Private Constrained PRFs (and More) from LWE

Private Constrained PRFs (and more) from LWE

Zvika Brakerski* Rotem Tsabary* Vinod Vaikuntanathan† Hoeteck Wee‡

Abstract

In a constrained PRF, the owner of the PRF key K can generate constrained keys K_f that allow anyone to evaluate the PRF on inputs x that satisfy the predicate f (namely, where $f(x)$ is "true") but reveal no information about the PRF evaluation on the other inputs. A private constrained PRF goes further by requiring that the constrained key K_f hides the predicate f.

Boneh, Kim and Montgomery (EUROCRYPT 2017) presented a construction of private constrained PRF for point function constraints, and Canetti and Chen (EUROCRYPT 2017) presented a completely different construction for NC^1 constraints. In this work, we show two constructions of LWE-based constraint-hiding constrained PRFs for *general predicates* described by polynomial-size circuits.

The two constructions are based on two distinct techniques that we show have further applicability by constructing weak attribute-hiding predicate encryption schemes. In a nutshell, the first construction imports the technique of *modulus switching* from the FHE world into the domain of trapdoor extension and homomorphism. The second construction shows how to use the duality between FHE secret-key/randomness and ABE randomness/secret-key to construct a scheme with *dual use* of the same values for both FHE and ABE purposes.

*Weizmann Institute of Science, {zvika.brakerski,rotem.tsabary}@weizmann.ac.il.

†MIT, vinodv@mit.edu.

‡ENS, hoeteck@di.ens.fr.

Contents

1 Introduction

Lattice-based cryptography, and in particular the construction of cryptographic primitives based on the learning with errors (LWE) assumption [Reg05], has seen a significant leap in recent years. Most notably, we now have a number of constructions of cryptographic primitives that "compute on encrypted data". For example, fully homomorphic encryption (FHE) [Gen09, BV11, BGV12, GSW13], which enables arbitrary computation on encrypted data without knowledge of the secret key; attribute-based encryption (ABE) [SW05, GPSW06, GVW13, BGG+14], which supports fine-grained access control of encrypted data via the creation of restricted secret keys; new forms of pseudo-random functions (PRF) such as constrained PRFs [BW13, KPTZ13, BGI14]; and many more.

In this paper, we continue this line of inquiry and develop two new constructions of private ABE schemes (also called predicate encryption [BW07, KSW08, BSW11, O'N10]) and two new constructions of private constrained PRFs [BLW17], private variants of ABE and constrained PRFs respectively, that take us further along in the quest to extend the limits of computing on encrypted data using LWE-based techniques. Our private constrained PRFs support polynomial-time computable constraints, generalizing the recent results of Boneh, Kim and Montgomery [BKM17] for point functions and Canetti and Chen [CC17] for NC^1 functions.

In constructing these schemes, we develop two new techniques that we believe are as interesting in their own right as the end results themselves. We proceed to introduce the protagonists of our work and describe our results and techniques.

Predicate Encryption. Predicate Encryption (PE) is a strengthening of ABE with additional privacy guarantees [BW07, KSW08, BSW11, O'N10]. In a predicate encryption scheme, ciphertexts are associated with descriptive attributes x and a plaintext M; secret keys are associated with Boolean functions f; and a secret key decrypts the ciphertext to recover M if $f(x)$ is true (henceforth, for convenience of notation later in the paper, we denote this by $f(x) = 0$).

The most basic security guarantee for attribute-based encryption as well as predicate encryption, called *payload hiding*, stipulates that M should remain private given its encryption under attributes x^* and an unbounded number of *unauthorized* keys, namely secret keys sk_f where $f(x^*)$ is false (we denote this $f(x^*) = 1$). The additional requirement in predicate encryption refers to hiding the attribute x^* (beyond leaking whether $f(x^*)$ is true or false). It turns out that this requirement, called attribute-hiding, can be formalized in two ways. The first is the definition of *weak attribute-hiding*, which stipulates that x^* remains hidden given an unbounded number of *unauthorized* keys. The second, called *strong attribute-hiding*, stipulates that x^* remains hidden given an unbounded number of keys, which may comprise of both authorized and unauthorized keys. Both these requirements can be formalized using simulation-based and indistinguishability-based definitions (simulation based strong attribute hiding is known to be impossible [AGVW13]); jumping ahead, we remark that our constructions will achieve the stronger simulation-based definition but for weak attribute hiding.

A sequence of works showed the surprising power of strong attribute-hiding predicate encryption [BV15a, AJ15, BKS16]. A strong attribute-hiding PE scheme (for sufficiently powerful classes of predicates) gives us a functional encryption scheme [BSW11], which in turn can be used to build an indistinguishability obfuscation (IO) scheme [BV15a, AJ15], which in turn has emerged as a very powerful "hub of cryptography" [GGH+16, SW14].

The only strong attribute-hiding predicate encryption schemes we have under standard cryptographic assumptions are for very simple functionalities related to the inner product predicate [KSW08, BW07, OT12], and build on bilinear groups. On the other hand, Gorbunov, Vaikuntanathan and Wee (GVW) [GVW15a] recently constructed a weak attribute-hiding predicate encryption scheme for all circuits (of an a-priori bounded polynomial depth) from the LWE assumption. They also pointed out two barriers, two sources of leakage, that prevent their construction from achieving the strong attribute-hiding guarantee. Indeed, Agrawal [Agr16] showed that both sources of leakage can be exploited to recover the private attribute x^* in the GVW scheme, under strong attribute-hiding attacks in the GVW scheme (that is, using both authorized and unauthorized secret keys).[1]

Private Constrained PRFs (CPRFs). Constrained Pseudorandom Functions (CPRFs) [BW13, KPTZ13, BGI14] are pseudorandom functions (PRF) where it is possible to delegate the computation of the PRF on a subset of the inputs. Specifically, an adversary can ask for a constrained key σ_f corresponding to a function f, which is derived from the (global) seed σ. Using σ_f it is possible to compute $\mathsf{PRF}_\sigma(x)$ for all x where $f(x)$ is true (in our notation, again, $f(x) = 0$). However, if $f(x) = 1$ then $\mathsf{PRF}_\sigma(x)$ is indistinguishable from uniform even for an adversary holding σ_f. The original definition considers the case of unbounded collusion, i.e. security against an adversary that can ask for many different σ_{f_i}, but this is currently only achievable for very simple function classes or under strong assumptions such as multilinear maps or indistinguishability obfuscation. Many of the applications of CPRFs (e.g. for broadcast encryption [BW13] and identity based key exchange [HKKW14]) rely on collusion resilience, but some (such as the puncturing paradigm [SW14]) only require releasing a single key. Brakerski and Vaikuntanathan [BV15b] showed that single-key CPRF is achievable for all functions with a-priori depth bound and non-uniformity bound under the LWE assumption.

Boneh, Lewi and Wu [BLW17] recently considered *constraint hiding* CPRFs (CH-CPRF or private CPRFs) where the constrained key σ_f does not reveal f (so, in a sense, the constrained key holder cannot tell whether it is computing the right value or not). They showed various applications for this new primitive, as well as constructions from multilinear maps and obfuscation for various function classes. Very recently, Boneh, Kim and Montgomery [BKM17] showed how to construct single-key private CPRFs for point functions, and Canetti and Chen [CC17] showed how to construct a single-key private CPRF for the class of NC^1 circuits (i.e. polynomial-size formulae). Both their constructions are secure under the LWE assumption. They also showed that even collusion resistance against 2-keys would imply indistinguishability obfuscation.

The technical core of these constructions is lattice-based constructions of PRFs, initiated by Banerjee, Peikert and Rosen [BPR12] and developed in a line of followup works [BP14, BLMR15, BFP+15, BV15b].

1.1 Our Results

In this work, we present two new techniques for achieving the attribute-hiding guarantee from the LWE assumption. We exemplify the novelty and usefulness of our techniques by showing that they can be used to derive new predicate encryption schemes and new constraint-hiding constrained

[1] In addition, we also have several constructions of *functional encryption* schemes for computing inner products over large fields [ABCP15, BJK15, ALS16] (as opposed to the inner product predicate) and for quadratic functions [Lin16, Gay16] from standard assumptions.

PRFs [BLW17, CC17]. In particular, under the (polynomial hardness of the subexponential noise rate) LWE assumption, we construct:

- Our main result is two constructions of single-key constraint-hiding constrained PRF families for all circuits (of an a-priori bounded polynomial depth). This generalizes recent results of [BKM17] who handle point functions and [CC17] who handle NC1 circuits. Our new techniques allow us to handle arbitrary polynomial-time constraints (of an a-priori bounded depth), which does not seem to follow from previous PE techniques, e.g., [GVW15a]. We describe constrained PRFs, constraint-hiding and our constructions in more detail in the sequel.

- Along the way to our main result, we also derive two new predicate encryption schemes that achieve the weak attribute-hiding security guarantee. Our predicate secret keys are shorter than in [GVW15a] by a $\mathsf{poly}(\lambda)$ factor. They also avoid the first source of leakage identified in [GVW15a, Agr16]. We will describe these features in more detail in the sequel.

Technical Background. Like [GVW15a], we build a predicate encryption scheme starting from an FHE and an ABE, following the "FHE+ABE" paradigm introduced in [GVW12, GKP+13] for the setting of a-priori bounded collusions. The idea is to first use FHE to produce an encryption Ψ of the attribute x, and use Ψ as the attribute in an ABE. This paradigm allows us to reduce the problem of protecting arbitrary polynomial-time computation f on a private attribute x to protecting a fixed computation, namely FHE decryption, on the FHE secret key. Henceforth, we suppress the issue of carrying out FHE homomorphic evaluation on the encrypted attribute, which can be handled via the underlying ABE as in [GVW15a], and focus on the issue of FHE decryption, which is where we depart from prior works.

With all LWE-based FHE schemes [BV11,BGV12,GSW13,BV14,AP14], decryption corresponds to computing an inner product modulo q followed by a threshold function. While constructing a *strongly* attribute hiding PE scheme for this function class is still beyond reach,[2] GVW construct an LWE-based weakly attribute hiding scheme by extending previous works [AFV11], and show how to attach it to the end of the decryption process of [BGG+14] ABE. Specifically, Agrawal, Freeman and Vaikuntanathan [AFV11] showed how to construct weakly attribute hiding PE for orthogonality checking modulo q, i.e. the class where attributes $\mathbf{x}$ and functions $f_{\mathbf{y}}$ correspond to vectors and decryption is possible if $\langle \mathbf{x}, \mathbf{y} \rangle = 0 \pmod{q}$. GVW rely on an additional feature of LWE-based FHE: that the value to be rounded after the inner product can be made polynomially bounded. Thus inner product plus rounding can be interpreted as a sequence of shifted inner products that are supported by [AFV11]. This in particular means that an authorized decryptor learns which of the shifts had been the successful one, a value that depends on the FHE randomness. This is one of the reasons why the GVW scheme is not strongly attribute hiding; there are others as described in [Agr16].

First New Technique: Dual Use. In this technique, we use the same LWE secret for the FHE and the ABE. Our main observation is that the structure of the [BGG+14] ABE scheme and that of the [GSW13] FHE scheme are so very similar that we can use the same LWE secret in both schemes. This can be viewed as encrypting the attribute under some FHE key, and then providing partly decrypted pieces as the ABE ciphertext. The PE decryption process first "puts the pieces together"

[2] There are constructions for function classes that semantically seem astonishingly similar, such as inner product over the integers followed by rounding [ALS16] but there appears to be a big technical gap between these classes.

according to the FHE homomorphic evaluation function, which makes the ABE ciphertext decrypt its own FHE component, leaving us with an ABE ciphertext which is ready to be decrypted using the ABE key. Proving security for this approach requires to delicately argue about the randomness used in the FHE encryption.

Second New Technique: Modulus Switching and HNF Lattice Trapdoors. In this technique, we attempt to implement the rounding post inner-product straightforwardly by rounding the resulting ciphertext. This does not work since the attribute is encoded in the ciphertext in a robust way, so it is not affected by rounding (this is why more sophisticated methods were introduced in the past). However, we show how to homomorphically modify the rounding in a way that makes the rounding effective for small noise, and yet preserves the most significant bits properly encoded. Interestingly, for the proof of security of our PE scheme, we utilize the ability of generating trapdoors for LWE lattices of the form $[\mathbf{I}\|\mathbf{A}]$ (which corresponds to Hermite Normal Form), even when generating a trapdoor for $\mathbf{A}$ itself is not possible.

We first construct predicate encryption schemes using our techniques, on the way to our main results, which are constructions of constraint-hiding CPRFs for general constraints. With this executive summary, we move on to a more in-depth technical discussion of our results and techniques.

2 Technical Overview

We provide a brief overview of the GVW predicate encryption scheme, along with our constructions, focusing on the points where they differ and supressing many technical details.

The [GVW15a] scheme. In the GVW scheme, the decryption algorithm on input an encryption of x and the secret key for f, computes a vector over $\mathbb{Z}_q$ of the form:

$$\mathbf{s}[\mathbf{A}\|\mathbf{A}_f - (f(x)\cdot t + \delta)\mathbf{G}] + \mathsf{noise} \tag{1}$$

where $\mathbf{A}_f$ is deterministically derived from the public parameters and f (the precise derivation is not relevant for the overview), and $f(x)\cdot t+\delta$ corresponds to the inner product of a FHE ciphertext (upon homomorphic evaluation) and the corresponding secret key. Here, δ is a small noise value bounded by B, and $t \gg B$ is a large constant, most commonly $t = \lfloor \frac{q}{2} \rceil$ (but we will also use other values, see below). As usual in LWE-based constructions, the vector $\mathbf{s}$ is an "LWE secret", and we use noise to denote non-specific low norm noise that is added to the ciphertext and accumulates as it is processed.[3]

Decryption should be permitted when $f(x) = 0$, which indicates that the policy f accepts the attribute x, (and forbidden when $f(x) = 1$). Therefore, the GVW scheme gives out trapdoors for the $2B + 1$ lattices

$$[\mathbf{A}\|\mathbf{A}_f - \beta\mathbf{G}], \quad \forall|\beta| \le B \ ,$$

and decryption tries all trapdoors until one works. This is called the "lazy OR" evaluation in [GVW15a] and has at least two problems: (1) In the context of a predicate encryption scheme, this ruins security by letting a successful decryption leak the FHE noise δ; and (2) Looking ahead, in the context of a constraint-hiding CPRF scheme (where one switches the function f and the input x), it ruins even correctness, preventing the holder of a constrained key from recovering the PRF value $\mathbf{s}[\mathbf{A}\|\mathbf{A}_x]$; rather, she only gets $\mathbf{s}[\mathbf{A}\|\mathbf{A}_x - \beta\mathbf{G}]$ for some small noise term β.

[3] A knowledgeable reader might notice that in [GVW15a] there is a plus sign in Eq. (1) instead of the minus sign. This alternative notation is equivalent and will be more useful for us.

Moving on, in the proof of security, a simulator needs to generate secret keys whenever $f(x) = 1$. To this end, the reduction knows a short $\mathbf{R}_f$ for which

$$\mathbf{A}\mathbf{R}_f = \mathbf{A}_f - (t + \delta^*)\mathbf{G} \tag{2}$$

We can then rewrite

$$[\mathbf{A} \| \mathbf{A}_f - \beta\mathbf{G}] = [\mathbf{A} \| \mathbf{A}\mathbf{R}_f + (\delta^* + t - \beta)\mathbf{G}]$$

and since $\beta - \delta^* - t \neq 0$, we will be able to generate trapdoors for this lattice knowing only $\mathbf{R}_f$ using the trapdoor extension techniques of [ABB10b, MP12].

2.1 Dual-Use of Secret and Randomness

Our first technique hinges on the key observation is that the structure of the [BGG+14] ABE scheme and that of the [GSW13] FHE scheme are so very similar that we can use the same LWE secret in both schemes; we refer to this as the "dual use" technique.

Instead of (1), we will compute a vector of the form

$$\mathbf{s}[\mathbf{B} \| \mathbf{B}_f - \overline{\Psi}_f] \tag{3}$$

Here, Ψ_f denotes the GSW FHE ciphertext upon homomorphic evaluation under the key $(\mathbf{s} \; -1)$. We stress that we are reusing $\mathbf{s}$ here. Concretely, Ψ_f can be written as:

$$\begin{pmatrix} \mathbf{B} \\ \mathbf{s}\mathbf{B} + \mathbf{e} \end{pmatrix} \mathbf{R}_f + f(x)\mathbf{G}$$

where $\mathbf{R}_f$ is small. $\overline{\Psi}_f$ refers to Ψ_f with the bottom row $\underline{\Psi}_f$ deleted so that it has the same height as $\mathbf{B}$. Again, we need to first explain how to arrive at a vector of this form, and second, how to generate secret keys for such vectors.

Compactification. Using techniques from ABE, we can generate vectors of the form:

$$\mathbf{s}[\mathbf{B} \| \mathbf{B}_{f_j} - \psi_{f,j}\overline{\mathbf{G}}]$$

where $\psi_{f,j} \in \mathbb{Z}_q$ are the entries of $\overline{\Psi}_j$ and $\overline{\mathbf{G}}$ refers to $\mathbf{G}$ with the bottom row deleted. In particular, we can write

$$\Psi_f = \sum_j \psi_{f,j} \cdot \mathbf{E}_j$$

where $\mathbf{E}_j$ is a $0, 1$-matrix whose j'th entry is 1 and 0 everywhere else. Then, we can write

$$\mathbf{B}_f + \overline{\Psi}_f = \sum_j (\mathbf{B}_{f_j} - \psi_{f,j}\overline{\mathbf{G}}) \cdot \overline{\mathbf{G}}^{-1}(\mathbf{E}_j)$$

Dual Use Decryption. The secret key for f is a trapdoor for the lattice $[\mathbf{B} \| \mathbf{B}_f]$. Observe that whenever $f(x) = 0$, we have $(\mathbf{s} \; -1)\Psi_f \approx \mathbf{0}$ (property of GSW FHE), which means we can compute

$$\mathbf{s}[\mathbf{B} \| \mathbf{B}_f - \overline{\Psi}_f] + [\mathbf{0} \| \underline{\Psi}_f] \approx \mathbf{s}[\mathbf{B} \| \mathbf{B}_f]$$

and thus decrypt.

In order to generate secret key whenever $f(x) = 1$ in the proof of security, the reduction knows a short $\mathbf{W}_f$ for which

$$\mathbf{B}\mathbf{W}_f = \mathbf{B}_f - \overline{\Psi}_f$$

We can then rewrite

$$[\mathbf{B}\|\mathbf{B}_f] = [\mathbf{B}\|\mathbf{B}\mathbf{R}_f + \overline{\Psi}_f] = [\mathbf{B}\|\mathbf{B}(\mathbf{R}_f - \mathbf{W}_f) + \overline{\mathbf{G}}]$$

and we will be able to generate trapdoors for this lattice knowing only $\mathbf{R}_f, \mathbf{W}_f$.

