsf{param}^F \in \mathsf{Params}^F_\lambda$ from it. Run $\mathsf{crs} \xleftarrow{\$} \mathsf{GenCRS}(1^\lambda, \mathsf{param}^F)$ and send crs to $\mathcal{A}$.

- **Query.** $\mathcal{A}$ chooses $f \in F_{\lambda,\mathsf{param}^F}$. Run $(\mathsf{param}_f, \mathsf{digest}_f) \xleftarrow{\$} \mathsf{Compress}(1^\lambda, \mathsf{crs}, f)$ and send digest_f to $\mathcal{A}$.

- **Challenge.** $\mathcal{A}$ chooses $x \in X_{\mathsf{param}_f}$. Run $\mathsf{ct} \xleftarrow{\$} \mathsf{Enc}(1^\lambda, \mathsf{crs}, \mathsf{param}_f, \mathsf{digest}_f, x)$.

- **Test.** Run $y \xleftarrow{\$} \mathsf{Dec}(1^\lambda, \mathsf{crs}, f, \mathsf{digest}_f, \mathsf{ct})$. If $y \neq f(x)$, the output of the experiment is 1. Otherwise, the output is 0.

$\mathsf{Exp}^{f\text{-sel}}_{\mathrm{LFE}\checkmark}$ is modified from $\mathsf{Exp}^{\mathrm{adptv}}_{\mathrm{LFE}\checkmark}$ by requiring $\mathcal{A}$ to choose f during **Setup** together with param^F (before it receives crs).

The scheme is *computationally (resp$_1$. statistically) adaptively (resp$_2$. f-selectively) correct* (default is computational)[7] if $\Pr[\mathsf{Exp}^{\mathrm{adptv}}_{\mathrm{LFE}\checkmark}(1^\lambda, \mathcal{A}) \to 1]$ (resp$_2$. for $\mathsf{Exp}^{f\text{-sel}}_{\mathrm{LFE}\checkmark}$) is negligible for all efficient $\mathcal{A}$ (resp$_1$. all, potentially inefficient, $\mathcal{A}$ whose total output length $(|\mathsf{param}^F| + |f| + |x|)$ is poly(λ)-bounded).

Definition 4 (LFE security)**.** Given LFE scheme (Definition 1) and a simulator $\widetilde{\mathsf{Enc}}$, consider $\mathsf{Exp}^{\mathrm{adptv},\beta}_{\mathrm{LFE}}(1^\lambda, \mathcal{A})$ for $\beta \in \{0,1\}$:

- **Setup.** Launch $\mathcal{A}(1^\lambda)$ and receive $\mathsf{param}^F \in \mathsf{Params}^F_\lambda$ from it. Run $\mathsf{crs} \xleftarrow{\$} \mathsf{GenCRS}(1^\lambda, \mathsf{param}^F)$ and send crs to $\mathcal{A}$.

- **Query.** $\mathcal{A}$ chooses $f \in F_{\lambda,\mathsf{param}^F}$. Run $(\mathsf{param}_f, \mathsf{digest}_f) \xleftarrow{\$} \mathsf{Compress}(1^\lambda, \mathsf{crs}, f)$ and send digest_f to $\mathcal{A}$.

- **Challenge.** $\mathcal{A}$ chooses $x \in X_{\mathsf{param}_f}$. Run

$$\mathsf{ct}_0 \xleftarrow{\$} \mathsf{Enc}(1^\lambda, \mathsf{crs}, \mathsf{param}_f, \mathsf{digest}_f, x), \qquad \mathsf{ct}_1 \xleftarrow{\$} \widetilde{\mathsf{Enc}}(1^\lambda, \mathsf{crs}, f, \mathsf{digest}_f, f(x)),$$

  and send ct_β to $\mathcal{A}$.

- **Guess.** $\mathcal{A}$ outputs a bit $\beta' \in \{0,1\}$, which is the output of the experiment.

$\mathsf{Exp}^{x\text{-sel}}_{\mathrm{LFE}}$ (resp. $\mathsf{Exp}^{\text{very-sel}}_{\mathrm{LFE}}$) is modified from $\mathsf{Exp}^{\mathrm{adptv}}_{\mathrm{LFE}}$ by requiring $\mathcal{A}$ to choose x (resp. x, f) during **Setup** together with param^F (before it receives crs).

The scheme is *adaptively (resp. x-selectively; very selectively) secure* if there exists an efficient simulator $\widetilde{\mathsf{Enc}}$ such that $\mathsf{Exp}^{\mathrm{adptv},0}_{\mathrm{LFE}} \approx \mathsf{Exp}^{\mathrm{adptv},1}_{\mathrm{LFE}}$ (resp. for $\mathsf{Exp}^{x\text{-sel}}_{\mathrm{LFE}}$; for $\mathsf{Exp}^{\text{very-sel}}_{\mathrm{LFE}}$).

[7] The strength of correctness is not the same as the nature of its proof. See Footnote 9.

2.2 Partially Hiding Functional Encryption

Definition 5 (PHFE)**.** Let $\Phi = \{\varphi_{\lambda,\mathsf{param}}\}_{\lambda\in\mathbb{N},\mathsf{param}\in\mathsf{Params}_\lambda}$ be a family of functionalities, where each $\varphi_{\lambda,\mathsf{param}}$ is a function $F_{\lambda,\mathsf{param}} \times X_{\lambda,\mathsf{param}} \times Y_{\lambda,\mathsf{param}} \to \{0,1\}^*$ and $\mathsf{Params} = \{\mathsf{Params}_\lambda\}_{\lambda\in\mathbb{N}}$ is a sequence of functionality description sets. A *partially hiding functional encryption (PHFE) scheme for* Φ consists of four efficient algorithms:

- $\mathsf{Setup}(1^\lambda, \mathsf{param})$ takes the functionality description $\mathsf{param} \in \mathsf{Params}_\lambda$ as input, and outputs a pair of master public/secret keys $(\mathsf{mpk}, \mathsf{msk})$.
- $\mathsf{KeyGen}(1^\lambda, \mathsf{msk}, f)$ takes as input msk and some $f \in F_{\lambda,\mathsf{param}}$. It outputs a secret key sk for f.
- $\mathsf{Enc}(1^\lambda, \mathsf{mpk}, x, y)$ takes as input mpk, public input $x \in X_{\lambda,\mathsf{param}}$, and private input $y \in Y_{\lambda,\mathsf{param}}$. It outputs a ciphertext ct of y tied to x.
- $\mathsf{Dec}(1^\lambda, \mathsf{mpk}, f, \mathsf{sk}, x, \mathsf{ct})$ is supposed to compute $\varphi_{\lambda,\mathsf{param}}(f, x, y)$.

A *key-policy (KP) ABE scheme for circuits of unbounded depth* is one with

$$\mathsf{Params}_\lambda = \{\, 1^L \mid L \in \mathbb{N} \,\}, \qquad F_{\lambda,1^L} = \{\, \text{circuit } C \mid C : \{0,1\}^L \to \{0,1\} \,\},$$
$$\mathbf{x} \in X_{\lambda,1^L} = \{0,1\}^L, \qquad \mu \in Y_{\lambda,1^L} = \{0,1\}, \qquad \varphi_{\lambda,1^L}(C,\mathbf{x},\mu) = \begin{cases} \mu, & \text{if } C(\mathbf{x}) = 0; \\ \bot, & \text{otherwise.} \end{cases}$$

A *KP-ABE scheme for circuits of unbounded depth and input length* is one with[8]

$$\mathsf{Params}_\lambda = \{\bot\}, \qquad F_{\lambda,\bot} = \{\, \text{circuit } C \mid C : \{0,1\}^L \to \{0,1\} \text{ for some } L < 2^\lambda \,\},$$
$$\mathbf{x} \in X_{\lambda,\bot} = \{0,1\}^{<2^\lambda}, \quad \mu \in Y_{\lambda,\bot} = \{0,1\}, \quad \varphi_{\lambda,\bot}(C,\mathbf{x},\mu) = \begin{cases} \mu, & \text{if } |\mathbf{x}| \geq L \text{ and } C(\mathbf{x}) = 0; \\ \bot, & \text{otherwise.} \end{cases}$$

A *functional encryption (FE) scheme for circuits of unbounded depth* is one with

$$\mathsf{Params}_\lambda = \{\, (1^L, 1^{L'}) \mid L, L' \in \mathbb{N} \,\}, \qquad F_{\lambda,1^L,1^{L'}} = \{\, \text{circuit } C \mid C : \{0,1\}^L \to \{0,1\}^{L'} \,\},$$
$$X_{\lambda,1^L} = \{\bot\}, \qquad \mathbf{x} \in Y_{\lambda,1^L} = \{0,1\}^L, \qquad \varphi_{\lambda,1^L}(C,\bot,\mathbf{x}) = C(\mathbf{x}).$$

Definition 6 (PHFE correctness)**.** Given a PHFE scheme (Definition 5), consider $\mathsf{Exp}^{\mathrm{adptv+}}_{\mathrm{PHFE}\checkmark}(1^\lambda, \mathcal{A})$:

- **Setup.** Launch $\mathcal{A}(1^\lambda)$ and receive $\mathsf{param} \in \mathsf{Params}_\lambda$ from it. Set up the scheme by running $(\mathsf{mpk}, \mathsf{msk}) \xleftarrow{\$} \mathsf{Setup}(1^\lambda, \mathsf{param})$ and send $(\mathsf{mpk}, \mathsf{msk})$ to $\mathcal{A}$.
- **Query I.** Repeat the following for arbitrarily many rounds determined by $\mathcal{A}$. In each round, $\mathcal{A}$ chooses $f_j \in F_{\lambda,\mathsf{param}}$. Run $\mathsf{sk}_j \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda, \mathsf{msk}, f_j)$ and send sk_j to $\mathcal{A}$.
- **Challenge.** $\mathcal{A}$ chooses $x \in X_{\lambda,\mathsf{param}}$ and $y \in Y_{\lambda,\mathsf{param}}$. Run $\mathsf{ct} \xleftarrow{\$} \mathsf{Enc}(1^\lambda, \mathsf{mpk}, x, y)$ and send ct to $\mathcal{A}$.
- **Query II.** Same as **Query I**.

[8] Unlike the case of monotone functions in many pairing-based schemes, here it is important to check the attribute length and the circuit input length so that security can be maintained for length-mismatching policy-attribute pairs. There are also variants where decryption is allowed if and only if the lengths match exactly, or where both the attribute and the policy circuit specify an index set over which they are defined and decryption considers the inclusion relation between the two sets.

- **Test.** Run $z_j \xleftarrow{\$} \mathsf{Dec}(1^\lambda, \mathsf{mpk}, f_j, \mathsf{sk}_j, x, \mathsf{ct})$ for all j. If $z_j \neq \varphi_{\lambda,\mathsf{param}}(f_j, x, y)$ for some j, the output of the experiment is 1. Otherwise, the output is 0.

Variants of $\mathsf{Exp}^{\mathrm{adptv+}}_{\mathrm{PHFE}\checkmark}$ are "xy-sel+", "f-sel+", "adptv", "xy-sel", "f-sel", "very-sel", where "+" (strong) means msk is given (default is non-strong) and "xy-sel" (resp. f-sel; very-sel) means (x, y) (resp. all f_j's; (x, y) and all f_j's) must be chosen together with param (before mpk is generated).

The scheme is *computationally (resp. statistically)* which*-correct* (default is computational)[9] if $\Pr[\mathsf{Exp}^{\mathsf{which}}_{\mathrm{PHFE}\checkmark}(1^\lambda, \mathcal{A}) \to 1]$ is negligible for for all efficient $\mathcal{A}$ (resp. all, potentially inefficient, $\mathcal{A}$ whose total output length ($|\mathsf{param}| + \sum_j |f_j| + |x| + |y|$) is poly($\lambda$)-bounded).

Definition 7 (PHFE security)**.** Given an PHFE scheme (Definition 5) and a stateful simulator Sim, consider $\mathsf{Exp}^{\mathrm{adptv},\beta}_{\mathrm{PHFE}}(1^\lambda, \mathcal{A})$ for $\beta \in \{0, 1\}$:

- **Setup.** Launch $\mathcal{A}(1^\lambda)$ and receive $\mathsf{param} \in \mathsf{Params}_\lambda$ from it. Run

$$\begin{aligned} (\mathsf{mpk}, \mathsf{msk}) &\xleftarrow{\$} \mathsf{Setup}(1^\lambda, \mathsf{param}), && \text{if } \beta = 0; \\ \mathsf{mpk} &\xleftarrow{\$} \mathsf{Sim}(1^\lambda, \mathsf{param}), && \text{if } \beta = 1; \end{aligned}$$

  and send mpk to $\mathcal{A}$.

- **Query I.** Repeat the following for arbitrarily many rounds determined by $\mathcal{A}$. In each round, $\mathcal{A}$ chooses $f_j \in F_{\lambda,\mathsf{param}}$. Run

$$\begin{aligned} \mathsf{sk}_j &\xleftarrow{\$} \mathsf{KeyGen}(1^\lambda, \mathsf{msk}, f_j), && \text{if } \beta = 0; \\ \mathsf{sk}_j &\xleftarrow{\$} \mathsf{Sim}(f_j), && \text{if } \beta = 1; \end{aligned}$$

  and send sk_j to $\mathcal{A}$.

- **Challenge.** $\mathcal{A}$ chooses $x \in X_{\lambda,\mathsf{param}}$ and $y \in Y_{\lambda,\mathsf{param}}$. Run

$$\begin{aligned} \mathsf{ct} &\xleftarrow{\$} \mathsf{Enc}(1^\lambda, \mathsf{mpk}, x, y), && \text{if } \beta = 0; \\ \mathsf{ct} &\xleftarrow{\$} \mathsf{Sim}(x, \{\varphi_{\lambda,\mathsf{param}}(f_j, x, y)\}_{j\text{'s so far}}), && \text{if } \beta = 1; \end{aligned}$$

  and send ct to $\mathcal{A}$.

- **Query II.** Same as **Query I** except when $\beta = 1$, the secret key is generated as

$$\mathsf{sk}_j \xleftarrow{\$} \mathsf{Sim}(f_j, \varphi_{\lambda,\mathsf{param}}(f_j, x, y)).$$

- **Guess.** $\mathcal{A}$ outputs a bit $\beta' \in \{0, 1\}$, which is the output of the experiment.

Variants of $\mathsf{Exp}^{\mathrm{adptv}}_{\mathrm{PHFE}}$ are "xy-sel", "f-sel", "very-sel", "1-", "ABE", where "xy-sel" (resp. f-sel; very-sel) means (x, y) (resp. all f_j's; (x, y) and all f_j's) must be chosen together with param (before mpk is generated), "1-" (1-key) means there is at most one f_j, and "ABE" (constrained; only applicable to and default for ABE) means $\varphi_{\lambda,\mathsf{param}}(f_j, x, y) = \perp$ must be satisfied for all j.

The scheme is *[1-key] [constrainedly]* which*-(simulation-)secure* if there exists some efficient[10] simulator such that $\mathsf{Exp}^{\mathsf{which},0}_{\mathrm{PHFE}} \approx \mathsf{Exp}^{\mathsf{which},1}_{\mathrm{PHFE}}$ [with subscript "1-" or "ABE", e.g., for $\mathsf{Exp}^{\mathsf{which}}_{\text{1-ABE}}$].

[9] The strength of correctness is not the same as the nature of its proof. Since there is no interaction in $\mathsf{Exp}^{\text{very-sel}}_{\mathrm{PHFE}\checkmark}$, computational very selective correctness against *non-uniform* adversaries is equivalent to statistical very selective correctness, and its proof might well rely on non-uniform computational hardness.

[10] We require that the simulator make $\mathsf{Exp}^{\mathsf{which},1}_{\mathrm{PHFE}}$ halt in polynomial time (counting the time of the adversary and that to generate keys and ciphertexts) whenever $\mathsf{Exp}^{\mathsf{which},0}_{\mathrm{PHFE}}$ does so. In case a stateful machine is modeled as passing the state as input, this prevents the simulator from gaining too much time by stretching its state (e.g., making it twice long in each round) across polynomially many rounds.

2.3 Reusable Garbled Circuits

Definition 8 (reusable garbling)**.** A *reusable garbling scheme* consists of three efficient algorithms:

- $\mathsf{Garble}(1^\lambda, C)$ takes a circuit $C : \{0,1\}^L \to \{0,1\}^{L'}$ as input. It outputs a garbled circuit $\widehat{C}$ and a public key pk.
- $\mathsf{Enc}(1^\lambda, \mathsf{pk}, \mathbf{x})$ takes as input pk and an input $\mathbf{x} \in \{0,1\}^L$. It outputs an encoding $\widehat{\mathbf{x}}$.
- $\mathsf{Eval}(1^\lambda, C, \widehat{C}, \mathsf{pk}, \widehat{\mathbf{x}})$ takes $C, \widehat{C}, \mathsf{pk}, \widehat{\mathbf{x}}$ as input. It is supposed to compute $C(\mathbf{x})$.

Definition 9 (garbling correctness)**.** Given a reusable garbling scheme (Definition 8), consider $\mathsf{Exp}_{\mathrm{GC}\checkmark}(1^\lambda, \mathcal{A})$:

- **Setup.** Launch $\mathcal{A}(1^\lambda)$ and receive a circuit $C : \{0,1\}^L \to \{0,1\}^{L'}$ from it. Set up the scheme by running $(\widehat{C}, \mathsf{pk}) \xleftarrow{\$} \mathsf{Garble}(1^\lambda, C)$ and send $(\widehat{C}, \mathsf{pk})$ to $\mathcal{A}$.
- **Challenge.** $\mathcal{A}$ chooses $\mathbf{x} \in \{0,1\}^L$. Run $\widehat{\mathbf{x}} \xleftarrow{\$} \mathsf{Enc}(1^\lambda, \mathsf{pk}, \mathbf{x})$.
- **Test.** Run $z \xleftarrow{\$} \mathsf{Eval}(1^\lambda, C, \widehat{C}, \mathsf{pk}, \widehat{\mathbf{x}})$. If $z \neq C(\mathbf{x})$, the experiment outputs 1. Otherwise, the output is 0.

The scheme is *computationally (resp. statistically) correct* (default is computational)[11] if for for all efficient $\mathcal{A}$ (resp. all, potentially inefficient, $\mathcal{A}$ whose total output length $(|C| + |\mathbf{x}|)$ is $\mathrm{poly}(\lambda)$-bounded), $\Pr[\mathsf{Exp}_{\mathrm{GC}\checkmark}(1^\lambda, \mathcal{A}) \to 1]$ is negligible.

Definition 10 (garbling security)**.** A reusable garbling scheme (Definition 8) is *selectively secure* if there exists an efficient simulator $\widetilde{\mathsf{Enc}}$ such that $\mathsf{Exp}^0_{\mathrm{GC}} \approx \mathsf{Exp}^1_{\mathrm{GC}}$, where $\mathsf{Exp}^\beta_{\mathrm{GC}}(1^\lambda, \mathcal{A})$ proceeds as follows:

- **Challenge.** Launch $\mathcal{A}(1^\lambda)$ and receive from it $C, \mathbf{x}$. Run

$$(\widehat{C}, \mathsf{pk}) \xleftarrow{\$} \mathsf{Garble}(1^\lambda, C), \qquad \widehat{\mathbf{x}}_0 \xleftarrow{\$} \mathsf{Enc}(1^\lambda, \mathsf{pk}, \mathbf{x}), \qquad \widehat{\mathbf{x}}_1 \xleftarrow{\$} \widetilde{\mathsf{Enc}}(1^\lambda, C, \mathsf{pk}, C(\mathbf{x})),$$

  and send $\widehat{C}, \mathsf{pk}, \widehat{\mathbf{x}}_\beta$ to $\mathcal{A}$.
- **Guess.** $\mathcal{A}$ outputs a bit $\beta' \in \{0,1\}$, which is the output of the experiment.

2.4 Lattices

Let $n, m \geq 1$ and $q \geq 2$ be integers such that $\log_2 q \leq \frac{m}{n+1} \in \mathbb{Z}$. Let

$$\mathbf{g} = (2^0, 2^1, \ldots, 2^{\frac{m}{n+1}-1})^\mathsf{T}, \qquad \mathbf{G} = \mathbf{I}_{n+1} \otimes \mathbf{g}^\mathsf{T} = \begin{pmatrix} \overline{\mathbf{G}} \\ \boldsymbol{\iota}^\mathsf{T}_{n+1} \otimes \mathbf{g}^\mathsf{T} \end{pmatrix}$$

be the gadget vector and the gadget matrix, with $\overline{\mathbf{G}}$ being the first n rows of $\mathbf{G}$. For $\mathbf{p} \in \mathbb{Z}_q^{n+1}$, we write $\mathbf{G}^{-1}(\mathbf{p})$ for the m-bit vector $(\mathsf{bits}(\mathbf{p}[1]), \ldots, \mathsf{bits}(\mathbf{p}[n+1]))^\mathsf{T}$, where $\mathsf{bits}(\mathbf{p}[i])$ are $\frac{m}{n+1}$ bits for each $i \in [n+1]$. The notation extends column-wise to matrices and it holds that $\mathbf{G}\mathbf{G}^{-1}(\mathbf{P}) = \mathbf{P}$.

Given $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{p} \in \mathbb{Z}_q^n$ such that $\mathbf{B}\mathbf{k} = \mathbf{p}$ has a solution $\mathbf{k}^* \in \mathbb{Z}^m$, we write

$$\Lambda^\perp_\mathbf{p}(\mathbf{B}) = \{\, \mathbf{k} \in \mathbb{Z}^m \mid \mathbf{B}\mathbf{k} = \mathbf{p} \,\} = \mathbf{k}^* + \Lambda^\perp_\mathbf{0}(\mathbf{B}),$$

which is a lattice coset. Let S be any lattice coset and $\sigma \geq 0$, we denote by $\mathcal{D}_{S,\sigma}$ the discrete Gaussian distribution [MP11] over S with width σ. We truncate $\mathcal{D}_{\mathbb{Z},\sigma}$ for ease of boundedness:

[11] The strength of correctness is not the same as the nature of its proof. See Footnote 9.

Lemma 1 (tail and truncation of $\mathcal{D}_{\mathbb{Z},\sigma}$)**.** *There exists $B_0 \in \Theta(\sqrt{\lambda})$ such that*

$$\Pr\left[\, x \xleftarrow{\$} \mathcal{D}_{\mathbb{Z},\sigma} \;:\; |x| > \sigma B_0(\lambda) \,\right] \le 2^{-\lambda} \qquad \text{for all } \sigma \ge 1 \text{ and } \lambda \in \mathbb{N}.$$

Let $B \ge 0$, the distribution $\mathcal{D}_{\mathbb{Z},\sigma,\le B}$ is sampled by first sampling $x \xleftarrow{\$} \mathcal{D}_{\mathbb{Z},\sigma}$, then returning x if $|x| \le B$, and 0 otherwise. Let $\sigma \ge 1$ and $B = \sigma \cdot \Theta(\sqrt{\lambda})$, then $\mathcal{D}_{\mathbb{Z},\sigma,\le B}$ is $2^{-\Omega(\lambda)}$-close to $\mathcal{D}_{\mathbb{Z},\sigma}$.

We will also need noise flooding with discrete Gaussian:

Lemma 2 (noise flooding with $\mathcal{D}_{\mathbb{Z},\sigma}$)**.** *Let $z \in \mathbb{Z}$, $\sigma \ge 2^{\lambda+6} z$, then $(z + \mathcal{D}_{\mathbb{Z},\sigma})$ is $2^{-\lambda}$-close to $\mathcal{D}_{\mathbb{Z},\sigma}$, and as a corollary, $(z + \mathcal{D}_{\mathbb{Z},\sigma,\le\sigma\sqrt{\lambda}})$ is $2^{-\Omega(\lambda)}$-close to $\mathcal{D}_{\mathbb{Z},\sigma,\le\sigma\sqrt{\lambda}}$.*

Assumption. We rely on the LWE assumption for small secrets with circular security.

