Multi-Input Correlation-Intractable Hash Functions via Shift-Hiding

Alex Lombardi* Vinod Vaikuntanathan†

November 3, 2020

Abstract

A hash function family $\mathcal{H}$ is correlation intractable for a t-input relation $\mathcal{R}$ if, given a random function h chosen from $\mathcal{H}$, it is hard to find $x_1, \ldots, x_t$ such that $\mathcal{R}(x_1, \ldots, x_t, h(x_1), \ldots, h(x_t))$ is true. Recent works have constructed correlation-intractable hash families for *single-input* relations from standard cryptographic assumptions. However, the case of multi-input relations (even for $t = 2$) is wide open: there are two known constructions, the first of which relies on a very strong "brute-force-is-best" type of hardness assumption (Holmgren and Lombardi, FOCS 2018); and the second only achieves the much weaker notion of output intractability (Zhandry, CRYPTO 2016).

Our main result is the construction of several *multi-input* correlation intractable hash families for large classes of interesting *input-dependent* relations from either the learning with errors (LWE) assumption or from indistinguishability obfuscation. Our constructions follow from a simple and modular approach to constructing correlation-intractable hash functions using shift-hiding shiftable functions (Peikert-Shiehian, PKC 2018). This approach also gives an alternative framework (as compared to Peikert-Shiehian, CRYPTO 2019) for achieving single-input correlation intractability (and NIZKs for NP) based on LWE.

---

*MIT. Email: `alexjl@mit.edu`. Research supported in part by an NDSEG fellowship. Research supported in part by NSF Grants CNS-1350619 and CNS-1414119, and by the Defense Advanced Research Projects Agency (DARPA) and the U.S. Army Research Office under contracts W911NF-15-C-0226 and W911NF-15-C-0236.

†MIT. Email: `vinodv@mit.edu`. Research supported in part by NSF Grants CNS-1350619 and CNS-1414119, and by the Defense Advanced Research Projects Agency (DARPA) and the U.S. Army Research Office under contracts W911NF-15-C-0226 and W911NF-15-C-0236.

Contents

1 Introduction

The random oracle model [BR94] is a powerful and controversial paradigm in cryptography where the proof of security of a cryptographic scheme assumes that a certain *publicly computable* function H that is used in the scheme behaves like a random function to the adversary. The random oracle model is hugely influential in designing concretely efficient cryptosystems, but is inherently problematic theoretically: how *could* a *public*, and therefore completely predictable, function behave in all aspects like a random function? Indeed, Canetti, Goldreich and Halevi [CGH98] came up with cryptographic schemes that one could prove secure in the random oracle model, but which are insecure no matter how one tries to instantiate the oracle with a concrete function (or even a function chosen at random from an exponential-size family). Nevertheless, this negative result and the notions introduced therein led to a long line of research that asked *what concrete properties* of a random oracle *are* instantiable in the standard model (see, e.g., [CMR98] for an early work in this direction), and opened the door to groundbreaking *positive* results two decades later [CCR16, KRR17, CCRR18, HL18, CCH+19, PS19].

The key notion introduced in [CGH98] is that of *correlation intractability*, which captures a general and powerful form of cryptographic hardness for a hash function family $\mathcal{H}$. For any relation $R(x_1, \ldots, x_t, y_1, \ldots y_t)$, a hash family $\mathcal{H}$ is *correlation-intractable for* R if it is computationally hard (given a hash function $h \leftarrow \mathcal{H}$) to find inputs $x_1, \ldots, x_t$ such that $(x_1, \ldots, x_t, h(x_1), \ldots, h(x_t)) \in R$. For this definition to make sense, we require that the relation R be sparse, that is, for any $\mathbf{x} = (x_1, \ldots, x_t)$, *all but a negligible fraction of* $\mathbf{y} = (y_1, \ldots, y_t)$ do *not* satisfy the relation with $\mathbf{x}$. Thus, if H were a random oracle, it would be correlation-intractable for *any sparse* R. On the other hand, [CGH98] showed that there exist sparse relations R for which no family $\mathcal{H}$ is correlation-intractable.

Let us examine the [CGH98] in more detail. What they show is that for every hash function family $\mathcal{H}$, there exists a sufficiently large $t = t(\lambda)$ along with a t-input (sparse) relation R for which $\mathcal{H}$ fails to be correlation-intractable (CI). This suggests a way to circumvent the negative result: flip the quantifiers! That is, perhaps for every *fixed* $t = t(\lambda)$, there is a hash family $\mathcal{H}$ that is CI with respect to t-input relations. To the best of our knowledge, with this relaxation, correlation intractability is potentially achievable in its full generality.