2.2 Modulus Switching and Trapdoor Extension in Hermite Normal Form

The crux of this technique is to replace Eq. (1) with a computation producing a vector of the form

$$\mathbf{s}[\mathbf{A}'\|\mathbf{A}'_f - f(x)\mathbf{G}'] + \mathsf{noise} \tag{4}$$

where $\mathbf{G}'$ is a different gadget matrix and $\mathbf{A}'_f$ is again deterministically derived from the public parameters and f. We will also make sure to sample a small $\mathbf{s}$, specifically from the LWE noise distribution (this is known as LWE in *Hermite Normal Form* (HNF) and was shown equivalent to the standard form [ACPS09]), the reason for doing so will be clear in a little bit. Next, we will address two challenges: first, how to arrive at a vector of this form, and second, how to generate secret keys for such vectors, both of which require new techniques.

Modulus Switching. We first describe how to get to Eq. (4) starting from Eq. (1) (to get to the latter, we will proceed as in GVW). We would like to use the magnitude gap between t and δ, and, inspired by modulus switching techniques in FHE [BV11,BGV12], "divide by t" to remove the dependence on δ. This seems odd at first since $t \cdot \mathbf{G}$ and $\delta \cdot \mathbf{G}$ actually have the same magnitude, so dividing by t will not eliminate the δ component. Therefore we will first find a linear transformation that maps $\delta\mathbf{G}$ into a matrix of small entries, while mapping $t \cdot \mathbf{G}$ into a gadget matrix with big entries. Recall that eventually this transformation is to be applied to the processed ciphertext from Eq. (1), so due to the noise component, we are only allowed linear operations with small coefficients (or more explicitly, multiplying on the right by a matrix with small values).

As we pointed out $\delta\mathbf{G}$ and $t\mathbf{G}$ have the same magnitude so it might seem odd that a low-magnitude linear transformation can shift them so far apart. However, since $\mathbf{G}$ is a matrix with public trapdoor, it is possible to convert $\mathbf{G}$ into any other matrix $\mathbf{M}$ using a small magnitude linear transformation which is denoted by $\mathbf{G}^{-1}(\mathbf{M})$ (note that this is just a formal notation, since $\mathbf{G}$ doesn't have an actual inverse). Specifically, we will multiply by $\mathbf{G}^{-1}(\mathbf{G}_p)$, where $\mathbf{G}_p$ is the gadget matrix w.r.t a smaller modulus $p = q/t$ (we assume that p is integer). Recall that our conceptual goal is to divide by t, and end up with a ciphertext in $\mathbb{Z}_p$, we can now reveal that indeed $\mathbf{G}' = \mathbf{G}_p$. Applying this transformation to the ciphertext results in

$$\mathbf{s}[\mathbf{A}\|\mathbf{A}_f\mathbf{G}^{-1}(\mathbf{G}_p) - f(x)t\mathbf{G}_p] - [0\|\delta\mathbf{s}\mathbf{G}_p] + \mathsf{noise}\ , \tag{5}$$

and indeed, since we use low-norm $\mathbf{s}$, we have that $\|\delta\mathbf{s}\mathbf{G}_p\| \ll q$, and we can now think about it as part of the noise. However, $t\mathbf{G}_p$ is still not a valid gadget matrix over $\mathbb{Z}_q$. Still, we can now divide the entire expression by t which results in

$$\mathbf{s}\big[\underbrace{\lfloor \mathbf{A}/t \rfloor}_{\mathbf{A}'} \| \underbrace{\lfloor \mathbf{A}_f\mathbf{G}^{-1}(\mathbf{G}_p)/t \rfloor}_{\mathbf{A}'_f} - f(x)\mathbf{G}_p\big] + \mathsf{noise} \pmod p\ , \tag{6}$$

as in Eq. (4). This technique is reminiscent of the one used by Boneh, Kim and Montgomery [BKM17] in constructing a private CPRF for point functions (but was obtained independently of theirs).

HNF Trapdoor Extension. The standard way to generate keys that decrypt whenever $f(x) = 0$ is to provide a trapdoor for $[\mathbf{A}'\|\mathbf{A}'_f]$ (over $\mathbb{Z}_p$) as in previous ABE schemes. Indeed, this will provide the required functionality, but introduce problems in the proof. As in Eq. (2), the simulator can find a low-magnitude $\mathbf{R}_f$ s.t. $\mathbf{A}\mathbf{R}_f = \mathbf{A}_f + (t + \delta^*)\mathbf{G}$, however, when applying our modulus switching from above, we get

$$\mathbf{A}'\mathbf{R}'_f = \mathbf{A}'_f - \mathbf{G}_p - \mathbf{E} \ ,$$

where $\mathbf{E}$ is a low-magnitude error matrix which is the result of the bias introduced by δ^* and various rounding errors (note that $\mathbf{E}$ is easily computable given $\mathbf{R}'_f$). Therefore, we have that

$$[\mathbf{A}'\|\mathbf{A}'_f] = [\mathbf{A}'\|\mathbf{A}'\mathbf{R}'_f + \mathbf{G}_p + \mathbf{E}] \ ,$$

which is no longer a form for which we can find a trapdoor using $\mathbf{R}'_f$.

To resolve this, we observe that we *can* find a trapdoor for the matrix $[\mathbf{I}\|\mathbf{A}'\|\mathbf{A}'_f] = [\mathbf{I}\|\mathbf{A}'\|\mathbf{A}'\mathbf{R}'_f + \mathbf{G}_p + \mathbf{E}]$, which corresponds to generating trapdoors for *lattices* in Hermite Normal Form. This follows from the trapdoor extension methods of [ABB10b, MP12] since

$$[\mathbf{I}\|\mathbf{A}'\|\mathbf{A}'\mathbf{R}'_f + \mathbf{G}_p + \mathbf{E}] \cdot \begin{bmatrix} -\mathbf{E} \\ -\mathbf{R}'_f \\ \mathbf{I} \end{bmatrix} = \mathbf{G}_p \ .$$

We will therefore change the way secret keys are generated in our scheme, and generate them as trapdoors for $[\mathbf{I}\|\mathbf{A}'\|\mathbf{A}'_f]$ instead of trapdoors for $[\mathbf{A}'\|\mathbf{A}'_f]$. This might seem problematic because our ciphertext processes to $\mathbf{s}[\mathbf{A}'\|\mathbf{A}'_f - f(x)\mathbf{G}'] + \mathsf{noise}$ as in Eq. (4) and not to $\mathbf{s}[\mathbf{I}\|\mathbf{A}'\|\mathbf{A}'_f - f(x)\mathbf{G}'] + \mathsf{noise}$. However, since $\mathbf{s}$ is short, the zero vector itself has the form $\mathbf{0} = \mathbf{s}\mathbf{I} + \mathsf{noise}$ (with $\mathsf{noise} = -\mathbf{s}$), and therefore we can always extend our ciphertext to this new form just by concatenating the zero vector.

Comparison with GVW15 Predicate Encryption. [GVW15a] pointed out that there are two barriers to achieving strongly attribute-hiding predicate encryption from LWE. First, multiple shifts approach to handle threshold inner product for FHE decryption leaks the exact inner product and therefore cannot be used to achieve full attribute-hiding. That is, authorized keys leak the FHE decryption key and in turn the private attribute x. Second, we do not currently know of a fully attribute-hiding inner product encryption scheme under the LWE assumption. Here, authorized keys leak the error terms used in the ciphertext. Indeed, Agrawal [Agr16] showed that both sources of leakage can be exploited to recover the private attribute x in the GVW scheme. Both of our new constructions do not explicitly contain the first source of leakage.

2.3 From PE to Constraint Hiding CPRF

It was shown in [BV15b] that the [BGG+14] ABE structure can be used to construct constrained PRFs for arbitrary bounded-uniformity bounded-depth functions, without collusion. Namely, a pseudorandom function where it is possible to produce a constrained key σ_f for a function f whose description length is a-priori bounded by ℓ and its depth is a-priori bounded by d, s.t. the constrained key can be used to compute $\mathsf{PRF}(x)$ for all x where $f(x) = 0$. At a high level, they considered a

set of public parameters for the ABE scheme, and some ciphertext randomness $\mathbf{s}$ (currently not corresponding to any concrete ciphertext). To compute the PRF at point x, the considered the circuit $\mathcal{U}_x$ which is the universal circuit that takes an ℓ-bit long description of a depth-d function, and evaluates it on x. Now, they compute $\mathsf{PRF}_{\mathbf{s}}(x) = \left\lfloor \frac{\mathbf{s}\mathbf{A}_{\mathcal{U}_x}}{T} \right\rceil$ for a sufficiently large T. This essentially the deterministic variant to setting $\mathsf{PRF}_{\mathbf{s}}(x) = \mathbf{s}\mathbf{A}_{\mathcal{U}_x} + \mathsf{noise}$ except here the noise is deterministic since the PRF computation needs to be deterministic. The matrix $\mathbf{A}_{\mathcal{U}_x}$ is exactly the matrix that would be computed in the ABE decryption process if given a key $\mathsf{sk}_{\mathcal{U}_x}$. The constrained key corresponds to an ABE ciphertext encrypting the description of f Therefore, constrained keys can be processed like ABE ciphertexts into the form $\mathbf{s}(\mathbf{A}_{\mathcal{U}_x} - \mathcal{U}_x(f)\mathbf{G}) + \mathsf{noise}$, for any circuit $\mathcal{U}_x$. Indeed, when $f(x) = 0$ the constrained key can be used to compute $\mathsf{PRF}(x)$. The construction itself is more complicated and contains additional features to ensure pseudorandomness in all of the points that cannot be computed using the constrained key.

This seems to be readily extendable to the PE setting, where the attribute hiding property should guarantee the constraint hiding of the CPRF. Indeed, now as in Eq. (1), the constrained key will only process to $\mathbf{s}(\mathbf{A}_{\mathcal{U}_x} - (tf(x)+\delta)\mathbf{G}) + \mathsf{noise}$. When $f(x) = 0$ this is equal to $\mathbf{s}(\mathbf{A}_{\mathcal{U}_x} - \delta\mathbf{G}) + \mathsf{noise}$ which does not allow to compute the correct value.

However, it is easy to see how using either of our new methods it is possible to overcome this issue. In a sense, in both methods the FHE noise which is embodied in the δ term is made small enough to be conjoined with the noise. The modulus switching technique allows to remove the δ term via multiplication by $\mathbf{G}^{-1}(\mathbf{G}_p)$ and dividing by t, and in the dual use method, the FHE noise is not multiplied by $\mathbf{G}$ to begin with. There are many other technical details to be dealt with, but they are resolved in ways inspired by [BV15b]. One technical difference between our solution and [BV15b] is that we do not use admissible hash functions to go from unpredictability to pseudorandomness, but instead we "compose" with the Banerjee-Peikert [BP14] pseudorandom function, which saves some complication as well as tightens the reduction somewhat. This could be used even in the setting of [BV15b] when constraint hiding is not sought.

Organization of the Paper. We start the rest of this paper with background information on lattices, LWE, trapdoors and FHE schemes in Section 3. Our first technique, namely dual-use, and the resulting PE and private CPRF scheme are presented in Section 4. Our second technique, namely HNF trapdoors and modulus switching, and the resulting PE and private CPRF schemes are presented in Section 5. These two sections can be read independently of each other. In each section, we first present the PE scheme and then the private CPRF scheme.

3 Preliminaries

3.1 Constrained Pseudo-Random Functions

In a constrained PRF family [BW13, BGI14, KPTZ13], the owner of a PRF key σ can compute a constrained PRF key σ_f corresponding to any Boolean circuit f. Given σ_f, anyone can compute the PRF on inputs x such that $f(x) = 0$. (As described before, our convention throughout this paper is that $f(x) = 0$ corresponds to the predicate f being satisfied). Furthermore, σ_f does not reveal any information about the PRF values at the other locations. A constrained PRF family is *constraint-hiding* if σ_f does not reveal any information about the internals of f. This requirement can be formalized through either an indistinguishability-based or simulation-based

definition [BLW17, CC17, BKM17]. Below, we present the definition of a constrained PRF adapted from [BV15b].

Definition 3.1 (Constrained PRF)**.** *A* constrained *pseudo-random function (PRF) family is defined by a tuple of algorithms* $(\mathsf{KeyGen}, \mathsf{Eval}, \mathsf{Constrain}, \mathsf{ConstrainEval})$ *where:*

- $\mathsf{KeyGen}(1^\lambda, 1^\ell, 1^d, 1^r)$ *is a* PPT *algorithm that takes as input the security parameter* λ*, a circuit max-length* ℓ*, a circuit max-depth* d *and an output space* r*, and outputs a PRF key* σ *and public parameters* pp*.*
- $\mathsf{Eval}_{\mathsf{pp}}(\sigma, x)$ *is a deterministic algorithm that takes as input a key* σ *and a string* $x \in \{0,1\}^*$*, and outputs* $y \in \mathbb{Z}_r$*;*
- $\mathsf{Constrain}_{\mathsf{pp}}(\sigma, f)$ *is a* PPT *algorithm that takes as input a PRF key* σ *and a circuit* $f : \{0,1\}^* \to \{0,1\}$*, and outputs a constrained key* σ_f*;*
- $\mathsf{ConstrainEval}_{\mathsf{pp}}(\sigma_f, x)$ *is a deterministic algorithm that takes as input a constrained key* σ_f *and a string* $x \in \{0,1\}^*$*, and outputs either a string* $y \in \mathbb{Z}_r$ *or* $\perp$*.*

Previous works define and analyze the correctness, pseudorandomness and constraint hiding properties separately. However, for our purposes it will be easiest to define a single game that captures all of these properties at the same time. This definition is equivalent to computational correctness and selective punctured pseudorandomness [BV15b], and selective constraint hiding [BLW15].

Definition 3.2. *Consider the following game between a PPT adversary* $\mathcal{A}$ *and a challenger:*

1. $\mathcal{A}$ *sends* $1^\ell, 1^d$ *and* $f_0, f_1 \in \{0,1\}^\ell$ *to the challenger.*
2. *The challenger generates* $(\mathsf{pp}, seed) \leftarrow \mathsf{Keygen}(1^\lambda, 1^\ell, 1^d, 1^r)$*. It flips three coins* $b_1, b_2, b_3 \xleftarrow{\$} \{0,1\}$*, intuitively* b_1 *selects whether* f_0 *or* f_1 *are used for the constraint,* b_2 *selects whether a real or random value is returned on queries non-constrained queries, and* b_3 *selects whether the actual or constrained value is returned on constrained queries.*

   The challenger creates $seed_f \leftarrow \mathsf{Constrain}_{\mathsf{pp}}(seed, f_{b_1})$*, and sends* $(\mathsf{pp}, seed_f)$ *to* $\mathcal{A}$*.*
3. $\mathcal{A}$ *adaptively sends unique queries* $x \in \{0,1\}^*$ *to the challenger (i.e. no* x *is queried more than once). The challenger returns:*

$$y = \begin{cases} \perp, & \text{if } f_0(x) \neq f_1(x). \\ U(\mathbb{Z}_r), & \text{if } (f_0(x) = f_1(x) = 1) \wedge (b_2 = 1). \\ \mathsf{ConstrainEval}_{\mathsf{pp}}(\sigma_f, x), & \text{if } (f_0(x) = f_1(x) = 0) \wedge (b_3 = 0). \\ \mathsf{Eval}_{\mathsf{pp}}(\sigma, x), & \text{otherwise.} \end{cases}$$

4. $\mathcal{A}$ *sends a guess* (i, b')*.*

The advantage of the adversary in this game is defined as $\mathrm{Adv}[\mathcal{A}] = |\Pr[b' = b_i] - 1/2|$*. A family of PRFs* $(\mathsf{KeyGen}, \mathsf{Eval}, \mathsf{Constrain}, \mathsf{ConstrainEval})$ *is a* single-key constraint-hiding selective-function constrained PRF *if for every PPT adversary* $\mathcal{A}$*,* $\mathsf{Adv}[\mathcal{A}] = \mathrm{negl}(\lambda)$*.*

3.2 Weakly Attribute Hiding Predicate Encryption

Following prior works, we associate $C(x) = 0$ as true and authorized, and $C(x) \neq 0$ as false and unauthorized.

Syntax. A Predicate Encryption scheme PE for input universe $\mathcal{X}$, a predicate universe $\mathcal{C}$, a message space $\mathcal{M}$, consists of four algorithms (PE.Setup, PE.Enc, PE.KeyGen, PE.Dec):

- $\text{PE.Setup}(1^\lambda, \mathcal{X}, \mathcal{C}, \mathcal{M}) \to (pp, msk)$. The setup algorithm gets as input the security parameter λ and a description of $(\mathcal{X}, \mathcal{C}, \mathcal{M})$ and outputs the public parameter pp, and the master key msk.
- $\text{PE.Enc}(pp, x, \mu) \to \mathsf{ct}$. The encryption algorithm gets as input pp, an attribute $x \in \mathcal{X}$ and a message $\mu \in \mathcal{M}$. It outputs a ciphertext ct.
- $\text{PE.KeyGen}(msk, C) \to \mathsf{sk}_C$. The key generation algorithm gets as input msk and a predicate $C \in \mathcal{C}$. It outputs a secret key sk_C.
- $\text{PE.Dec}((\mathsf{sk}_C, C), \mathsf{ct}) \to \mu$. The decryption algorithm gets as input the secret key sk_C, a predicate C, and a ciphertext ct. It outputs a message $\mu \in \mathcal{M}$ or $\perp$.

Correctness. We require that for all $\text{PE.Setup}(1^\lambda, \mathcal{X}, \mathcal{C}, \mathcal{M}) \to (pp, msk)$, for all $(x, C) \in \mathcal{X} \times \mathcal{C}$ such that $C(x) = 0$, for all $\mu \in \mathcal{M}$,

$$\Pr\left[\text{PE.Dec}((\mathsf{sk}_C, C), \mathsf{ct}) = \mu\right] \geq 1 - \text{negl}(\lambda),$$

where the probabilities are taken over the coin of PE.Setup, $\mathsf{sk}_C \leftarrow \text{PE.KeyGen}(msk, C)$, $\mathsf{ct} \leftarrow \text{PE.Enc}(pp, x, \mu)$.