Assumption 1 ((circular) (small-secret) LWE)**.** Let $n, m, m', q, \sigma, \sigma'$ be functions of λ and

$$\begin{array}{lll}
\overline{\mathbf{A}}_{\mathrm{fhe}} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, & \overline{\mathbf{A}}' \xleftarrow{\$} \mathbb{Z}_q^{n\times m'}, & \boxed{\mathbf{r} \xleftarrow{\$} \mathcal{D}^n_{\mathbb{Z},\sigma,\le\sigma\sqrt{\lambda}},} \qquad \mathbf{s} \leftarrow (\mathbf{r}^\intercal, -1)^\intercal, \\
\mathbf{e}_{\mathrm{fhe}} \xleftarrow{\$} \mathcal{D}^m_{\mathbb{Z},\sigma,\le\sigma\sqrt{\lambda}}, & \mathbf{e}' \xleftarrow{\$} \mathcal{D}^{m'}_{\mathbb{Z},\sigma',\le\sigma'\sqrt{\lambda}}, & \mathbf{R} \xleftarrow{\$} \{0,1\}^{m\times(n+1)\lceil\log_2 q\rceil m}, \\
\boldsymbol{\delta}_{\mathrm{fhe}} \xleftarrow{\$} \mathbb{Z}_q^m, & \boldsymbol{\delta}' \xleftarrow{\$} \mathbb{Z}_q^{m'}, & \boldsymbol{\Delta} \xleftarrow{\$} \mathbb{Z}_q^{(n+1)\times(n+1)\lceil\log_2 q\rceil m}.
\end{array}$$

The *circular small-secret LWE assumption* $\mathsf{csLWE}_{n,m,m',q,\sigma,\sigma'}$ states that

$$\begin{aligned}
&\left\{\left(1^\lambda, \begin{pmatrix}\overline{\mathbf{A}}_{\mathrm{fhe}} \\ \mathbf{r}^\intercal\overline{\mathbf{A}}_{\mathrm{fhe}} + \mathbf{e}_{\mathrm{fhe}}^\intercal\end{pmatrix}, \begin{pmatrix}\overline{\mathbf{A}}_{\mathrm{fhe}} \\ \mathbf{r}^\intercal\overline{\mathbf{A}}_{\mathrm{fhe}} + \mathbf{e}_{\mathrm{fhe}}^\intercal\end{pmatrix}\mathbf{R} - \mathsf{bits}(\mathbf{s}) \otimes \mathbf{G}, \overline{\mathbf{A}}', \mathbf{r}^\intercal\overline{\mathbf{A}}' + (\mathbf{e}')^\intercal\right)\right\}_{\lambda\in\mathbb{N}} \\
\approx &\left\{\left(1^\lambda, \underbrace{\begin{pmatrix}\overline{\mathbf{A}}_{\mathrm{fhe}} \\ \boldsymbol{\delta}_{\mathrm{fhe}}^\intercal\end{pmatrix}, \boldsymbol{\Delta}}_{\text{circular terms}}, \overline{\mathbf{A}}', (\boldsymbol{\delta}')^\intercal\right)\right\}_{\lambda\in\mathbb{N}}.
\end{aligned}$$

For the *small-secret LWE assumption* $\mathsf{sLWE}_{n,m',q,\sigma,\sigma'}$, the circular terms are removed from both distributions. For the *LWE assumption* $\mathsf{LWE}_{n,m',q,\sigma'}$, the distribution of $\mathbf{r}$ is changed to $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^n$ and the circular terms are removed.

Lemma 3. *The following hardness implications are true:*

- $\mathsf{csLWE}_{n,m,m',q,\sigma,\sigma'} \Longrightarrow \mathsf{sLWE}_{n,m',q,\sigma,\sigma'} \Longrightarrow \mathsf{LWE}_{n,m',q,\sigma'}$;
- $\mathsf{LWE}_{n,\lambda n+m',q,\sigma'} \Longrightarrow \mathsf{sLWE}_{n,m',q,\sigma',\sigma'}$ [ACPS09];
- $\mathsf{LWE}_{n,2\lambda+n\lceil\log_2 q\rceil,q,2^{-\lambda}\sigma'} \Longrightarrow \mathsf{LWE}_{n,m',q,\sigma'}$ for all $m' = \mathrm{poly}(\lambda)$;
- $\mathsf{sLWE}_{n,2\lambda+n\lceil\log_2 q\rceil,q,\sigma,2^{-\lambda}\sigma'} \Longrightarrow \mathsf{sLWE}_{n,m',q,\sigma,\sigma'}$ for all $m' = \mathrm{poly}(\lambda)$;
- $\mathsf{csLWE}_{n,m,2\lambda+n\lceil\log_2 q\rceil,q,\sigma,2^{-\lambda}\sigma'} \Longrightarrow \mathsf{csLWE}_{n,m,m',q,\sigma,\sigma'}$ for all $m' = \mathrm{poly}(\lambda)$.

We note that due to the presence of circular encryption, the reductions in Lemma 3 cannot alter the distribution of $\mathbf{r}$ in the last implication.

Parameters. We rely on csLWE with the following parameters:

- n is a fixed polynomial in λ and $m = 3(n+1)\lceil \log_2 q \rceil$;[12]
- m' varies over all possible polynomials in λ;
- $\log_2 q$ is an integer and fixed polynomial in n;
- σ, σ' are fixed functions of n such that $\frac{q}{\sigma\sqrt{n}}, \frac{q}{\sigma'\sqrt{n}} = \Omega(2^{n^\theta})$ for some constant $0 < \theta < 1$.

By Lemma 3, they reduce to a single $\mathsf{csLWE}_{n,m,2\lambda+n\lceil \log_2 q \rceil,q,\sigma,2^{-\lambda}\sigma'}$ assumption, which also implies all $\mathsf{sLWE}_{n,\mathrm{poly}(\lambda),q,\sigma,\sigma'}$ and $\mathsf{LWE}_{n,\mathrm{poly}(\lambda),q,\sigma'}$. In our derivation, we might use more specific choices of parameters.

We remark that we could solely rely on quasi-polynomial modulus-to-noise ratio. Clearly, the correctness error would not go below quasi-polynomial. There is a less obvious cost. Since our security proof relies on correctness, the provable advantage bound would also be quasi-polynomial, even if the assumptions have subexponential security. We choose to not go this route. In contrast, we obtain subexponential correctness and security from subexponential modulus-to-noise ratio if the assumptions are subexponentially secure, because our reductions are polynomial-time.

2.5 Homomorphic Encryption and Evaluation à la [GSW13]

We rely on the (leveled fully) homomorphic encryption due to [GSW13]. Since we use it as a building block and the security proof of our construction will be different, we only recall the format (without the distribution) of its components and the correctness property:

Lemma 4 ([GSW13])**.** *The leveled FHE scheme works as follows:*

- *The keys are*

$$\text{(public)} \quad \mathbf{A}_{\mathrm{fhe}} = \begin{pmatrix} \overline{\mathbf{A}}_{\mathrm{fhe}} \\ \mathbf{r}^\intercal \overline{\mathbf{A}}_{\mathrm{fhe}} + \mathbf{e}_{\mathrm{fhe}}^\intercal \end{pmatrix} \in \mathbb{Z}_q^{(n+1)\times m}, \qquad \text{(secret)} \quad \mathbf{s}^\intercal = (\mathbf{r}^\intercal, -1) \in \mathbb{Z}^{n+1},$$

  where $\mathbf{r} \in \mathbb{Z}^n$, $\overline{\mathbf{A}}_{\mathrm{fhe}} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{e}_{\mathrm{fhe}} \in \mathbb{Z}^m$.

- *A ciphertext of* $x \in \{0,1\}$ *is*

$$\mathbf{X} = \mathbf{A}_{\mathrm{fhe}}\mathbf{R} - x\mathbf{G} \in \mathbb{Z}_q^{(n+1)\times m},$$

  where $\mathbf{R} \in \mathbb{Z}^{m\times m}$ *is the encryption randomness. The decryption equation is*

$$\mathbf{s}^\intercal\mathbf{X} = -\mathbf{e}_{\mathrm{fhe}}^\intercal\mathbf{R} - x\mathbf{s}^\intercal\mathbf{G} \in \mathbb{Z}_q^m,$$

  which can be used to extract x *via multiplication by* $\mathbf{G}^{-1}(\lfloor q/2 \rceil \boldsymbol{\iota}_{n+1})$.

- *There is an efficient algorithm*

$$\mathsf{MakeHEvalCkt}(1^n, 1^m, q, C) = \mathsf{HEval}_C$$

  that takes as input n, m, q *and a circuit* $C : \{0,1\}^L \to \{0,1\}$ *and outputs a circuit*

$$\mathsf{HEval}_C(\mathbf{X}_1, \ldots, \mathbf{X}_L) = \mathbf{C}$$

  taking L *ciphertexts as input and outputting a new ciphertext* $\mathbf{C}$.

[12] This choice of m enables the reduction of *non-circular* IND-CPA security of the [GSW13] FHE scheme (used in the circular terms) to $\mathsf{LWE}_{n,m,q,\sigma'}$, providing heuristic justification for our assumption. It also satisfies the constraint in Lemma 21 for trapdoor generation.

– *The depth of* HEval_C *is* $d\,\mathrm{O}(\log m \log\log q)$,[13] *where* d *is the depth of* C.
– *Suppose* $\mathbf{X}_\ell = \mathbf{A}_{\mathrm{fhe}}\mathbf{R}_\ell - \mathbf{x}[\ell]\mathbf{G}$ *for* $\ell \in [L]$ *with* $\mathbf{x} \in \{0,1\}^L$, *then*

$$\mathbf{C} = \mathbf{A}_{\mathrm{fhe}}\mathbf{R}_C - C(\mathbf{x})\mathbf{G},$$

where $\|\mathbf{R}_C^\intercal\| \le (m+2)^d \max_{\ell\in[L]} \|\mathbf{R}_\ell^\intercal\|$.

Additionally, in the circular version, ciphertexts of $\mathsf{bits}(\mathbf{s})$ *are published.*

It will be convenient for us to extend the homomorphic evaluation procedure for vector-valued functions similarly to [BTVW17].

Lemma 5 (homomorphic evaluation for vector-valued functions; ¶)**.** *For the scheme in Lemma 4, there is an efficient algorithm*

$$\mathsf{MakeVEvalCkt}(n, m, q, C) = \mathsf{VEval}_C$$

that takes as input n, m, q *and a vector-valued circuit* $C : \{0,1\}^L \to \mathbb{Z}_q^{1\times m'}$ *and outputs a circuit*

$$\mathsf{VEval}_C(\mathbf{X}_1, \ldots, \mathbf{X}_L) = \mathbf{C}$$

taking L *ciphertexts as input and outputting a new ciphertext* $\mathbf{C}$ *of different format.*

- *The depth of* VEval_C *is* $\big(d\,\mathrm{O}(\log m \log\log q) + \mathrm{O}(\log^2 \log q)\big)$[14] *for* C *of depth* d.
- *Suppose* $\mathbf{X}_\ell = \mathbf{A}_{\mathrm{fhe}}\mathbf{R}_\ell - \mathbf{x}[\ell]\mathbf{G}$ *for* $\ell \in [L]$ *with* $\mathbf{x} \in \{0,1\}^L$, *then*

$$\mathbf{C} = \mathbf{A}_{\mathrm{fhe}}\mathbf{R}_C - \begin{pmatrix} \mathbf{0}_{n\times m'} \\ C(\mathbf{x}) \end{pmatrix} \in \mathbb{Z}_q^{(n+1)\times m'},$$

where $\|\mathbf{R}_C^\intercal\| \le (m+2)^d \lceil \log_2 q \rceil \max_{\ell\in[L]} \|\mathbf{R}_\ell^\intercal\|$. *The new decryption equation is*

$$\mathbf{s}^\intercal \mathbf{C} = -\mathbf{e}_{\mathrm{fhe}}^\intercal \mathbf{R}_C + C(\mathbf{x}) \in \mathbb{Z}_q^{1\times m'}.$$

Proof (Lemma 5). Recall that the codomain of C being $\mathbb{Z}_q^{1\times m'}$ is just conceptual — pragmatically, the output is computed bit by bit. Let $C_{u,v} : \{0,1\}^L \to \{0,1\}$ be the subcircuit of C computing the v^{th} bit (low to high) of the u^{th} scalar output, i.e., $C_{u,v}(\mathbf{x}) = \mathsf{bits}(C(\mathbf{x})[1,u])[1,v]$. The algorithm $\mathsf{MakeVEvalCkt}$ works as follows:

1. Given n, m, q, C, it splits C into $C_{u,v}$'s and runs

$$\mathsf{HEval}_{C_{u,v}} \leftarrow \mathsf{MakeHEvalCkt}(n, m, q, C_{u,v}) \qquad \text{for all } u \in [m'] \text{ and } v \in [0, \log_2 q).$$

2. It constructs and outputs the following circuit VEval_C.
   (a) Given $\mathbf{X}_1, \ldots, \mathbf{X}_L$, first use $\mathsf{HEval}_{C_{u,v}}$ to obtain $\mathbf{C}_{u,v}$ for all u, v in parallel.
   (b) Then compute, for all u in parallel,

$$\mathbf{C}_u = \sum_v \mathbf{C}_{u,v}\mathbf{G}^{-1}(2^v \boldsymbol{\iota}_{n+1}).$$

[13] A better bound is $\mathrm{O}(\log m + \log\log q)$. Consequently, some parameters can be tighter. We choose to keep the bound used during the initial write-up to avoid an overhaul at this time.
[14] A better bound is $\big(d\,\mathrm{O}(\log m + \log\log q) + \mathrm{O}(\log\log q)\big)$. See Footnote 13.

(c) Lastly, output $\mathbf{C} = (\mathbf{C}_1, \ldots, \mathbf{C}_{m'})$.

To analyze the depth of VEval_C, let $d_{u,v}$ be the depth of $C_{u,v}$, then $d_{u,v} \le d$ since $C_{u,v}$ is a subcircuit of C, which is of depth d. Step 2a thus is of depth $d\,\mathrm{O}(\log m \log\log q)$ by Lemma 4. In Step 2b, multiplication by $\mathbf{G}^{-1}(2^v \boldsymbol{\iota}_{n+1})$, which is a standard basis vector known to $\mathsf{MakeHEvalCkt}$ as a constant, can be implemented by wiring the correct entries (namely, the $(n \cdot \frac{m}{n+1} + v + 1)^{\text{st}}$ column) of $\mathbf{C}_{u,v}$, hence requires no additional depth. Summing up $\lceil \log_2 q \rceil$ values in $\mathbb{Z}_q$ takes $\mathrm{O}(\log^2 \log q)$[15] depth. Step 2c simply specifies the output, making no contribution to the depth. The bound for the total depth follows.

To see the new decryption equation and analyze $\|\mathbf{R}_C^\intercal\|$, we have $\mathbf{C}_{u,v} = \mathbf{A}_{\text{fhe}}\mathbf{R}_{C_{u,v}} - C_{u,v}(\mathbf{x})\mathbf{G}$ by Lemma 4, where

$$\|\mathbf{R}_{C_{u,v}}^\intercal\| \le (m+2)^{d_{u,v}} \max_{\ell \in [L]} \|\mathbf{R}_\ell^\intercal\| \le (m+2)^{d} \max_{\ell \in [L]} \|\mathbf{R}_\ell^\intercal\|.$$

For each $u \in [m']$,

$$\begin{aligned}\mathbf{C}_u = \sum_v \mathbf{C}_{u,v}\mathbf{G}^{-1}(2^v \boldsymbol{\iota}_{n+1}) &= \mathbf{A}_{\text{fhe}} \sum_v \mathbf{R}_{C_{u,v}}\mathbf{G}^{-1}(2^v \boldsymbol{\iota}_{n+1}) - \sum_v C_{u,v}(\mathbf{x})\mathbf{G}\mathbf{G}^{-1}(2^v \boldsymbol{\iota}_{n+1}) \\ &= \mathbf{A}_{\text{fhe}} \sum_v \mathbf{R}_{C_{u,v}}\mathbf{G}^{-1}(2^v \boldsymbol{\iota}_{n+1}) - \begin{pmatrix} \mathbf{0}_{n\times 1} \\ C(\mathbf{x})[1,u] \end{pmatrix},\end{aligned}$$

and therefore,

$$\mathbf{C} = (\mathbf{C}_1, \ldots, \mathbf{C}_{m'}) = \mathbf{A}_{\text{fhe}}\mathbf{R}_C - \begin{pmatrix} \mathbf{0}_{n\times m'} \\ C(\mathbf{x}) \end{pmatrix},$$

$$\text{with} \qquad \mathbf{R}_C = \left(\sum_v \mathbf{R}_{C_{1,v}}\mathbf{G}^{-1}(2^v \boldsymbol{\iota}_{n+1}),\ \ldots,\ \sum_v \mathbf{R}_{C_{m',v}}\mathbf{G}^{-1}(2^v \boldsymbol{\iota}_{n+1}) \right).$$

Note that every row of $\mathbf{R}_C^\intercal$ is a sum of $\lceil \log_2 q \rceil$ rows among all the rows of $\mathbf{R}_{C_{u,v}}^\intercal$, from which the desired norm bound follows. □

2.6 Attribute Encoding and Homomorphic Evaluation aux [BGG+14,BTVW17]

We use the attribute encoding and its homomorphic evaluation in [BGG+14], with the extension to matrix-valued circuits in [BTVW17]. We further extend the dual-use technique in [BTVW17] to circular encryption and vector-valued circuits.

Lemma 6 ([BGG+14,BTVW17])**.** *The attribute encoding and its homomorphic evaluation work as follows:*

- *For L-bit input, the public parameter is $\mathbf{A}_{\text{attr}} \in \mathbb{Z}_q^{(n+1)\times(L+1)m}$.*
- *The encoding of $\mathbf{x} \in \{0,1\}^L$ is*

$$\mathbf{s}^\intercal(\mathbf{A}_{\text{attr}} - (1, \mathbf{x}^\intercal) \otimes \mathbf{G}) + \mathbf{e}_{\text{attr}}^\intercal,$$

where $\mathbf{s}^\intercal = (\mathbf{r}^\intercal, -1)$ with $\mathbf{r} \in \mathbb{Z}^n$ and $\mathbf{e}_{\text{attr}} \in \mathbb{Z}^{(L+1)m}$.

[15] A better bound is $\mathrm{O}(\log\log q)$. See Footnote 13.

- *There are efficient deterministic algorithms [BGG+14]*

$$\mathsf{EvalC}(\mathbf{A}_{\mathrm{attr}}, C) = \mathbf{H}_C \qquad \text{and} \qquad \mathsf{EvalCX}(\mathbf{A}_{\mathrm{attr}}, C, \mathbf{x}) = \mathbf{H}_{C,\mathbf{x}}$$

  that take as input $\mathbf{A}_{\mathrm{attr}}$*, a circuit* $C : \{0,1\}^L \to \{0,1\}$*, and (for* EvalCX*) some* $\mathbf{x} \in \{0,1\}^L$*, and output some matrix in* $\mathbb{Z}^{(L+1)m \times m}$*.*

  - *Suppose* C *is of depth* d*, then* $\|\mathbf{H}_C^\intercal\|, \|\mathbf{H}_{C,\mathbf{x}}^\intercal\| \le (m+2)^d$*.*
  - *They satisfy encoding homomorphism,* $(\mathbf{A}_{\mathrm{attr}} - (1, \mathbf{x}^\intercal) \otimes \mathbf{G})\mathbf{H}_{C,\mathbf{x}} = \mathbf{A}_{\mathrm{attr}}\mathbf{H}_C - C(\mathbf{x})\mathbf{G}$*.*

- *There are efficient deterministic algorithms [BTVW17]*

$$\mathsf{MEvalC}(\mathbf{A}_{\mathrm{attr}}, C) = \mathbf{H}_C \qquad \text{and} \qquad \mathsf{MEvalCX}(\mathbf{A}_{\mathrm{attr}}, C, \mathbf{x}) = \mathbf{H}_{C,\mathbf{x}}$$

  that take as input $\mathbf{A}_{\mathrm{attr}}$*, a matrix-valued circuit* $C : \{0,1\}^L \to \mathbb{Z}_q^{(n+1)\times m'}$*, and (for* MEvalCX*) some* $\mathbf{x} \in \{0,1\}^L$*, and output some matrix in* $\mathbb{Z}^{(L+1)m \times m'}$*.*

  - *Suppose* C *is of depth* d*, then* $\|\mathbf{H}_C^\intercal\|, \|\mathbf{H}_{C,\mathbf{x}}^\intercal\| \le (m+2)^d \lceil \log_2 q \rceil$*.*
  - *The matrix encoding homomorphism is* $(\mathbf{A}_{\mathrm{attr}} - (1, \mathbf{x}^\intercal) \otimes \mathbf{G})\mathbf{H}_{C,\mathbf{x}} = \mathbf{A}_{\mathrm{attr}}\mathbf{H}_C - C(\mathbf{x})$*.*

Dual-Use Technique and Extension. In [BTVW17], the attribute encoded with secret $\mathbf{s}^\intercal$ is FHE ciphertexts under key $\mathbf{s}^\intercal$ (the same, "dual-use") and the circuit being MEvalCX'ed is some HEval_C. This leads to *automatic decryption*. Let C be a circuit with Boolean output, $\mathbf{x}$ an input, $\mathbf{X}$ a bunch of FHE ciphertexts of $\mathsf{bits}(\mathbf{x})$ under $\mathbf{s}^\intercal$, and $\mathbf{e}_{\mathrm{attr}}, \mathbf{e}', \mathbf{e}''$ some unspecified noises, then

$$\begin{aligned}
& \big(\mathbf{s}^\intercal(\mathbf{A}_{\mathrm{attr}} - (1, \mathsf{bits}(\mathbf{X})) \otimes \mathbf{G}) + \mathbf{e}_{\mathrm{attr}}^\intercal\big)\mathbf{H}_{\mathsf{HEval}_C,\mathbf{x}} \\
(\mathsf{MEvalCX}) \quad & = \mathbf{s}^\intercal \mathbf{A}_{\mathrm{attr}} \mathbf{H}_{\mathsf{HEval}_C} - \mathbf{s}^\intercal \mathsf{HEval}_C(\mathbf{X}) + (\mathbf{e}')^\intercal \\
(\mathsf{HEval}\text{ decryption}) \quad & = \mathbf{s}^\intercal \mathbf{A}_{\mathrm{attr}} \mathbf{H}_{\mathsf{HEval}_C} - \mathbf{s}^\intercal C(\mathbf{x})\mathbf{G} + (\mathbf{e}'')^\intercal \\
& = \mathbf{s}^\intercal(\mathbf{A}_{\mathrm{attr}} \mathbf{H}_{\mathsf{HEval}_C} - C(\mathbf{x})\mathbf{G}) + (\mathbf{e}'')^\intercal.
\end{aligned}$$

To extend the dual-use technique to vector-valued circuits, let the codomain of C be $\mathbb{Z}_q^{1\times m'}$, then VEval_C is $\mathbb{Z}_q^{(n+1)\times m'}$-valued and

$$\begin{aligned}
& \big(\mathbf{s}^\intercal(\mathbf{A}_{\mathrm{attr}} - (1, \mathsf{bits}(\mathbf{X})) \otimes \mathbf{G}) + \mathbf{e}_{\mathrm{attr}}^\intercal\big)\mathbf{H}_{\mathsf{VEval}_C,\mathbf{x}} \\
(\mathsf{MEvalCX}) \quad & = \mathbf{s}^\intercal \mathbf{A}_{\mathrm{attr}} \mathbf{H}_{\mathsf{VEval}_C} - \mathbf{s}^\intercal \mathsf{VEval}_C(\mathbf{X}) + (\mathbf{e}')^\intercal \\
(\mathsf{VEval}\text{ decryption}) \quad & = \mathbf{s}^\intercal \mathbf{A}_{\mathrm{attr}} \mathbf{H}_{\mathsf{VEval}_C} - C(\mathbf{x}) + (\mathbf{e}'')^\intercal.
\end{aligned}$$

Extension to circular encryption means setting $\mathbf{x} = \mathbf{s}$, for which we say $\mathbf{S}, \mathbf{A}_{\mathrm{circ}}$ in place of $\mathbf{X}, \mathbf{A}_{\mathrm{attr}}$.

3 Bootstrapping Homomorphic Evaluation

We first introduce some lemmas that will come handy later.

Lemma 7 (¶)**.** *Let* q *and* $\mathbf{h} \in \mathbb{Z}^{m'}$ *be fixed. Let* $g = \gcd(q, \mathbf{h}[1], \ldots, \mathbf{h}[m'])$ *and* $\mathbf{d} \xleftarrow{\$} \mathbb{Z}_q^{m'}$*, then* $\mathbf{d}^\intercal\mathbf{h}$ *is uniformly random over*

$$g\mathbb{Z}_q = \{\underbrace{g \cdot x}_{\text{multiplication over } \mathbb{Z}_q} \mid x \in \mathbb{Z}_q\} = \{\underbrace{g \cdot (x \bmod (q/g))}_{\substack{\text{multiplication over } \mathbb{Z} \\ \text{then sent into } \mathbb{Z}_q}} \mid x \in \mathbb{Z}_{q/g}\}.$$

Proof (Lemma 7). Let i run through $[m']$. By Bézout's lemma, there exist $z_0, z_1, \ldots, z_{m'} \in \mathbb{Z}$ with

$$z_0 q + \sum_i z_i \mathbf{h}[i] = g.$$

Let $x \xleftarrow{\$} \mathbb{Z}_q$ and $\mathbf{d}'[i] = \mathbf{d}[i] - x z_i$, then $\mathbf{d}'$ is independent of x. Note that

$$\begin{aligned}\mathbf{d}^\mathsf{T}\mathbf{h} = gx + \mathbf{d}^\mathsf{T}\mathbf{h} - gx &= gx + \sum_i \mathbf{d}[i]\mathbf{h}[i] - x\left(z_0 q + \sum_i z_i \mathbf{h}[i]\right) \\ &= gx - \underbrace{qxz_0}_{=0} + \sum_i (\mathbf{d}[i] - xz_i)\mathbf{h}[i] = gx + (\mathbf{d}')^\mathsf{T}\mathbf{h}.\end{aligned}$$

Clearly, $(\mathbf{d}')^\mathsf{T}\mathbf{h} \in g\mathbb{Z}_q$. Since gx is uniformly random over $g\mathbb{Z}_q$ and $(\mathbf{d}')^\mathsf{T}\mathbf{h}$ is independent of x, we conclude that $\mathbf{d}^\mathsf{T}\mathbf{h} = gx + (\mathbf{d}')^\mathsf{T}\mathbf{h}$ follows the uniform distribution over $g\mathbb{Z}_q$. □

Lemma 8 (¶). *Let $B, B', n, m', q, \sigma, \sigma', p > 0$ and $h : \mathbb{Z}_q^{(n+1)\times m'} \xrightarrow{\$} \mathbb{Z}^{m'}$, all of which are efficiently computable. Suppose $p \mid q$ and $\Pr[\,\|(h(\mathbf{A}))^\mathsf{T}\| \le B'\,] = 1$, then under $\mathsf{sLWE}_{n,m',q,\sigma,\sigma'}$ (Assumption 1),*

$$\Pr\left[\begin{array}{l}\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{(n+1)\times m'} \\ \mathbf{r} \xleftarrow{\$} \mathcal{D}^n_{\mathbb{Z},\sigma,\le\sigma\sqrt{\lambda}} \\ \mathbf{h} \xleftarrow{\$} h(\mathbf{A})\end{array} : \begin{array}{r}((\mathbf{r}^\mathsf{T}, -1)\mathbf{A}\mathbf{h} \bmod p) \\ \in \left[-\frac{p}{2} + B, \frac{p}{2} - B\right)\end{array}\right] \ge 1 - \frac{2B + (2\sigma'\sqrt{\lambda} + 1)B'}{p} - \mathrm{negl}(\lambda).$$

Proof (Lemma 8). Let

$$\mathbf{A} = \begin{pmatrix}\overline{\mathbf{A}}' \\ \underline{\mathbf{a}}^\mathsf{T}\end{pmatrix}, \quad \mathbf{e} \xleftarrow{\$} \mathcal{D}^{m'}_{\mathbb{Z},\sigma',\le\sigma'\sqrt{\lambda}}, \quad \boldsymbol{\delta} \xleftarrow{\$} \mathbb{Z}_q^{m'}, \quad \widetilde{B} = B + B'\sigma'\sqrt{\lambda}, \quad E_0 : |\mathbf{e}^\mathsf{T}\mathbf{h}| \le B'\sigma'\sqrt{\lambda},$$

$$\begin{aligned} E_1 &: ((\mathbf{r}^\mathsf{T}, -1)\mathbf{A}\mathbf{h} \bmod p) \in \left[-\frac{p}{2} + B, \frac{p}{2} - B\right), \\ E_2 &: ((\mathbf{r}^\mathsf{T}\overline{\mathbf{A}}' + \mathbf{e}^\mathsf{T} - \underline{\mathbf{a}}^\mathsf{T})\mathbf{h} \bmod p) \in \left[-\frac{p}{2} + \widetilde{B}, \frac{p}{2} - \widetilde{B}\right), \\ E_3 &: ((\boldsymbol{\delta}^\mathsf{T} - \underline{\mathbf{a}}^\mathsf{T})\mathbf{h} \bmod p) \in \left[-\frac{p}{2} + \widetilde{B}, \frac{p}{2} - \widetilde{B}\right). \end{aligned}$$

Note that $(\mathbf{r}^\mathsf{T}\overline{\mathbf{A}}' + \mathbf{e}^\mathsf{T} - \underline{\mathbf{a}}^\mathsf{T})\mathbf{h} = (\mathbf{r}^\mathsf{T}, -1)\mathbf{A}\mathbf{h} + \mathbf{e}^\mathsf{T}\mathbf{h}$, so $E_1 \supseteq E_0 \cap E_2$. Together with $\Pr[E_0] = 1$,

$$\Pr[E_1] \ge \Pr[E_0 \cap E_2] = \Pr[E_2].$$

It follows from $\mathsf{sLWE}_{n,m',q,\sigma,\sigma'}$ that $\Pr[E_2] \ge \Pr[E_3] - \mathrm{negl}(\lambda)$. (The reduction algorithm receives $\overline{\mathbf{A}}'$ and either $(\mathbf{r}^\mathsf{T}\overline{\mathbf{A}}' + \mathbf{e}^\mathsf{T})$ or $\boldsymbol{\delta}^\mathsf{T}$, and performs the range testing after sampling $\underline{\mathbf{a}}, \mathbf{h}$.)

Moving to modulus p, let $g = \gcd(p, \mathbf{h}[1], \ldots, \mathbf{h}[m'])$ and $\mathbf{d} \in \mathbb{Z}_p^{m'}$ such that $\mathbf{d} = (\boldsymbol{\delta} - \underline{\mathbf{a}}) \bmod p$, then $\mathbf{d}$ is uniformly random and independent of $\mathbf{h}, g$. By Lemma 7, conditioned on g, the quantity $\mathbf{d}^\mathsf{T}\mathbf{h}$ is uniform over $g\mathbb{Z}_p$. Since $\|\mathbf{h}^\mathsf{T}\| \le B'$, either $g = p$ (if $\mathbf{h} = \mathbf{0}$) or $g \le \max|\mathbf{h}[i]| \le B'$. We proceed with a case analysis:

- ($g = p$.) In this case, $\mathbf{d}^\mathsf{T}\mathbf{h} \bmod p = 0$, so $\Pr[\,E_3 \mid g = p\,] = 1 \ge 1 - \frac{B' + 2\widetilde{B}}{p}$.

- ($g = g_0 \le B'$.) In this case, $\frac{\mathbf{d}^\intercal \mathbf{h} \bmod p}{g_0}$ is uniformly random over $\left[-\frac{p}{2g_0}, \frac{p}{2g_0}\right)$, so

$$\begin{aligned}\Pr[\,E_3 \mid g = g_0\,] &= \Pr\left[u \xleftarrow{\$} \left[-\frac{p}{2g_0}, \frac{p}{2g_0}\right) \,:\, u \in \left[-\frac{p}{2g_0} + \frac{\widetilde{B}}{g_0}, \frac{p}{2g_0} - \frac{\widetilde{B}}{g_0}\right)\right] \\ &\ge \frac{\lceil (p - 2\widetilde{B})/2g_0 \rceil - \lceil -(p - 2\widetilde{B})/2g_0 \rceil}{p/g_0} > \frac{2(p - 2\widetilde{B})/2g_0 - 1}{p/g_0} \\ &\ge 1 - \frac{g_0 + 2\widetilde{B}}{p} \ge 1 - \frac{B' + 2\widetilde{B}}{p}.\end{aligned}$$

Therefore,

$$\begin{aligned}\Pr[E_1] \ge \Pr[E_2] \ge \Pr[E_3] - \mathrm{negl}(\lambda) &= \mathbb{E}[\,\Pr[\,E_3 \mid g\,]\,] - \mathrm{negl}(\lambda) \\ &\ge 1 - \frac{B' + 2\widetilde{B}}{p} - \mathrm{negl}(\lambda) = 1 - \frac{2B + (2\sigma'\sqrt{\lambda} + 1)B'}{p} - \mathrm{negl}(\lambda).\end{aligned}$$

□

Lemma 9 (¶). *For all $u, v \in \mathbb{Z}$ and $w \in \mathbb{Z}_+$,*

$$\left\lfloor \frac{u+v}{w} \right\rfloor = \left\lfloor \frac{u}{w} \right\rfloor \qquad \text{if and only if} \qquad (u+v) \bmod w = (u \bmod w) + v.$$

Proof (Lemma 9). Taking the difference between the two sides,

$$\begin{aligned}\left\lfloor \frac{u+v}{w} \right\rfloor - \left\lfloor \frac{u}{w} \right\rfloor &= \frac{(u+v) - \big((u+v) \bmod w\big)}{w} - \frac{u - (u \bmod w)}{w} \\ &= \frac{\big((u \bmod w) + v\big) - \big((u+v) \bmod w\big)}{w}.\end{aligned}$$

The lemma follows from the equivalence of being zero for both sides. □

Notations and Parameters. Let q be the modulus, M the rounding resolution, B the bound of removable noises (also a loose bound on the secret), B' the bound of homomorphism matrices for one step of computation, and σ, σ' Gaussian widths. For the purpose of this section, they must satisfy these relations:

- M is an exact power of two (so that M divides a large portion of $\mathbf{G}$) and $M \mid q$;
- $M^2 B/q$, $\sigma' B'/q$, B/M, $\sigma' B'/M$ are negligible;
- B' is $2^{\Omega(\log^6 \lambda)}$ and $\sigma\sqrt{\lambda} \le 2^{-\lambda} B$ and $\sigma'\sqrt{\lambda} \le 2^{-\lambda} B$.

For the applications except KP-ABE, we need another Gaussian width σ_{msg} and require

$$\sigma_{\mathrm{msg}}\sqrt{\lambda} + Bm + 1 < q/4, \qquad \sigma_{\mathrm{msg}} \ge 2^{\lambda+6}(Bm + \sigma'\sqrt{\lambda}).$$

For KP-ABE, we need $\sigma_{-1}, \sigma_{\mathrm{pre}}, \sigma_{\mathrm{post}}$ and require

$$\begin{aligned}&m \cdot \sigma_{\mathrm{post}}\sqrt{\lambda} \cdot \sigma_{-1}\sqrt{m} + \sigma'\sqrt{\lambda} + Bm + 1 < q/4, \\ &\sigma_{\mathrm{pre}} \ge 2^{\lambda+6}(Bm + \sigma'\sqrt{\lambda}), \qquad \sigma_{\mathrm{post}} \ge \sigma_{\mathrm{pre}}.\end{aligned}$$

We rely on $\mathsf{csLWE}_{n,m,2\lambda+n\lceil\log_2 q\rceil,q,\sigma,2^{-\lambda}\sigma'}$ (Assumption 1) for some $n \le \mathrm{poly}(\lambda)$ (dependent on θ; see discussion after Lemma 3) and $m = 3(n+1)\lceil\log_2 q\rceil$. By Lemma 3, it implies $\mathsf{csLWE}_{n,m,m',q,\sigma,\sigma''}$, $\mathsf{sLWE}_{n,m',q,\sigma,\sigma''}$, $\mathsf{LWE}_{n,m',q,\sigma''}$ for all $m \le \mathrm{poly}(\lambda)$ and $\sigma'' \ge \sigma'$.

We always write $\mathbf{s}^\mathsf{T} = (\mathbf{r}^\mathsf{T}, -1)$ for $\mathbf{r} \in \mathbb{Z}^n$. We define "left/right" gadget vectors/matrices

$$\begin{aligned} \mathbf{g}_\mathrm{L}^\mathsf{T} &= (2^0, 2^1, 2^2, \ldots, M/2), & \mathbf{G}_\mathrm{L} &= \mathbf{I}_{n+1} \otimes \mathbf{g}_\mathrm{L}^\mathsf{T}, \\ \mathbf{g}_\mathrm{R}^\mathsf{T} &= (M, 2M, 4M, \ldots, 2^{\frac{m}{n+1}-1}), & \mathbf{G}_\mathrm{R} &= \mathbf{I}_{n+1} \otimes \mathbf{g}_\mathrm{R}^\mathsf{T}, \end{aligned}$$

then $\mathbf{g}^\mathsf{T} = (\mathbf{g}_\mathrm{L}^\mathsf{T}, \mathbf{g}_\mathrm{R}^\mathsf{T})$ and $\mathbf{G} = (\mathbf{G}_\mathrm{L}, \mathbf{G}_\mathrm{R})\mathbf{Q}$ for some efficiently computable permutation matrix $\mathbf{Q}$.

Concretely, we can set (exact numbers)

$$\begin{aligned} q &= 2^{15\lambda}, \quad M = 2^{5\lambda}, \quad B = 2^{4\lambda}, \quad B' = 2^\lambda, \quad \sigma = 2^\lambda, \quad \sigma' = 2^{2\lambda}, \\ &\sigma_\mathrm{msg} = 2^{6\lambda}, \quad \sigma_{-1} = 2^\lambda, \quad \sigma_\mathrm{pre} = 2^{6\lambda}, \quad \sigma_\mathrm{post} = 2^{7\lambda}, \end{aligned}$$

and choose n appropriately. (The proofs in this section work with the general relations, not this particular version, and some are stated for conditions even weaker than the general ones set above.)

3.1 Noise Removal

In this subsection, we present the procedure $\mathsf{RemoveNoise}(\cdot)$. It takes as input an attribute encoding $(\mathbf{s}^\mathsf{T}(\mathbf{A} - x\mathbf{G}) + \mathbf{e}^\mathsf{T}) \in \mathbb{Z}_q^m$ with large noise $\mathbf{e}$. The deterministic algorithm, with overwhelming probability over the input, outputs a noiseless encoding $(\mathsf{RndPad}_\mathbf{A}(\mathbf{s}) - x\mathbf{s}^\mathsf{T}\mathbf{G}) \in \mathbb{Z}_q^m$, where the function $\mathsf{RndPad}_\mathbf{A}(\cdot)$ is fully described by $\mathbf{A}$ (not dependent on $x, \mathbf{e}$).

Construction 1 (noise removal)**.** $\mathsf{RemoveNoise}(\mathbf{u}^\mathsf{T} \in \mathbb{Z}_q^{1\times m})$ computes

$$\mathbf{v}_\mathrm{L}^\mathsf{T} \leftarrow \mathbf{u}^\mathsf{T}\mathbf{G}^{-1}(M\mathbf{G}_\mathrm{L}), \quad \mathbf{v}_\mathrm{R}^\mathsf{T} \leftarrow \mathbf{u}^\mathsf{T}\mathbf{G}^{-1}(\mathbf{G}_\mathrm{R}), \quad \mathbf{w}^\mathsf{T} \leftarrow \left(\left\lfloor \frac{\mathbf{v}_\mathrm{L}^\mathsf{T} \bmod q}{M} \right\rceil, M \left\lfloor \frac{\mathbf{v}_\mathrm{R}^\mathsf{T} \bmod q}{M} \right\rceil \right) \mathbf{Q},$$

and outputs $\mathbf{w}^\mathsf{T}$ as a $\mathbb{Z}_q^m$-element. The function $\mathsf{RndPad}_\mathbf{A}(\mathbf{s})$ is $\mathsf{RemoveNoise}(\mathbf{s}^\mathsf{T}\mathbf{A})$.

Lemma 10. *A canonical Boolean circuit of* $\mathsf{RndPad}_\mathbf{A}(\cdot)$ *is of depth* $\mathrm{O}(\log n \log\log q)$[16] *and can be efficiently generated from* $\mathbf{A} \in \mathbb{Z}_q^{(n+1)\times m}$.

$\mathsf{RemoveNoise}$ satisfies the following conditional correctness property.

Theorem 11 (¶)**.** *Let* $\mathbf{\Gamma} = \mathbf{G}^{-1}(M\mathbf{G}_\mathrm{L}, \mathbf{G}_\mathrm{R})$. *For all* $\mathbf{s}, \mathbf{A}$ *such that* $\|\mathbf{s}\| \le B$, *if both*

$$(\mathbf{s}^\mathsf{T}\mathbf{A}\mathbf{\Gamma} \bmod q)^\mathsf{T} \in \left[-\frac{q}{2} + M^2B, \frac{q}{2} - M^2B\right)^m \quad \text{and} \quad (\mathbf{s}^\mathsf{T}\mathbf{A}\mathbf{\Gamma} \bmod M)^\mathsf{T} \in \left[-\frac{M}{2} + B, \frac{M}{2} - B\right)^m$$

hold, then for all $x \in \{0,1\}$ *and* $\mathbf{e}$ *such that* $\|\mathbf{e}\| \le B$,

$$\mathsf{RemoveNoise}(\mathbf{s}^\mathsf{T}(\mathbf{A} - x\mathbf{G}) + \mathbf{e}^\mathsf{T}) = \mathsf{RndPad}_\mathbf{A}(\mathbf{s}) - x\mathbf{s}^\mathsf{T}\mathbf{G}.$$

Proof (Theorem 11). The proof works by expanding $\mathsf{RemoveNoise}$. Letting $\mathbf{u}^\mathsf{T} = \mathbf{s}^\mathsf{T}(\mathbf{A} - x\mathbf{G}) + \mathbf{e}^\mathsf{T}$ and $\mathbf{v}^\mathsf{T} = \mathbf{u}^\mathsf{T}\mathbf{\Gamma}$, we can rewrite $\mathbf{v}$ as

$$\begin{aligned} (\mathbf{v}_\mathrm{L}^\mathsf{T}, \mathbf{v}_\mathrm{R}^\mathsf{T}) &= (\mathbf{s}^\mathsf{T}(\mathbf{A} - x\mathbf{G}) + \mathbf{e}^\mathsf{T})\mathbf{G}^{-1}(M\mathbf{G}_\mathrm{L}, \mathbf{G}_\mathrm{R}) \\ &= (\mathbf{s}^\mathsf{T}\mathbf{A}_\mathrm{L} - Mx\mathbf{s}^\mathsf{T}\mathbf{G}_\mathrm{L} + \mathbf{e}_\mathrm{L}^\mathsf{T}, \mathbf{s}^\mathsf{T}\mathbf{A}_\mathrm{R} - x\mathbf{s}^\mathsf{T}\mathbf{G}_\mathrm{R} + \mathbf{e}_\mathrm{R}^\mathsf{T}), \end{aligned} \tag{2}$$

[16] A better bound is $\mathrm{O}(\log n + \log\log q)$. See Footnote 13.

where $(\mathbf{A}_{\mathrm{L}}, \mathbf{A}_{\mathrm{R}}) = \mathbf{A}\boldsymbol{\Gamma}$ and $(\mathbf{e}_{\mathrm{L}}^{\mathsf{T}}, \mathbf{e}_{\mathrm{R}}^{\mathsf{T}}) = \mathbf{e}^{\mathsf{T}}\boldsymbol{\Gamma}$ are blocked correspondingly. Because M is an exact power of two and $M^2 \le q$, each column of $\boldsymbol{\Gamma} = \mathbf{G}^{-1}(M\mathbf{G}_{\mathrm{L}}, \mathbf{G}_{\mathrm{R}})$ is a standard basis vector, which implies $\|\boldsymbol{\Gamma}^{\mathsf{T}}\| \le 1$. Since $(\mathbf{G}_{\mathrm{L}}, \mathbf{G}_{\mathrm{R}})\mathbf{Q} = \mathbf{G}$, it suffices to prove both of

$$\text{(left part)} \qquad \left\lfloor \frac{\mathbf{v}_{\mathrm{L}}^{\mathsf{T}} \bmod q}{M} \right\rceil = \left\lfloor \frac{\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{L}} \bmod q}{M} \right\rceil - x\mathbf{s}^{\mathsf{T}}\mathbf{G}_{\mathrm{L}},$$

$$\text{(right part)} \qquad M\left\lfloor \frac{\mathbf{v}_{\mathrm{R}}^{\mathsf{T}} \bmod q}{M} \right\rceil \equiv M\left\lfloor \frac{\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{R}} \bmod q}{M} \right\rceil - x\mathbf{s}^{\mathsf{T}}\mathbf{G}_{\mathrm{R}} \pmod q.$$

For the left part, observe that

$$\|(-Mx\mathbf{s}^{\mathsf{T}}\mathbf{G}_{\mathrm{L}} + \mathbf{e}_{\mathrm{L}}^{\mathsf{T}})^{\mathsf{T}}\| \le M \cdot \|\mathbf{s}\| \cdot \|\mathbf{G}_{\mathrm{L}}^{\mathsf{T}}\| + \|\mathbf{e}\| \cdot \|\boldsymbol{\Gamma}^{\mathsf{T}}\| \le M \cdot B \cdot M/2 + B \cdot 1 \le M^2 B.$$

Combining it with the first premise, we obtain

$$(\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{L}} - Mx\mathbf{s}^{\mathsf{T}}\mathbf{G}_{\mathrm{L}} + \mathbf{e}_{\mathrm{L}}^{\mathsf{T}}) \bmod q = (\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{L}} \bmod q) - Mx\mathbf{s}^{\mathsf{T}}\mathbf{G}_{\mathrm{L}} + \mathbf{e}_{\mathrm{L}}^{\mathsf{T}}. \tag{3}$$

By the second premise and the fact that $\|\mathbf{e}_{\mathrm{L}}\| \le \|\mathbf{e}\| \cdot \|\boldsymbol{\Gamma}^{\mathsf{T}}\| \le B$,

$$\begin{aligned}((\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{L}} \bmod q) + \mathbf{e}_{\mathrm{L}}^{\mathsf{T}}) \bmod M &= ((\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{L}} \bmod M) + \mathbf{e}_{\mathrm{L}}^{\mathsf{T}}) \bmod M \\ &= (\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{L}} \bmod M) + \mathbf{e}_{\mathrm{L}}^{\mathsf{T}} = ((\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{L}} \bmod q) \bmod M) + \mathbf{e}_{\mathrm{L}}^{\mathsf{T}}.\end{aligned}$$

Applying Lemma 9 to it,

$$\left\lfloor \frac{(\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{L}} \bmod q) + \mathbf{e}_{\mathrm{L}}^{\mathsf{T}}}{M} \right\rceil = \left\lfloor \frac{\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{L}} \bmod q}{M} \right\rceil. \tag{4}$$

Now we have

$$\begin{aligned}\left\lfloor \frac{\mathbf{v}_{\mathrm{L}}^{\mathsf{T}} \bmod q}{M} \right\rceil &\overset{(2)}{=\!=} \left\lfloor \frac{(\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{L}} - Mx\mathbf{s}^{\mathsf{T}}\mathbf{G}_{\mathrm{L}} + \mathbf{e}_{\mathrm{L}}^{\mathsf{T}}) \bmod q}{M} \right\rceil \overset{(3)}{=\!=} \left\lfloor \frac{(\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{L}} \bmod q) - Mx\mathbf{s}^{\mathsf{T}}\mathbf{G}_{\mathrm{L}} + \mathbf{e}_{\mathrm{L}}^{\mathsf{T}}}{M} \right\rceil \\ &= \left\lfloor \frac{(\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{L}} \bmod q) + \mathbf{e}_{\mathrm{L}}^{\mathsf{T}}}{M} \right\rceil - x\mathbf{s}^{\mathsf{T}}\mathbf{G}_{\mathrm{L}} \overset{(4)}{=\!=} \left\lfloor \frac{\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{L}} \bmod q}{M} \right\rceil - x\mathbf{s}^{\mathsf{T}}\mathbf{G}_{\mathrm{L}}.\end{aligned}$$

This completes the proof for the left part.