Definition 3.3 (PE (Weak) Attribute-Hiding)**.** *Fix* (PE.Setup, PE.Enc, PE.KeyGen, PE.Dec)*. For every stateful p.p.t. adversary* Adv*, and a p.p.t. simulator* Sim*, consider the following two experiments:*

$\underline{\exp^{\mathsf{real}}_{\mathcal{PE},\text{Adv}}(1^\lambda)}$**:**	$\underline{\exp^{\mathsf{ideal}}_{\mathcal{PE},\text{Sim}}(1^\lambda)}$**:**
1: $x \leftarrow \text{Adv}(1^\lambda, \mathcal{X}, \mathcal{C}, \mathcal{M})$	1: $x \leftarrow \text{Adv}(1^\lambda, \mathcal{X}, \mathcal{C}, \mathcal{M})$
2: $(pp, msk) \leftarrow$ $\text{PE.Setup}(1^\lambda, \mathcal{X}, \mathcal{C}, \mathcal{M})$	2: $(pp, msk) \leftarrow$ $\text{PE.Setup}(1^\lambda, \mathcal{X}, \mathcal{C}, \mathcal{M})$
3: $\mu \leftarrow \text{Adv}^{\text{PE.KeyGen}(msk,\cdot)}(pp)$	3: $\mu \leftarrow \text{Adv}^{\text{PE.KeyGen}(msk,\cdot)}(pp)$
4: $\mathsf{ct} \leftarrow \text{PE.Enc}(pp, x, \mu)$	4: $\mathsf{ct} \leftarrow \text{Sim}(\mathsf{mpk}, \mathcal{X}, \mathcal{M})$
5: $\alpha \leftarrow \text{Adv}^{\text{PE.KeyGen}(msk,\cdot)}(\mathsf{ct})$	5: $\alpha \leftarrow \text{Adv}^{\text{PE.KeyGen}(msk,\cdot)}(\mathsf{ct})$
6: *Output* (x, μ, α)	6: *Output* (x, μ, α)

We say an adversary Adv *is admissible if all oracle queries that it makes $C \in \mathcal{C}$ satisfy $C(x) \neq 0$ (i.e. false). The Predicate Encryption scheme $\mathcal{PE}$ is then said to be* (weak) attribute-hiding *if there is a p.p.t. simulator* Sim *such that for every stateful p.p.t. adversary* Adv*, the following two distributions are computationally indistinguishable:*

$$\left\{\exp^{\mathsf{real}}_{\mathcal{PE},\text{Adv}}(1^\lambda)\right\}_{\lambda\in\mathbb{N}} \overset{c}{\approx} \left\{\exp^{\mathsf{ideal}}_{\mathcal{PE},\text{Sim}}(1^\lambda)\right\}_{\lambda\in\mathbb{N}}$$

3.3 Learning With Errors

The *Learning with Errors* (LWE) problem was introduced by Regev [Reg05]. Our scheme relies on the hardness of its decisional version.

Definition 3.4 (Decisional LWE (DLWE) [Reg05] and its HNF [ACPS09])**.** *Let λ be the security parameter, $n = n(\lambda)$ and $q = q(\lambda)$ be integers and let $\chi = \chi(\lambda)$ be a probability distribution over $\mathbb{Z}$. The $\mathrm{DLWE}_{n,q,\chi}$ problem states that for all $m = \mathrm{poly}(n)$, letting $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \chi^m$, and $\mathbf{u} \leftarrow \mathbb{Z}_q^m$, it holds that $(\mathbf{A}, \mathbf{sA}+\mathbf{e})$ and $(\mathbf{A}, \mathbf{u})$ are computationally indistinguishable. The problem is equally hard in its "Hermite Normal Form": when sampling $\mathbf{s} \leftarrow \chi^n$.*

In this work we only consider the case where $q \leq 2^n$. Recall that GapSVP_γ is the (promise) problem of distinguishing, given a basis for a lattice and a parameter d, between the case where the lattice has a vector shorter than d, and the case where the lattice doesn't have any vector shorter than $\gamma \cdot d$. SIVP is the search problem of finding a set of "short" vectors. The best known algorithms for GapSVP_γ ([Sch87]) require at least $2^{\tilde{\Omega}(n/\log\gamma)}$ time. We refer the reader to [Reg05, Pei09] for more information.

There are known reductions between $\mathrm{DLWE}_{n,q,\chi}$ and those problems, which allows us to appropriately choose the LWE parameters for our scheme. We summarize in the following corollary (which addresses the regime of sub-exponential modulus-to-noise ratio).

Corollary 3.1 ([Reg05, Pei09, MM11, MP12, BLP+13])**.** *For any function $B = B(n) \geq \widetilde{O}(\sqrt{n})$ there exists a B-bounded distribution ensemble $\chi = \chi(n)$ over the integers s.t. for all $q = q(n)$, letting $\gamma = \widetilde{O}(\sqrt{n}q/B)$, it holds that $\mathrm{DLWE}_{n,q,\chi}$ is at least as hard as the quantum hardness of GapSVP_γ and SIVP_γ. Classical hardness GapSVP_γ follows if $q(n) \geq 2^{n/2}$ or for other values of q for $\tilde{\Omega}(\sqrt{n})$ dimensional lattices and approximation factor $q/B \cdot \mathrm{poly}(n\lceil\log q\rceil)$.*

3.4 Trapdoors and Discrete Gaussians

Let $n, q \in \mathbb{Z}$,

$$\mathbf{g} = (1, 2, 4, \ldots, 2^{\lceil\log q\rceil - 1}) \in \mathbb{Z}_q^{\lceil\log q\rceil}$$

and $m = n\lceil\log q\rceil$. The *gadget matrix* $\mathbf{G}$ is defined as the diagonal concatenation of $\mathbf{g}$ n times. Formally, $\mathbf{G} = \mathbf{g} \otimes \mathbf{I}_n \in \mathbb{Z}_q^{n\times m}$. For any $t \in \mathbb{Z}$, the function $\mathbf{G}^{-1} : \mathbb{Z}_q^{n\times t} \to \{0,1\}^{m\times t}$ expands each entry $a \in \mathbb{Z}_q$ of the input matrix into a column of size $\lceil\log q\rceil$ consisting of the bit-representation of a. For any matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times t}$, it holds that $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{A}) = \mathbf{A} \pmod q$.

The (centered) discrete Gaussian distribution over $\mathbb{Z}^m$ with parameter τ, denoted $D_{\mathbb{Z}^m,\tau}$, is the distribution over $\mathbb{Z}^m$ where for all $\mathbf{x}$, $\Pr[\mathbf{x}] \propto e^{-\pi\|\mathbf{x}\|^2/\tau^2}$.

Let $n, m, q \in \mathbb{N}$ and consider a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$. For all $\mathbf{v} \in \mathbb{Z}_q^n$ we let $\mathbf{A}_\tau^{-1}(\mathbf{v})$ denote the random variable whose distribution is the Discrete Gaussian $D_{\mathbb{Z}^m,\tau}$ conditioned on $\mathbf{A} \cdot \mathbf{A}_\tau^{-1}(\mathbf{v}) = \mathbf{v} \pmod q$. If $\mathbf{h} \xleftarrow{\$} \mathbf{A}_\tau^{-1}(\mathbf{v})$ then $\|\mathbf{h}\| \leq k\tau\sqrt{m}$ with probability at least $1 - e^{-\Omega(k^2)}$.

A τ-trapdoor for $\mathbf{A}$ is a procedure that can sample from a distribution within 2^{-n} statistical distance of $\mathbf{A}_\tau^{-1}(\mathbf{v})$ in time $\mathrm{poly}(n, m, \log q)$, for any $\mathbf{v} \in \mathbb{Z}_q^n$. We slightly overload notation and denote a τ-trapdoor for $\mathbf{A}$ by $\mathbf{A}_\tau^{-1}$. The following properties have been established in a long sequence of works.

Corollary 3.2 (Trapdoor Generation [Ajt96, MP12])**.** *There is a probabilistic polynomial-time algorithm $\mathsf{TrapGen}(1^n, q, m)$ that for all $m \geq m_0 = m_0(n, q) = O(n \log q)$, outputs $(\mathbf{A}, \mathbf{A}_{\tau_0}^{-1})$ s.t. $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ is within statistical distnace 2^{-n} from uniform and $\tau_0 = O(\sqrt{n \log q \log n})$.*

We use the most general form of trapdoor extension as formalized in [MP12].

Theorem 3.3 (Trapdoor Extension [ABB10b, MP12])**.** *Given* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$*, with a trapdoor* $\mathbf{A}_\tau^{-1}$*, and letting* $\mathbf{B} \in \mathbb{Z}_q^{n \times m'}$ *be s.t.* $\mathbf{A} = \mathbf{BS} \pmod q$ *where* $\mathbf{S} \in \mathbb{Z}^{m' \times m}$ *with largest singular value* $s_1(\mathbf{S}) \leq \sigma$*, then* $(\mathbf{A}_\tau^{-1}, \mathbf{S})$ *can be used to sample from* $\mathbf{B}_{\sigma\tau}^{-1}$*.*

Note that since only an upper bound on the singular value is required, this theorem implies that $\mathbf{A}_{\tau'}^{-1}$ is derived from $\mathbf{A}_\tau^{-1}$ whenever $\tau \leq \tau'$. A few additional important corollaries are derived from this theorem. We recall that $s_1(\mathbf{S}) \leq \sqrt{nm}\,\|\mathbf{S}\|_\infty$ and that a trapdoor $\mathbf{G}_{O(1)}^{-1}$ is trivial.

The first is a trapdoor extension that follows by taking $\mathbf{S} = [\mathbf{I} \parallel \mathbf{0}]$.

Corollary 3.4. *Given* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$*, with a trapdoor* $\mathbf{A}_\tau^{-1}$*, it is efficient to sample from* $[\mathbf{A}\|\mathbf{B}]_\tau^{-1}$ *for all* $\mathbf{B}$*.*

Next is a trapdoor extension that had been used extensively in prior work. It follows from Theorem 3.3 with $\mathbf{S} = [-\mathbf{R}^T \| \mathbf{I}]^T$.

Corollary 3.5. *Given* $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times m'}$*, and* $\mathbf{R} \in \mathbb{Z}^{m' \times m}$ *with* $m = n\lceil \log q \rceil$*, it is efficient to sample from* $[\bar{\mathbf{A}} \| \bar{\mathbf{A}}\mathbf{R} + \mathbf{G}]_\tau^{-1}$ *for* $\tau = O(\sqrt{mm'}\,\|\mathbf{R}\|_\infty)$*.*

Note that by taking $\bar{\mathbf{A}}$ uniform and $\mathbf{R}$ to be a high entropy small matrix, e.g. uniform in $\{-1, 0, 1\}$ and relying on the leftover hash lemma, Corollary 3.2 is in fact a special case of this one.

The following shows a different method for trapdoor extension which corresponds to matrices in Hermite Normal Form. This trapdoor generation method is mentioned in passing in [MP12] as a method for improving parameters by relying on computational assumptions. Our use of this property is quite different. Technically it follows from Theorem 3.3 with $\mathbf{S} = [-\mathbf{E}^T \| -\mathbf{R}^T \| \mathbf{I}]^T$.

Corollary 3.6 (Trapdoor Extension in HNF)**.** *Let* $n, q, m' \geq 1$ *and let* $m = n\lceil \log q \rceil$*. Given* $\bar{\mathbf{A}} \xleftarrow{\$} \mathbb{Z}_q^{n \times m'}$*,* $\mathbf{R} \in \mathbb{Z}^{m' \times m}$ *and* $\mathbf{E} \in \mathbb{Z}^{n \times m}$*, the trapdoor* $[\mathbf{I} \| \bar{\mathbf{A}} \| \bar{\mathbf{A}}\mathbf{R} + \mathbf{G} + \mathbf{E}]_\tau^{-1}$ *is efficiently computable for* $\tau = O(\sqrt{mm'}\,\|\mathbf{R}\|_\infty + \sqrt{mn}\,\|\mathbf{E}\|_\infty)$*.*

3.5 Lattice Evolution

The following is an abstraction of the evaluation procedure in recent LWE based FHE and ABE schemes that developed in a long sequence of works [ABB10b, MP12, GSW13, AP14, BGG+14, GVW15b]. We use a similar formalism as in [BV15b, BCTW16] but slightly rename the functions.

Theorem 3.7. *There exist efficient deterministic algorithms* EvalF *and* EvalFX *such that for all* $n, q, \ell \in \mathbb{N}$*, and for any sequence of matrices* $(\mathbf{A}_1, \ldots, \mathbf{A}_\ell) \in (\mathbb{Z}_q^{n \times n\lceil \log q \rceil})^\ell$*, for any depth-d Boolean circuit* $f : \{0,1\}^\ell \to \{0,1\}$ *and for every* $\mathbf{x} = (x_1, \ldots, x_\ell) \in \{0,1\}^\ell$*, the following properties hold.*

- *The outputs* $\mathbf{H}_f = \mathsf{EvalF}(f, \mathbf{A}_1, \ldots, \mathbf{A}_\ell)$ *and* $\mathbf{H}_{f,x} = \mathsf{EvalFX}(f, x, \mathbf{A}_1, \ldots, \mathbf{A}_\ell)$ *are both matrices in* $\mathbb{Z}^{(\ell n \lceil \log q \rceil) \times n \lceil \log q \rceil}$*;*
- *It holds that* $\|\mathbf{H}_f\|_\infty, \|\mathbf{H}_{f,x}\|_\infty \leq (n \log q)^{O(d)}$*.*
- *It holds that*

$$[\mathbf{A}_1 - x_1\mathbf{G} \| \mathbf{A}_2 - x_2\mathbf{G} \| \ldots \| \mathbf{A}_\ell - x_\ell\mathbf{G}] \cdot \mathbf{H}_{f,\mathbf{x}} = [\mathbf{A}_1 \| \mathbf{A}_2 \| \ldots \| \mathbf{A}_\ell] \cdot \mathbf{H}_f - f(\mathbf{x})\mathbf{G} \pmod q \quad (7)$$

We will call this the "key equation" for matrix evolution.

For a proof of this theorem, we refer the reader to [BV15b]. This evolution method was extended by [AFV11,GVW15a] to show that in the case of the inner product function it is possible to compute EvalFX with only one of the two operands.

Theorem 3.8. *There exist efficient deterministic algorithms* EvalF^{ip} *and* EvalFX^{ip} *as follows. Let* $n, q, \ell, \vec{\mathbf{A}} = (\mathbf{A}_1, \ldots, \mathbf{A}_\ell), \mathbf{x}$ *be as above. Let* $\ell' \in \mathbb{N}$ *and* $\vec{\mathbf{B}} = (\mathbf{B}_1, \ldots, \mathbf{B}_{\ell'}) \in (\mathbb{Z}_q^{n \times n\lceil \log q \rceil})^{\ell'}$*, and let* $f : \{0,1\}^\ell \to \{0,1\}^{\ell'}$ *be a depth* d *boolean circuit with* ℓ' *output bits. Then:*

- $\mathbf{H}_f = \mathsf{EvalF}^{ip}(f, \vec{\mathbf{A}}, \vec{\mathbf{B}})$ *and* $\mathbf{H}_{f,x} = \mathsf{EvalFX}^{ip}(f, \mathbf{x}, \vec{\mathbf{A}}, \vec{\mathbf{B}})$ *are both in* $\mathbb{Z}^{((\ell+\ell')n\lceil \log q \rceil) \times n\lceil \log q \rceil}$*;*
- *It holds that* $\|\mathbf{H}_f\|_\infty, \|\mathbf{H}_{f,x}\|_\infty \leq \ell'(n \log q)^{O(d)}$*;*
- *It holds that for all* $\mathbf{y} \in \mathbb{Z}^{\ell'}$

$$\left([\vec{\mathbf{A}} \| \vec{\mathbf{B}}] - [\mathbf{x} \| \mathbf{y}] \otimes \mathbf{G}\right) \cdot \mathbf{H}_{f,\mathbf{x}} = [\vec{\mathbf{A}} \| \vec{\mathbf{B}}] \cdot \mathbf{H}_f - \langle f(\mathbf{x}), \mathbf{y} \rangle \mathbf{G} \pmod q \ , \tag{8}$$

  where the inner product is over the integers (or equivalently modulo q*).*

We note that EvalFX^{ip} does not take $\mathbf{y}$ as input and furthermore that $\mathbf{y}$ can have arbitrary integer values (not necessarily binary). We will later extend these theorems to functions that output matrices in Section .

3.6 Fully Homomorphic Encryption (FHE)

A (secret-key) homomorphic encryption (HE) scheme w.r.t a function class $\mathcal{F}$ is a semantically secure encryption scheme adjoined with an additional p.p.t. algorithm Eval s.t. for all $f \in \mathcal{F}$ and $\mathbf{x} \in \{0,1\}^\ell$ it holds that if sk is properly generated and $\mathsf{ct}_i = \mathsf{Enc}_{\mathsf{sk}}(x_i)$, then $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Eval}(f, \mathsf{ct}_1, \ldots, \mathsf{ct}_\ell)) = f(\mathbf{x})$ with all but negligible probability. The following is a corollary of the [GSW13] encryption scheme. We note that the common use of the scheme is with $t = q/2$ but we will use $t \approx \sqrt{q}$ in this work.

Lemma 3.9 (Leveled FHE [GSW13])**.** *Let* $q, n, t, d \geq 1$ *and let* χ *be* B*-bounded. If* $q > 2t \geq 4B(n\lceil \log q \rceil)^{O(d)}$ *then there exists an FHE scheme for the class* $\mathcal{F}_d$ *of depth* d *circuits based on* $\mathrm{DLWE}_{n,q,\chi}$ *with the following properties.*

- *The ciphertext length is* $\ell_c = \mathrm{poly}(n\lceil \log q \rceil)$*.*
- *Decryption involves* (i) *preprocessing the ciphertext (independently of the secret key) into a* binary *vector* $\mathbf{c} \in \{0,1\}^{\ell_s}$ *for* $\ell_s = \mathrm{poly}(n\lceil \log q \rceil)$*;* (ii) *taking inner product* $\langle \mathbf{c}, \mathbf{s} \rangle \pmod q$ *for an integer secret-key vector* $\mathbf{s}$*, which results in* $t\mu + \delta$ *with* $|\delta| \leq B(n\lceil \log q \rceil)^{O(d)}$*;* (iii) *extracting the output* μ *from the above expression.*

  Moreover, for any $f \in \mathcal{F}_d$*, the depth of* $f'(\cdot) = \mathrm{FHE.Eval}(f, \cdot)$ *is at most* $d' = d \cdot \mathrm{polylog}(n\lceil \log q \rceil)$*.*

3.7 The Banerjee-Peikert Pseudorandom Function

Banerjee and Peikert [BP14] introduced an LWE-based key homomorphic pseudorandom function which was the basis for the [BV15b] constrained PRF. While [BV15b] only drew from the ideas in [BP14], we use their construction explicitly as a building block, which simplifies our analysis. We present their construction using the our instance evolution terminology.