For the right part, since $M \mid q$, for all $k, u \in \mathbb{Z}$, it holds that

$$M\left\lfloor \frac{u}{M} \right\rceil \equiv M\left\lfloor \frac{u}{M} \right\rceil + kq = M\left\lfloor \frac{u}{M} + k \cdot \frac{q}{M} \right\rceil \equiv M\left\lfloor \frac{u + kq}{M} \right\rceil \pmod q. \tag{5}$$

Therefore,

$$\begin{aligned}M\left\lfloor \frac{\mathbf{v}_{\mathrm{R}}^{\mathsf{T}} \bmod q}{M} \right\rceil &\overset{(2)}{=\!=} M\left\lfloor \frac{(\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{R}} - x\mathbf{s}^{\mathsf{T}}\mathbf{G}_{\mathrm{R}} + \mathbf{e}_{\mathrm{R}}^{\mathsf{T}}) \bmod q}{M} \right\rceil \\ &\overset{(5)}{=\!=} M\left\lfloor \frac{(\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{R}} \bmod q) - x\mathbf{s}^{\mathsf{T}}\mathbf{G}_{\mathrm{R}} + \mathbf{e}_{\mathrm{R}}^{\mathsf{T}}}{M} \right\rceil \\ \text{(by } M \mid \mathbf{G}_{\mathrm{R}}) \quad &= M\left\lfloor \frac{(\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{R}} \bmod q) + \mathbf{e}_{\mathrm{R}}^{\mathsf{T}}}{M} \right\rceil - x\mathbf{s}^{\mathsf{T}}\mathbf{G}_{\mathrm{R}} \pmod q.\end{aligned}$$

By the second premise, the fact that $\|\mathbf{e}_{\mathrm{R}}\| \le \|\mathbf{e}\| \cdot \|\boldsymbol{\Gamma}^{\mathsf{T}}\| \le B$, and Lemma 9,

$$\left\lfloor \frac{(\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{R}} \bmod q) + \mathbf{e}_{\mathrm{R}}^{\mathsf{T}}}{M} \right\rceil = \left\lfloor \frac{\mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{R}} \bmod q}{M} \right\rceil.$$

This completes the proof for the right part. □

3.2 Bootstrapping

In the previous subsection, we showed how to remove noise from an attribute encoding, yet the noiseless encoding is not amenable to further homomorphism. To support unbounded evaluation, we transform it back to an attribute encoding with smaller noise (magnitude independent of that of the old encoding). This is done with the help of a circular encoding of the secret.

Theorem 12 (bootstrapping; ¶)**.** *It works as follows:*

- *The secret is* $\mathbf{s} = (\mathbf{r}^\mathsf{T}, -1)^\mathsf{T} \in \mathbb{Z}^{n+1}$ *with* $\mathsf{bits}(\mathbf{s}) \in \{0,1\}^{(n+1)\lceil \log_2 q \rceil}$*. The circular ciphertext is*

$$\mathbf{S} = \begin{pmatrix} \overline{\mathbf{A}}_{\mathrm{fhe}} \\ \mathbf{r}^\mathsf{T}\overline{\mathbf{A}}_{\mathrm{fhe}} + \mathbf{e}_{\mathrm{fhe}}^\mathsf{T} \end{pmatrix} (\mathbf{R}_1, \ldots, \mathbf{R}_{(n+1)\lceil \log_2 q \rceil}) - \mathsf{bits}(\mathbf{s}) \otimes \mathbf{G} \in \mathbb{Z}_q^{(n+1)\times m(n+1)\lceil \log_2 q \rceil},$$

  where $\overline{\mathbf{A}}_{\mathrm{fhe}} \in \mathbb{Z}_q^{n\times m}$*,* $\mathbf{e}_{\mathrm{fhe}} \in \mathbb{Z}^m$*, and* $\mathbf{R}_\ell \in \{0,1\}^{m\times m}$ *for* $1 \le \ell \le (n+1)\lceil \log_2 q \rceil$*.*
  Let $L_{\mathrm{S}} = m(n+1)^2\lceil \log_2 q \rceil^2$ *be the bit-length of* $\mathbf{S}$ *so that* $\mathsf{bits}(\mathbf{S}) \in \{0,1\}^{1\times L_{\mathrm{S}}}$*.*

- *The circular encoding is*

$$\mathbf{s}^\mathsf{T}(\mathbf{A}_{\mathrm{circ}} - (1, \mathsf{bits}(\mathbf{S})) \otimes \mathbf{G}) + \mathbf{e}_{\mathrm{circ}}^\mathsf{T},$$

  where $\mathbf{A}_{\mathrm{circ}} \in \mathbb{Z}_q^{(n+1)\times(L_{\mathrm{S}}+1)m}$ *and* $\mathbf{e}_{\mathrm{circ}} \in \mathbb{Z}^{(L_{\mathrm{S}}+1)m}$*.*

- *There are efficient deterministic algorithms*

$$\mathsf{EvalRndPad}(\mathbf{A}_{\mathrm{circ}}, \mathbf{A}) = \mathbf{H}_{\mathbf{A}}^{\mathsf{RndPad}} \qquad \text{and} \qquad \mathsf{EvalRndPadS}(\mathbf{A}_{\mathrm{circ}}, \mathbf{A}, \mathbf{S}) = \mathbf{H}_{\mathbf{A},\mathbf{S}}^{\mathsf{RndPad}}$$

  that take as input $\mathbf{A}_{\mathrm{circ}}$*, a target matrix* $\mathbf{A} \in \mathbb{Z}_q^{(n+1)\times m}$*, and (for* $\mathsf{EvalRndPadS}$*) some* $\mathbf{S}$*, and output some matrix in* $\mathbb{Z}^{(L_{\mathrm{S}}+1)m\times m}$*. It holds that*

$$\|(\mathbf{H}_{\mathbf{A}}^{\mathsf{RndPad}})^\mathsf{T}\|, \|(\mathbf{H}_{\mathbf{A},\mathbf{S}}^{\mathsf{RndPad}})^\mathsf{T}\| \le (m+2)^{\mathrm{O}(\log n \log m \log^2 \log q)} \le 2^{\mathrm{O}(\log^5 \lambda)}.$$

  Moreover, when $\mathbf{S}$ *is indeed of the correct form,*

$$\mathbf{s}^\mathsf{T}(\mathbf{A}_{\mathrm{circ}} - (1, \mathsf{bits}(\mathbf{S})) \otimes \mathbf{G})\mathbf{H}_{\mathbf{A},\mathbf{S}}^{\mathsf{RndPad}} = \mathbf{s}^\mathsf{T}\mathbf{A}_{\mathrm{circ}}\mathbf{H}_{\mathbf{A}}^{\mathsf{RndPad}} - \mathsf{RndPad}_{\mathbf{A}}(\mathbf{s}) + \mathbf{e}_{\mathrm{fhe}}^\mathsf{T}\mathbf{R}_{\mathsf{RndPad}_{\mathbf{A}}},$$

  for $\mathsf{RndPad}_{\mathbf{A}}$ *defined in Construction 1 with*

$$\|\mathbf{R}_{\mathsf{RndPad}_{\mathbf{A}}}^\mathsf{T}\| \le (m+2)^{\mathrm{O}(\log n \log\log q)} \cdot \mathrm{O}(\log q) \le 2^{\mathrm{O}(\log^4 \lambda)}.$$

Before proceeding to the proof, we remark that the procedure, combined with $\mathsf{RemoveNoise}$, indeed reduces the encoding noise. Suppose $\|\mathbf{e}_{\mathrm{fhe}}\|, \|\mathbf{e}_{\mathrm{circ}}\| \le 2^{-\lambda}B$, then the output encoding is

$$\begin{aligned} &(\mathbf{s}^\mathsf{T}(\mathbf{A}_{\mathrm{circ}} - (1, \mathsf{bits}(\mathbf{A})) \otimes \mathbf{G}) + \mathbf{e}_{\mathrm{circ}}^\mathsf{T})\mathbf{H}_{\mathbf{A},\mathbf{S}}^{\mathsf{RndPad}} \\ &= \mathbf{s}^\mathsf{T}\mathbf{A}_{\mathrm{circ}}\mathbf{H}_{\mathbf{A}}^{\mathsf{RndPad}} - \mathsf{RndPad}_{\mathbf{A}}(\mathbf{s}) + \mathbf{e}_{\mathrm{fhe}}^\mathsf{T}\mathbf{R}_{\mathsf{RndPad}_{\mathbf{A}}} + \mathbf{e}_{\mathrm{circ}}^\mathsf{T}\mathbf{H}_{\mathbf{A},\mathbf{S}}^{\mathsf{RndPad}}, \end{aligned}$$

whose noise is bounded by

$$\|\mathbf{e}_{\mathrm{fhe}}\| \cdot \|\mathbf{R}_{\mathsf{RndPad}_{\mathbf{A}}}^\mathsf{T}\| + \|\mathbf{e}_{\mathrm{circ}}\| \cdot \|(\mathbf{H}_{\mathbf{A},\mathbf{S}}^{\mathsf{RndPad}})^\mathsf{T}\| \le B \cdot 2^{-\lambda + \mathrm{poly}(\log \lambda)}.$$

Adding it to the output of $\mathsf{RemoveNoise}$, we restore the format of attribute encoding with the noise level reduced from close to B to much lower below it, which can be done after each step of homomorphic evaluation for unbounded homomorphism.

Proof (Theorem 12). We employ the dual-use technique in Section 2.6. By Lemmas 5 and 6, define

$$\begin{aligned}\mathsf{VEval}_{\mathsf{RndPad}_{\mathbf{A}}} &= \mathsf{MakeVEvalCkt}(n, m, q, \mathsf{RndPad}_{\mathbf{A}}),\\ \mathsf{EvalRndPad}(\mathbf{A}_{\mathrm{circ}}, \mathbf{A}) &= \mathsf{MEvalC}(\mathbf{A}_{\mathrm{circ}}, \mathsf{VEval}_{\mathsf{RndPad}_{\mathbf{A}}}),\\ \mathsf{EvalRndPadS}(\mathbf{A}_{\mathrm{circ}}, \mathbf{A}, \mathbf{S}) &= \mathsf{MEvalCX}(\mathbf{A}_{\mathrm{circ}}, \mathsf{VEval}_{\mathsf{RndPad}_{\mathbf{A}}}, \mathsf{bits}(\mathbf{S})),\end{aligned}$$

all of which can be efficiently computed. By Lemma 10, the depth of $\mathsf{RndPad}_{\mathbf{A}}$ is $\mathrm{O}(\log n \log\log q)$, so by Lemma 5, the circuit $\mathsf{VEval}_{\mathsf{RndPad}_{\mathbf{A}}}$ is of depth

$$\mathrm{O}(\log n \log\log q) \cdot \mathrm{O}(\log m \log\log q) + \mathrm{O}(\log^2 \log q) = \mathrm{O}(\log n \log m \log^2 \log q).$$

The desired norm bounds of $\mathbf{H}_{\mathbf{A}}^{\mathsf{RndPad}}$ and $\mathbf{H}_{\mathbf{A},\mathbf{S}}^{\mathsf{RndPad}}$ follow by Lemma 6.

When $\mathbf{S}$ is in the correct form,

$$\begin{aligned}&\mathbf{s}^{\mathsf{T}}(\mathbf{A}_{\mathrm{circ}} - (1, \mathsf{bits}(\mathbf{S})) \otimes \mathbf{G})\mathbf{H}_{\mathbf{A},\mathbf{S}}^{\mathsf{RndPad}}\\ \text{(Lemma 6)}\quad &= \mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{circ}}\mathbf{H}_{\mathbf{A}}^{\mathsf{RndPad}} - \mathbf{s}^{\mathsf{T}}\mathsf{VEval}_{\mathsf{RndPad}_{\mathbf{A}}}(\mathbf{S})\\ \text{(Lemma 5)}\quad &= \mathbf{s}^{\mathsf{T}}\mathbf{A}_{\mathrm{circ}}\mathbf{H}_{\mathbf{A}}^{\mathsf{RndPad}} - \mathsf{RndPad}_{\mathbf{A}}(\mathbf{s}) + \mathbf{e}_{\mathrm{fhe}}^{\mathsf{T}}\mathbf{R}_{\mathsf{RndPad}_{\mathbf{A}}}.\end{aligned}$$

Lastly, $\mathbf{R}_{\mathsf{RndPad}_{\mathbf{A}}}$ satisfies (again by Lemma 5)

$$\begin{aligned}\|\mathbf{R}_{\mathsf{RndPad}_{\mathbf{A}}}^{\mathsf{T}}\| &\le (m+2)^{\mathrm{O}(\log n \log\log q)\lceil \log_2 q\rceil} \cdot \overbrace{\max_{1\le \ell \le (n+1)\lceil \log_2 q\rceil} \|\mathbf{R}_{\ell}^{\mathsf{T}}\|}^{\le m \le m+2}\\ &\le (m+2)^{\mathrm{O}(\log n \log\log q)} \cdot \mathrm{O}(\log q).\end{aligned}$$

□

3.3 Unbounded Homomorphic Evaluation

In this subsection, we present homomorphic evaluation for circuits of unbounded depth. Unlike Lemma 6, our evaluation procedure is non-linear, and we cannot formulate them as producing low-norm linear transformation.

Construction 2 (unbounded homomorphic evaluation of attribute encoding)**.** It works as follows.

- $\mathsf{UEvalC}(\mathbf{A}_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, C)$ takes as input the public matrices $\mathbf{A}_{\mathrm{attr}} \in \mathbb{Z}_q^{(n+1)\times(L+1)m}$ (for attribute encoding) and $\mathbf{A}_{\mathrm{circ}} \in \mathbb{Z}_q^{(n+1)\times(L_{\mathrm{S}}+1)m}$ (for circular encoding, with $L_{\mathrm{S}} = m(n+1)^2\lceil \log_2 q\rceil^2$), and a circuit $C : \{0,1\}^L \to \{0,1\}^{1\times L'}$ of arbitrary size and depth. It does the following:

  1. Let C_ℓ (for $\ell \in [L]$) be the input gate of C corresponding to $\mathbf{x}[\ell]$, and $C_{L+1}, \ldots, C_{|C|}$ the other gates of C in topological order. Let $\mathbf{A}_{\mathrm{attr}} = (\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_L)$, where $\mathbf{A}_0$ corresponds to the constant 1 and $\mathbf{A}_\ell$ (for $\ell \in [L]$), the input $\mathbf{x}[\ell]$, in the attribute encoding.
  2. For $i = L+1, \ldots, |C|$, let fan-ins of C_i be fan-outs of $C_{i'}, C_{i''}$ for some $i', i'' < i$ (for negation gates, ignore i''). Use Lemma 6 to compute[17]

     $$\mathbf{H}'_{C_i} \leftarrow \mathsf{EvalC}((\mathbf{A}_0, \mathbf{A}_{i'}, \mathbf{A}_{i''}), \text{gate } C_i \text{ as one-gate circuit}), \qquad \mathbf{A}'_i \leftarrow (\mathbf{A}_0, \mathbf{A}_{i'}, \mathbf{A}_{i''})\mathbf{H}'_{C_i},$$

     then use Theorem 12 to compute

     $$\mathbf{H}_{\mathbf{A}'_i}^{\mathsf{RndPad}} \leftarrow \mathsf{EvalRndPad}(\mathbf{A}_{\mathrm{circ}}, \mathbf{A}'_i), \qquad \mathbf{A}_i \leftarrow \mathbf{A}_{\mathrm{circ}}\mathbf{H}_{\mathbf{A}'_i}^{\mathsf{RndPad}}.$$

[17] The notations are different from those used in the technical overview. Here, primes are put on matrices before bootstrapping (an implementation detail) so that matrices defined for gates (the interface if the gate is an output) do not use primes (contrary to technical overview, where primes are attached to matrices appearing later in text).

3. Let $i_1, \ldots, i_{L'}$ be the indices of the output gates. Set $\mathbf{A}_{C:\ell} \leftarrow \mathbf{A}_{i_\ell}$ for all $\ell \in [L']$ and output $\mathbf{A}_C = (\mathbf{A}_{C:1}, \ldots, \mathbf{A}_{C:L'})$.

- $\mathsf{UEvalCX}(\mathbf{A}_{\mathrm{attr}}, \mathbf{c}_{\mathrm{attr}}^\intercal, \mathbf{A}_{\mathrm{circ}}, \mathbf{c}_{\mathrm{circ}}^\intercal, C, \mathbf{x}, \mathbf{S})$ takes as input all the input to UEvalC, attribute encoding $\mathbf{c}_{\mathrm{attr}} \in \mathbb{Z}_q^{(L+1)m}$, circular encoding $\mathbf{c}_{\mathrm{circ}} \in \mathbb{Z}_q^{(L_{\mathrm{S}}+1)m}$, attribute $\mathbf{x} \in \{0,1\}^L$, and circular ciphertext $\mathbf{S} \in \mathbb{Z}_q^{(n+1)\times m(n+1)\lceil \log_2 q \rceil}$. It does the following:

  1. Parse C into $C_1, \ldots, C_L, C_{L+1}, \ldots, C_{|C|}$ and $\mathbf{A}_{\mathrm{attr}}$ into $(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_L)$ as in UEvalC. Split $\mathbf{c}_{\mathrm{attr}}^\intercal$ into $(\mathbf{c}_0^\intercal, \mathbf{c}_1^\intercal, \ldots, \mathbf{c}_L^\intercal)$, where $\mathbf{c}_0$ is the encoding of constant 1 and $\mathbf{c}_\ell$ (for $\ell \in [L]$), that of input $\mathbf{x}[\ell]$.
  2. For $i = L+1, \ldots, |C|$, let fan-ins of C_i be fan-outs of $C_{i'}, C_{i''}$ for some $i', i'' < i$ (for negation gates, ignore i''), and $\mathbf{x}[i'], \mathbf{x}[i'']$ the wire values out of $C_{i'}, C_{i''}$. Evaluate the gate and store the result into $\mathbf{x}[i]$ (extending $\mathbf{x}$). Use Lemma 6 to compute

  $$\mathbf{H}'_{C_i,\mathbf{x}[i'],\mathbf{x}[i'']} \leftarrow \mathsf{EvalCX}((\mathbf{A}_0, \mathbf{A}_{i'}, \mathbf{A}_{i''}), \text{gate } C_i \text{ as one-gate circuit}, (\mathbf{x}[i'], \mathbf{x}[i''])),$$

  and set $(\mathbf{c}'_i)^\intercal \leftarrow (\mathbf{c}_0^\intercal, \mathbf{c}_{i'}^\intercal, \mathbf{c}_{i''}^\intercal)\mathbf{H}'_{C_i,\mathbf{x}[i'],\mathbf{x}[i'']}$. Next, run Construction 1,

  $$(\mathbf{c}''_i)^\intercal \leftarrow \mathsf{RemoveNoise}((\mathbf{c}'_i)^\intercal).$$

  Then, use Theorem 12 to compute

  $$\mathbf{H}^{\mathsf{RndPad}}_{\mathbf{A}'_i,\mathbf{S}} \leftarrow \mathsf{EvalRndPad}(\mathbf{A}_{\mathrm{circ}}, \mathbf{A}'_i, \mathbf{S}),$$

  and set $\mathbf{c}_i^\intercal \leftarrow (\mathbf{c}''_i)^\intercal + \mathbf{c}_{\mathrm{circ}}^\intercal \mathbf{H}^{\mathsf{RndPad}}_{\mathbf{A}'_i,\mathbf{S}}$.
  3. Let $i_1, \ldots, i_{L'}$ be the indices of the output gates. Set $\mathbf{c}_{C,\mathbf{x}:\ell} \leftarrow \mathbf{c}_{i_\ell}$ for all $\ell \in [L']$ and output $\mathbf{c}_{C,\mathbf{x}}^\intercal = (\mathbf{c}_{C,\mathbf{x}:1}^\intercal, \ldots, \mathbf{c}_{C,\mathbf{x}:L'}^\intercal)$.

We expect the homomorphism property — with overwhelming probability over the input, the output satisfies $\mathbf{c}_{C,\mathbf{x}}^\intercal = \mathbf{s}^\intercal(\mathbf{A}_C - C(\mathbf{x}) \otimes \mathbf{G}) + \mathbf{e}_{C,\mathbf{x}}^\intercal$ for small $\mathbf{e}_{C,\mathbf{x}}$. We consider this in two steps. We first show a sufficient condition (barring "bad event") for correctness, and then prove that "bad event" happens with negligible probability.

Theorem 13 (¶). *Let λ be sufficiently large. For $C : \{0,1\}^L \to \{0,1\}^{1\times L'}$ and $\mathbf{x} \in \{0,1\}^L$, suppose*

$$\mathbf{R} \in \{0,1\}^{m\times m(n+1)\lceil \log_2 q \rceil}, \qquad \mathbf{S} = \begin{pmatrix} \overline{\mathbf{A}}_{\mathrm{fhe}} \\ \mathbf{s}^\intercal \overline{\mathbf{A}}_{\mathrm{fhe}} + \mathbf{e}_{\mathrm{fhe}}^\intercal \end{pmatrix} \mathbf{R} - \mathsf{bits}(\mathbf{s}) \otimes \mathbf{G},$$

$$\mathbf{c}_{\mathrm{attr}}^\intercal = \mathbf{s}^\intercal(\mathbf{A}_{\mathrm{attr}} + (1, \mathbf{x}^\intercal) \otimes \mathbf{G}) + \mathbf{e}_{\mathrm{attr}}^\intercal, \qquad \mathbf{c}_{\mathrm{circ}}^\intercal = \mathbf{s}^\intercal(\mathbf{A}_{\mathrm{circ}} + (1, \mathsf{bits}(\mathbf{S})) \otimes \mathbf{G}) + \mathbf{e}_{\mathrm{circ}}^\intercal,$$

$\|\mathbf{s}\| \le B$, and $\|\mathbf{e}_{\mathrm{fhe}}\|, \|\mathbf{e}_{\mathrm{circ}}\|, \|\mathbf{e}_{\mathrm{circ}}\| \le 2^{-\lambda} B$. Define $\mathbf{\Gamma} = \mathbf{G}^{-1}(M\mathbf{G}_{\mathrm{L}}, \mathbf{G}_{\mathrm{R}})$ as in Theorem 11 and let $\mathbf{A}_i, \mathbf{A}'_i$ be those in Construction 2. If both premises

$$(\mathbf{s}^\intercal \mathbf{A}'_i \mathbf{\Gamma} \bmod q) \in \left[-\frac{q}{2} + M^2 B, \frac{q}{2} - M^2 B\right)^m \quad \text{and} \quad (\mathbf{s}^\intercal \mathbf{A}'_i \mathbf{\Gamma} \bmod M) \in \left[-\frac{M}{2} + B, \frac{M}{2} - B\right)^m$$

hold for all $i \in [L+1, |C|]$, then the output

$$\mathbf{A}_C = \mathsf{UEvalC}(\mathbf{A}_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, C)$$
$$\text{and} \quad \mathbf{c}_{C,\mathbf{x}}^\intercal = \mathsf{UEvalCX}(\mathbf{A}_{\mathrm{attr}}, \mathbf{c}_{\mathrm{attr}}^\intercal, \mathbf{A}_{\mathrm{circ}}, \mathbf{c}_{\mathrm{circ}}^\intercal, C, \mathbf{x}, \mathbf{S}) = \mathbf{s}^\intercal(\mathbf{A}_C - C(\mathbf{x}) \otimes \mathbf{G}) + \mathbf{e}_{C,\mathbf{x}}^\intercal$$

satisfy $\|\mathbf{e}_{C,\mathbf{x}}\| \le B$.