For all $x \in \{0,1\}^\ell$, consider the circuit (more precisely, arithmetic formula) $\mathcal{T}_x(y_0, y_1)$ which computes the product $\prod_{i\in[\ell]} y_{x_i}$ using a balanced binary multiplication tree. Note that we are never actually computing $\mathcal{T}_x$ on any input. We are only using its formal combinatorial structure for the purpose of evolution as described next.

Corollary 3.10 (follows from [BP14, Theorems 3.7, 3.8])**.** *Let $n, p, \ell \geq 1$ be integers, let χ be B-bounded and assume* $\mathrm{DLWE}_{n,p,\chi}$*. Then there exists an efficiently computable randomized function $E : \{0,1\}^\ell \to \mathbb{Z}^{n\lceil \log p\rceil}$ with bounded norm $\|E\|_\infty \leq B\sqrt{\ell}\cdot(n\lceil\log p\rceil)^{\log \ell}$, such that, letting $\mathbf{C}_0, \mathbf{C}_1 \xleftarrow{\$} \mathbb{Z}_p^{n\times n\lceil\log p\rceil}$ and denoting $\vec{\mathbf{C}} = (\mathbf{C}_0, \mathbf{C}_1)$, $\mathbf{C}_x = \mathsf{EvalF}(\mathcal{T}_x, \vec{\mathbf{C}})$ for all x.*

$$F_{\mathbf{s}}(x) = \mathbf{s}\mathbf{C}_x + E(x) \pmod p$$

is pseudorandom, where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^n$. Furthermore, the same holds for

$$F'_{\mathbf{d}}(x) = \mathbf{d}\mathbf{G}_p^{-1}(\mathbf{C}_x) + E(x) \pmod p$$

where $\mathbf{d} \xleftarrow{\$} \mathbb{Z}_p^{n\lceil\log p\rceil}$ and $\mathbf{C}_x, E$ as above.

4 Our First Construction: The Dual-Use Technique

In this section, we present the *dual-use technique* and construct a new weakly attribute-hiding PE scheme and a constraint-hiding constrained PRF based on LWE. We will use the machinery for lattice evolution developed in Section 3.5. First, in Section 4.1, we extend this machinery to work for computations that output not just scalars but matrices. Then, in sections 4.2 and 4.3, we describe our weakly attribute-hiding PE scheme and a constraint-hiding constrained PRF scheme, respectively.

4.1 Lattice Evolution of Matrix-Valued Functions

We first extend evolution of matrices from Section 3.5 to deal with functions whose output is a matrix instead of a bit (we still treat the input as bits).

Notation. Given a matrix $\mathbf{X} \in \mathbb{Z}_q^{n\times n\log q}$, we will index its $n^2\log q$ entries by numbers, for convenience of notation (as opposed to the standard practice of using a pair of numbers to index the row and column separately). We use $x_{j,\tau} \in \{0,1\}$ where $j \in [n^2\log q], \tau \in [\log q]$ to denote the τ'th bit of the j'th entry of $\mathbf{X}$. This means that we can write

$$\mathbf{X} = \sum_{j,\tau} x_{j,\tau}\cdot 2^\tau \mathbf{E}_j$$

where $\mathbf{E}_j$ is a $0,1$-matrix whose j'th entry is 1 and 0 everywhere else. Throughout, we use $j \in [n^2\log q], \tau \in [\log q]$ and $i \in [\ell]$ and we avoid explicitly quantifying over these variables.

Matrix computation. Suppose $f : x_1, \ldots, x_\ell \mapsto \mathbf{X}_f$ where these matrices have the same dimensions as $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_\ell$. Then, we require the following key relation between $\mathbf{H}_f$ and $\mathbf{H}_{f,\mathbf{x}}$:

$$\left[\mathbf{A}_1 - x_1\mathbf{G}\middle|\ \cdots\ \middle|\mathbf{A}_\ell - x_\ell\mathbf{G}\right]\cdot\mathbf{H}_{f,\mathbf{x}} = \left[\mathbf{A}_1\middle|\ \cdots\ \middle|\mathbf{A}_\ell\right]\cdot\mathbf{H}_f - \mathbf{X}_f \tag{9}$$

Constructing $\mathbf{H}_{f,\mathbf{x}}$ and $\mathbf{H}_f$. Let $f_{j,\tau} : x_1, \ldots, x_\ell \mapsto \{0,1\}$ denote the function that outputs τ'th bit of the j'th entry of $\mathbf{X}_f$. Then, we define $\mathbf{H}_f$ as follows.

$$\mathbf{H}_{f,j,\tau} := \mathsf{EvalF}(f_{j,\tau}, \{\mathbf{A}_i\}), \qquad \mathbf{H}_f := \sum_{j,\tau} \mathbf{H}_{f,j,\tau} \cdot \mathbf{G}^{-1}(2^\tau \mathbf{E}_j)$$

Then, the key relation (Equation 9) follows readily from the following relations:

$$\begin{aligned} \left[\mathbf{A}_1 - x_1\mathbf{G}\middle| \cdots \middle| \mathbf{A}_\ell - x_\ell\mathbf{G}\right] \cdot \mathbf{H}_{f,j,\tau,\mathbf{x}} &= \left[\mathbf{A}_1\middle| \cdots \middle| \mathbf{A}_\ell\right] \cdot \mathbf{H}_{f,j,\tau} - x_{f,j,\tau}\mathbf{G} \\ \text{and} \quad \sum_{j,\tau} x_{f,j,\tau}\mathbf{G} \cdot \mathbf{G}^{-1}(2^\tau \mathbf{E}_j) &= \mathbf{X}_f \end{aligned}$$

where the first equation is the key relation for functions with scalar output. These two relations together show us that the setting of

$$\mathbf{H}_f := \sum_{j,\tau} \mathbf{H}_{f,j,\tau}\mathbf{G}^{-1}(2^\tau \mathbf{E}_j), \quad \mathbf{H}_{f,\mathbf{x}} := \sum_{j,\tau} \mathbf{H}_{f,j,\tau,\mathbf{x}}\mathbf{G}^{-1}(2^\tau \mathbf{E}_j)$$

satisfies equation 9.

4.2 Weakly Attribute-Hiding Predicate Encryption

In this section, we describe the dual use technique and use it to construct a weakly attribute-hiding predicate encryption scheme.

Notation. We use gadget matrices $\mathbf{G} \in \mathbb{Z}_q^{(n+1)\times(n+1)\log q}$ and we write $\overline{\mathbf{G}} \in \mathbb{Z}_q^{n\times(n+1)\log q}$ to denote all but the last row of $\mathbf{G}$. Given a circuit computing a function $f : \{0,1\}^\ell \to \{0,1\}$, and GSW FHE encryptions $\Psi := (\Psi_1, \ldots, \Psi_\ell)$ of $x_1, \ldots, x_\ell$, we write Ψ_f to denote $\mathsf{fhe.eval}(f, \Psi)$. Noting that Ψ_f is a matrix, we let $\underline{\Psi}_f$ denote the last row of Ψ_f, and $\overline{\Psi}_f$ to denote all but the last row of Ψ_f. In addition, we write $\hat{f}$ to denote the circuit that computes $\Psi \mapsto \overline{\Psi}_f$, namely it takes as input the bits of Ψ and outputs the matrix $\overline{\Psi}_f$.

We let $\mathbf{e} \xleftarrow{\sigma} \mathbb{Z}^m$ denote the process of sampling a vector $\mathbf{e}$ where each of its entries is drawn independently from the discrete Gaussian with mean 0 and standard deviation σ over $\mathbb{Z}$.

Our predicate encryption scheme works as follows.

- $\mathsf{Setup}(1^\lambda, 1^\ell, 1^d)$: sample $(\mathbf{B}, T_\mathbf{B})$ where $\mathbf{B} \in \mathbb{Z}_q^{n\times(n+1)\log q}$ and $T_\mathbf{B}$ denotes the trapdoor for $\mathbf{B}$. Pick $\mathbf{B}_j \xleftarrow{\$} \mathbb{Z}_q^{n\times(n+1)\log q}$ and $\mathbf{p} \xleftarrow{\$} \mathbb{Z}_q^n$. Output

$$\begin{aligned} \mathsf{mpk} &:= \Big(\mathbf{B}, \{\mathbf{B}_j\}_{j\in[L]}, \mathbf{p} \Big), \\ \mathsf{msk} &:= \Big(\mathbf{T}_\mathbf{B} \Big) \end{aligned}$$

  where $L = \ell \cdot (n+1)^2 \log^2 q$.

- $\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}, M \in \{0,1\})$: pick $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e}, \mathbf{e}_0, \mathbf{e}_j \xleftarrow{\sigma} \mathbb{Z}^m, e' \xleftarrow{\sigma} \mathbb{Z}, \mathbf{R}_i \in \{0,1\}^{(n+1)\log q \times (n+1)\log q}$ and compute

$$\Psi_i := \begin{pmatrix} \mathbf{B} \\ \mathbf{s}^T\mathbf{B} + \mathbf{e}^T \end{pmatrix} \mathbf{R}_i + x_i \mathbf{G}$$

Let $\psi_1, \ldots, \psi_L$ denote the binary representation of $\Psi := [\Psi_1 \mid \cdots \mid \Psi_\ell]$. Compute

$$\mathbf{c}_0^T := \mathbf{s}^T\mathbf{B} + \mathbf{e}_0^T, \qquad \mathbf{c}_j^T := \mathbf{s}^T[\mathbf{B}_j - \psi_j \overline{\mathbf{G}}] + \mathbf{e}_j^T$$

and $\kappa := \mathbf{s}^T\mathbf{p} + e' + M \cdot \lfloor q/2 \rfloor \pmod q$.
The PE ciphertext consists of the FHE ciphertext Ψ and the ABE ciphertexts computed as above. That is,

$$\mathsf{ct} := \left(\Psi, \mathbf{c}_0, \{\mathbf{c}_j\}_{j\in[L]}, \kappa \right)$$

- $\mathsf{KeyGen}(\mathsf{msk}, f)$: Let $\hat{f}$ denote the circuit computing $\Psi \mapsto \overline{\Psi}_f$ and

$$\mathbf{H}_{\hat{f}} := \mathsf{EvalF}(\hat{f}, \{\mathbf{B}_j\}_{j\in[L]}),\ \mathbf{B}_{\hat{f}} := [\mathbf{B}_1 \mid \cdots \mid \mathbf{B}_L] \cdot \mathbf{H}_{\hat{f}}$$

Sample a short sk_f using $\mathbf{T}_\mathbf{B}$ such that

$$[\mathbf{B} \mid \mathbf{B}_{\hat{f}}] \cdot \mathsf{sk}_f = \mathbf{p}$$

Output sk_f.

- $\mathsf{Dec}((\mathsf{sk}_f, f), \mathsf{ct})$: Let $\hat{f}$ denote the circuit computing $\Psi \mapsto \overline{\Psi}_f$ and parse the ciphertext ct as $(\Psi, \mathbf{c}_0, \{\mathbf{c}_j\}_{j\in L}, \kappa)$. Compute:

$$\begin{aligned} \Psi_f &:= \hat{f}(\Psi) \\ \mathbf{H}_{\hat{f},\Psi} &:= \mathsf{EvalFX}(\hat{f}, \Psi, \{\mathbf{B}_j\}_{j\in[L]}) \\ \mathbf{c}_{\hat{f}} &:= [\mathbf{c}_1 \mid \cdots \mid \mathbf{c}_L] \cdot \mathbf{H}_{\hat{f},\Psi} + \underline{\Psi}_f \end{aligned}$$

Compute

$$\kappa' := [\mathbf{c}_0 \mid \mathbf{c}_{\hat{f}}] \cdot \mathsf{sk}_f$$

and output the MSB of $\kappa - \kappa'$.

We now analyze the correctness of the PE scheme (in the process setting the parameters) and prove its (selective) security under the polynomial hardness of LWE with a sub-exponential modulus-to-noise ratio.

Theorem 4.1 (Correctness)**.** *The* PE *construction above is correct as per Definition 3.2.*

Proof. The key relation tells us that

$$[\mathbf{B}_1 - \psi_1\overline{\mathbf{G}} \mid \cdots \mid \mathbf{B}_L - \psi_L\overline{\mathbf{G}}] \cdot \mathbf{H}_{\hat{f},\Psi} = [\mathbf{B}_1 \mid \cdots \mid \mathbf{B}_L] \cdot \mathbf{H}_{\hat{f}} - \overline{\Psi}_f = \mathbf{B}_{\hat{f}} - \overline{\Psi}_f$$

Multiplying both sides by $\mathbf{s}^T$, we have

$$\begin{aligned} \mathbf{c}_{\hat{f}} &\approx \mathbf{s}^T[\mathbf{B}_1 - \psi_1\overline{\mathbf{G}} \mid \cdots \mid \mathbf{B}_L - \psi_L\overline{\mathbf{G}}] \cdot \mathbf{H}_{\hat{f},\Psi} + \underline{\Psi}_f \\ &= \mathbf{s}^T\mathbf{B}_{\hat{f}} - \mathbf{s}^T\overline{\Psi}_f + \underline{\Psi}_f \\ &= \mathbf{s}^T\mathbf{B}_{\hat{f}} - [\mathbf{s}^T \mid -1] \cdot \Psi_f \\ &\approx \mathbf{s}^T\mathbf{B}_{\hat{f}} - f(\mathbf{x}) \cdot [\mathbf{s}^T \mid -1] \cdot \mathbf{G} \end{aligned}$$

where the first approximate equality is because of the accumulated error which is a product of the LWE errors and the low-norm matrix $\mathbf{H}_{\hat{f},\Psi}$, the second equality is because of the key relation, and the final approximate equality is because of the decryption equation of the GSW FHE scheme. Then, when $f(\mathbf{x}) = 0$,

$$\kappa' := [\mathbf{c}_0 \mid \mathbf{c}_{\hat{f}}] \cdot \mathsf{sk}_f \approx \mathbf{s}^T [\mathbf{B} \mid \mathbf{B}_{\hat{f}}] \cdot \mathsf{sk}_f = \mathbf{s}^T \mathbf{p}$$

Now, decryption succeeds in recovering M since $\kappa := \mathbf{s}^T \mathbf{p} + e' + M \cdot \lfloor q/2 \rfloor \pmod q$.

Setting Parameters. The error growth on FHE evaluation is by a multiplicative factor of $(n \log q)^{O(d_f)}$ where d_f is the depth of the circuit computing f. Furthermore, the error growth on ABE evaluation has magnitude at most $(n \log q)^{O(d_{\hat{f}})}$ where $d_{\hat{f}}$ is the depth of the circuit that performs GSW FHE evaluation for the function f. We know that $d_{\hat{f}} = d \cdot \mathsf{poly}(\log n, \log \log q)$. The total error growth thus has magnitude $(n \log q)^{d \cdot \mathsf{poly}(\log n, \log \log q)}$ which should be at most $q/4$ for correctness.

On the other hand, we would like to set $q = O(2^{n^\epsilon})$ for some constant ϵ so as to rely on the hardness of sub-exponential-error LWE. It is possible to find a setting of parameters that satisfy all these conditions, analogous to Section 5.1. □

Theorem 4.2 (Security)**.** *The scheme* PE *is secure as per Definition 3.3 under the* $\mathrm{LWE}_{n,q,\chi}$ *assumption, and thus under the worst case hardness of approximating* GapSVP, SIVP *to within a* $2^{\widetilde{O}(n^\epsilon)}$ *factor in polynomial time.*

Proof. We provide a proof sketch for selective security of the PE scheme.

First, we describe a set of auxiliary algorithms consisting of alternative algorithms $(\mathsf{Setup}^*, \mathsf{KeyGen}^*, \mathsf{Enc}^*)$ that will be used in the proof of security. We are given $\mathbf{A} = \binom{\mathbf{B}}{\mathbf{c}}, \mathbf{p}, p'$ and the selective challenge $\mathbf{x}^*$. Here, $(\mathbf{c}, p')$ is either $(\mathbf{s}^T\mathbf{B} + \mathbf{e}, \mathbf{s}^T\mathbf{p} + e')$ or uniformly random.

$\mathsf{Setup}^*(\mathbf{B}, \mathbf{p}, \mathbf{x}^*)$: pick $\mathbf{W}'_j \xleftarrow{\$} \{0,1\}^{n \times (n+1)\log q}, \mathbf{R}_i \in \{0,1\}^{(n+1)\log q \times (n+1)\log q}$. Compute

$$\begin{aligned} \Psi_i &:= \mathbf{A}\mathbf{R}_i + x^*_i \mathbf{G} \\ \mathbf{B}_j &= \mathbf{B}\mathbf{W}'_j + \psi_j \overline{\mathbf{G}} \end{aligned}$$

where, as before, ψ_j denote the bits of $\Psi = [\Psi_1 \mid \cdots \mid \Psi_\ell]$. Output

$$\begin{aligned} \mathsf{mpk} &:= \Big(\mathbf{B}, \{\mathbf{B}_j\}_{j \in [L]}, \mathbf{p} \Big), \\ \mathsf{msk}^* &:= \Big(\{\mathbf{W}'_j\}_{j \in [L]} \Big) \end{aligned}$$

$\mathsf{Enc}^*(\mathbf{B}, \mathbf{p}, \mathbf{x}^*)$: Compute

$$\mathbf{c}_0^T := \mathbf{c}^T, \qquad \mathbf{c}_j^T := \mathbf{c}^T \mathbf{W}'_j$$

Output

$$\mathsf{ct} := \big(\Psi, \mathbf{c}_0, \{\mathbf{c}_j\}_{j \in [L]}, p' + M \cdot q/2 \big)$$

$\mathsf{KeyGen}^*(\mathsf{msk}^*, f)$: On input f such that $f(\mathbf{x}^*) \neq 0$,

$$\begin{aligned} \mathbf{B}_{\hat{f}} &= [\mathbf{B}\mathbf{W}'_1 + \psi_1 \overline{\mathbf{G}} \mid \cdots \mid \mathbf{B}\mathbf{W}'_L + \psi_L \overline{\mathbf{G}}] \cdot \mathbf{H}_{\hat{f}} \\ &= [\mathbf{B}\mathbf{W}'_1 \mid \cdots \mid \mathbf{B}\mathbf{W}'_L] \cdot \mathbf{H}_{\hat{f},\Psi} + \overline{\Psi}_f \\ &= \mathbf{B}(\mathbf{W}'_{\hat{f}} + \mathbf{R}_f) + f(\mathbf{x}^*)\overline{\mathbf{G}} \end{aligned}$$

where

$$\mathbf{W}'_{\hat{f}} := [\mathbf{W}'_1 \mid \cdots \mid \mathbf{W}'_L] \cdot \mathbf{H}_{\hat{f},\Psi}, \quad \Psi_f = \mathbf{A}\mathbf{R}_f + f(\mathbf{x}^*)\mathbf{G}$$

We can then sample a short sk_f using $\mathbf{W}'_{\hat{f}} + \mathbf{R}_f$ such that

$$[\mathbf{B} \mid \mathbf{B}_{\hat{f}}] \cdot \mathsf{sk}_f = \mathbf{p}$$

Output sk_f.

We now proceed to describe a sketch of the proof of security through a sequence of games, using the auxiliary algorithms described above.

Game 0. Real world.

Game 1. Switch to $\mathsf{Setup}^*, \mathsf{Enc}^*$ that are given $\mathbf{A} = \binom{\mathbf{B}}{\mathbf{c}}$ and use $\mathbf{W}'_j$. When $\mathbf{c}$ is the LWE vector relative to $\mathbf{B}$, game 0 and game 1 are statistically close by an application of the leftover hash lemma. (In this proof sketch, we ignore the issue of smoothing the errors in the ciphertext, which can be done by noise flooding). Note that in this game, the challenger does not know the LWE secret $\mathbf{s}$.

Game 2. Switch to KeyGen^* that uses $(\mathbf{W}'_j, \mathbf{R}_i)$ instead of $\mathbf{T}_\mathbf{B}$. The difference between game 1 and game 2 is that in the former, secret keys are generated using $\mathbf{T}_\mathbf{B}$ whereas in the latter, they are generated using $\mathbf{W}'_{\hat{f}} + \mathbf{R}_f$, by employing the ABB trick [ABB10a]. Thus, games 1 and 2 are statistically indistinguishable.

Game 3. Switch $\mathbf{c}$ in $\mathbf{A}$ from $\mathbf{s}^T\mathbf{B} + \mathbf{e}$ to a random $\mathbf{c}$ (this changes both $\mathsf{abe.ct}$ and Ψ). Games 2 and 3 are computationally indistinguishable by the LWE assumption.

Game 4. switch from KeyGen^* back to KeyGen. Games 3 and 4 are statistically indistinguishable by the same argument as Games 1 versus 2.

Now, in game 4, we argue that $x_1^*, \ldots, x_n^*$ is information-theoretically hidden, as follows:

- First, note that the distribution of the NO keys only depends on $[\mathbf{B} \mid \mathbf{B}_{\hat{f}}]$, that is, on $(\mathsf{mpk}, f, \mathbf{T}_\mathbf{B})$, and leak no information about the FHE encryption randomness $\mathbf{R}_1, \ldots, \mathbf{R}_n$.
- Secondly, mpk and the ciphertext depend on the ψ_i's and the $\mathbf{W}'_j$'s, but not on the FHE encryption randomness $\mathbf{R}_1, \ldots, \mathbf{R}_n$.
- Using these two observations, we argue that ψ_i hides x_i^*. Indeed, by left-over hash lemma, we know that $\mathbf{A}\mathbf{R}_i$ is statistically close to uniform given $\mathbf{A} = \binom{\mathbf{B}}{\mathbf{c}}$, and therefore completely hides x_i^*.

□

Remark: Relation to the GVW15 Security Proof. Many of the steps in the proof are analogous to what happens in GVW15. The crucial difference is that in GVW15, the leftover hash lemma (LHL) was used to hide the FHE secret key which is embedded as part of the ABE attributes. Using the fact that NO keys do not leak any information about *the randomness* $\mathbf{W}_j$ *used to simulate ABE ciphertext*, one can apply LHL to this randomness and therefore, hide the FHE secret key, and consequently, hiding the attributes. In our scheme, LHL is applied to *the randomness* $\mathbf{R}_j$ *used for FHE encryption*, and not on the randomness $\mathbf{W}'_j$ used to simulate the ABE ciphertext.

4.3 Constraint Hiding Constrained PRF

We now present a Constraint Hiding CPRF construction that relies on the [BV15b] CPRF together with the dual use technique from Section 4.2.

Our constraint hiding CPRF scheme works as follows.

- $\mathsf{CPRF.Keygen}(1^\lambda, 1^\ell, 1^{\ell_x}, 1^d)$ takes as input the security parameter λ, the maximum description length ℓ of constraint functions, their input length ℓ_x and depth d, and outputs public parameters pp and a secret key σ for the CPRF scheme. Let $L = \ell \cdot (n+1)^2 \log^2 q$.

  Sample $\mathbf{B}, \mathbf{B}_1, \ldots, \mathbf{B}_L \xleftarrow{\$} \mathbb{Z}_q^{n \times (n+1)\log q}$ and $\mathbf{D}, \mathbf{C}_1, \ldots, \mathbf{C}_{\ell_x} \in \mathbb{Z}_q^{n \times m}$ for some $m = \Omega(n \log q)$. Sample a uniformly random vector $\mathbf{s} \in \mathbb{Z}_q^n$. Output

$$\mathsf{pp} := \Big(\mathbf{B}, \{\mathbf{B}_j\}_{j\in[L]}, \{\mathbf{C}_j\}_{j\in[\ell_x]}, \mathbf{D} \Big),$$
$$\sigma := \mathbf{s}$$

- $\mathsf{CPRF.Eval}_{\mathsf{pp}}(\sigma, x)$ outputs the evaluation of the PRF on an input x.

  Let $\mathcal{U}_x : \{0,1\}^\ell \to \{0,1\}$ be the circuit that takes as input a description of a function f and outputs $f(x)$. Now consider the circuit $\widehat{\mathcal{U}}_x : \{0,1\}^L \to \mathbb{Z}_q^{n \times (n+1)\log q}$ that takes as input a GSW encryption $\hat{f}$ of the description of f and outputs $\overline{\Psi}_{\mathbf{x}}$ where $\Psi_{\mathbf{x}} = \mathrm{FHE.Eval}(\mathcal{U}_x, \hat{f})$.

  Let $\widehat{\mathcal{U}}_x$ denote the circuit computing $\Psi \mapsto \overline{\Psi}_{\mathbf{x}}$ and

$$\mathbf{H}_{\widehat{\mathcal{U}}_x} := \mathsf{EvalF}(\widehat{\mathcal{U}}_x, \{\mathbf{B}_j\}_{j\in[L]}),\ \mathbf{B}_{\widehat{\mathcal{U}}_x} := [\mathbf{B}_1 \mid \cdots \mid \mathbf{B}_L] \cdot \mathbf{H}_{\widehat{\mathcal{U}}_x}$$

  Compute $\mathbf{C}_x = \mathsf{EvalF}(\mathcal{T}_x, \mathbf{C}_1, \ldots, \mathbf{C}_{\ell_x})$ (as defined in Section 3.7) and fix $\mathbf{M}_x = \mathbf{D}\mathbf{G}^{-1}(\mathbf{C}_x)$. The PRF output is

$$y = \left\lfloor \mathbf{s}^T \cdot \mathbf{B}_{\widehat{\mathcal{U}}_x} \mathbf{G}^{-1}(\mathbf{M}_x) \right\rceil .$$

- $\mathsf{CPRF.Constrain}_{\mathsf{pp}}(\sigma, f)$ outputs a constrained key σ_f.

  Pick $\mathbf{e}, \mathbf{e}_0, \mathbf{e}_j \xleftarrow{\sigma} \mathbb{Z}^m, \mathbf{R}_i \in \{0,1\}^{(n+1)\log q \times (n+1)\log q}$ and compute GSW ciphertexts

$$\Psi_i := \begin{pmatrix} \mathbf{B} \\ \mathbf{s}^T\mathbf{B} + \mathbf{e}^T \end{pmatrix} \mathbf{R}_i + f_i \mathbf{G}$$

  where $(f_1, \ldots, f_\ell)$ is the description of the function f.

Let $\psi_1, \ldots, \psi_L$ denote the binary representation of $\Psi := [\Psi_1 \mid \cdots \mid \Psi_\ell]$. Compute

$$\mathbf{c}_0^T := \mathbf{s}^T\mathbf{B} + \mathbf{e}_0^T, \qquad \mathbf{c}_j^T := \mathbf{s}^T[\mathbf{B}_j - \psi_j\overline{\mathbf{G}}] + \mathbf{e}_j^T$$

The constrained key consists of the FHE ciphertext Ψ and the "ABE ciphertexts" computed as above. That is,

$$\mathsf{ct} := \left(\Psi, \mathbf{c}_0, \{\mathbf{c}_j\}_{j\in[L]}\right)$$

- $\mathsf{CPRF.ConstrainEval}_{\mathsf{pp}}(\sigma_f, x)$ takes as input a constrained key σ_f and an input x and outputs a (potential) PRF output.

  Let $\hat{f}$ denote the circuit computing $\Psi \mapsto \overline{\Psi}_{\mathbf{x}}$ (as above) and parse the constrained key ct as $(\Psi, \mathbf{c}_0, \{\mathbf{c}_j\}_{j\in L})$. Compute:

$$\begin{aligned}
\overline{\Psi}_{\mathbf{x}} &:= \widehat{\mathcal{U}}_x(\Psi) \\
\mathbf{H}_{\widehat{\mathcal{U}}_x,\Psi} &:= \mathsf{EvalFX}(\widehat{\mathcal{U}}_x, \Psi, \{\mathbf{B}_j\}_{j\in[L]}) \\
\mathbf{c}_{\widehat{\mathcal{U}}_x} &:= [\mathbf{c}_1 \mid \cdots \mid \mathbf{c}_L] \cdot \mathbf{H}_{\widehat{\mathcal{U}}_x,\Psi} + \underline{\Psi}_{\mathbf{x}}
\end{aligned}$$

  Output

$$y' = \left\lfloor \mathbf{c}_{\widehat{\mathcal{U}}_x} \mathbf{G}^{-1}(\mathbf{M}_x) \right\rceil$$

Theorem 4.3 (Correctness, Pseudorandomness, Constraint Hiding)**.** *Under the* $\mathrm{DLWE}_{n,q,\chi}$ *hardness assumption,* CPRF *is correct, pseudorandom and constraint hiding.*

Proof. Correctness follows from a computation similar to the one in Section 4.2. In particular, the key relation tells us that

$$[\mathbf{B}_1 - \psi_1\overline{\mathbf{G}} \mid \cdots \mid \mathbf{B}_L - \psi_L\overline{\mathbf{G}}] \cdot \mathbf{H}_{\widehat{\mathcal{U}}_x,\Psi} = [\mathbf{B}_1 \mid \cdots \mid \mathbf{B}_L] \cdot \mathbf{H}_{\widehat{\mathcal{U}}_x} - \overline{\Psi}_{\mathbf{x}} = \mathbf{B}_{\widehat{\mathcal{U}}_x} - \overline{\Psi}_{\mathbf{x}}$$

Multiplying both sides by $\mathbf{s}^T$, we have

$$\begin{aligned}
\mathbf{c}_{\widehat{\mathcal{U}}_x} &\approx \mathbf{s}^T[\mathbf{B}_1 - \psi_1\overline{\mathbf{G}} \mid \cdots \mid \mathbf{B}_L - \psi_L\overline{\mathbf{G}}] \cdot \mathbf{H}_{\widehat{\mathcal{U}}_x,\Psi} + \underline{\Psi}_{\mathbf{x}} \\
&= \mathbf{s}^T\mathbf{B}_{\widehat{\mathcal{U}}_x} - \mathbf{s}^T\overline{\Psi}_{\mathbf{x}} + \underline{\Psi}_{\mathbf{x}} \\
&= \mathbf{s}^T\mathbf{B}_{\widehat{\mathcal{U}}_x} - [\mathbf{s}^T \mid -1] \cdot \Psi_{\mathbf{x}} \\
&\approx \mathbf{s}^T\mathbf{B}_{\widehat{\mathcal{U}}_x} - f(\mathbf{x}) \cdot [\mathbf{s}^T \mid -1] \cdot \mathbf{G}
\end{aligned}$$

Then, when $f(\mathbf{x}) = 0$, the constrained evaluation algorithm outputs

$$y = \left\lfloor \mathbf{c}_{\widehat{\mathcal{U}}_x} \mathbf{G}^{-1}(\mathbf{M}_x) \right\rceil = \left\lfloor \mathbf{s}^T\mathbf{B}_{\widehat{\mathcal{U}}_x} \mathbf{G}^{-1}(\mathbf{M}_x) \right\rceil$$

which is indeed the PRF output on $\mathbf{x}$. The error growth behaves as in the PE scheme and thus, the parameters are set as in Theorem 4.1.

The proof of security closely follows the outline of Theorem 5.5 for our modulus-switching based private CPRF construction. We omit the details from this version. □

5 Our Second Technique: Modulus Switching in HNF

This section contains our PE and CH-CPRF constructions based on the modulus switching method. We start with a technical lemma that explains how rounding is used to push the FHE noise into the ABE noise, as explained in the introduction. This is followed by our construction of a Weakly Attribute Hiding Predicate Encryption in Section 5.1 and our construction of Constraint Hiding Constrained PRF in Section 5.2.

Throughout this section we denote $\lfloor x \rceil_p = \left\lfloor \frac{x}{q/p} \right\rceil$ when the operand is $x \in \mathbb{Z}_q$ and output in $\mathbb{Z}_p$, for q, p that will be defined appropriately in the relevant sections. We extend this operator to vectors and matrices by applying it element-wise. We start with the aforementioned rounding lemma.

Lemma 5.1. *Let n, m', t, p be integers and consider $q = t \cdot p$. Let* FHE *be the scheme guaranteed in Lemma 3.9, with some depth bound d, let d', B as in the lemma statement, and assume that t conforms with the conditions of the lemma. Denote $m = n\lceil \log q \rceil$.*

Let $\mathsf{sk} \in \mathbb{Z}_q^{\ell_s} \leftarrow \text{FHE.}\mathsf{Keygen}(1^\lambda)$ and $\tilde{x} \in \mathbb{Z}_q^{\ell_p} \leftarrow \text{FHE.}\mathsf{Enc}(\mathsf{sk}, x)$ for some $x \in \{0,1\}^\ell$, and for any circuit $f : \{0,1\}^\ell \to \{0,1\}$ define the circuit $f' : \{0,1\}^{\ell_p} \to \{0,1\}^{\ell_s}$ as $f'(\cdot) = \text{FHE.}\mathsf{Eval}(f, \cdot)$. Let $\mathbf{M} \in \mathbb{Z}_p^{n \times m'}, \vec{\mathbf{A}} \in \mathbb{Z}_q^{n \times \ell_p m}, \vec{\mathbf{B}} \in \mathbb{Z}_q^{n \times \ell_s m}$. Denote

$$\mathbf{A}_f = [\vec{\mathbf{A}} \parallel \vec{\mathbf{B}}] \cdot \mathbf{H}_f, \qquad \Psi_f = [\vec{\mathbf{A}} - \tilde{x} \otimes \vec{\mathbf{G}} \parallel \vec{\mathbf{B}} - \mathsf{sk} \otimes \vec{\mathbf{G}}] \cdot \mathbf{H}_{f,\mathbf{x}}$$

where $\mathbf{H}_f = \mathsf{EvalF}^{ip}(f', \vec{\mathbf{A}}, \vec{\mathbf{B}})$ and $\mathbf{H}_{f,\mathbf{x}} = \mathsf{EvalFX}^{ip}(f', \tilde{\mathbf{x}}, \vec{\mathbf{A}}, \vec{\mathbf{B}})$. Then

1. *$\Psi_f = \mathbf{A}_f - (f(x) \cdot t + e)\mathbf{G}$ where $|e| \le B_{\text{FHE}} = B(n\lceil \log q \rceil)^{O(d)}$.*
2. *$\left\lfloor \Psi_f \mathbf{G}^{-1}(\mathbf{M}) \right\rceil_p = \left\lfloor \mathbf{A}_f \mathbf{G}^{-1}(\mathbf{M}) \right\rceil_p - f(x)\mathbf{M} + \mathbf{E}$ where $\|\mathbf{E}\|_\infty \le 2 + \frac{B_{\text{FHE}} \|\mathbf{M}\|_\infty}{t}$.*

Proof. By Theorem 3.8,

$$\begin{aligned}
\Psi_f &= \left[\vec{\mathbf{A}} - \tilde{x} \otimes \vec{\mathbf{G}} \parallel \vec{\mathbf{B}} - \mathsf{sk} \otimes \vec{\mathbf{G}}\right] \cdot \mathbf{H}_{f,\mathbf{x}} \\
&= [\vec{\mathbf{A}} \| \vec{\mathbf{B}}] \cdot \mathbf{H}_f - \langle f'(\tilde{x}), \mathsf{sk} \rangle \mathbf{G} \\
&= \mathbf{A}_f - \langle \text{FHE.}\mathsf{Eval}(f, \tilde{x}), \mathsf{sk} \rangle \mathbf{G}
\end{aligned}$$

where by Lemma 3.9, $\langle \text{FHE.}\mathsf{Eval}(f, \tilde{x}), \mathsf{sk} \rangle = t \cdot f(x) + e$ with $|e| \le B(n\lceil \log q \rceil)^{O(d)}$, so (1) follows. Moreover,

$$\begin{aligned}
\left\lfloor \Psi_f \mathbf{G}^{-1}(\mathbf{M}) \right\rceil_p &= \left\lfloor (\mathbf{A}_f - (t \cdot f(x) + e)\mathbf{G}) \mathbf{G}^{-1}(\mathbf{M}) \right\rceil_p \\
&= \left\lfloor \mathbf{A}_f \mathbf{G}^{-1}(\mathbf{M}) - t \cdot f(x)\mathbf{M} - e\mathbf{M} \right\rceil_p \\
&= \left\lfloor \mathbf{A}_f \mathbf{G}^{-1}(\mathbf{M}) - e\mathbf{M} \right\rceil_p - f(x)\mathbf{M} \\
&= \left\lfloor \mathbf{A}_f \mathbf{G}^{-1}(\mathbf{M}) \right\rceil_p - f(x)\mathbf{M} - \mathbf{E}
\end{aligned}$$

where $\mathbf{E} = (e/t)\mathbf{M} + \mathbf{\Delta}$ for a rounding-errors matrix $\|\mathbf{\Delta}\|_\infty \le 2$, and therefore $\|\mathbf{E}\|_\infty \le 2 + |e| \cdot (\|\mathbf{M}\|_\infty / t)$. □

5.1 Weakly Attribute Hiding Predicate Encryption

The scheme is parameterized by $\epsilon \in (0,1)$ which governs the lattice hardness assumption that underly the construction. Essentially, with parameter ϵ the scheme will be secure under the polynomial hardness of approximating lattice problems to within a $2^{\widetilde{O}(n^{\epsilon})}$-factor.