Proof (Theorem 13). Evaluate $C(\mathbf{x})$ and store the intermediate wire values into the extended $\mathbf{x}$ as done in $\mathsf{UEvalCX}$. It suffices to show the stronger statement that for all $i \in [0, |C|]$,

$$\|\mathbf{e}_i\| \le 2^{-\lambda/2}B \qquad \text{with} \quad \mathbf{c}_i^\mathsf{T} = \mathbf{s}^\mathsf{T}(\mathbf{A}_i - \mathbf{x}[i]\mathbf{G}) + \mathbf{e}_i^\mathsf{T}.$$

We prove it by induction on i. The basis case of $i \in [0, L]$ is by assumption. For the inductive case, suppose $i \in [L+1, |C|]$. By the definitions of $\mathsf{UEvalC}, \mathsf{UEvalCX}$ and Lemma 6,

$$\begin{aligned}
(\mathbf{c}'_i)^\mathsf{T} &= (\mathbf{c}_0^\mathsf{T}, \mathbf{c}_{i'}^\mathsf{T}, \mathbf{c}_{i''}^\mathsf{T})\mathbf{H}'_{C_i,\mathbf{x}[i'],\mathbf{x}[i'']} \\
&= \big(\mathbf{s}^\mathsf{T}\big((\mathbf{A}_0, \mathbf{A}_{i'}, \mathbf{A}_{i''}) - (1, \mathbf{x}[i'], \mathbf{x}[i'']) \otimes \mathbf{G}\big) + (\mathbf{e}_0^\mathsf{T}, \mathbf{e}_{i'}^\mathsf{T}, \mathbf{e}_{i''}^\mathsf{T})\big)\mathbf{H}'_{C_i,\mathbf{x}[i'],\mathbf{x}[i'']} \\
&= \mathbf{s}^\mathsf{T}(\mathbf{A}'_i - \mathbf{x}[i]\mathbf{G}) + \underbrace{(\mathbf{e}_0^\mathsf{T}, \mathbf{e}_{i'}^\mathsf{T}, \mathbf{e}_{i''}^\mathsf{T})\mathbf{H}'_{C_i,\mathbf{x}[i'],\mathbf{x}[i'']}}_{(\mathbf{e}'_i)^\mathsf{T}}.
\end{aligned}$$

Since a single gate C_i is of depth 1, by Lemma 6, it holds that $\|\mathbf{H}'_{C_i,\mathbf{x}[i'],\mathbf{x}[i'']}\| \le m + 2$, and

$$\|\mathbf{e}'_i\| \le \max\{\|\mathbf{e}_0\|, \|\mathbf{e}_{i'}\|, \|\mathbf{e}_{i''}\|\} \cdot \|\mathbf{H}^\mathsf{T}_{C_i,\mathbf{x}[i'],\mathbf{x}[i'']}\| \le 2^{-\lambda/2}B \cdot (m+2) < B,$$

where the second-last inequality invokes the induction hypothesis on $0, i', i'' < i$. Combining the premises with $\|\mathbf{e}'_i\|, \|\mathbf{s}\| \le B$, Theorem 11 yields

$$(\mathbf{c}''_i)^\mathsf{T} = \mathsf{RemoveNoise}((\mathbf{c}'_i)^\mathsf{T}) = \mathsf{RemoveNoise}(\mathbf{s}^\mathsf{T}(\mathbf{A}'_i - \mathbf{x}[i]\mathbf{G}) + (\mathbf{e}'_i)^\mathsf{T}) = \mathsf{RndPad}_{\mathbf{A}'_i}(\mathbf{s}) - \mathbf{x}[i]\mathbf{s}^\mathsf{T}\mathbf{G}.$$

By Theorem 12, we have

$$\begin{aligned}
\mathbf{c}_{\mathrm{circ}}^\mathsf{T}\mathbf{H}^{\mathsf{RndPad}}_{\mathbf{A}'_i,\mathbf{S}} &= (\mathbf{s}^\mathsf{T}(\mathbf{A}_{\mathrm{circ}} - (1, \mathsf{bits}(\mathbf{S})) \otimes \mathbf{G}) + \mathbf{e}_{\mathrm{circ}}^\mathsf{T})\mathbf{H}^{\mathsf{RndPad}}_{\mathbf{A}'_i,\mathbf{S}} \\
&= \mathbf{s}^\mathsf{T}\mathbf{A}_{\mathrm{circ}}\mathbf{H}^{\mathsf{RndPad}}_{\mathbf{A}'_i} - \mathsf{RndPad}_{\mathbf{A}'_i}(\mathbf{s}) + \mathbf{e}_{\mathrm{fhe}}^\mathsf{T}\mathbf{R}_{\mathsf{RndPad}_{\mathbf{A}'_i}} + \mathbf{e}_{\mathrm{circ}}^\mathsf{T}\mathbf{H}^{\mathsf{RndPad}}_{\mathbf{A}'_i,\mathbf{S}}.
\end{aligned}$$

Because we set $\mathbf{A}_i = \mathbf{A}_{\mathrm{circ}}\mathbf{H}^{\mathsf{RndPad}}_{\mathbf{A}'_i}$, it follows that

$$\begin{aligned}
\mathbf{e}_i^\mathsf{T} &= \mathbf{c}_i^\mathsf{T} - \mathbf{s}^\mathsf{T}(\mathbf{A}_i - \mathbf{x}[i]\mathbf{G}) = (\mathbf{c}''_i)^\mathsf{T} + \mathbf{c}_{\mathrm{circ}}^\mathsf{T}\mathbf{H}^{\mathsf{RndPad}}_{\mathbf{A}'_i,\mathbf{S}} + (-\mathbf{s}^\mathsf{T}\mathbf{A}_i + \mathbf{x}[i]\mathbf{s}^\mathsf{T}\mathbf{G}) \\
&= (\mathsf{RndPad}_{\mathbf{A}'_i}(\mathbf{s}) - \mathbf{x}[i]\mathbf{s}^\mathsf{T}\mathbf{G}) + \big(\mathbf{s}^\mathsf{T}\mathbf{A}_{\mathrm{circ}}\mathbf{H}^{\mathsf{RndPad}}_{\mathbf{A}'_i} - \mathsf{RndPad}_{\mathbf{A}'_i}(\mathbf{s}) \\
&\qquad + \mathbf{e}_{\mathrm{fhe}}^\mathsf{T}\mathbf{R}_{\mathsf{RndPad}_{\mathbf{A}'_i}} + \mathbf{e}_{\mathrm{circ}}^\mathsf{T}\mathbf{H}^{\mathsf{RndPad}}_{\mathbf{A}'_i,\mathbf{S}}\big) + (-\mathbf{s}^\mathsf{T}\mathbf{A}_i + \mathbf{x}[i]\mathbf{s}^\mathsf{T}\mathbf{G}) \\
&= \mathbf{e}_{\mathrm{fhe}}^\mathsf{T}\mathbf{R}_{\mathsf{RndPad}_{\mathbf{A}'_i}} + \mathbf{e}_{\mathrm{circ}}^\mathsf{T}\mathbf{H}^{\mathsf{RndPad}}_{\mathbf{A}'_i,\mathbf{S}}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\|\mathbf{e}_i\| &\le \|\mathbf{e}_{\mathrm{fhe}}\| \cdot \|\mathbf{R}^\mathsf{T}_{\mathsf{RndPad}_{\mathbf{A}'_i}}\| + \|\mathbf{e}_{\mathrm{circ}}\| \cdot \|(\mathbf{H}^{\mathsf{RndPad}}_{\mathbf{A}'_i,\mathbf{S}})^\mathsf{T}\| \\
&\le 2^{-\lambda}B \cdot 2^{\mathrm{O}(\log^4\lambda)} + 2^{-\lambda}B \cdot 2^{\mathrm{O}(\log^5\lambda)} < 2^{-\lambda/2}B,
\end{aligned}$$

where the second-last inequality uses Theorem 12 and the assumption that $\|\mathbf{e}_{\mathrm{fhe}}\|, \|\mathbf{e}_{\mathrm{circ}}\| \le 2^{-\lambda}B$. This completes the induction proof. □

Assuming small-secret LWE, the premises of Theorem 13 hold with overwhelming probability:

Theorem 14 (¶). *Let $\mathcal{A}$ be an efficient adversary choosing a circuit $C : \{0,1\}^L \to \{0,1\}^{1\times L'}$, and $\overline{L}$ a polynomial upper bound of L that could ever be chosen by $\mathcal{A}$. Under $\mathsf{sLWE}_{n,(\overline{L}+L_S+2)m,q,\sigma,\sigma'}$ (Assumption 1) and our choice of parameters,*

$$\Pr\left[\begin{array}{l} C \xleftarrow{\$} \mathcal{A}() \\ \mathbf{A}_{\mathrm{attr}} \xleftarrow{\$} \mathbb{Z}_q^{(n+1)\times(L+1)m} \\ \mathbf{A}_{\mathrm{circ}} \xleftarrow{\$} \mathbb{Z}_q^{(n+1)\times(L_S+1)m} \\ \mathbf{r} \xleftarrow{\$} \mathcal{D}^n_{\mathbb{Z},\sigma,\le\sigma\sqrt{\lambda}} \end{array} : \begin{array}{l} \text{there exists } i \in [L+1,|C|] \text{ such that} \\ (\mathbf{s}^\intercal \mathbf{A}_i' \mathbf{\Gamma} \bmod q) \notin \left[-\frac{q}{2}+M^2B, \frac{q}{2}-M^2B\right)^m \\ \text{or } (\mathbf{s}^\intercal \mathbf{A}_i' \mathbf{\Gamma} \bmod M) \notin \left[-\frac{M}{2}+B, \frac{M}{2}-B\right)^m \end{array}\right]$$

is negligible, where $\mathbf{A}_i, \mathbf{A}_i'$ are those in Construction 2.

Combined with Theorem 13 and the fact that $\|\mathbf{r}\| \le \sigma\sqrt{\lambda} \le 2^{-\lambda}B$ (by our choice of parameters), it follows as a corollary that

$$\Pr\left[\begin{array}{l} \text{for all } \mathbf{e}_{\mathrm{fhe}}, \mathbf{e}_{\mathrm{attr}}, \mathbf{e}_{\mathrm{circ}} \text{ such that } \|\mathbf{e}_{\mathrm{fhe}}\|, \|\mathbf{e}_{\mathrm{attr}}\|, \|\mathbf{e}_{\mathrm{circ}}\| \le 2^{-\lambda}B, \\ \quad \text{all } \mathbf{R} \in \{0,1\}^{m\times m(n+1)\lceil\log_2 q\rceil}, \text{ and all } \mathbf{x} \in \{0,1\}^L, \\ \qquad\qquad \text{it holds that } \|\mathbf{e}_{C,\mathbf{x}}\| \le B. \end{array}\right]$$

is overwhelming, where the probability is taken over $C, \mathbf{A}_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, \mathbf{r}$ as in the previous one and the variables follow the relations in Theorem 13.

We emphasize that the correctness is strong in the sense that it does *not* depend on the randomness of $\mathbf{e}$'s or $\mathbf{R}$, nor the choice of $\mathbf{x}$ — only reliant on $\mathbf{s}$ (i.e, $\mathbf{r}$) and $\mathbf{A}$'s following the right distribution. Moreover, in case of non-uniform assumptions, it holds for any C of polynomially bounded size (by letting the "worst" C be the advice of $\mathcal{A}$).

Proof (Theorem 14). Augment the probability space in the statement with $i^* \xleftarrow{\$} [L+1,|C|]$ and $j^* \xleftarrow{\$} [m]$, define events (with i, j running through $[L+1,|C|]$ and $[m]$)

$$\begin{aligned} E_{i,j,1} &: \quad ((\mathbf{s}^\intercal \mathbf{A}_i' \mathbf{\Gamma})[1,j] \bmod q) \notin \left[-\frac{q}{2}+M^2B, \frac{q}{2}-M^2B\right), \\ E_{i,j,2} &: \quad ((\mathbf{s}^\intercal \mathbf{A}_i' \mathbf{\Gamma})[1,j] \bmod M) \notin \left[-\frac{M}{2}+B, \frac{M}{2}-B\right), \\ E &: \quad \bigvee_{i=L+1}^{|C|} \bigvee_{j=1}^{m} (E_{i,0} \vee E_{i,1}), \end{aligned}$$

and let $\overline{C} > 0$ be a polynomial upper bound of the size of C that could ever be chosen by $\mathcal{A}$, then

$$\begin{aligned} & \Pr[\, E_{i^*,j^*,1} \vee E_{i^*,j^*,2} \mid E, C \,] \ge 1 \,/\, (|C| - L)m \ge 1 \,/\, \overline{C}m \\ \Longrightarrow \quad & \Pr[\, E_{i^*,j^*,1} \vee E_{i^*,j^*,2} \,] = \mathbb{E}\big[\, \Pr[\, E_{i^*,j^*,1} \vee E_{i^*,j^*,2} \mid C \,] \,\big] \\ & \qquad\qquad \ge \mathbb{E}\left[\, \Pr[\, E \mid C \,] \,/\, \overline{C}m \,\right] = \Pr[E] \,/\, \overline{C}m \\ \Longrightarrow \quad & \Pr[E] \le \overline{C}m \Pr[\, E_{i^*,j^*,1} \vee E_{i^*,j^*,2} \,] \\ & \qquad\qquad \le \overline{C}m \Pr[E_{i^*,j^*,1}] + \overline{C}m \Pr[E_{i^*,j^*,2}]. \end{aligned}$$

It suffices to show that both $\Pr[E_{i^*,j^*,1}]$ and $\Pr[E_{i^*,j^*,2}]$ are negligible. We prove it by reduction to Lemma 8. Let $m' = (\overline{L} + L_S + 2)m$ and define $h : \mathbb{Z}_q^{(n+1)\times m'} \xrightarrow{\$} \mathbb{Z}^{m'}$ as follows. On input $\mathbf{A}$:

1. Sample $C \xleftarrow{\$} \mathcal{A}()$ to obtain $C : \{0,1\}^L \to \{0,1\}^{1\times L'}$. Sample $i^* \xleftarrow{\$} [L+1, |C|]$ and $j^* \xleftarrow{\$} [m]$.
2. Parse $\mathbf{A}$ as $(\mathbf{A}_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, \dots)$, where $\mathbf{A}_{\mathrm{attr}} \in \mathbb{Z}_q^{(n+1)\times(L+1)m}$ and $\mathbf{A}_{\mathrm{circ}} \in \mathbb{Z}_q^{(n+1)\times(L_{\mathrm{S}}+1)m}$, with the rest ("...") ignored.
3. Run $\mathsf{UEvalC}(\mathbf{A}_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, C)$ and use the intermediate values to find (low-norm) $\mathbf{H}$ such that $\mathbf{AH} = \mathbf{A}'_{i^*}$. Let i', i'' be the fan-in indices of C_{i^*}, then $\mathbf{A}'_{i^*} = (\mathbf{A}_0, \mathbf{A}_{i'}, \mathbf{A}_{i''})\mathbf{H}'_{C_{i^*}}$. Here, $\mathbf{A}_0$ is part of $\mathbf{A}_{\mathrm{attr}}$, and the first third (by rows) of $\mathbf{H}'_{C_{i^*}}$ becomes part of $\mathbf{H}$. When $i' \le L$, the matrix $\mathbf{A}_{i'}$ is again part of $\mathbf{A}_{\mathrm{attr}}$. When $i' > L$, by definition, $\mathbf{A}_{i'} = \mathbf{A}_{\mathrm{circ}}\mathbf{H}^{\mathsf{RndPad}}_{\mathbf{A}'_{i'}}$, and $\mathbf{H}^{\mathsf{RndPad}}_{\mathbf{A}'_{i'}}$ multiplied by the second third of $\mathbf{H}'_{C_{i^*}}$ contributes to $\mathbf{H}$. Similar handling applies to i'', and most parts of $\mathbf{H}$ are zero-padded.
4. Split $\mathbf{H\Gamma}$ by column into $(\mathbf{h}_1, \dots, \mathbf{h}_m)$ and output $\mathbf{h} = \mathbf{h}_{j^*}$.

Clearly, h is efficient. Recall that B' is $2^{\Omega(\log^6 \lambda)}$ in our choice of parameters. By construction,

$$\|\mathbf{h}^\mathsf{T}\| \le \|\mathbf{H}^\mathsf{T}\| \cdot \|\mathbf{\Gamma}^\mathsf{T}\| \le 3 \cdot 2^{\mathrm{O}(\log^4 \lambda)} \cdot (m+2) \cdot 1 \le B',$$

where the factors come from addition among the three parts, $\mathbf{H}^{\mathsf{RndPad}}_{\mathbf{A}'_{i'}}$, $\mathbf{H}'_{C_{i^*}}$, $\mathbf{\Gamma}$, respectively. Note also that $\mathbf{s}^\mathsf{T}\mathbf{h} = (\mathbf{s}^\mathsf{T}\mathbf{A}'_{i^*}\mathbf{\Gamma})[1, j^*]$. Invoking Lemma 8 on h with its (p, B) being our (q, M^2B) and (M, B) yields that under $\mathsf{sLWE}_{n,m',q,\sigma,\sigma'}$,

$$\Pr[E_{i^*,j^*,1}] \le \frac{2M^2B + (2\sigma'\sqrt{\lambda}+1)B'}{q} + \mathrm{negl}(\lambda), \quad \Pr[E_{i^*,j^*,2}] \le \frac{2B + (2\sigma'\sqrt{\lambda}+1)B'}{M} + \mathrm{negl}(\lambda),$$

both of which are negligible under our choice of parameters. This completes the proof. □

3.4 Stronger Correctness

The primary version in the previous section requires that $\mathbf{A}$'s be chosen independently of C. It can be avoided by applying a random shift to each $\mathsf{RemoveNoise}$. We explain the modifications:

- In Construction 1, define $\mathsf{RemoveNoise}'(\mathbf{u}^\mathsf{T}, \mathbf{w}^\mathsf{T})$ to be

$$\left\lceil \frac{\mathbf{u}^\mathsf{T}\mathbf{G}^{-1}(M\mathbf{G}_{\mathrm{L}}, \mathbf{G}_{\mathrm{R}}) + \mathbf{w}^\mathsf{T}}{M} \right\rfloor \begin{pmatrix} \mathbf{I} & \\ & M\mathbf{I} \end{pmatrix} \mathbf{Q},$$

  and define $\mathsf{RndPad}'_{\mathbf{A},\mathbf{w}}(\mathbf{s})$ to be $\mathsf{RemoveNoise}'(\mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{w}^\mathsf{T})$. In all the succeeding constructions, propositions, and proofs, replace $\mathsf{RemoveNoise}, \mathsf{RndPad}$ by $\mathsf{RemoveNoise}', \mathsf{RndPad}'$.
- In Construction 2, the algorithm UEvalC is randomized. It samples $\mathbf{w}_{\mathbf{A}'_i}$ for each $\mathbf{A}'_i$ and outputs the information necessary to recover them. The algorithm $\mathsf{UEvalCX}$ additionally takes that information as input so that it can use the same $\mathbf{w}$'s. The $\mathbf{w}$'s can be either truly random or pseudorandom — in the latter case, the information can be just a PRF key of length some fixed $\mathrm{poly}(\lambda)$.
- In Theorems 11 and 13, change $\mathbf{s}^\mathsf{T}\mathbf{A\Gamma}$ to $(\mathbf{s}^\mathsf{T}\mathbf{A\Gamma} + \mathbf{w}^\mathsf{T})$.
- In Theorem 14, the adversary $\mathcal{A}$ is allowed to choose C and $\mathbf{A}_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, \mathbf{r}$, but not $\mathbf{w}$'s. The probability is taken over the randomness of $\mathcal{A}$ and $\mathbf{w}$'s.

– If $\mathbf{w}$'s are truly random, the statement holds unconditionally against worst-case $C, \mathbf{A}_{\mathrm{attr}}$, $\mathbf{A}_{\mathrm{circ}}, \mathbf{r}$ of polynomial total length.
– If $\mathbf{w}$'s are pseudorandom (against uniform adversaries), it holds against uniform $\mathcal{A}$'s.
– If $\mathbf{w}$'s are pseudorandom against non-uniform adversaries, it holds against worst-case C, $\mathbf{A}_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, \mathbf{r}$ of polynomial total length.

The stronger correctness makes the applications *adaptively* (instead of only *selectively*) correct and enables proofs of *selective* (instead of only *very selective*) security.

4 Applications

In this section, we show how our new homomorphic evaluation algorithm helps achieving various primitives for circuits of unbounded depth. This includes laconic function evaluation, 1-key secure attribute-based encryption, 1-key secure functional encryption, and reusable garbling schemes.

4.1 Laconic Function Evaluation

The (AB-)LFE scheme of [QWW18a] is obtained by modifying the ABE scheme of [BGG+14] and its proof does not rely on the techniques for multi-key security. Essentially the same construction using our unbounded homomorphic evaluation yields AB-LFE for circuits of unbounded depth.

Construction 3 (AB-LFE)**.** Our AB-LFE for circuits of unbounded depth works as follows:

- $\mathsf{GenCRS}(1^L)$ takes the attribute length L as input. It samples

$$\mathbf{A}_{\mathrm{attr}} \xleftarrow{\$} \mathbb{Z}_q^{(n+1)\times(L+1)m}, \qquad \mathbf{A}_{\mathrm{circ}} \xleftarrow{\$} \mathbb{Z}_q^{(n+1)\times(L_{\mathrm{S}}+1)m},$$

  and outputs $\mathsf{crs} = (\mathbf{A}_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}})$.

- $\mathsf{Compress}(\mathsf{crs}, C)$ takes as input crs and a circuit $C : \{0,1\}^L \to \{0,1\}^{1\times L'}$. It runs

$$\mathsf{digest}_C = (\mathbf{A}_{C:1}, \ldots, \mathbf{A}_{C:L'}) = \mathbf{A}_C \leftarrow \mathsf{UEvalC}(\mathbf{A}_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, C),$$

  and outputs $\mathsf{param}_C = (L, L')$ and digest_C.

- $\mathsf{Enc}(\mathsf{crs}, \mathsf{param}_C, \mathsf{digest}_C, (\mathbf{x}, \boldsymbol{\mu}))$ takes as input crs, param_C, digest_C, an attribute $\mathbf{x} \in \{0,1\}^L$, and a multi-bit message $\boldsymbol{\mu} \in \{0,1\}^{L'}$. It operates in the two-phase manner.

  1. $\mathsf{EncX}(\mathsf{crs}, (\mathbf{x}, \boldsymbol{\mu}))$ samples $\mathbf{r} \xleftarrow{\$} \mathcal{D}^n_{\mathbb{Z},\sigma,\le\sigma\sqrt{\lambda}}$ and sets $\mathbf{s} \leftarrow (\mathbf{r}^\intercal, -1)^\intercal$. The algorithm creates a circular encryption by

$$\overline{\mathbf{A}}_{\mathrm{fhe}} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \qquad \mathbf{e}_{\mathrm{fhe}} \xleftarrow{\$} \mathcal{D}^m_{\mathbb{Z},\sigma,\le\sigma\sqrt{\lambda}}, \qquad \mathbf{R} \xleftarrow{\$} \{0,1\}^{m\times m(n+1)\lceil\log_2 q\rceil},$$
$$\mathbf{A}_{\mathrm{fhe}} \leftarrow \begin{pmatrix} \overline{\mathbf{A}}_{\mathrm{fhe}} \\ \mathbf{r}^\intercal\overline{\mathbf{A}}_{\mathrm{fhe}} + \mathbf{e}_{\mathrm{fhe}}^\intercal \end{pmatrix}, \qquad \mathbf{S} \leftarrow \mathbf{A}_{\mathrm{fhe}}\mathbf{R} - \mathsf{bits}(\mathbf{s}) \otimes \mathbf{G},$$

     and the attribute/circular encodings by

$$\mathbf{e}_{\mathrm{attr}} \xleftarrow{\$} \mathcal{D}^{(L+1)m}_{\mathbb{Z},\sigma',\le\sigma'\sqrt{\lambda}}, \qquad \mathbf{c}_{\mathrm{attr}}^\intercal \leftarrow \mathbf{s}^\intercal(\mathbf{A}_{\mathrm{attr}} - (1, \mathbf{x}^\intercal) \otimes \mathbf{G}) + \mathbf{e}_{\mathrm{attr}}^\intercal,$$
$$\mathbf{e}_{\mathrm{circ}} \xleftarrow{\$} \mathcal{D}^{(L_{\mathrm{S}}+1)m}_{\mathbb{Z},\sigma',\le\sigma'\sqrt{\lambda}}, \qquad \mathbf{c}_{\mathrm{circ}}^\intercal \leftarrow \mathbf{s}^\intercal(\mathbf{A}_{\mathrm{circ}} - (1, \mathsf{bits}(\mathbf{S})) \otimes \mathbf{G}) + \mathbf{e}_{\mathrm{circ}}^\intercal.$$

     It outputs $\mathsf{ct}_{\mathrm{off}} = (\mathbf{x}, \mathbf{S}, \mathbf{c}_{\mathrm{attr}}, \mathbf{c}_{\mathrm{circ}})$ and $\mathsf{st} = (\boldsymbol{\mu}, \mathbf{s})$.

2. $\mathsf{EncD}(\mathsf{st}, \mathsf{param}_C, \mathsf{digest}_C)$ samples and sets

$$\mathbf{z}_1, \ldots, \mathbf{z}_{L'} \xleftarrow{\$} \mathbb{Z}_q^{n+1}, \qquad \mathbf{A}_{\mathrm{msg}} \leftarrow (\mathbf{A}_{C:1}\mathbf{G}^{-1}(\mathbf{z}_1), \ldots, \mathbf{A}_{C:L'}\mathbf{G}^{-1}(\mathbf{z}_{L'})).$$

The algorithm creates the message encoding by

$$\mathbf{e}_{\mathrm{msg}} \xleftarrow{\$} \mathcal{D}^{L'}_{\mathbb{Z},\sigma_{\mathrm{msg}},\le\sigma_{\mathrm{msg}}\sqrt{\lambda}}, \qquad \mathbf{c}_{\mathrm{msg}}^\intercal \leftarrow \mathbf{s}^\intercal\mathbf{A}_{\mathrm{msg}} + \mathbf{e}_{\mathrm{msg}}^\intercal + \lfloor q/2 \rceil \cdot \boldsymbol{\mu}^\intercal.$$

It outputs $\mathsf{ct}_{\mathrm{on}} = (\mathbf{z}_1, \ldots, \mathbf{z}_{L'}, \mathbf{c}_{\mathrm{msg}})$.

- $\mathsf{Dec}(\mathsf{crs}, C, \mathsf{digest}_C, \mathsf{ct})$ evaluates $C(\mathbf{x})$, computes

$$\begin{aligned}(\mathbf{c}_{C,\mathbf{x}:1}^\intercal, \ldots, \mathbf{c}_{C,\mathbf{x}:L'}^\intercal) = \mathbf{c}_{C,\mathbf{x}}^\intercal &\leftarrow \mathsf{UEvalCX}(\mathbf{A}_{\mathrm{attr}}, \mathbf{c}_{\mathrm{attr}}^\intercal, \mathbf{A}_{\mathrm{circ}}, \mathbf{c}_{\mathrm{circ}}^\intercal, C, \mathbf{x}, \mathbf{S}),\\ (\mathbf{c}')^\intercal &\leftarrow \left(\mathbf{c}_{\mathrm{msg}}^\intercal - (\mathbf{c}_{C,\mathbf{x}:1}^\intercal\mathbf{G}^{-1}(\mathbf{z}_1), \ldots, \mathbf{c}_{C,\mathbf{x}:L'}^\intercal\mathbf{G}^{-1}(\mathbf{z}_{L'}))\right) \bmod q,\end{aligned}$$

and sets for all $\ell \in [L']$,

$$\boldsymbol{\mu}'[\ell] \leftarrow \begin{cases} 0, & \text{if } C(\mathbf{x})[1,\ell] = 1;\\ 0, & \text{if } C(\mathbf{x})[1,\ell] = 0 \text{ and } \mathbf{c}'[\ell] \in [-q/4, q/4);\\ 1, & \text{otherwise.}\end{cases}$$

The algorithm outputs $(\mathbf{x}, (\boldsymbol{\mu}')^\intercal)$.

This scheme has a deterministic $\mathsf{Compress}$, and satisfies

$$\begin{aligned} |\mathsf{crs}| &= \mathrm{O}(L), & |\mathsf{digest}_C| &= \mathrm{O}(1), & |\mathsf{ct}_{\mathrm{off}}| &= \mathrm{O}(L), & |\mathsf{ct}_{\mathrm{on}}| &= \mathrm{O}(L'),\\ T_{\mathsf{GenCRS}} &= \mathrm{O}(L), & T_{\mathsf{Compress}}, T_{\mathsf{Dec}} &= \mathrm{O}(|C|), & T_{\mathsf{EncX}} &= \mathrm{O}(L+L'), & T_{\mathsf{EncD}} &= \mathrm{O}(L').\end{aligned}$$

Theorem 15 (¶)**.** *Under* sLWE *(Assumption 1) suitable for Theorem 14, Construction 3 is computationally f-selectively correct (Definition 3; C must be chosen before seeing* crs*). If the assumption holds against non-uniform adversaries, then the correctness becomes statistical (still f-selective).*

Proof (Theorem 15). Let $\mathcal{A}$ be an efficient adversary against $\mathsf{Exp}^{f\text{-sel}}_{\mathrm{LFE}\checkmark}$ and write

$$\mathbf{c}_{C,\mathbf{x}}^\intercal = \mathbf{s}^\intercal(\mathbf{A}_C - C(\mathbf{x}) \otimes \mathbf{G}) + \mathbf{e}_{C,\mathbf{x}}^\intercal, \qquad \mathbf{c}_{C,\mathbf{x}:\ell}^\intercal = \mathbf{s}^\intercal(\mathbf{A}_{C:\ell} - C(\mathbf{x})[1,\ell]\mathbf{G}) + \mathbf{e}_{C,\mathbf{x}:\ell}^\intercal \quad \text{for } \ell \in [L'],$$

then $\mathbf{e}_{C,\mathbf{x}}^\intercal = (\mathbf{e}_{C,\mathbf{x}:1}^\intercal, \ldots, \mathbf{e}_{C,\mathbf{x}:L'}^\intercal)$. For each $\ell \in [L']$, it suffices to consider the case of $C(\mathbf{x})[1,\ell] = 0$, in which during decryption,

$$\begin{aligned}\mathbf{c}'[\ell] &\equiv (\mathbf{s}^\intercal\mathbf{A}_{C:\ell}\mathbf{G}^{-1}(\mathbf{z}_\ell) + \mathbf{e}_{\mathrm{msg}}[\ell] + \boldsymbol{\mu}[\ell] \cdot \lfloor q/2 \rceil)\\ &\quad - (\mathbf{s}^\intercal(\mathbf{A}_{C:\ell} - C(\mathbf{x})[1,\ell]\mathbf{G}) + \mathbf{e}_{C,\mathbf{x}:\ell}^\intercal)\mathbf{G}^{-1}(\mathbf{z}_\ell)\\ &\equiv (\mathbf{e}_{\mathrm{msg}}[\ell] - \mathbf{e}_{C,\mathbf{x}:\ell}^\intercal\mathbf{G}^{-1}(\mathbf{z}_\ell)) + \boldsymbol{\mu}[\ell] \cdot \lfloor q/2 \rceil \pmod q.\end{aligned}$$

Suppose $\|\mathbf{e}_{C,\mathbf{x}}\| \le B$, then $\|\mathbf{e}_{C,\mathbf{x}:\ell}\| \le \|\mathbf{e}_{C,\mathbf{x}}\| \le B$ and by our choice of parameters,

$$\begin{aligned}|\mathbf{e}_{\mathrm{msg}}[\ell]| + |\mathbf{e}_{C,\mathbf{x}:\ell}^\intercal\mathbf{G}^{-1}(\mathbf{z}_\ell)| + |\boldsymbol{\mu}[\ell] \cdot (q/2 - \lfloor q/2 \rceil)| &\le \sigma_{\mathrm{msg}}\sqrt{\lambda} + \|\mathbf{e}_{C,\mathbf{x}:\ell}\| \cdot \|(\mathbf{G}^{-1}(\mathbf{z}_\ell))^\intercal\| + 1\\ &\le \sigma_{\mathrm{msg}}\sqrt{\lambda} + B \cdot m + 1 < q/4,\end{aligned}$$

so $\mathbf{c}'[\ell] \in [-q/4, q/4)$ (i.e., $\boldsymbol{\mu}'[\ell] = 0$) if and only if $\boldsymbol{\mu}[\ell] = 0$. It remains to bound $\Pr[\,\|\mathbf{e}_{C,\mathbf{x}:\ell}\| > B\,]$. Note that

$$\|\mathbf{e}_{\mathrm{fhe}}\| \le \sigma\sqrt{\lambda} \le 2^{-\lambda}B, \qquad \|\mathbf{e}_{\mathrm{attr}}\|, \|\mathbf{e}_{\mathrm{circ}}\| \le \sigma'\sqrt{\lambda} \le 2^{-\lambda}B,$$

so applying Theorem 14 (corollary part) to the first phase of $\mathcal{A}$ (choosing $1^L, C$) yields the desired bound.

Statistical f-selective correctness follows from non-uniform assumption, because Theorem 14 guarantees simultaneous correctness for all $\mathbf{x}$ (not necessarily efficiently computable after choosing C and seeing $\mathbf{A}$'s) with overwhelming probability. □

Theorem 16 (¶)**.** *Under* csLWE *(Assumption 1) with our choice of parameters, Construction 3 is very selectively secure (Definition 4).*

Proof (Theorem 16). The simulator $\widetilde{\mathsf{Enc}}(\mathsf{crs}, C, \mathsf{digest}_C, (\mathbf{x}, (\boldsymbol{\mu}')^\intercal))$ works as follows.

1. Sample $\mathbf{z}_1, \dots, \mathbf{z}_{L'}, \mathbf{S}, \mathbf{c}_{\mathrm{attr}}, \mathbf{c}_{\mathrm{circ}}$ uniformly at random and $\mathbf{e}_{\mathrm{msg}} \xleftarrow{\$} \mathcal{D}^{L'}_{\mathbb{Z},\sigma_{\mathrm{msg}},\le\sigma_{\mathrm{msg}}\sqrt{\lambda}}$.
2. Compute

$$(\mathbf{c}^\intercal_{C,\mathbf{x}:1}, \dots, \mathbf{c}^\intercal_{C,\mathbf{x}:L'}) \leftarrow \mathsf{UEvalCX}(\mathbf{A}_{\mathrm{attr}}, \mathbf{c}^\intercal_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, \mathbf{c}^\intercal_{\mathrm{circ}}, C, \mathbf{x}, \mathbf{S}),$$

$$\text{for each } \ell \in [L']\text{:} \quad \mathbf{c}_{\mathrm{msg}}[\ell] \begin{cases} \leftarrow \mathbf{c}^\intercal_{C,\mathbf{x}:\ell}\mathbf{G}^{-1}(\mathbf{z}_\ell) + \mathbf{e}_{\mathrm{msg}}[\ell] + \boldsymbol{\mu}'[\ell]\cdot\lfloor q/2\rceil, & \text{if } C(\mathbf{x})[1,\ell] = 0; \\ \xleftarrow{\$} \mathbb{Z}_q, & \text{if } C(\mathbf{x})[1,\ell] = 1. \end{cases}$$

3. Output $(\mathbf{z}_1, \dots, \mathbf{z}_{L'}, \mathbf{x}, \mathbf{S}, \mathbf{c}_{\mathrm{attr}}, \mathbf{c}_{\mathrm{circ}}, \mathbf{c}_{\mathrm{msg}})$.

We describe the hybrids by their changes to the previous one.

- H_0 is $\mathsf{Exp}^{\text{very-sel},0}_{\mathrm{LFE}}$, where the ciphertext sent to $\mathcal{A}$ is from Enc. We write

$$\begin{aligned}&(\mathbf{c}^\intercal_{C,\mathbf{x}:1}, \dots, \mathbf{c}^\intercal_{C,\mathbf{x}:L'}) = \mathsf{UEvalCX}(\mathbf{A}_{\mathrm{attr}}, \mathbf{c}^\intercal_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, \mathbf{c}^\intercal_{\mathrm{circ}}, C, \mathbf{x}, \mathbf{S}) \\ &= (\mathbf{s}^\intercal(\mathbf{A}_{C:1} - C(\mathbf{x})[1,1]\mathbf{G}) + \mathbf{e}^\intercal_{C,\mathbf{x}:1}, \dots, \mathbf{s}^\intercal(\mathbf{A}_{C:L'} - C(\mathbf{x})[1,L']\mathbf{G}) + \mathbf{e}^\intercal_{C,\mathbf{x}:L'}).\end{aligned}$$

- In H_1, for each $\ell \in [L']$, the message encoding component is computed as

$$\mathbf{c}_{\mathrm{msg}}[\ell] = \begin{cases} \mathbf{c}^\intercal_{C,\mathbf{x}:\ell}\mathbf{G}^{-1}(\mathbf{z}_\ell) & - \mathbf{e}^\intercal_{C,\mathbf{x}:\ell}\mathbf{G}^{-1}(\mathbf{z}_\ell) + \mathbf{e}_{\mathrm{msg}}[\ell] + \boldsymbol{\mu}'[\ell]\cdot\lfloor q/2\rceil, & \text{if } C(\mathbf{x})[1,\ell] = 0; \\ \mathbf{c}^\intercal_{C,\mathbf{x}:\ell}\mathbf{G}^{-1}(\mathbf{z}_\ell) + \mathbf{s}^\intercal\mathbf{z}_\ell & - \mathbf{e}^\intercal_{C,\mathbf{x}:\ell}\mathbf{G}^{-1}(\mathbf{z}_\ell) + \mathbf{e}_{\mathrm{msg}}[\ell] + \boldsymbol{\mu}[\ell]\cdot\lfloor q/2\rceil, & \text{if } C(\mathbf{x})[1,\ell] = 1. \end{cases}$$

  Note that $\boldsymbol{\mu}'[\ell] = \boldsymbol{\mu}[\ell]$ when $C(\mathbf{x})[1,\ell] = 0$. This is just rewriting, so $\mathsf{H}_0 \equiv \mathsf{H}_1$.

- In H_2, the experiment aborts if $\|\mathbf{e}_{C,\mathbf{x}}\| > B$. We have $\mathsf{H}_1 \approx \mathsf{H}_2$ by (the proof of) Theorem 15.[18]

- In H_3, using $\mathbf{e}_{\mathrm{msg}}$ for flooding, the terms $\mathbf{e}^\intercal_{C,\mathbf{x}:\ell}\mathbf{G}^{-1}(\mathbf{z}_\ell)$ are removed from $\mathbf{c}_{\mathrm{msg}}$, and a small noise $\mathbf{e}'_{\mathrm{msg}}[\ell]$ is inserted into $\mathbf{c}_{\mathrm{msg}}[\ell]$ if $C(\mathbf{x})[1,\ell] = 1$, i.e.,

$$\mathbf{c}_{\mathrm{msg}}[\ell] = \begin{cases} \mathbf{c}^\intercal_{C,\mathbf{x}:\ell}\mathbf{G}^{-1}(\mathbf{z}_\ell) & + \mathbf{e}_{\mathrm{msg}}[\ell] + \boldsymbol{\mu}'[\ell]\cdot\lfloor q/2\rceil, & \text{if } C(\mathbf{x})[1,\ell] = 0; \\ \mathbf{c}^\intercal_{C,\mathbf{x}:\ell}\mathbf{G}^{-1}(\mathbf{z}_\ell) + \mathbf{s}^\intercal\mathbf{z}_\ell + \mathbf{e}'_{\mathrm{msg}}[\ell] & + \mathbf{e}_{\mathrm{msg}}[\ell] + \boldsymbol{\mu}[\ell]\cdot\lfloor q/2\rceil, & \text{if } C(\mathbf{x})[1,\ell] = 1; \end{cases}$$

[18] This relies on $\mathcal{A}$ being selective in C.

where $\mathbf{e}'_{\mathrm{msg}} \xleftarrow{\$} \mathcal{D}^{L'}_{\mathbb{Z},\sigma',\le\sigma'\sqrt{\lambda}}$. We let $\mathbf{w}[\ell] = \mathbf{s}^\mathsf{T}\mathbf{z}_\ell + \mathbf{e}'_{\mathrm{msg}}[\ell]$ for $\ell \in [L']$. When the experiment does not abort, we have

$$\begin{aligned} 2^{\lambda+6}(|\mathbf{e}'[\ell]| + |\mathbf{e}^\mathsf{T}_{C,\mathbf{x}:\ell}\mathbf{G}^{-1}(\mathbf{z}_\ell)|) &\le 2^{\lambda+6}(|\mathbf{e}'[\ell]| + \|\mathbf{e}_{C,\mathbf{x}}\| \cdot \|(\mathbf{G}^{-1}(\mathbf{z}_\ell))^\mathsf{T}\|) \\ &\le 2^{\lambda+6}(\sigma'\sqrt{\lambda} + Bm) \le \sigma_{\mathrm{msg}} \end{aligned}$$

by our choice of parameters, so $\mathsf{H}_2 \approx_\mathrm{s} \mathsf{H}_3$ by Lemma 2.

- In H_4, the experiment no longer tests whether $\|\mathbf{e}_{C,\mathbf{x}}\| > B$ nor aborts. $\mathsf{H}_3 \approx \mathsf{H}_4$ by (the proof of) Theorem 15.
- H_5 is $\mathsf{Exp}^{\text{very-sel},1}_{\mathrm{LFE}}$, where $\mathbf{S}, \mathbf{c}_{\mathrm{attr}}, \mathbf{c}_{\mathrm{circ}}$ are uniformly random and

$$\mathbf{c}_{\mathrm{msg}}[\ell] = \begin{cases} \mathbf{c}^\mathsf{T}_{C,\mathbf{x}:\ell}\mathbf{G}^{-1}(\mathbf{z}_\ell) + \mathbf{e}_{\mathrm{msg}}[\ell] + \boldsymbol{\mu}'[\ell] \cdot \lfloor q/2 \rceil, & \text{if } C(\mathbf{x})[1,\ell] = 0; \\ \text{random}, & \text{if } C(\mathbf{x})[1,\ell] = 1. \end{cases}$$

To prove $\mathsf{H}_4 \approx \mathsf{H}_5$, we invoke csLWE while setting[19]

$$(\mathbf{A}_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, \mathbf{z}_1, \ldots, \mathbf{z}_{L'}) = \begin{pmatrix} \overline{\mathbf{A}}' \\ \underline{\mathbf{a}}^\mathsf{T} \end{pmatrix} + \big((1, \mathbf{x}^\mathsf{T}, 1, \mathsf{bits}(\mathbf{S})) \otimes \mathbf{G},\ \mathbf{0}_{(n+1)\times L'}\big),$$

where $\overline{\mathbf{A}}', \mathbf{S}$ are public matrix and circular ciphertext received from csLWE, and $\underline{\mathbf{a}}$ is sampled by the reduction algorithm. Note that in H_4,

$$(\mathbf{c}^\mathsf{T}_{\mathrm{attr}}, \mathbf{c}^\mathsf{T}_{\mathrm{circ}}, \mathbf{w}^\mathsf{T}) = \mathbf{r}^\mathsf{T}\overline{\mathbf{A}}' + (\mathbf{e}')^\mathsf{T} - \underline{\mathbf{a}}^\mathsf{T} - \big((1, \mathbf{x}^\mathsf{T}, 1, \mathsf{bits}(\mathbf{S})) \otimes \boldsymbol{\iota}^\mathsf{T}_{n+1} \otimes \mathbf{g}^\mathsf{T},\ \mathbf{0}_{1\times L'}\big),$$

$$\text{for each } \ell \in [L']\text{:}\quad \mathbf{c}_{\mathrm{msg}}[\ell] = \begin{cases} \mathbf{c}^\mathsf{T}_{C,\mathbf{x}:\ell}\mathbf{G}^{-1}(\mathbf{z}_\ell) + \mathbf{e}_{\mathrm{msg}}[\ell] + \boldsymbol{\mu}'[\ell] \cdot \lfloor q/2 \rceil, & \text{if } C(\mathbf{x})[1,\ell] = 0; \\ \mathbf{c}^\mathsf{T}_{C,\mathbf{x}:\ell}\mathbf{G}^{-1}(\mathbf{z}_\ell) + \mathbf{w}[\ell] + \mathbf{e}_{\mathrm{msg}}[\ell] + \boldsymbol{\mu}[\ell] \cdot \lfloor q/2 \rceil, & \text{if } C(\mathbf{x})[1,\ell] = 1. \end{cases}$$

The components change into the distribution in H_5 once csLWE is applied.

By hybrid argument, $\mathsf{Exp}^{\text{very-sel},0}_{\mathrm{LFE}} \equiv \mathsf{H}_0 \approx \mathsf{H}_5 \equiv \mathsf{Exp}^{\text{very-sel},1}_{\mathrm{LFE}}$. □

AB-LFE can be lifted to LFE generically [QWW18a] using FHE:

Corollary 17 (LFE)**.** *Under* csLWE *(Assumption 1) under our choice of parameters, there exists an f-selectively correct (Definition 3), very selectively secure (Definition 4) LFE scheme for circuits of unbounded depth (Definition 1) with*

$$\begin{aligned} |\mathsf{crs}| &= \mathrm{O}(L), & |\mathsf{digest}_C| &= \mathrm{O}(L'), & |\mathsf{ct}_{\mathrm{off}}| &= \mathrm{O}(L), & |\mathsf{ct}_{\mathrm{on}}| &= \mathrm{O}(L'), \\ T_{\mathsf{GenCRS}} &= \mathrm{O}(L), & T_{\mathsf{Compress}}, T_{\mathsf{Dec}} &= \mathrm{O}(|C|), & T_{\mathsf{EncX}} &= \mathrm{O}(L), & T_{\mathsf{EncD}} &= \mathrm{O}(L'), \end{aligned}$$

where L, L' are the input/output lengths and whose $\mathsf{Compress}$ *is deterministic.*

Corollary 17 follows by applying Section 4.4 of [QWW18b] to Construction 3 using a non-leveled FHE (which exists under csLWE, e.g., a circular version of [GSW13]). Although the transformation in [QWW18b] is presented for perfectly correct and selectively secure schemes, it is readily verified that it preserves the lesser correctness and security considered here.

To derive the stated two-phase encryption efficiency, we have to break the abstraction. The offline phase of Corollary 17 performs homomorphically encrypts $\mathbf{x}$ and runs EncX of AB-LFE on the ciphertext of $\mathbf{x}$, crucially leaving the AB-LFE message length indeterminate, a feature supported by Construction 3. The state consists of the FHE key and the AB-LFE state. During the online phase, the FHE decryption circuit is garbled, and EncD of AB-LFE is executed to transfer the labels.

[19] Embedding $\mathbf{x}$ into $\mathbf{A}_{\mathrm{attr}}$ relies on $\mathcal{A}$ being selective in x.

Stronger Correctness and Security. When using unbounded homomorphic evaluation with stronger correctness, the (AB-)LFE schemes become *adaptively* correct (computationally or statistically) and *selectively* secure. They will use randomized $\mathsf{Compress}$ (see Definition 1).

4.2 1-Key Functional Encryption and Attribute-Based Encryption

LFE generically implies 1-key FE:

Corollary 18 (1-key FE)**.** *Under* csLWE *(Assumption 1), there exists a strongly f-selectively correct (Definition 6), very selectively 1-key simulation-secure (Definition 7) FE scheme for circuits of unbounded depth (Definition 5) with*

$$|\mathsf{mpk}| = \mathrm{O}(L + L'), \qquad |\mathsf{sk}_C| = \mathrm{O}(L'), \qquad |\mathsf{ct}(\mathbf{x})| = \mathrm{O}(L + L'),$$
$$T_{\mathsf{Setup}} = \mathrm{O}(L + L'), \qquad T_{\mathsf{KeyGen}}, T_{\mathsf{Dec}} = \mathrm{O}(|C|), \qquad T_{\mathsf{Enc}} = \mathrm{O}(L + L'),$$

where L, L' are the input/output lengths.