- $\mathsf{PE.Setup}(1^{\lambda}, 1^{d}) \to (\mathsf{pp}, \mathsf{msk})$. Define $\ell = \lambda$ (this is the supported attribute length). Set $n = (\lambda d)^{1/\epsilon}$. Let χ be the $B = \widetilde{O}(\sqrt{n})$-bounded distribution from Corollary 3.1. Let p, τ be integer parameters set such that $\tau \geq z_1$, $p \geq 4z_2 \cdot \tau$ for parameters $z_1, z_2 = 2^{d \cdot \mathrm{polylog}(n)}$ that will be specified throughout the analysis. Let $t = \Theta(p)$ and $q = p \cdot t$. Denote $m = n\lceil \log q \rceil$. Recall Corollary 3.2 and let $m_0 = m_0(n, q)$ as in the corollary statement. Let FHE be the scheme from Lemma 3.9 with depth parameter d, define ℓ_s, ℓ_c, d' as in the lemma statement, and let $\ell_p = \ell \cdot \ell_c$.

  Recall Corollary 3.2 and let $m_0 = m_0(n, p)$ as in the corollary statement. Consider $m' = \max\{(n+1)\lceil \log q \rceil + 2\lambda, m_0\}$ (note that m_0 is w.r.t p but m' needs to be larger than $(n+1)\lceil \log q \rceil$). Generate a matrix with a trapdoor $(\mathbf{A}, \mathbf{A}_{\tau_0}^{-1}) \leftarrow \mathsf{TrapGen}(1^n, p, m')$, i.e. $\mathbf{A} \in \mathbb{Z}_p^{n \times m'}$. Sample a uniform $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_p^n$. Generate uniform $\vec{\mathbf{A}} \xleftarrow{\$} (\mathbb{Z}_q^{n \times m})^{\ell_p}$ and $\vec{\mathbf{B}} \xleftarrow{\$} (\mathbb{Z}_q^{n \times m})^{\ell_s}$.

  Let $\mathsf{msk} = \mathbf{A}_{\tau_0}^{-1}$ and $\mathsf{pp} = (\mathbf{A}, \mathbf{v}, \vec{\mathbf{A}}, \vec{\mathbf{B}})$.

- $\mathsf{PE.Enc}_{\mathsf{pp}}(\mu, x) \to \mathsf{ct}$. Generate $\mathsf{sk} \leftarrow \mathrm{FHE.Keygen}(1^{\lambda})$, s.t. $\mathsf{sk} \in \mathbb{Z}_p^{\ell_s}$ and compute $\tilde{\mathbf{x}} \leftarrow \mathrm{FHE.Enc}(\mathsf{sk}, x)$. Sample a vector $\mathbf{s} \xleftarrow{\$} \chi^n$, an error vector $\mathbf{e} \xleftarrow{\$} \chi^{m'}$ and an error scalar $e \xleftarrow{\$} \chi$. Sample $\mathbf{R}_A \xleftarrow{\$} \{0,1\}^{(m' \times m)\ell_p}$ and $\mathbf{R}_B \xleftarrow{\$} \{0,1\}^{(m' \times m)\ell_s}$. Sample a matrix $\mathbf{A}_t \xleftarrow{\$} \mathbb{Z}_t^{n \times m'}$ and a vector $\mathbf{v}_t \xleftarrow{\$} \mathbb{Z}_t^n$. Encrypt as follows:

$$
\begin{aligned}
\mathbf{u}_0 &= \mathbf{s}\mathbf{A} + \lfloor \mathbf{s}\mathbf{A}_t + \mathbf{e} \rceil_p \pmod p \\
\mathbf{u}_\mu &= \mathbf{s}\mathbf{v} + \lfloor \mathbf{s}\mathbf{v}_t + e \rceil_p + \mu \lfloor p/2 \rceil \pmod p \\
\vec{\mathbf{a}} &= \mathbf{s}(\vec{\mathbf{A}} - \tilde{\mathbf{x}} \otimes \mathbf{G}_q) + \mathbf{e}\mathbf{R}_A \pmod q \\
\vec{\mathbf{b}} &= \mathbf{s}(\vec{\mathbf{B}} - \mathsf{sk} \otimes \mathbf{G}_q) + \mathbf{e}\mathbf{R}_B \pmod q
\end{aligned}
$$

  Output $\mathsf{ct} = (\tilde{\mathbf{x}}, \mathbf{u}_0, \mathbf{u}_\mu, \vec{\mathbf{a}}, \vec{\mathbf{b}})$.

- $\mathsf{PE.Keygen}_{\mathsf{msk}}(f) \to \mathsf{sk}_f$. Define $f'(\cdot) = \mathrm{FHE.Eval}(f, \cdot)$ and compute $\mathbf{A}_f = [\vec{\mathbf{A}} \| \vec{\mathbf{B}}] \cdot \mathbf{H}_f$, where $\mathbf{H}_f \leftarrow \mathsf{EvalF}^{ip}(f', \vec{\mathbf{A}}, \vec{\mathbf{B}})$. Compute $\widehat{\mathbf{A}}_f = \left\lfloor \mathbf{A}_f \mathbf{G}^{-1}(\mathbf{G}_p) \right\rceil_p$. Use $\mathbf{A}_{\tau_0}^{-1}$ to sample $[\mathbf{h}_f \| \mathbf{k}_f] = [\mathbf{I} \| \mathbf{A} \| \widehat{\mathbf{A}}_f]_{\tau}^{-1}(\mathbf{v})$, i.e. s.t. $[\mathbf{A} \| \widehat{\mathbf{A}}_f]\mathbf{k}_f = \mathbf{v} - \mathbf{h}_f \pmod p$. Output $\mathsf{sk}_f = \mathbf{k}_f$.

- $\mathsf{PE.Dec}_{\mathsf{pp}}(\mathsf{ct}, \mathsf{sk}_f) \to \mu$. Compute $\mathbf{H}_{f,x} \leftarrow \mathsf{EvalFX}^{ip}(f', \tilde{\mathbf{x}}, \vec{\mathbf{A}}, \vec{\mathbf{B}})$ and set $\mathbf{a}_{f,x} = [\vec{\mathbf{a}} \| \vec{\mathbf{b}}]\mathbf{H}_{f,x}$. Compute $\widehat{\mathbf{a}}_{f,x} = (1/t)(\mathbf{a}_{f,x}\mathbf{G}^{-1}(\mathbf{G}_p))$ and $b' = \mathbf{u}_\mu - [\mathbf{u}_0 \| \widehat{\mathbf{a}}_{f,x}]\mathbf{k}_f \pmod p$. Return 0 if $|b'| < \frac{p}{4}$ and 1 otherwise.

Analysis. Correctness and security are stated and proven next. We note that since $q \leq 2^n$ regardless of the exact manner we choose p, τ we have that any polynomial of the form $\mathrm{poly}(\lambda, B, (n\lceil \log q \rceil)^{O(d')})$ is upper bounded by a function of the form $2^{d \cdot \mathrm{polylog}(n)}$. This is since $n\lceil \log q \rceil \leq n^2$, $\lambda < n$ and $d' = d \cdot \mathrm{polylog}(n\lceil \log q \rceil) = d \cdot \mathrm{polylog}(n)$.

Theorem 5.2 (Correctness)**.** *The* PE *construction above is correct as per Definition 3.2.*

Proof. Let ct be an encryption of message μ under attribute x and let $\mathbf{k}_f$ be a secret key for a function f. Let $\mathbf{H}_f = \mathsf{EvalF}^{ip}(f', \vec{\mathbf{A}}, \vec{\mathbf{B}})$, $\mathbf{H}_{f,\mathbf{x}} = \mathsf{EvalFX}^{ip}(f', \vec{\mathbf{A}}, \vec{\mathbf{B}})$, $\mathbf{A}_f = [\vec{\mathbf{A}} \| \vec{\mathbf{B}}] \cdot \mathbf{H}_f$, and denote $\Psi_f = [\vec{\mathbf{A}} - \tilde{x}\vec{\mathbf{G}} \| \vec{\mathbf{B}} - \mathsf{sk}\vec{\mathbf{G}}] \cdot \mathbf{H}_{f,\mathbf{x}}$. By Lemma 5.1, $\Psi_f = \mathbf{A}_f - (f(x) \cdot t + e)\mathbf{G}$ where $|e| \leq B_{\mathrm{FHE}} = B(n\lceil \log q \rceil)^{O(d)}$. Then

$$\begin{aligned}
\mathbf{a}_{f,x} &= [\vec{\mathbf{a}} \| \vec{\mathbf{b}}] \mathbf{H}_{f,x} \\
&= \left(\mathbf{s}([\vec{\mathbf{A}} \| \vec{\mathbf{B}}] - [\tilde{\mathbf{x}} \| \mathsf{sk}] \otimes \mathbf{G}) + \mathbf{e}[\mathbf{R}_A \| \mathbf{R}_B] \right) \mathbf{H}_{f,x} \\
&= \mathbf{s}\Psi_f + \mathbf{e}[\mathbf{R}_A \| \mathbf{R}_B] \mathbf{H}_{f,x} \\
&= \mathbf{s}\left(\mathbf{A}_f - (f(x) \cdot t + e)\mathbf{G}\right) + \mathbf{e}[\mathbf{R}_A \| \mathbf{R}_B] \mathbf{H}_{f,x}
\end{aligned}$$

Therefore,

$$\begin{aligned}
\widehat{\mathbf{a}}_{f,x} &= \frac{\mathbf{a}_{f,x} \mathbf{G}^{-1}(\mathbf{G}_p)}{t} \\
&= \frac{\mathbf{s}\left(\mathbf{A}_f - (f(x) \cdot t + e)\mathbf{G}\right) \mathbf{G}^{-1}(\mathbf{G}_p)}{t} + \underbrace{\frac{\mathbf{e}[\mathbf{R}_A \| \mathbf{R}_B] \mathbf{H}_{f,x} \mathbf{G}^{-1}(\mathbf{G}_p)}{t}}_{\mathbf{e}_1} \\
&= \frac{\mathbf{s}\mathbf{A}_f \mathbf{G}^{-1}(\mathbf{G}_p)}{t} - f(x)\mathbf{s}\mathbf{G}_p \underbrace{-(e/t)\mathbf{s}\mathbf{G}_p}_{\mathbf{e}_2} + \mathbf{e}_1 \\
&= \mathbf{s} \cdot \left\lfloor \frac{\mathbf{A}_f \mathbf{G}^{-1}(\mathbf{G}_p)}{t} \right\rceil - f(x)\mathbf{s}\mathbf{G}_p + \mathbf{e}_1 + \mathbf{e}_2 + \underbrace{\mathbf{s}\boldsymbol{\Delta}}_{\mathbf{e}_3},
\end{aligned}$$

where $\boldsymbol{\Delta}$ is the matrix of rounding errors, i.e. $\|\boldsymbol{\Delta}\|_\infty \leq 1/2$. We can bound the error $\mathbf{e}' = \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3$ as follows: $\|\mathbf{e}_1\|_\infty \leq Bm'(\ell_p + \ell_s)(n\lceil \log q \rceil)^{O(d')} n \lceil \log p \rceil / t$, $\|\mathbf{e}_2\|_\infty \leq nBp(n\lceil \log q \rceil)^{O(d)}/t$, $\|\mathbf{e}_3\|_\infty \leq nB/2$. Note that $\ell_p, \ell_s = \mathrm{poly}(n\lceil \log q \rceil)$, hence $\|\mathbf{e}'\|_\infty \leq \mathrm{poly}(\lambda, B, (n\lceil \log q \rceil)^{O(d')})$.

It follows that if indeed $f(x) = 0$ then $\widehat{\mathbf{a}}_{f,x} = \mathbf{s}\widehat{\mathbf{A}}_f + \mathbf{e}'$. Now, recall that the distribution of $\mathbf{k}_f, \mathbf{h}_f$ is Gaussian with parameter τ subject to $[\mathbf{A} \| \widehat{\mathbf{A}}_f]\mathbf{k}_f = \mathbf{v} - \mathbf{h}_f \pmod p$. Therefore $\|\mathbf{k}_f\|_\infty \leq \tau\sqrt{\lambda(m + m')}$ and $\|\mathbf{h}_f\|_\infty \leq \tau\sqrt{\lambda n}$ with all but $2^{-\lambda} = \mathrm{negl}(\lambda)$ probability. By definition,

$$\mathbf{u}_0 = \mathbf{s}\mathbf{A} + \lfloor \mathbf{s}\mathbf{A}_t + \mathbf{e} \rceil_p, \qquad \mathbf{u}_\mu = \mathbf{s}\mathbf{v} + \lfloor \mathbf{s}\mathbf{v}_t + e \rceil_p + \mu\lfloor p/2 \rceil$$

Denote $\mathbf{e}_0 = \lfloor \mathbf{s}\mathbf{A}_t + \mathbf{e} \rceil_p$ and $e_\mu = \lfloor \mathbf{s}\mathbf{v}_t + e \rceil_p$, then $\|\mathbf{e}_0\|_\infty, |e_\mu| \leq (n+1)B$. Therefore,

$$\begin{aligned}
b' &= \mathbf{u}_\mu - [\mathbf{u}_0 \| \widehat{\mathbf{a}}_{f,x}]\mathbf{k}_f \\
&= \mathbf{s}\mathbf{v} + e_\mu + \mu\lfloor p/2 \rceil - \mathbf{s}[\mathbf{A} \| \widehat{\mathbf{A}}_f - f(x)\mathbf{G}_p]\mathbf{k}_f - [\mathbf{e}_0 \| \mathbf{e}']\mathbf{k}_f \\
&= \mu\lfloor p/2 \rceil + \underbrace{e_\mu - \mathbf{s}\mathbf{h}_f - [\mathbf{e}_0 \| \mathbf{e}']\mathbf{k}_f}_{e''} + f(x)\mathbf{s}[\mathbf{0} \| \mathbf{G}_p]\mathbf{k}_f
\end{aligned}$$

where $|e''| < \tau \cdot \mathrm{poly}(\lambda, B, (n\lceil \log q \rceil)^{O(d')})$. Therefore there exists some $z_2 = 2^{d\mathrm{polylog}(n)}$ s.t. when we set $p > 4z_2\tau$ we get that $|e''| < \frac{p}{4}$. Hence, if $f(x) = 0$ then $b' = \mu\lfloor p/2 \rceil + e'' \in \mu\lfloor p/2 \rceil \pm \frac{p}{4}$ and in particular $\mu = 0$ implies $|b'| < \frac{p}{4}$ and $\mu = 1$ implies $|b'| > \frac{p}{4}$. □ □

Theorem 5.3 (Security)**.** *The scheme* PE *is secure as per Definition 3.3 under the* $\mathrm{LWE}_{n,q,\chi}$ *assumption, and thus under the worst case hardness of approximating* $\mathsf{GapSVP}, \mathsf{SIVP}$ *to within a* $2^{\widetilde{O}(n^\epsilon)}$ *factor in polynomial time.*

Sketch. Define the simulator $\mathrm{Sim}(\mathsf{pp}) \to \mathsf{ct}$ that generates $\mathsf{ct} = (\tilde{x}, \mathbf{u}_0, \mathbf{u}_\mu, \vec{\mathbf{a}}, \vec{\mathbf{b}})$ by computing $\tilde{x} \leftarrow \mathrm{FHE.Enc}(\mathsf{sk}, 0^\ell)$ and sampling all the other ct parts uniformly from $\mathbb{Z}_q$ as required. We now show a sequence of hybrids, where the first hybrid corresponds to $\exp_{real}$ and the last hybrid corresponds to $\exp_{ideal}$ with the simulator Sim we just defined.

Hybrid $\mathcal{H}_0$. This is $\exp_{real}$.

Hybrid $\mathcal{H}_1$. We change the Setup algorithm, specificaly the generation of $\vec{\mathbf{A}}, \vec{\mathbf{B}}$: Let x be the attribute declared by the adversary. Generate $\mathsf{sk} \leftarrow \mathrm{FHE.Keygen}(1^\lambda)$ and compute $\tilde{x} \leftarrow \mathrm{FHE.Enc}(\mathsf{sk}, x)$. Sample $\mathbf{R}_A \xleftarrow{\$} \{0,1\}^{m' \times (m\ell_p)}$ and $\mathbf{R}_B \xleftarrow{\$} \{0,1\}^{m' \times (m\ell_s)}$, and define

$$\vec{\mathbf{A}} = (t\mathbf{A} + \mathbf{A}_t)\mathbf{R}_A + \tilde{\mathbf{x}} \otimes \mathbf{G}_q, \qquad \vec{\mathbf{B}} = (t\mathbf{A} + \mathbf{A}_t)\mathbf{R}_B + \mathsf{sk} \otimes \mathbf{G}_q \ .$$

$\mathbf{A}$ is statistically close to uniform in $\mathbb{Z}_p^{n \times m'}$ and $\mathbf{A}_t$ is uniform in $\mathbb{Z}_t^{n \times m'}$, therefore the matrix $t\mathbf{A} + \mathbf{A}_t$ is close to uniform in $\mathbb{Z}_q$. Since each $\mathbf{R}_A, \mathbf{R}_B$ are sampled uniformly and independently and $m' \geq (n+1)\lceil \log q \rceil + 2\lambda$, indistinguishability follows from the extended leftover hash lemma.