Corollary 18 follows from a tweaked version of Theorem C.1 of [QWW18b] to remove the additive dependency on L in sk_C. Recall that in [QWW18b], sk_C contains an underlying 1-key FE key for the LFE encryption circuit (whose size is linear in the LFE encryption circuit size), making sk_C be of size linear in $|\mathsf{Enc}|$, which is $\Omega(L + L')$. We instead only use the underlying 1-key FE perform EncD so that the key size is linear is $|\mathsf{EncD}| = \mathrm{O}(L')$. Although the transformation was presented for perfectly correct and selectively secure schemes, its tweaked version (and itself) preserves strong f-selective correctness and very selective security considered here.

Corollary 18 implies 1-key ABE with constant-size keys, since the ABE functionality has 1-bit output:

Corollary 19 (1-key ABE)**.** *Under* csLWE *(Assumption 1), there exists a strongly f-selectively correct (Definition 6), very selectively 1-key secure (Definition 7) ABE scheme for circuits of unbounded depth (Definition 5) with*

$$|\mathsf{mpk}| = \mathrm{O}(L), \qquad |\mathsf{sk}_C| = \mathrm{O}(1), \qquad |\mathsf{ct}_{\mathbf{x}}| = \mathrm{O}(L),$$
$$T_{\mathsf{Setup}} = \mathrm{O}(L), \qquad T_{\mathsf{KeyGen}}, T_{\mathsf{Dec}} = \mathrm{O}(|C|), \qquad T_{\mathsf{Enc}} = \mathrm{O}(L),$$

where L is the attribute length.

Alternatively, Corollary 19 can be obtained applying Section C of [QWW18b] to Construction 3, again with the trick of reducing ciphertext size using the two-phase encryption structure.

Stronger Correctness and Security. When using unbounded homomorphic evaluation with stronger correctness, the 1-key FE/ABE schemes become *adaptively* correct (computationally or statistically) and *selectively* secure.

4.3 Reusable Garbled Circuits

From [GKP+13b], it is known that 1-key FE implies reusable garbling.

Corollary 20. *Assuming* csLWE *(Assumption 1), there is a selectively secure reusable garbling scheme (Definitions 8, 9, and 10) with*

$$|\widehat{C}| = \mathrm{O}(L'), \qquad |\mathsf{pk}| = \mathrm{O}(L + L'), \qquad |\widehat{\mathbf{x}}| = \mathrm{O}(L + L'),$$

where L, L' are the input/output lengths of C.

Note that in [GKP+13b], the garbled circuit $\widehat{C}$ hides C (and input garbling is secret-key), whereas in our scheme it does not, in exchange of better efficiency. We tweak Section 4.1 of [GKP+13b] so that $\widehat{C}$ is a 1-FE key for

$$C'(b, \mathbf{x}, \mathbf{y}) = \begin{cases} C(\mathbf{x}), & \text{if } b = 0; \\ \mathbf{y} & \text{if } b = 1; \end{cases}$$

and the input encoding $\widehat{\mathbf{x}}$ is a 1-FE ciphertext for $(0, \mathbf{x}, \mathbf{0})$. The simulator outputs $\widehat{\mathbf{x}}$ as a 1-FE ciphertext for $(1, \mathbf{0}, C(\mathbf{x}))$.

Using the unbounded homomorphic evaluation algorithms with stronger correctness does not improve correctness or security of our reusable garbling scheme from csLWE.

5 KP-ABE for Circuits of Unbounded Depth

In this section, we show how to achieve full-fledged ABE with our unbounded homomorphic evaluation algorithms by additionally assuming the evasive (circular small-secret) LWE assumption.

5.1 Lattice Trapdoors and Evasive LWE Assumption

We introduce additional preliminaries needed for KP-ABE.

Trapdoor Generation and Gaussian Preimage Sampling. We rely on the following:

Lemma 21 ([MP12; Theorem 2])**.** *There are efficient algorithms* $\mathsf{TrapGen}$ *and* SampD *and functions* $m_0 \in \Theta(n \log q)$ *and* $\sigma_0 \in \omega(\sqrt{m \log m}) \cap \mathrm{O}(m)$ *satisfying these conditions:*

- $\mathsf{TrapGen}(1^n, 1^m, q)$ *takes as input* $n \geq 1$, $q \geq 2$, *and* $m \geq m_0(n, q)$. *It outputs* $(\mathbf{B}, \tau)$ *such that* $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ *and* $\mathbf{B}$ *is* $\mathrm{negl}(n)$*-close to uniform over* $\mathbb{Z}_q^{n \times m}$.
- $\mathsf{SampD}(\mathbf{B}, \tau, \mathbf{p}, \sigma)$ *takes as input* $\mathbf{B}, \tau$ *from* $\mathsf{TrapGen}$, *some* $\mathbf{p} \in \mathbb{Z}_q^n$, *and* $\sigma \geq \sigma_0(n, m)$. *It outputs* $\mathbf{k} \in \mathbb{Z}^m$ *such that* $\mathbf{B}\mathbf{k} = \mathbf{p}$, $\|\mathbf{k}\| \leq \sigma\sqrt{m}$, *and* $\mathbf{k}$ *is* $\mathrm{negl}(n)$*-close to* $\mathcal{D}_{\Lambda_{\mathbf{p}}^{\perp}(\mathbf{B}), \sigma}$.

In particular (not aiming for optimality), m_0 *can be taken as* $3(n+1)\lceil \log_2 q \rceil$.

Batch Notation. It is convenient to extend SampD to process multiple $\mathbf{p}$'s in one shot. Let $\mathbf{P} = (\mathbf{p}_1, \ldots, \mathbf{p}_{m'})$ be a matrix or a batch of vectors, then $\mathsf{SampD}(\mathbf{B}, \tau, \mathbf{P}, \sigma)$ is

$$\mathbf{K} \xleftarrow{\$} \big(\mathsf{SampD}(\mathbf{B}, \tau, \mathbf{p}_1, \sigma), \ \ldots, \ \mathsf{SampD}(\mathbf{B}, \tau, \mathbf{p}_{m'}, \sigma)\big),$$

with fresh randomness for each call to SampD on the right-hand side.

Assumption. We formulate the variant of evasive LWE assumption needed for our KP-ABE. While it appears complicated due to incorporation of circular ciphertext and encoding, the assumption follows the same rationale as existing variants of evasive LWE. More discussion follows the definition.

Assumption 2 (evcsLWE)**.** Let $\mathcal{S}(1^\lambda; \mathsf{aux})$ be an algorithm that, given randomness aux, outputs

$$\overline{\mathbf{A}}_{\mathrm{circ}} \in \mathbb{Z}_q^{n \times (m(n+1)^2 \lceil \log_2 q \rceil^2 + 1)m}, \quad \overline{\mathbf{A}}' \in \mathbb{Z}_q^{n \times m'}, \quad \mathbf{P} \in \mathbb{Z}_q^{n \times J}, \quad \sigma, \quad \sigma', \quad \sigma_{-1}, \quad \sigma_{\mathrm{post}}, \quad \sigma_{\mathrm{pre}},$$

where $m \geq m_0(n, q)$ and $\sigma_{-1} \geq \sigma_0(n, m)$ are constrained by Lemma 21 and $\sigma_{\mathrm{post}} \geq \sigma_{\mathrm{pre}}$. Suppose

$$\overline{\mathbf{A}}_{\mathrm{fhe}} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}, \qquad (\mathbf{B}, \tau) \xleftarrow{\$} \mathsf{TrapGen}(1^n, 1^m, q), \qquad \mathbf{K} \xleftarrow{\$} \mathsf{SampD}(\mathbf{B}, \tau, \mathbf{P}, \sigma_{-1}),$$

$$\mathbf{e}_{\mathrm{fhe}} \xleftarrow{\$} \mathcal{D}^{m}_{\mathbb{Z},\sigma,\le\sigma\sqrt{\lambda}}, \quad \mathbf{e}_{\mathrm{circ}} \xleftarrow{\$} \mathcal{D}^{(m(n+1)^2\lceil\log_2 q\rceil^2+1)m}_{\mathbb{Z},\sigma',\le\sigma'\sqrt{\lambda}}, \quad \mathbf{e}' \xleftarrow{\$} \mathcal{D}^{m'}_{\mathbb{Z},\sigma',\le\sigma'\sqrt{\lambda}}, \quad \mathbf{e}_{\mathrm{B}} \in \mathbb{Z}^m, \quad \mathbf{e}_{\mathrm{P}} \in \mathbb{Z}^J,$$
$$\boldsymbol{\delta}_{\mathrm{fhe}} \xleftarrow{\$} \mathbb{Z}_q^m, \quad \boldsymbol{\delta}_{\mathrm{circ}} \xleftarrow{\$} \mathbb{Z}_q^{(m(n+1)^2\lceil\log_2 q\rceil^2+1)m}, \quad \boldsymbol{\delta}' \xleftarrow{\$} \mathbb{Z}_q^{m'}, \quad \boldsymbol{\delta}_{\mathrm{B}} \xleftarrow{\$} \mathbb{Z}_q^m, \quad \boldsymbol{\delta}_{\mathrm{P}} \xleftarrow{\$} \mathbb{Z}_q^J,$$
$$\mathbf{r} \xleftarrow{\$} \mathcal{D}^{n}_{\mathbb{Z},\sigma,\le\sigma\sqrt{\lambda}}, \quad \mathbf{R} \xleftarrow{\$} \{0,1\}^{m\times(n+1)\lceil\log_2 q\rceil m}, \quad \mathbf{S} = \begin{pmatrix} \overline{\mathbf{A}}_{\mathrm{fhe}} \\ \mathbf{r}^\intercal\overline{\mathbf{A}}_{\mathrm{fhe}} + \mathbf{e}_{\mathrm{fhe}}^\intercal \end{pmatrix} \mathbf{R} - \mathsf{bits}(\mathbf{s}) \otimes \mathbf{G}.$$
$$\mathbf{s} \leftarrow (\mathbf{r}^\intercal, -1)^\intercal, \quad \boldsymbol{\Delta} \xleftarrow{\$} \mathbb{Z}_q^{(n+1)\times(n+1)\lceil\log_2 q\rceil m},$$

In the precondition, the entries of $\mathbf{e}_{\mathrm{B}}, \mathbf{e}_{\mathrm{P}}$ are independent and follow $\mathcal{D}_{\mathbb{Z},\sigma_{\mathrm{pre}},\le\sigma_{\mathrm{pre}}\sqrt{\lambda}}$, and $\mathsf{evcsLWE}^{\mathcal{S}}_{\mathrm{pre}}$ states that

$$\left\{ \begin{pmatrix} 1^\lambda, \mathsf{aux}, \overline{\mathbf{A}}_{\mathrm{fhe}}, \mathbf{B}, \mathbf{r}^\intercal\overline{\mathbf{A}}_{\mathrm{fhe}} + \mathbf{e}_{\mathrm{fhe}}^\intercal, \mathbf{S}, \\ \mathbf{r}^\intercal(\overline{\mathbf{A}}_{\mathrm{circ}} - (1, \mathsf{bits}(\mathbf{S})) \otimes \overline{\mathbf{G}}) + \mathbf{e}_{\mathrm{circ}}^\intercal, \\ \mathbf{r}^\intercal\overline{\mathbf{A}}' + (\mathbf{e}')^\intercal, \mathbf{r}^\intercal\mathbf{B} + \mathbf{e}_{\mathrm{B}}^\intercal, \mathbf{r}^\intercal\mathbf{P} + \mathbf{e}_{\mathrm{P}}^\intercal \end{pmatrix} \right\}_{\lambda\in\mathbb{N}} \approx \left\{ \begin{pmatrix} 1^\lambda, \mathsf{aux}, \overline{\mathbf{A}}_{\mathrm{fhe}}, \mathbf{B}, \boldsymbol{\delta}_{\mathrm{fhe}}^\intercal, \boldsymbol{\Delta}, \\ \boldsymbol{\delta}_{\mathrm{circ}}^\intercal, \\ (\boldsymbol{\delta}')^\intercal, \boldsymbol{\delta}_{\mathrm{B}}^\intercal, \boldsymbol{\delta}_{\mathrm{P}}^\intercal \end{pmatrix} \right\}_{\lambda\in\mathbb{N}}.$$

In the postcondition, the entries of $\mathbf{e}_{\mathrm{B}}$ are independent and follow $\mathcal{D}_{\mathbb{Z},\sigma_{\mathrm{post}},\le\sigma_{\mathrm{post}}\sqrt{\lambda}}$, and $\mathsf{evcsLWE}^{\mathcal{S}}_{\mathrm{post}}$ states that

$$\left\{ \begin{pmatrix} 1^\lambda, \mathsf{aux}, \overline{\mathbf{A}}_{\mathrm{fhe}}, \mathbf{B}, \mathbf{r}^\intercal\overline{\mathbf{A}}_{\mathrm{fhe}} + \mathbf{e}_{\mathrm{fhe}}^\intercal, \mathbf{S}, \\ \mathbf{r}^\intercal(\overline{\mathbf{A}}_{\mathrm{circ}} - (1, \mathsf{bits}(\mathbf{S})) \otimes \overline{\mathbf{G}}) + \mathbf{e}_{\mathrm{circ}}^\intercal, \\ \mathbf{r}^\intercal\overline{\mathbf{A}}' + (\mathbf{e}')^\intercal, \mathbf{r}^\intercal\mathbf{B} + \mathbf{e}_{\mathrm{B}}^\intercal, \quad \mathbf{K} \end{pmatrix} \right\}_{\lambda\in\mathbb{N}} \approx \left\{ \begin{pmatrix} 1^\lambda, \mathsf{aux}, \overline{\mathbf{A}}_{\mathrm{fhe}}, \mathbf{B}, \boldsymbol{\delta}_{\mathrm{fhe}}^\intercal, \boldsymbol{\Delta}, \\ \boldsymbol{\delta}_{\mathrm{circ}}^\intercal, \\ (\boldsymbol{\delta}')^\intercal, \boldsymbol{\delta}_{\mathrm{B}}^\intercal, \mathbf{K} \end{pmatrix} \right\}_{\lambda\in\mathbb{N}}.$$

The *evasive circular small-secret LWE assumption* states that $\mathsf{evcsLWE}^{\mathcal{S}}_{\mathrm{pre}}$ implies $\mathsf{evcsLWE}^{\mathcal{S}}_{\mathrm{post}}$ for all efficient sampler $\mathcal{S}$.

Remarks. As in [Wee22,WWW22], our evasive circular small-secret LWE is only for public-coin samplers, enforced by providing the sampler randomness to the distinguisher. This is weaker than the versions [Tsa22,VWW22] used for witness encryption, which considers private-coin samplers and the distinguisher, instead of the sampler randomness, gets the auxiliary information produced as an additional output of the sampler. Assuming evasive LWE only for public-coin samplers avoids obfuscation-based counterexamples, where $\mathbf{P}$ is sampled with a trapdoor and the auxiliary information contains an obfuscated program with the trapdoor hardwired.

As suggested in [Wee22], the noise magnitude in the postcondition can be made larger than that in the precondition for a more conservative assumption. We can implement it by requiring evasive LWE to hold only for samplers with a certain gap between σ_{post} and σ_{pre}. We additionally allow a worse Gaussian preimage $\mathbf{K}$. Unlike [Wee22,WWW22,Tsa22,VWW22], we do not require an exact advantage relation between the the precondition and postcondition distinguishers (i.e., they share the same values of λ for which indistinguishability fails). Other than those two aspects, our assumption is stronger than the version in [Wee22], and is incomparable to that in [WWW22].

We note that in our formulation, in contrast to $\mathbf{e}_{\mathrm{B}}, \mathbf{e}_{\mathrm{P}}$, the distributions of $\mathbf{r}, \mathbf{e}_{\mathrm{fhe}}, \mathbf{e}_{\mathrm{circ}}, \mathbf{e}'$ are invariant across $\mathsf{evcsLWE}_{\mathrm{pre}}$ and $\mathsf{evcsLWE}_{\mathrm{post}}$. Intuitively, $\mathbf{e}_{\mathrm{B}}, \mathbf{e}_{\mathrm{P}}$ being smaller in the precondition compensates for the fact that in the postcondition, the noise attached to $\mathbf{r}^\intercal\mathbf{BK}$ is not an independent $\mathbf{e}_{\mathrm{P}}$ but correlated with $\mathbf{e}_{\mathrm{B}}$. However, this does not apply to $\mathbf{r}, \mathbf{e}_{\mathrm{fhe}}, \mathbf{e}_{\mathrm{circ}}, \mathbf{e}'$. In the presence of circular encryption, a distribution-varying $\mathbf{r}$ is difficult to interpret. Lastly, the formulation must consider different magnitudes for $\mathbf{e}', \mathbf{e}_{\mathrm{P}}$ — for correctness, $\mathbf{e}_{\mathrm{fhe}}, \mathbf{e}_{\mathrm{circ}}, \mathbf{e}'$ (the last contains attribute encoding noises) must be small to be successfully refreshed; operationally, the (refreshed) homomorphic noises are larger than $\mathbf{e}_{\mathrm{circ}}$; for security, $\mathbf{e}_{\mathrm{P}}$ must be large to flood the homomorphic noises. We thus settle for the version above.

The sampler is allowed to choose $\overline{\mathbf{A}}_{\mathrm{circ}}$. Since the public matrices for circuits depend on it, it must be known to the sampler, either (here) chosen by or (alternatively) given to the sampler. In

both cases, the structure of $(\overline{\mathbf{A}}_{\mathrm{circ}} - (1, \mathsf{bits}(\mathbf{S})) \otimes \mathbf{G})$ must be incorporated into the assumption, because the sampler cannot choose $\mathbf{S}$. We settle for one possible version here.

5.2 Construction of KP-ABE

Construction 4 (KP-ABE for circuits of unbounded depth)**.** It works as follows.

- $\mathsf{Setup}(1^L)$ defines appropriate $n, m, q, \sigma, \sigma', \sigma_{-1}, \sigma_{\mathrm{post}}$ as described in Section 3, and sets $L_{\mathrm{S}} = m(n+1)^2\lceil \log_2 q \rceil^2$. The algorithm samples

$$\begin{aligned} \mathbf{A}_{\mathrm{attr}} &\xleftarrow{\$} \mathbb{Z}_q^{(n+1)\times(L+1)m}, & (\mathbf{B}, \tau) &\xleftarrow{\$} \mathsf{TrapGen}(1^n, 1^m, q), \\ \mathbf{A}_{\mathrm{circ}} &\xleftarrow{\$} \mathbb{Z}_q^{(n+1)\times(L_{\mathrm{S}}+1)m}, & \mathbf{z} &\xleftarrow{\$} \mathbb{Z}_q^n. \end{aligned}$$

  It outputs $\mathsf{mpk} = (n, m, q, \sigma, \sigma', \sigma_{-1}, \sigma_{\mathrm{post}}, \mathbf{A}_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, \mathbf{B}, \mathbf{z})$ and $\mathsf{msk} = (\mathsf{mpk}, \tau)$.

- $\mathsf{KeyGen}(\mathsf{msk}, C)$ takes as input msk and some circuit $C : \{0,1\}^L \to \{0,1\}$. It computes $\mathbf{A}_C \leftarrow \mathsf{UEvalC}(\mathbf{A}_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, C)$ and samples $\mathbf{z}' \xleftarrow{\$} \mathbb{Z}_q^{n+1}$. The algorithm generates a trapdoor

$$\mathbf{k} \xleftarrow{\$} \mathsf{SampD}(\mathbf{B}, \tau, \overline{\mathbf{A}}_C\mathbf{G}^{-1}(\mathbf{z}') + \mathbf{z}, \sigma_{-1}),$$

  where $\overline{\mathbf{A}}_C \in \mathbb{Z}_q^{n\times m}$ is the the first n rows of $\mathbf{A}_C$. The algorithm outputs $\mathsf{sk}_C = (\mathbf{z}', \mathbf{k})$.

- $\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}, \mu)$ takes as input mpk, attribute $\mathbf{x} \in \{0,1\}^L$, and message $\mu \in \{0,1\}$. It samples $\mathbf{r} \xleftarrow{\$} \mathcal{D}^n_{\mathbb{Z},\sigma,\le\sigma\sqrt{\lambda}}$ and sets $\mathbf{s} \leftarrow (\mathbf{r}^\mathsf{T}, -1)^\mathsf{T}$. The algorithm creates a circular encryption by

$$\begin{aligned} \overline{\mathbf{A}}_{\mathrm{fhe}} &\xleftarrow{\$} \mathbb{Z}_q^{n\times m}, & \mathbf{e}_{\mathrm{fhe}} &\xleftarrow{\$} \mathcal{D}^m_{\mathbb{Z},\sigma,\le\sigma\sqrt{\lambda}}, & \mathbf{R} &\xleftarrow{\$} \{0,1\}^{m\times m(n+1)\lceil\log_2 q\rceil}, \\ \mathbf{A}_{\mathrm{fhe}} &\leftarrow \begin{pmatrix} \overline{\mathbf{A}}_{\mathrm{fhe}} \\ \mathbf{r}^\mathsf{T}\overline{\mathbf{A}}_{\mathrm{fhe}} + \mathbf{e}_{\mathrm{fhe}}^\mathsf{T} \end{pmatrix}, & & & \mathbf{S} &\leftarrow \mathbf{A}_{\mathrm{fhe}}\mathbf{R} - \mathsf{bits}(\mathbf{s}) \otimes \mathbf{G}, \end{aligned}$$

  and the attribute/circular encodings by

$$\begin{aligned} \mathbf{e}_{\mathrm{attr}} &\xleftarrow{\$} \mathcal{D}^{(L+1)m}_{\mathbb{Z},\sigma',\le\sigma'\sqrt{\lambda}}, & \mathbf{c}_{\mathrm{attr}}^\mathsf{T} &\leftarrow \mathbf{s}^\mathsf{T}(\mathbf{A}_{\mathrm{attr}} - (1, \mathbf{x}^\mathsf{T}) \otimes \mathbf{G}) + \mathbf{e}_{\mathrm{attr}}^\mathsf{T}, \\ \mathbf{e}_{\mathrm{circ}} &\xleftarrow{\$} \mathcal{D}^{(L_{\mathrm{S}}+1)m}_{\mathbb{Z},\sigma',\le\sigma'\sqrt{\lambda}}, & \mathbf{c}_{\mathrm{circ}}^\mathsf{T} &\leftarrow \mathbf{s}^\mathsf{T}(\mathbf{A}_{\mathrm{circ}} - (1, \mathsf{bits}(\mathbf{S})) \otimes \mathbf{G}) + \mathbf{e}_{\mathrm{circ}}^\mathsf{T}. \end{aligned}$$

  It also generates the message encoding as

$$\begin{aligned} \mathbf{e}_{\mathrm{B}} &\xleftarrow{\$} \mathcal{D}^m_{\mathbb{Z},\sigma_{\mathrm{post}},\le\sigma_{\mathrm{post}}\sqrt{\lambda}}, & \mathbf{c}_{\mathrm{B}}^\mathsf{T} &\leftarrow \mathbf{r}^\mathsf{T}\mathbf{B} + \mathbf{e}_{\mathrm{B}}^\mathsf{T}, \\ e_{\mathrm{msg}} &\xleftarrow{\$} \mathcal{D}_{\mathbb{Z},\sigma',\le\sigma'\sqrt{\lambda}}, & c_{\mathrm{msg}} &\leftarrow \mathbf{r}^\mathsf{T}\mathbf{z} + e_{\mathrm{msg}} + \mu \cdot \lfloor q/2 \rceil. \end{aligned}$$

  The algorithm outputs $\mathsf{ct}_{\mathbf{x}} = (\mathbf{S}, \mathbf{c}_{\mathrm{attr}}, \mathbf{c}_{\mathrm{circ}}, \mathbf{c}_{\mathrm{B}}, c_{\mathrm{msg}})$.