Hybrid $\mathcal{H}_2$. We change the Enc algorithm. Sample $\mathbf{s} \leftarrow \chi_q^n$, $\mathbf{e} \leftarrow \chi_q^{m'}$ and $e \leftarrow \chi_q$ as in the original encryption algorithm, then compute

$$\mathbf{u}_0' = \mathbf{s}(t\mathbf{A} + \mathbf{A}_t) + \mathbf{e}, \qquad \mathbf{u}_\mu' = \mathbf{s}(t\mathbf{v} + \mathbf{v}_t) + e \ .$$

Encrypt as follows:

$$\mathbf{u}_0 = \left\lfloor \mathbf{u}_0' \right\rceil_p, \qquad \mathbf{u}_\mu = \left\lfloor \mathbf{u}_\mu' \right\rceil_p, \qquad \vec{\mathbf{a}} = \mathbf{u}_0' \mathbf{R}_A, \qquad \vec{\mathbf{b}} = \mathbf{u}_0' \mathbf{R}_B.$$

The distributions remain as in the original scheme so statistical indistinguishability is maintained:

$$\begin{aligned}
\mathbf{u}_0 &= \left\lfloor \mathbf{u}_0' \right\rceil_p = \left\lfloor \mathbf{s}(t\mathbf{A} + \mathbf{A}_t) + \mathbf{e} \right\rceil_p = \mathbf{s}\mathbf{A} + \left\lfloor \mathbf{s}\mathbf{A}_t + \mathbf{e} \right\rceil_p \\
\mathbf{u}_\mu &= \left\lfloor \mathbf{u}_\mu' \right\rceil_p = \left\lfloor \mathbf{s}(t\mathbf{v} + \mathbf{v}_t) + e \right\rceil_p = \mathbf{s}\mathbf{v} + \left\lfloor \mathbf{s}\mathbf{v}_t + e \right\rceil_p \\
\vec{\mathbf{a}} &= \mathbf{u}_0' \mathbf{R}_A = (\mathbf{s}(t\mathbf{A} + \mathbf{A}_t) + \mathbf{e})\mathbf{R}_A = \mathbf{s}(\vec{\mathbf{A}} - \tilde{\mathbf{x}} \otimes \mathbf{G}_q) + \mathbf{e}\mathbf{R}_A \\
\vec{\mathbf{b}} &= \mathbf{u}_0' \mathbf{R}_B = (\mathbf{s}(t\mathbf{A} + \mathbf{A}_t) + \mathbf{e})\mathbf{R}_B = \mathbf{s}(\vec{\mathbf{B}} - \mathsf{sk} \otimes \mathbf{G}_q) + \mathbf{e}\mathbf{R}_B
\end{aligned}$$

Hybrid $\mathcal{H}_3$. We change the Keygen algorithm. We're only required to generate keys for f s.t. $f(x) = 1$, otherwise the adversary is not admissible. Recall that in $\mathsf{PE.Keygen}$ we sample from $[\mathbf{I} \| \mathbf{A} \| \widehat{\mathbf{A}}_f]_\tau^{-1}(\mathbf{v})$, where $\widehat{\mathbf{A}}_f = \left\lfloor \mathbf{A}_f \mathbf{G}^{-1}(\mathbf{G}_p) \right\rceil_p$ and $\mathbf{A}_f = [\vec{\mathbf{A}} \| \vec{\mathbf{B}}] \cdot \mathbf{H}_f$. Using the notation

$$\Psi_f = [\vec{\mathbf{A}} - \tilde{x} \otimes \mathbf{G}_q \| \vec{\mathbf{B}} - \mathsf{sk} \otimes \mathbf{G}_q] \cdot \mathbf{H}_{f,\mathbf{x}} \ ,$$

after the changes that were made in the previous hybrid, we have:

$$\Psi_f = [\vec{\mathbf{A}} - \tilde{x} \otimes \mathbf{G}_q \| \vec{\mathbf{B}} - \mathsf{sk} \otimes \mathbf{G}_q] \cdot \mathbf{H}_{f,x} = (t\mathbf{A} + \mathbf{A}_t)[\mathbf{R}_A \| \mathbf{R}_B] \cdot \mathbf{H}_{f,x} \ .$$

so

$$\begin{aligned}\left\lfloor \Psi_f \mathbf{G}^{-1}(\mathbf{G}_p)\right\rceil_p &= \left\lfloor (t\mathbf{A}+\mathbf{A}_t)[\mathbf{R}_A\|\mathbf{R}_B]\cdot \mathbf{H}_{f,x}\mathbf{G}^{-1}(\mathbf{G}_p)\right\rceil_p \\ &= \mathbf{A}[\mathbf{R}_A\|\mathbf{R}_B]\cdot \mathbf{H}_{f,x}\mathbf{G}^{-1}(\mathbf{G}_p) + \left\lfloor \mathbf{A}_t[\mathbf{R}_A\|\mathbf{R}_B]\cdot \mathbf{H}_{f,x}\mathbf{G}^{-1}(\mathbf{G}_p)\right\rceil_p \\ &= \mathbf{A}[\mathbf{R}_A\|\mathbf{R}_B]\cdot \mathbf{H}_{f,x}\mathbf{G}^{-1}(\mathbf{G}_p) + \mathbf{E}' \qquad \left\|\mathbf{E}'\right\|_\infty \le (n\lceil\log q\rceil)^{O(d')}\end{aligned}$$

and by Lemma 5.1,

$$\left\lfloor \Psi_f \mathbf{G}^{-1}(\mathbf{G}_p)\right\rceil_p = \left\lfloor \mathbf{A}_f \mathbf{G}^{-1}(\mathbf{G}_p)\right\rceil_p - f(x)\mathbf{G}_p + \mathbf{E} \qquad \|\mathbf{E}\|_\infty \le 2 + \frac{B_{\mathrm{FHE}}\cdot p}{t}$$

Therefore, when $f(x)=1$,

$$\begin{aligned}\widehat{\mathbf{A}}_f = \left\lfloor \mathbf{A}_f \mathbf{G}^{-1}(\mathbf{G}_p)\right\rceil_p &= \left\lfloor \Psi_f \mathbf{G}^{-1}(\mathbf{G}_p)\right\rceil_p + \mathbf{G}_p - \mathbf{E} \\ &= \mathbf{A}[\mathbf{R}_A\|\mathbf{R}_B]\cdot \mathbf{H}_{f,x}\mathbf{G}^{-1}(\mathbf{G}_p) + \mathbf{G}_p + \mathbf{E}' - \mathbf{E}\end{aligned}$$

where $\|\mathbf{E}'-\mathbf{E}\|_\infty \le \mathrm{poly}(\lambda, B, (n\lceil\log q\rceil)^{O(d')})$. Given $[\mathbf{R}_A\|\mathbf{R}_B]\mathbf{H}_{f,x}\mathbf{G}^{-1}(\mathbf{G}_p)$ we can also compute $\mathbf{E}'-\mathbf{E}$, and then, by Corollary 3.6, we can compute the trapdoor $[\mathbf{I}\|\mathbf{A}\|\widehat{\mathbf{A}}_f]_\tau^{-1}$ for any $\tau \ge z_1$ for

$$\begin{aligned}z_1 &= O(\sqrt{mm'}\left\|[\mathbf{R}_A\|\mathbf{R}_B]\mathbf{H}_{f,x}\mathbf{G}^{-1}(\mathbf{G}_p)\right\|_\infty + \sqrt{mn}\left\|\mathbf{E}'-\mathbf{E}\right\|_\infty) \\ &\le \mathrm{poly}(\lambda, B, (n\lceil\log q\rceil)^{O(d')}) \le 2^{d\cdot\mathrm{polylog}(n)}\ .\end{aligned}$$

We will choose our parameters so that indeed $\tau \ge z_1$ which will allow us to sample from $[\mathbf{I}_n\|\mathbf{A}\|\widehat{\mathbf{A}}_f]_\tau^{-1}(\mathbf{v})$. Note that in this hybrid $\mathbf{A}_{\tau_0}^{-1}$ is no longer used.

Hybrid $\mathcal{H}_4$. In Setup: Generate $\mathbf{A}$ uniformly instead generating it with a trapdoor. Statistical indistinguishability holds by Corollary 3.2.

Hybrid $\mathcal{H}_5$. In Enc: Generate $\mathbf{u}_0', \mathbf{u}_\mu'$ uniformly in $\mathbb{Z}_q^n, \mathbb{Z}_q$ respectively. This is indistinguishable assuming hardness of $\mathrm{DLWE}_{q,n,\chi}$. Note that now $\mathbf{u}_0 = \left\lfloor \mathbf{u}_0'\right\rceil_p$ and $\mathbf{u}_\mu = \left\lfloor \mathbf{u}_\mu'\right\rceil_p$ are uniform in $\mathbb{Z}_p^n, \mathbb{Z}_p$ as well.

Hybrid $\mathcal{H}_6$. In Enc: Genrate $\vec{\mathbf{a}}$ and $\vec{\mathbf{b}}$ uniformly from $\mathbb{Z}_p^m$. This is indistinguishable by the extended leftover hash lemma since $\mathbf{u}_0'$ is uniform, $\mathbf{R}_A, \mathbf{R}_B$ were randomly and independently generated and $m' \ge (n+1)\lceil\log q\rceil + 2\lambda$. The only information that ct reveals now is $\tilde{x}$.

Hybrid $\mathcal{H}_7$. In Setup: Generate $\mathbf{A}$ together with a trapdoor (the opposite of Hybrid 4). Statistical indistinguishability holds by Corollary 3.2.

Hybrid $\mathcal{H}_8$. In Keygen: Generate keys with $\mathbf{A}_{\tau_0}^{-1}$ (the opposite of Hybrid 3). Indistinguishability holds since the keys are sampled from the same distribution.

Hybrid $\mathcal{H}_9$. In Setup: Generate the matrics $\vec{\mathbf{A}}, \vec{\mathbf{B}}$ as in the real Setup algorithm (the opposite of Hybrid 1). Indistinguishability holds by the leftover hash lemma.

Hybrid $\mathcal{H}_{10}$. Change $\tilde{x}$ to $\tilde{x} \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}, 0^\ell)$. By Lemma 3.9, those hybrids are indistinguishable under $\mathrm{DLWE}_{n,q,\chi}$. In this hybrid the Enc algorithm is equivalent to the simulator Sim that was defined at the beginning of the proof, therefore it is equivalent to $\exp_{ideal}$. □ □

5.2 Constraint Hiding Constrained PRF

We present a constraint hiding constrained PRF scheme that supports all functions expressible by boolean circuits of depth d, input length k and description length ℓ, for predefined polynomials ℓ, k, d. We will rely on the hardness of LWE with sub-exponential noise to modulus ratio, as in our predicate encryption scheme. Working with a predefined polynomial input length k makes the analysis much simpler than [BV15b], however we note that relying on a different hardness assumption (a variant of one dimensional SIS) it is possible to support a-priori unbounded inputs as in [BV15b].

- $\mathsf{CPRF.Keygen}(1^\lambda, 1^\ell, 1^k, 1^d) \to (\mathsf{pp}, \sigma)$. We let n be a parameter to be chosen later as a function of λ, ℓ, k, d. We let $q = p \cdot t$ and t' be s.t. $t'|p$. If we wish to rely on the hardness of lattice problems with approximation ratio $2^{\tilde{O}(n^\epsilon)}$, then all values p, t, t' will be of size $2^{\tilde{O}(n^\epsilon)}$ as well. The resulting constrained PRF scheme will support constraint functions of description length ℓ, input length k and depth d. The PRF itself outputs random elements in $\mathbb{Z}_{p/t'}$, i.e. $\log(p/t')$ bits of randomness.

  Denote $m = n\lceil \log q \rceil$ and $m' = n\lceil \log p \rceil$. Let FHE be the scheme from Lemma 3.9 with depth parameter d, define ℓ_c, ℓ_s, d' as in the lemma statement, where ℓ_c is the FHE ciphertext length, ℓ_s is the FHE key length and d' is the max depth of $\mathsf{FHE.Eval}_{\mathsf{pp}}(f, \cdot)$ for any f of depth at most d. Denote $\ell_p = \ell \cdot \ell_c$. Let $\mathbf{G}_q$ and $\mathbf{G}_p$ denote the gadget matrices of dimensions $n \times n\lceil \log q \rceil$ and $n \times n\lceil \log p \rceil$ respectively.

  Generate $\vec{\mathbf{A}} \xleftarrow{\$} (\mathbb{Z}_q^{n \times m})^{\ell_p}$ and $\vec{\mathbf{B}} \xleftarrow{\$} (\mathbb{Z}_q^{n \times m})^{\ell_s}$. Generate $\mathbf{D} \xleftarrow{\$} \mathbb{Z}_p^{n \times m'}$ and $\vec{\mathbf{C}} = [\mathbf{C}_0 \| \mathbf{C}_1] \xleftarrow{\$} (\mathbb{Z}_p^{n \times m'})^2$. Sample a vector $\mathbf{s} \xleftarrow{\$} \chi^n$ and compute $\mathsf{sk} \leftarrow \mathsf{FHE.Keygen}(1^\lambda)$. Sample an error vector $\mathbf{e}_b \xleftarrow{\$} \chi^{m\ell_s}$ and let $\vec{\mathbf{b}} = \mathbf{s}(\vec{\mathbf{B}} - \mathsf{sk} \otimes \mathbf{G}_q) + \mathbf{e}_b$. The public parameters are $\mathsf{pp} = (\vec{\mathbf{A}}, \vec{\mathbf{B}}, \vec{\mathbf{C}}, \mathbf{D}, \vec{\mathbf{b}})$ and the master seed is $\sigma = (\mathbf{s}, \mathsf{sk})$.

- $\mathsf{CPRF.Eval}_{\mathsf{pp}}(\sigma, x) \to y \in \mathbb{Z}_{p/t'}$. Let $\mathcal{U}_x : \{0,1\}^\ell \to \{0,1\}$ be the circuit that takes as input a description of a function f and outputs $f(x)$. Now consider the circuit $\mathcal{U}'_x : \{0,1\}^{\ell_p} \to \{0,1\}^{\ell_s}$ that takes as input an encryption of a description of f, i.e. $\tilde{f} = \mathsf{FHE.Enc}(\mathsf{sk}, f)$, and outputs $\mathsf{FHE.Eval}(\mathcal{U}_x, \tilde{f})$, i.e. an FHE encryption of $f(x)$. Compute $\mathbf{A}_x = [\vec{\mathbf{A}} \| \vec{\mathbf{B}}] \cdot \mathbf{H}_x$, where $\mathbf{H}_x \leftarrow \mathsf{EvalF}^{ip}(\mathcal{U}'_x, \vec{\mathbf{A}}, \vec{\mathbf{B}})$. Compute $\mathbf{C}_x = \mathsf{EvalF}(\mathcal{T}_x, \vec{\mathbf{C}})$ (as defined in Section 3.7) and fix $\mathbf{M}_x = \mathbf{D}\mathbf{G}_p^{-1}(\mathbf{C}_x) \mod p$. Output

$$y = \left\lfloor \mathbf{s} \cdot \frac{\mathbf{A}_x \mathbf{G}_q^{-1}(\mathbf{M}_x)}{t' \cdot t} \right\rceil .$$

- $\mathsf{CPRF.Constrain}_{\mathsf{pp}}(\sigma, f) \to \sigma_f$. Compute $\tilde{f} = \mathsf{FHE.Enc}(\mathsf{sk}, f)$. Sample an error vector $\mathbf{e}_a \xleftarrow{\$} \chi^{m\ell_p}$ and compute $\vec{\mathbf{a}} = \mathbf{s}(\vec{\mathbf{A}} - \tilde{f} \otimes \mathbf{G}_q) + \mathbf{e}_a$. Output $\sigma_f = (\vec{\mathbf{a}}, \tilde{f})$.

- $\mathsf{CPRF.ConstrainEval}_{\mathsf{pp}}(\sigma_f, x) \to y' \in \mathbb{Z}_r$. Compute $\mathbf{a}_{f,x} = [\vec{\mathbf{a}} \| \vec{\mathbf{b}}] \cdot \mathbf{H}_{f,x}$, where $\mathbf{H}_{f,x} \leftarrow \mathsf{EvalFX}^{ip}(\mathcal{U}'_x, \tilde{f}, \vec{\mathbf{A}}, \vec{\mathbf{B}})$, and output

$$y' = \left\lfloor \frac{\mathbf{a}_{f,x} \mathbf{G}_q^{-1}(\mathbf{M}_x)}{t' \cdot t} \right\rceil$$

Analysis. The following will be useful in the security and correctness proof.

Lemma 5.4. *Let d' denote the depth of the circuit $\mathcal{U}'_x$. Consider $\mathbf{a}_{f,x}$ and $\mathbf{A}_x$ as defined in* CPRF.ConstrainEval *and* CPRF.Eval*, then:*

$$\frac{\mathbf{a}_{f,x} \mathbf{G}_q^{-1}(\mathbf{M}_x)}{t} = \mathbf{s} \frac{\mathbf{A}_x \mathbf{G}_q^{-1}(\mathbf{M}_x)}{t} - f(x)\mathbf{s}\mathbf{M}_x + \mathbf{e}''$$

where $\|\mathbf{e}''\|_\infty \leq \mathrm{poly}(\lambda, B, (n\lceil \log q \rceil)^{O(d')})$.

Proof. Recall that $\|[\mathbf{e}_a \| \mathbf{e}_b]\|_\infty \leq B$ and $\|\mathbf{H}_{f,x}\|_\infty \leq (n\lceil \log q \rceil)^{O(d')}$. Hence

$$\begin{aligned}
\mathbf{a}_{f,x} &= [\vec{\mathbf{a}} \| \vec{\mathbf{b}}] \cdot \mathbf{H}_{f,x} \\
&= \mathbf{s} \underbrace{[\vec{\mathbf{A}} - \tilde{f} \otimes \mathbf{G}_q \| \vec{\mathbf{B}} - \mathsf{sk} \otimes \mathbf{G}_q] \cdot \mathbf{H}_{f,x}}_{\Psi_x} + \underbrace{[\mathbf{e}_a \| \mathbf{e}_b] \cdot \mathbf{H}_{f,x}}_{\mathbf{e}}
\end{aligned}$$

where $\|\mathbf{e}\|_\infty \leq \mathrm{poly}(\lambda, B, (n\lceil \log q \rceil)^{O(d')})$. Therefore

$$\frac{\mathbf{a}_{f,x} \mathbf{G}_q^{-1}(\mathbf{M}_x)}{t} = \frac{(\mathbf{s}\Psi_x + \mathbf{e}) \mathbf{G}_q^{-1}(\mathbf{M}_x)}{t} = \mathbf{s} \cdot \frac{\Psi_x \mathbf{G}_q^{-1}(\mathbf{M}_x)}{t} + \underbrace{\mathbf{e}/t \cdot \mathbf{G}_q^{-1}(\mathbf{M}_x)}_{\mathbf{e}'} \tag{10}$$

where $\|\mathbf{e}'\|_\infty \leq \mathrm{poly}(\lambda, B, (n\lceil \log q \rceil)^{O(d')})$.

By Lemma 5.1, $\Psi_x = \mathbf{A}_x - (f(x) \cdot t + e)\mathbf{G}_q$ where $|e| \leq B_{\mathrm{FHE}} = B(n\lceil \log q \rceil)^{O(d)}$, therefore

$$\frac{\Psi_x \mathbf{G}_q^{-1}(\mathbf{M}_x)}{t} = \frac{\mathbf{A}_x \mathbf{G}_q^{-1}(\mathbf{M}_x)}{t} - f(x)\mathbf{M}_x - \underbrace{e/t\mathbf{M}_x}_{\mathbf{E}} \qquad \|\mathbf{E}\|_\infty \leq B_{\mathrm{FHE}} \cdot (p/t) \tag{11}$$

From Equations 10 and 11, we get

$$\begin{aligned}
\frac{\mathbf{a}_{f,x} \mathbf{G}_q^{-1}(\mathbf{M}_x)}{t} &= \mathbf{s} \cdot \frac{\Psi_x \mathbf{G}_q^{-1}(\mathbf{M}_x)}{t} + \mathbf{e}' \\
&= \mathbf{s} \cdot \left(\frac{\mathbf{A}_x \mathbf{G}_q^{-1}(\mathbf{M}_x)}{t} - f(x)\mathbf{M}_x - \mathbf{E} \right) + \mathbf{e}' \\
&= \mathbf{s} \frac{\mathbf{A}_x \mathbf{G}_q^{-1}(\mathbf{M}_x)}{t} - f(x)\mathbf{s}\mathbf{M}_x + \underbrace{-\mathbf{s}\mathbf{E} + \mathbf{e}'}_{\mathbf{e}''}
\end{aligned}$$

where $\|\mathbf{e}''\|_\infty \leq \mathrm{poly}(\lambda, B, (n\lceil \log q \rceil)^{O(d')})$. □

Theorem 5.5 (Correctness, Pseudorandomness, Constraint Hiding)**.** *Under the* $\mathrm{DLWE}_{n,q,\chi}$ *hardness assumption,* CPRF *is correct, pseudorandom and constraint hiding.*

Proof. Let $\mathcal{A}$ be a PPT adversary against CPRF and consider the game from Definiton 3.2. The proof proceeds with a sequence of hybrids.

Hybrid $\mathcal{H}_0$. The game from the definition.

Hybrid $\mathcal{H}_1$. Change the way that the vectors $\vec{\mathbf{a}}$ and $\vec{\mathbf{b}}$ are computed in $\mathsf{Constrain}$ and Keygen respectively: Define the matrices $\widehat{\mathbf{A}} = \vec{\mathbf{A}} - \tilde{f} \otimes \mathbf{G}_q$ and $\widehat{\mathbf{B}} = \vec{\mathbf{B}} - \mathsf{sk} \otimes \mathbf{G}_q$. Then let $\vec{\mathbf{a}} = \mathbf{s}\widehat{\mathbf{A}} + \mathbf{e}_a$ and $\vec{\mathbf{b}} = \mathbf{s}\widehat{\mathbf{B}} + \mathbf{e}_b$ where $\mathbf{e}_a \xleftarrow{\$} \chi^{m\ell_p}$, $\mathbf{e}_b \xleftarrow{\$} \chi^{m\ell_s}$. This is simply a change in notation.

Hybrid $\mathcal{H}_2$. Change the Eval algorithm. Up to this hybrid, in Eval we computed $\mathbf{M}_x = \mathbf{D}\mathbf{G}_p^{-1}(\mathbf{C}_x)$ and the output was

$$y = \left\lfloor \mathbf{s} \cdot \frac{\mathbf{A}_x \mathbf{G}_q^{-1}(\mathbf{M}_x)}{t' \cdot t} \right\rceil .$$

Consider the vector $\mathbf{d} = \mathbf{s}\mathbf{D} + \mathbf{e}_d$ where $\mathbf{e}_d \leftarrow \chi^{n\lceil \log p \rceil}$. In this hybrid the output of Eval will be

$$y^* = \left\lfloor \frac{\mathbf{v}}{t'} \right\rceil \qquad \text{where} \qquad \mathbf{v} = \frac{\mathbf{a}_{f,x}\mathbf{G}_q^{-1}(\mathbf{M}_x)}{t} + f(x)\left(\mathbf{d}\mathbf{G}_p^{-1}(\mathbf{C}_x) + E(x)\right)$$

and $E(\cdot)$ is the function from Corollary 3.10, and in particular $|E(x)| \le B\sqrt{k} \cdot (n\lceil \log p \rceil)^{\log k}$.

We analyse now the event that $y^* \neq y$. Note that

$$\mathbf{d}\mathbf{G}_p^{-1}(\mathbf{C}_x) = \mathbf{s}\underbrace{\mathbf{D}\mathbf{G}_p^{-1}(\mathbf{C}_x)}_{\mathbf{M}_x} + \underbrace{\mathbf{e}_d\mathbf{G}_p^{-1}(\mathbf{C}_x)}_{\mathbf{e}} = \mathbf{s}\mathbf{M}_x + \mathbf{e} \qquad \|\mathbf{e}\|_\infty \le B \cdot n\lceil \log p \rceil$$

By Lemma 5.4,

$$\frac{\mathbf{a}_{f,x}\mathbf{G}^{-1}(\mathbf{M}_x)}{t} = \mathbf{s} \cdot \frac{\mathbf{A}_x\mathbf{G}^{-1}(\mathbf{M}_x)}{t} - f(x)\mathbf{s}\mathbf{M}_x + \mathbf{e}'' \qquad \left\|\mathbf{e}''\right\|_\infty \le \mathrm{poly}(\lambda, B, (n\lceil \log q \rceil)^{O(d')})$$

Hence

$$\begin{aligned}
y &= \left\lfloor \mathbf{s} \cdot \frac{\mathbf{A}_x\mathbf{G}^{-1}(\mathbf{M}_x)}{t' \cdot t} \right\rceil \\
&= \left\lfloor \frac{1}{t'}\left(\frac{\mathbf{a}_{f,x}\mathbf{G}^{-1}(\mathbf{M}_x)}{t} + f(x)\mathbf{s}\mathbf{M}_x - \mathbf{e}'' \right) \right\rceil \\
&= \left\lfloor \frac{1}{t'}\left(\frac{\mathbf{a}_{f,x}\mathbf{G}^{-1}(\mathbf{M}_x)}{t} + f(x)\left(\mathbf{d}\mathbf{G}_p^{-1}(\mathbf{C}_x) - \mathbf{e}\right) - \mathbf{e}'' \right) \right\rceil \\
&= \left\lfloor \frac{1}{t'}\left(\frac{\mathbf{a}_{f,x}\mathbf{G}^{-1}(\mathbf{M}_x)}{t} + f(x)\left(\mathbf{d}\mathbf{G}_p^{-1}(\mathbf{C}_x) + E(x)\right) - \underbrace{\left(f(x)E(x) + f(x)\mathbf{e} + \mathbf{e}''\right)}_{\mathbf{e}'''} \right) \right\rceil \\
&= \left\lfloor \frac{1}{t'}\left(\mathbf{v} - \mathbf{e}'''\right) \right\rceil
\end{aligned}$$

where $\|\mathbf{e}'''\|_\infty$ is bounded by a value $E' = \mathrm{poly}(\lambda, B, (n\lceil \log q \rceil)^{O(d')}, B\sqrt{k} \cdot (n\lceil \log p \rceil)^{\log k})$. Therefore $y^* \neq y$ only when there exists $i \in [n\lceil \log p \rceil]$ such that the ith entry of the vector $\mathbf{v}$ is E'-close to $t'\mathbb{Z} + t'/2$, i.e. when the ith entry of the vector $t\mathbf{v}$ is tE'-close to $(t \cdot t')\mathbb{Z} + (t \cdot t')/2$. Let $\mathrm{Borderline}_x$ denote this event, then $\neg\mathrm{Borderline}_x \implies y^* = y$. We can bound the advantage in distinguishing between this hybrid and the previous one by the probability of $\mathrm{Borderline} = \bigvee_x \mathrm{Borderline}_x$:

$$|\mathrm{Adv}_{\mathcal{H}_2}(\mathcal{A}) - \mathrm{Adv}_{\mathcal{H}_1}(\mathcal{A})| \le \Pr_{\mathcal{H}_2}[\mathrm{Borderline}]$$

Lemma 5.6. *The following holds:*

$$\Pr\Big[\bigvee_{x\in\{0,1\}^k} \text{Borderline}_x\Big] \leq n\lceil\log p\rceil 2^k E'/t' = \text{negl}(\lambda)\ , \tag{12}$$

where the probability is over the randomness of the key generation algorithm in $\mathcal{H}_2$.

Proof. Fix an arbitrary value for x and some coordinate $i \in [n\lceil\log p\rceil]$ and note that

$$t\mathbf{v} = \mathbf{a}_{f,x}\mathbf{G}_q^{-1}(\mathbf{M}_x) + f(x)t\left(\mathbf{d}\mathbf{G}_p^{-1}(\mathbf{C}_x) + E(x)\right)$$

where $\mathbf{a}_{f,x} = [\vec{\mathbf{a}}\|\vec{\mathbf{b}}]\mathbf{H}_{f,x} = \mathbf{s}[\widehat{\mathbf{A}}\|\widehat{\mathbf{B}}]\mathbf{H}_{f,x} + [\mathbf{e}_a\|\mathbf{e}_b]\mathbf{H}_{f,x}$. Recall that $\|\mathbf{s}\|_\infty \leq B < t < p$, where p, t are prime and $q = p \cdot t$, so each entry of $\mathbf{s}$ is a unit in $\mathbb{Z}_q$. Simillarly, $\left\|\mathbf{H}_{f,x}\mathbf{G}_q^{-1}(\mathbf{M}_x)\right\| \leq (n\lceil\log q\rceil)^{O(d')} < t \leq p$ and so each entry of $\mathbf{H}_{f,x}\mathbf{G}_q^{-1}(\mathbf{M}_x)$ is a unit in $\mathbb{Z}_q$.

Since $[\widehat{\mathbf{A}}\|\widehat{\mathbf{B}}]$ is uniform over $\mathbb{Z}_q^{n\times m(\ell_p+\ell_s)}$, it follows that each entry of $\mathbf{s}[\widehat{\mathbf{A}}\|\widehat{\mathbf{B}}]\mathbf{H}_{f,x}\mathbf{G}_q^{-1}(\mathbf{M}_x)$ is uniform over $\mathbb{Z}_q$ and so the marginal distribution of the ith entry of $t\mathbf{v}$ as a function of the randomness of Keygen is uniform over $\mathbb{Z}_q$. Therefore, the probability of this value being tE'-close to $(t\cdot t')\mathbb{Z} + (t\cdot t')/2$ is at most E'/t'. Applying the union bound over all possible values of x and i, the lemma follows. □

Note that in this hybrid, if $f(x) = 0$ then the output of Eval is identical to the output of $\mathsf{ConstrainEval}$, so the adversary has no advantage in guessing b_3.

Hybrid $\mathcal{H}_3$. Change $\mathbf{d}$: sample it uniformly from $\mathbb{Z}_p^{n\lceil\log p\rceil}$. This change is computationally indistinguishable under $\text{DLWE}_{n,p,\chi}$.

Hybrid $\mathcal{H}_4$. Change again Eval: compute $\mathbf{v}$ by first sampling a vector $\mathbf{u}_x \xleftarrow{\$} \mathbb{Z}_p^m$ and setting

$$\mathbf{v} = \frac{\mathbf{a}_{f,x}\mathbf{G}_q^{-1}(\mathbf{M}_x)}{t} + f(x)\mathbf{u}_x\ .$$

Recall that the adversary can query each distinct x once. By Corollary 3.10, those hybrids are indistinguishable under $\text{DLWE}_{n,p,\chi}$.

In this hybrid, if $f(x) = 1$ then the output of Eval is uniformly distributed over $\mathbb{Z}_p^m$, so the adversary has no advantage in guessing b_2.

Hybrid $\mathcal{H}_5$. Change $\mathsf{Constrain}$: compute $\tilde{f}$ as $\tilde{f} \leftarrow \text{FHE.}\mathsf{Enc}(\mathsf{sk}, 0)$. By Lemma 3.9, those hybrids are indistinguishable under $\text{DLWE}_{n,q,\chi}$. At this stage the adversary has no information about f and therefore it has no advantage in guessing b_1, which completes the proof. □

Choice of parameters. In order to satisfy the requirements in the above proof, we require that $n\lceil\log p\rceil 2^k E'/t' = \text{negl}(\lambda)$. For the sake of concreteness, we will set $\text{negl}(\lambda)$ to $2^{-\lambda}$. Recalling that $E' = \text{poly}(\lambda, B, (n\lceil\log q\rceil)^{O(d')}, B\sqrt{k}\cdot(n\lceil\log p\rceil)^{\log k})$, we get $t' \geq 2^{O(\lambda+k+(d+\log k)\cdot\text{polylog}(n))}$. This can be satisfied by setting $n = (\lambda k d)^{1/\epsilon}$ and setting $t' = 2^{\tilde{O}(n^\epsilon)}$ appropriately. Then p, t can be chosen to be polynomially related in size to t' s.t. $t, t', p/t'$ are prime.

References

[ABB10a] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, pages 553–572, 2010.

[ABB10b] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *CRYPTO*, pages 98–115, 2010.

[ABCP15] Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Simple functional encryption schemes for inner products. In Jonathan Katz, editor, *Public-Key Cryptography - PKC 2015 - 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30 - April 1, 2015, Proceedings*, volume 9020 of *Lecture Notes in Computer Science*, pages 733–751. Springer, 2015.

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, pages 595–618, 2009.

[AFV11] S. Agrawal, D. M. Freeman, and V. Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In *ASIACRYPT*, 2011.

[Agr16] Shweta Agrawal. Interpolating predicate and functional encryption from learning with errors. *IACR Cryptology ePrint Archive*, 2016:654, 2016.

[AGVW13] S. Agrawal, S. Gorbunov, V. Vaikuntanathan, and H. Wee. Functional encryption: New perspectives and lower bounds. CRYPTO, 2013.

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In *CRYPTO*, pages 308–326, 2015.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108, 1996.

[ALS16] Shweta Agrawal, Benoît Libert, and Damien Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, pages 333–362, 2016.

[AP14] Jacob Alperin-Sheriff and Chris Peikert. Faster bootstrapping with polynomial error. In Garay and Gennaro [GG14], pages 297–314.

[BCTW16] Zvika Brakerski, David Cash, Rotem Tsabary, and Hoeteck Wee. Targeted homomorphic attribute-based encryption. In *TCC*, pages 330–360, 2016.

[BFP+15] Abhishek Banerjee, Georg Fuchsbauer, Chris Peikert, Krzysztof Pietrzak, and Sophie Stevens. Key-homomorphic constrained pseudorandom functions. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, pages 31–60, 2015.

[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 *EUROCRYPT*, pages 533–556, 2014.

[BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings*, pages 501–519, 2014.

[BGV12] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, 2012.

[BJK15] Allison Bishop, Abhishek Jain, and Lucas Kowalczyk. Function-hiding inner product encryption. In *ASIACRYPT*, pages 470–491, 2015.

[BKM17] Dan Boneh, Sam Kim, and Hart William Montgomery. Private puncturable prfs from standard lattice assumptions. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, pages 415–445, 2017.

[BKS16] Zvika Brakerski, Ilan Komargodski, and Gil Segev. Multi-input functional encryption in the private-key setting: Stronger security from weaker assumptions. In *EUROCRYPT*, pages 852–880, 2016.

[BLMR15] Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key homomorphic prfs and their applications. *IACR Cryptology ePrint Archive*, 2015:220, 2015.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Boneh et al. [BRF13], pages 575–584.

[BLW15] Dan Boneh, Kevin Lewi, and David J. Wu. Constraining pseudorandom functions privately. *IACR Cryptology ePrint Archive*, 2015:1167, 2015.

[BLW17] Dan Boneh, Kevin Lewi, and David Wu. Constraining pseudorandom functions privately. In *Public Key Cryptography (PKC) Conference*, 2017. To appear.

[BP14] Abhishek Banerjee and Chris Peikert. New and improved key-homomorphic pseudorandom functions. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 353–370, 2014.

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, pages 719–737, 2012.

[BRF13] Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors. *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*. ACM, 2013.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In *TCC*, pages 253–273, 2011.

[BV11] Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. In *FOCS*, 2011.

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Lattice-based FHE as secure as PKE. In Moni Naor, editor, *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pages 1–12. ACM, 2014.

[BV15a] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 171–190. IEEE Computer Society, 2015.

[BV15b] 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, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 1–30. Springer, 2015.

[BW07] Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In Salil P. Vadhan, editor, *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, volume 4392 of *Lecture Notes in Computer Science*, pages 535–554. Springer, 2007.

[BW13] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II*, pages 280–300, 2013.

[CC17] Ran Canetti and Yilei Chen. Constraint-hiding constrained prfs for nc^1 from lwe. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, pages 446–476, 2017.

[CHKP12] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *J. Cryptology*, 25(4):601–639, 2012.

[Gay16] Romain Gay. Functional encryption for quadratic functions, and applications to predicate encryption. *IACR Cryptology ePrint Archive*, 2016:1106, 2016.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178. ACM, 2009.

[GG14] Juan A. Garay and Rosario Gennaro, editors. *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*. Springer, 2014.

[GGH+16] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM J. Comput.*, 45(3):882–929, 2016.

[GKP+13] Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In *STOC*, pages 555–564, 2013.

[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, *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006*, pages 89–98. ACM, 2006.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 197–206. ACM, 2008.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92. Springer, 2013.

[GVW12] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In *CRYPTO*, pages 162–179, 2012.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Boneh et al. [BRF13], pages 545–554.

[GVW15a] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 503–523, 2015.

[GVW15b] Sergey Gorbunov, Vinod Vaikuntanathan, and Daniel Wichs. Leveled fully homomorphic signatures from standard lattices. In Rocco A. Servedio and Ronitt Rubinfeld,

editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 469–477. ACM, 2015.

[HKKW14] Dennis Hofheinz, Akshay Kamath, Venkata Koppula, and Brent Waters. Adaptively secure constrained pseudorandom functions. Cryptology ePrint Archive, Report 2014/720, 2014. http://eprint.iacr.org/.

[KPTZ13] Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, pages 669–684. ACM, 2013.

[KSW08] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 146–162. Springer, 2008.

[Lin16] Huijia Lin. Indistinguishability obfuscation from constant-degree graded encoding schemes. *IACR Cryptology ePrint Archive*, 2016:257, 2016.

[MM11] Daniele Micciancio and Petros Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, pages 465–484, 2011.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, pages 700–718, 2012.

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

[OT12] Tatsuaki Okamoto and Katsuyuki Takashima. Adaptively attribute-hiding (hierarchical) inner product encryption. In *EUROCRYPT*, pages 591–608, 2012. Also, Cryptology ePrint Archive, Report 2011/543.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 333–342, 2009.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 84–93, 2005.

[Sch87] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987.

[SW05] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, 2005.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 475–484. ACM, 2014.