- $\mathsf{Dec}(\mathsf{mpk}, C, \mathsf{sk}_C, \mathbf{x}, \mathsf{ct}_{\mathbf{x}})$ takes $\mathsf{mpk}, C, \mathsf{sk}_C, \mathbf{x}, \mathsf{ct}_{\mathbf{x}}$ as input. It computes

$$\begin{aligned} \begin{pmatrix} \overline{\mathbf{A}}_C \\ \underline{\mathbf{a}}_C^\mathsf{T} \end{pmatrix} = \mathbf{A}_C &\leftarrow \mathsf{UEvalC}(\mathbf{A}_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, C), \\ \mathbf{c}_{C,\mathbf{x}}^\mathsf{T} &\leftarrow \mathsf{UEvalCX}(\mathbf{A}_{\mathrm{attr}}, \mathbf{c}_{\mathrm{attr}}^\mathsf{T}, \mathbf{A}_{\mathrm{circ}}, \mathbf{c}_{\mathrm{circ}}^\mathsf{T}, C, \mathbf{x}, \mathbf{S}), \\ c' &\leftarrow c_{\mathrm{msg}} + \mathbf{c}_{C,\mathbf{x}}^\mathsf{T}\mathbf{G}^{-1}(\mathbf{z}') - (\mathbf{c}_{\mathrm{B}}^\mathsf{T}\mathbf{k} - \underline{\mathbf{a}}_C^\mathsf{T}\mathbf{G}^{-1}(\mathbf{z}')). \end{aligned}$$

  The algorithm outputs $\mu' = 0$ if $c' \in [-q/4, q/4)$, and $\mu' = 1$ otherwise.

The efficiency parameters of the scheme are

$$\begin{aligned} |\mathsf{mpk}| &= \mathrm{O}(L), & |\mathsf{sk}_C| &= \mathrm{O}(1), & |\mathsf{ct}_{\mathbf{x}}| &= \mathrm{O}(L), \\ T_{\mathsf{Setup}} &= \mathrm{O}(L), & T_{\mathsf{KeyGen}}, T_{\mathsf{Dec}} &= \mathrm{O}(|C|), & T_{\mathsf{Enc}} &= \mathrm{O}(L). \end{aligned}$$

Theorem 22 (¶). *Under* sLWE *(Assumption 1), Construction 4 is strongly f-selectively correct (Definition 6).*

Proof (Theorem 22). The proof resembles that of Theorem 15. When $C_j(\mathbf{x}) = 0$, we write

$$\mathbf{c}_{C_j,\mathbf{x}}^{\mathsf{T}} = \mathbf{s}^{\mathsf{T}}(\mathbf{A}_{C_j} - C_j(\mathbf{x})\mathbf{G}) + \mathbf{e}_{C_j,\mathbf{x}}^{\mathsf{T}} = \mathbf{s}^{\mathsf{T}}\mathbf{A}_{C_j} + \mathbf{e}_{C_j,\mathbf{x}}^{\mathsf{T}}.$$

By definition,

$$\begin{aligned} \mathbf{c}_{\mathrm{B}}^{\mathsf{T}}\mathbf{k}_j - \underline{\mathbf{a}}_{C_j}^{\mathsf{T}}\mathbf{G}^{-1}(\mathbf{z}'_j) &= \mathbf{r}^{\mathsf{T}}\overline{\mathbf{A}}_{C_j}\mathbf{G}^{-1}(\mathbf{z}'_j) + \mathbf{r}^{\mathsf{T}}\mathbf{z} + \mathbf{e}_{\mathrm{B}}^{\mathsf{T}}\mathbf{k}_j - \underline{\mathbf{a}}_{C_j}^{\mathsf{T}}\mathbf{G}^{-1}(\mathbf{z}'_j) \\ &= \mathbf{s}^{\mathsf{T}}\mathbf{A}_{C_j}\mathbf{G}^{-1}(\mathbf{z}'_j) + \mathbf{r}^{\mathsf{T}}\mathbf{z} + \mathbf{e}_{\mathrm{B}}^{\mathsf{T}}\mathbf{k}_j, \\ \implies \quad c' &= c_{\mathrm{msg}} + \mathbf{c}_{C_j,\mathbf{x}}^{\mathsf{T}}\mathbf{G}^{-1}(\mathbf{z}'_j) - (\mathbf{c}_{\mathrm{B}}^{\mathsf{T}}\mathbf{k}_j - \underline{\mathbf{a}}_{C_j}^{\mathsf{T}}\mathbf{G}^{-1}(\mathbf{z}'_j)) \\ &= (\mathbf{r}^{\mathsf{T}}\mathbf{z} + e_{\mathrm{msg}} + \mu \cdot \lfloor q/2 \rceil) + (\mathbf{s}^{\mathsf{T}}\mathbf{A}_{C_j}\mathbf{G}^{-1}(\mathbf{z}'_j) + \mathbf{e}_{C_j,\mathbf{x}}^{\mathsf{T}}\mathbf{G}^{-1}(\mathbf{z}'_j)) \\ &\quad - (\mathbf{s}^{\mathsf{T}}\mathbf{A}_{C_j}\mathbf{G}^{-1}(\mathbf{z}'_j) + \mathbf{r}^{\mathsf{T}}\mathbf{z} + \mathbf{e}_{\mathrm{B}}^{\mathsf{T}}\mathbf{k}_j) \\ &= \mu \cdot q/2 + (\mathbf{e}_{\mathrm{B}}^{\mathsf{T}}\mathbf{k}_j + e_{\mathrm{msg}} + \mathbf{e}_{C_j,\mathbf{x}}^{\mathsf{T}}\mathbf{G}^{-1}(\mathbf{z}'_j) + \mu \cdot (\lfloor q/2 \rceil - q/2)). \end{aligned}$$

When $\|\mathbf{e}_{C_j,\mathbf{x}}\| \le B$, according to our choice of parameters, the total error is bounded by

$$\begin{aligned} & m \cdot \|\mathbf{e}_{\mathrm{B}}\| \cdot \|\mathbf{k}_j\| + |e_{\mathrm{msg}}| + \|\mathbf{e}_{C_j,\mathbf{x}}\| \cdot \|(\mathbf{G}^{-1}(\mathbf{z}'_j))^{\mathsf{T}}\| + 1 \\ & \qquad \le m \cdot \sigma_{\mathrm{post}}\sqrt{\lambda} \cdot \sigma_{-1}\sqrt{m} + \sigma'\sqrt{\lambda} + B \cdot m + 1 < q/4, \end{aligned}$$

in which case $\mu' = \mu$. From Theorem 14, it follows that $\Pr[\,\|\mathbf{e}_{C_j,\mathbf{x}}\| \le B \text{ for all } j\,]$ is overwhelming by a standard guessing reduction. This completes the proof. □

Stronger Correctness. When using unbounded homomorphic evaluation with stronger correctness, our KP-ABE scheme becomes (computationally or statistically) *adaptively* correct. We note that it does *not* become more secure due to the reliance on evcsLWE.

5.3 Security of KP-ABE

Theorem 23 (¶). *Under* csLWE *(Assumption 1) and* evcsLWE *(Assumption 2), Construction 4 is very selectively secure (Definition 7).*

Proof (Theorem 23). Technically, we are considering simulation security (the constrained variant, so there is no decrypting key). In general, this can be done by using the normal Setup, KeyGen to generate keys and encrypting any message chosen by the simulator under the known attribute, but it is more convenient in our proof to simulate the ciphertext as uniformly random.

Let $\mathcal{A}$ be an efficient adversary against the very selective security of Construction 4. Consider the following sampler $\mathcal{S}(\mathsf{aux})$:

- Parse aux into (r, r').
- Launch $\mathcal{A}(r)$ to obtain $(1^L, \mathbf{x}, \{C_j\}_{j\in[J]})$, where $\mathbf{x} \in \{0,1\}^L$ and $C_j : \{0,1\}^L \to \{0,1\}$ with $C_j(\mathbf{x}) = 1$ for all $j \in [J]$.

- Pick $n, m, q, \sigma', \sigma', \sigma_{-1}, \sigma_{\mathrm{post}}$ as in $\mathsf{Setup}(1^L)$. Pick an appropriate σ_{pre}.
- Sample $\mathbf{A}_{\mathrm{attr}} = \begin{pmatrix} \overline{\mathbf{A}}_{\mathrm{attr}} \\ \underline{\mathbf{a}}_{\mathrm{attr}}^\mathsf{T} \end{pmatrix}, \mathbf{A}_{\mathrm{circ}} = \begin{pmatrix} \overline{\mathbf{A}}_{\mathrm{circ}} \\ \underline{\mathbf{a}}_{\mathrm{circ}}^\mathsf{T} \end{pmatrix}, \mathbf{z}, \{\mathbf{z}'_j\}_{j\in[J]}$ in a straight-forward way[20] using r'.
- Compute and set

$$\begin{pmatrix} \overline{\mathbf{A}}_{C_j} \\ \underline{\mathbf{a}}_{C_j}^\mathsf{T} \end{pmatrix} = \mathbf{A}_{C_j} \leftarrow \mathsf{UEvalC}(\mathbf{A}_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, C_j), \qquad \mathbf{p}_j \leftarrow \overline{\mathbf{A}}_{C_j}\mathbf{G}^{-1}(\mathbf{z}'_j) + \mathbf{z},$$
$$\overline{\mathbf{A}}' \leftarrow (\overline{\mathbf{A}}_{\mathrm{attr}} - (1, \mathbf{x}^\mathsf{T}) \otimes \overline{\mathbf{G}}, \mathbf{z}), \qquad \mathbf{P} \leftarrow (\mathbf{p}_1, \ldots, \mathbf{p}_J),$$

  where $\overline{\mathbf{G}}$ is the first n rows of $\mathbf{G}$.
- Output $(\overline{\mathbf{A}}_{\mathrm{circ}}, \overline{\mathbf{A}}', \mathbf{P}, \sigma, \sigma', \sigma_{-1}, \sigma_{\mathrm{post}}, \sigma_{\mathrm{pre}})$.

We first show that $\mathsf{evcsLWE}^{\mathcal{S}}_{\mathrm{post}}$ implies ABE security against $\mathcal{A}$ by proving that in $\mathsf{Exp}^{f\text{-sel+},0}_{\mathrm{PHFE}}$, the challenge ciphertext is pseudorandom. We can rewrite the view of $\mathcal{A}$ in $\mathsf{Exp}^{f\text{-sel+},0}_{\mathrm{PHFE}}$ as

$$\mathsf{mpk} = (n, m, q, \sigma, \sigma', \sigma_{-1}, \sigma_{\mathrm{post}}, \mathbf{A}_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, \boxed{\mathbf{B}}, \mathbf{z}), \quad \{\mathsf{sk}_j\}_{j\in[J]} = (\{\mathbf{z}'_j\}_{j\in[J]}, \boxed{\mathbf{K}}),$$
$$\mathsf{ct}_{\mathbf{x}} = (\boxed{\mathbf{S}}, \mathbf{c}_{\mathrm{attr}}, \mathbf{c}_{\mathrm{circ}}, \boxed{\mathbf{c}_{\mathrm{B}}}, c_{\mathrm{msg}}), \quad \text{where}$$
$$\mathbf{c}_{\mathrm{attr}}^\mathsf{T} = \mathbf{s}^\mathsf{T}(\mathbf{A}_{\mathrm{attr}} - (1, \mathbf{x}^\mathsf{T}) \otimes \mathbf{G}) + \mathbf{e}_{\mathrm{attr}}^\mathsf{T}$$
$$= \boxed{\mathbf{r}^\mathsf{T}(\overline{\mathbf{A}}_{\mathrm{attr}} - (1, \mathbf{x}^\mathsf{T}) \otimes \overline{\mathbf{G}}) + \mathbf{e}_{\mathrm{attr}}^\mathsf{T}} - \underline{\mathbf{a}}_{\mathrm{attr}}^\mathsf{T} + (1, \mathbf{x}^\mathsf{T}) \otimes \boldsymbol{\iota}_{n+1}^\mathsf{T} \otimes \mathbf{g}^\mathsf{T},$$
$$\mathbf{c}_{\mathrm{circ}}^\mathsf{T} = \mathbf{s}^\mathsf{T}(\mathbf{A}_{\mathrm{circ}} - (1, \mathsf{bits}(\mathbf{S})) \otimes \mathbf{G}) + \mathbf{e}_{\mathrm{circ}}^\mathsf{T}$$
$$= \boxed{\mathbf{r}^\mathsf{T}(\overline{\mathbf{A}}_{\mathrm{circ}} - (1, \mathsf{bits}(\mathbf{S})) \otimes \overline{\mathbf{G}}) + \mathbf{e}_{\mathrm{circ}}^\mathsf{T}} - \underline{\mathbf{a}}_{\mathrm{circ}}^\mathsf{T} + (1, \mathsf{bits}(\boxed{\mathbf{S}})) \otimes \boldsymbol{\iota}_{n+1}^\mathsf{T} \otimes \mathbf{g}^\mathsf{T},$$
$$c_{\mathrm{msg}} = \boxed{\mathbf{r}^\mathsf{T}\mathbf{z} + e_{\mathrm{msg}}} + \mu \cdot \lfloor q/2 \rceil.$$

Here, the boxed terms are from the non-aux part of the first distribution in $\mathsf{evcsLWE}^{\mathcal{S}}_{\mathrm{post}}$. In the right distribution of $\mathsf{evcsLWE}^{\mathcal{S}}_{\mathrm{post}}$, all components of $\mathsf{ct}_{\mathbf{x}}$ are one-time padded by a random value due to the boxed terms. From that we conclude the very selective security of Construction 4.

It remains to prove $\mathsf{evcsLWE}^{\mathcal{S}}_{\mathrm{pre}}$ using csLWE. This is highly analogous to the proof of Theorem 16 except there is one more level of "one-time pad indirection". We omit the details here. □

5.4 Attribute-Unbounded Depth-Unbounded KP-ABE

We can apply the result of [GKW16] to obtain KP-ABE unbounded in both attribute length and circuit depth:

Corollary 24 (KP-ABE for circuits of unbounded depth and input length)**.** *Under* csLWE *(Assumption 1) and* $\mathsf{evcsLWE}$ *(Assumption 2), there exists a strongly f-selectively correct (Definition 6), very selectively secure (Definition 7) KP-ABE scheme for circuits of unbounded depth and input length (Definition 5) with*

$$|\mathsf{mpk}| = \mathrm{O}(1), \qquad |\mathsf{sk}_C| = \mathrm{O}(L), \qquad |\mathsf{ct}_{\mathbf{x}}| = \mathrm{O}(|\mathbf{x}|),$$
$$T_{\mathsf{Setup}} = \mathrm{O}(1), \qquad T_{\mathsf{KeyGen}}, T_{\mathsf{Dec}} = \mathrm{O}(|C|), \qquad T_{\mathsf{Enc}} = \mathrm{O}(|\mathbf{x}|),$$

where L is the input length of C (not $|\mathbf{x}|$).

[20] Precisely speaking, r' conditioned on $\mathbf{A}_{\mathrm{attr}}, \mathbf{A}_{\mathrm{circ}}, \mathbf{z}, \{\mathbf{z}'_j\}_{j\in[J]}$ should be efficiently sampleable given them so that r' does not contain a trapdoor breaking LWE for those public matrices.

The generic transformation of [GKW16] only handles equal-length matching, i.e., $\mathbf{x}$ must be of exactly the input length of C for decryption to succeed. It can be extended to prefix matching when based on Construction 4 with two changes:

- For correctness, we use a PRF to generate an exponentially large $\mathbf{A}_{\text{attr}}$ matrix (instead of unrelated $\mathbf{A}_{\text{attr}}$ matrices for different lengths), and use the relevant parts of it in key generation and encryption. This ensures that sk_C and $\mathsf{ct}_{\mathbf{x}}$ use the same $\mathbf{A}_{\text{attr}}$ for decryption to succeed.
- For security, we transform C and $\mathbf{x}$ into C' and $(L_{\mathrm{x}}, \mathbf{x})$, where $L_{\mathrm{x}} = |\mathbf{x}|$ is the λ-bit binary representation of the length of $\mathbf{x}$ and

$$C'(L_{\mathrm{x}}, \mathbf{x}') = \begin{cases} C(\mathbf{x}'), & \text{if } L_{\mathrm{x}} \geq L; \\ 1, & \text{if } L_{\mathrm{x}} < L; \end{cases}$$

  with L being the input length of C and $\mathbf{x}'$ (hopefully) a prefix of $\mathbf{x}$. In the security proof, during reduction to Construction 4, we choose $(|\mathbf{x}|, (\mathbf{x}, \mathbf{0}))$ for $\mathbf{0}$ of appropriate length when generating the challenge ciphertext (part of which is thrown away) so that the challenge has the correct length as far as Construction 4 is concerned and policy circuits taking overlong input are sure to reject decryption.

When using unbounded homomorphic evaluation with stronger correctness, the scheme becomes (computationally or statistically) *adaptively* correct.

Acknowledgments. The authors were supported by NSF grants CNS-1936825 (CAREER), CNS-2026774, a JP Morgan AI Research Award, a Cisco Research Award, and a Simons Collaboration on the Theory of Algorithmic Fairness. The views expressed are those of the authors and do not reflect the official policy or position of the funding agencies. The authors thank the anonymous reviewers of FOCS 2023 for their valuable comments.

References

[ABG+13] Prabhanjan Ananth, Dan Boneh, Sanjam Garg, Amit Sahai, and Mark Zhandry. Differing-inputs obfuscation and applications. Cryptology ePrint Archive, Report 2013/689, 2013. https://eprint.iacr.org/2013/689.

[ACFQ22] Prabhanjan Ananth, Kai-Min Chung, Xiong Fan, and Luowen Qian. Collusion-resistant functional encryption for RAMs. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part I*, volume 13791 of *LNCS*, pages 160–194. Springer, Heidelberg, December 2022.

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618. Springer, Heidelberg, August 2009.

[AJS17] Prabhanjan Ananth, Abhishek Jain, and Amit Sahai. Indistinguishability obfuscation for turing machines: Constant overhead and amortization. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 252–279. Springer, Heidelberg, August 2017.

[AM18] Shweta Agrawal and Monosij Maitra. FE and iO for turing machines from minimal assumptions. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 473–512. Springer, Heidelberg, November 2018.

[AS16] Prabhanjan Vijendra Ananth and Amit Sahai. Functional encryption for turing machines. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 125–153. Springer, Heidelberg, January 2016.

[BF11] Dan Boneh and David Mandell Freeman. Homomorphic signatures for polynomial functions. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 149–168. Springer, Heidelberg, May 2011.

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 533–556. Springer, Heidelberg, May 2014.

[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, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Heidelberg, August 2001.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *ITCS 2012*, pages 309–325. ACM, January 2012.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, Heidelberg, March 2011.

[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, Part I*, volume 10677 of *LNCS*, pages 264–302. Springer, Heidelberg, November 2017.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd FOCS*, pages 97–106. IEEE Computer Society Press, October 2011.

[BV15] Zvika Brakerski and Vinod Vaikuntanathan. Constrained key-homomorphic PRFs from standard lattice assumptions - or: How to secretly embed a circuit in your PRF. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 1–30. Springer, Heidelberg, March 2015.

[BV16] Zvika Brakerski and Vinod Vaikuntanathan. Circuit-ABE from LWE: Unbounded attributes and semi-adaptive security. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 363–384. Springer, Heidelberg, August 2016.

[BW13] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 280–300. Springer, Heidelberg, December 2013.

[CHL+15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 3–12. Springer, Heidelberg, April 2015.

[CLTV15] Ran Canetti, Huijia Lin, Stefano Tessaro, and Vinod Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 468–497. Springer, Heidelberg, March 2015.

[CVW18] Yilei Chen, Vinod Vaikuntanathan, and Hoeteck Wee. GGH15 beyond permutation branching programs: Proofs, attacks, and candidates. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 577–607. Springer, Heidelberg, August 2018.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DQV+21] Lalita Devadas, Willy Quach, Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Succinct LWE sampling, random polynomials, and obfuscation. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part II*, volume 13043 of *LNCS*, pages 256–287. Springer, Heidelberg, November 2021.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.

[GGH+13a] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.

[GGH+13b] Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and Brent Waters. Attribute-based encryption for circuits from multilinear maps. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 479–499. Springer, Heidelberg, August 2013.

[GKP+13a] Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. How to run turing machines on encrypted data. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 536–553. Springer, Heidelberg, August 2013.

[GKP+13b] Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 555–564. ACM Press, June 2013.

[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, Part II*, volume 9986 of *LNCS*, pages 361–388. Springer, Heidelberg, October / November 2016.

[GP21] Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. In Samir Khuller and Virginia Vassilevska Williams, editors, *53rd ACM STOC*, pages 736–749. ACM Press, June 2021.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309.

[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, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 75–92. Springer, Heidelberg, August 2013.

[GVW12] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 162–179. Springer, Heidelberg, August 2012.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 545–554. ACM Press, June 2013.

[GVW15a] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 503–523. Springer, Heidelberg, August 2015.

[GVW15b] Sergey Gorbunov, Vinod Vaikuntanathan, and Daniel Wichs. Leveled fully homomorphic signatures from standard lattices. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th ACM STOC*, pages 469–477. ACM Press, June 2015.

[GWZ22] Jiaxin Guan, Daniel Wichs, and Mark Zhandry. Incompressible cryptography. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part I*, volume 13275 of *LNCS*, pages 700–730. Springer, Heidelberg, May / June 2022.

[HJL21] Samuel B. Hopkins, Aayush Jain, and Huijia Lin. Counterexamples to new circular security assumptions underlying iO. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 673–700, Virtual Event, August 2021. Springer, Heidelberg.

[JLL23] Aayush Jain, Huijia Lin, and Ji Luo. On the optimal succinctness and efficiency of functional encryption and attribute-based encryption. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 479–510. Springer, Heidelberg, April 2023.

[JLLS23] Aayush Jain, Huijia Lin, Paul Lou, and Amit Sahai. Polynomial-time cryptanalysis of the subspace flooding assumption for post-quantum $i\mathcal{O}$. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part I*, volume 14004 of *LNCS*, pages 205–235. Springer, Heidelberg, April 2023.

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams, editors, *53rd ACM STOC*, pages 60–73. ACM Press, June 2021.

[JLS22] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from LPN over $\mathbb{F}_p$, DLIN, and PRGs in NC^0. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part I*, volume 13275 of *LNCS*, pages 670–699. Springer, Heidelberg, May / June 2022.

[KNTY19] Fuyuki Kitagawa, Ryo Nishimaki, Keisuke Tanaka, and Takashi Yamakawa. Adaptively secure and succinct functional encryption: Improving security and efficiency, simultaneously. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 521–551. Springer, Heidelberg, August 2019.

[LLL22] Hanjun Li, Huijia Lin, and Ji Luo. ABE for circuits with constant-size secret keys and adaptive security. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 680–710. Springer, Heidelberg, November 2022.

[MP11] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. Cryptology ePrint Archive, Report 2011/501, 2011. https://eprint.iacr.org/2011/501.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Heidelberg, April 2012.

[O'N10] Adam O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010. https://eprint.iacr.org/2010/556.

[QWW18a] Willy Quach, Hoeteck Wee, and Daniel Wichs. Laconic function evaluation and applications. In Mikkel Thorup, editor, *59th FOCS*, pages 859–870. IEEE Computer Society Press, October 2018.

[QWW18b] Willy Quach, Hoeteck Wee, and Daniel Wichs. Laconic function evaluation and applications. Cryptology ePrint Archive, Report 2018/409, 2018. https://eprint.iacr.org/2018/409.

[RAD78] Ronald L. Rivest, Len Adleman, and Michael L. Dertouzos. On data banks and privacy homomorphisms. *Foundations of Secure Computation*, pages 169–180, 1978.

[SS10] Amit Sahai and Hakan Seyalioglu. Worry-free encryption: functional encryption with public keys. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM CCS 2010*, pages 463–472. ACM Press, October 2010.

[SW05] Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Heidelberg, May 2005.

[Tsa22] Rotem Tsabary. Candidate witness encryption from lattice techniques. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 535–559. Springer, Heidelberg, August 2022.

[VWW22] Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Witness encryption and null-IO from evasive LWE. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part I*, volume 13791 of *LNCS*, pages 195–221. Springer, Heidelberg, December 2022.

[Wee22] Hoeteck Wee. Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 217–241. Springer, Heidelberg, May / June 2022.

[WW21] Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious LWE sampling. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part III*, volume 12698 of *LNCS*, pages 127–156. Springer, Heidelberg, October 2021.

[WWW22] Brent Waters, Hoeteck Wee, and David J. Wu. Multi-authority ABE from lattices without random oracles. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 651–679. Springer, Heidelberg, November 2022.