Correlation intractability for relations with a fixed number of inputs is very useful. Indeed, if there were a CI family for even (the class of all) *single-input* relations, that would suffice to instantiate the storied Fiat-Shamir transform [FS87] in the standard model [DNRS99, CCRR18] for any constant-round public-coin interactive proof system. The Fiat-Shamir transform is a generic transformation from a public-coin interactive proof (or argument) system into a non-interactive one using a hash function to compute the verifier's challenges. The challenge with such a transformation is to show that the resulting non-interactive protocol remains sound (indeed, by reducing interaction, we are taking away the leverage from the verifier). The soundness of the Fiat-Shamir transform was shown in the random oracle model [BR94, PS96, BCS16], but the question of whether – for *any* specific protocol of interest – soundness can be achieved in the plain model by instantiating the hash family appropriately was open for a long time.

In large part due to the Fiat-Shamir connection, single-input CI has been the subject of much recent work [CCR16, KRR17, CCRR18, HL18, CCH+19, PS19, BKM20, LV20]. In particular, this line of work has led to new *feasibility results* for correlation intractability for a wide class of relations. Most notably, [CCH+19, PS19] construct (under standard cryptographic assumptions) a hash family $\mathcal{H}$ that is CI for relations R defined by any (a priori-bounded) poly-size circuits C, where $R(x, y) = 1$

iff $y = C(x)$. In other words, the hash family $\mathcal{H}$ has the property that for any efficiently computable function f, given $h \leftarrow \mathcal{H}$, it is provably hard to find an input x such that $h(x) = f(x)$.

Multi-Input Correlation Intractability. In contrast to the single-input case, *multi-input* correlation intractability (for any $t \geq 2$) is a far less well-understood primitive. Perhaps the simplest interesting example of multi-input CI is for the relation R where $R(x_1, x_2, y_1, y_2) = 1$ if and only if $y_1 = y_2$ but $x_1 \neq x_2$. A CI hash family for R is precisely a collision-resistant hash family. CI for more general multi-input relations also has interesting applications, including:

1. As a useful tool for the *untrusted setup* of public parameters [CCR16, Zha16]: Multi-input CI hash functions allow n parties $P_1, \ldots, P_n$ with inputs $x_1, \ldots, x_n$ to compute public outputs $y_i = H(x_i)$ that can be used to generate public parameters for a multi-party protocol. Correlation intractability of H is necessary to ensure that a "bad CRS" is not accidentally (or maliciously) agreed on.

2. As a hash function in proof-of-work protocols [CCR16, CCRR18]: In the bitcoin protocol [Nak08], a miner succeeds in adding a block to the blockchain when she finds an x such that $y = H(x||B_i)$ starts with a specified number of zeroes (here, B_i is the i-th block and once found, y is placed in the next block B_{i+1}). A very desirable property in this setting is that a single miner (or collection of colluding miners) cannot find *multiple consecutive blocks* with significantly less effort than finding them sequentially. This property can be formalized as a quantitatively precise variant of multi-input CI. For example, in the case of two consecutive blocks, simplifying the setting a little, we require a 2-input CI for the relation R where $R(x_1, x_2, y_1, y_2) = 1$ iff y_1 and y_2 start with a pre-specified number ℓ of zeroes, and y_1 is a suffix of x_2.

Unfortunately, multi-input CI has so far proved hard to achieve. In particular, the constructions of [CCR16, KRR17, CCRR18, CCH+19, PS19, BKM20] are only known to achieve single-input CI. Holmgren and Lombardi [HL18] do achieve multi-input CI for a large class of relations that they call *locally sampleable* relations. However, they require both an indistinguishability obfuscation (iO) scheme [BGI+01] as well as an "optimally-secure" one-way product function [HL18]. While iO can now be achieved under relatively standard assumptions [BDGM20, GP20, JLS20], the latter is a very strong "brute force is optimal"-type assumption. Zhandry [Zha16] constructed a hash family satisfying a very special form of multi-input CI called "output intractability". Output intractability is a form of CI for relations $R(x_1, \ldots, x_t, y_1, \ldots, y_t)$ that depend only on the y_i, which captures some variants of application (1) above. On the plus side, the construction is based on the exponential hardness of the Diffie-Hellman problem.[1] To sum up, multi-input CI is either known for a small class of relations under standard assumptions, or for a larger class of relations under very strong assumptions. We refer the reader to Section 1.2 for more details and further comparisons.

We remark that constructing (single- or multi-input) CI hash functions even assuming indistinguishability obfuscation is far from straightforward. Indeed, the initial works [CCR16, KRR17, HL18] in this line all made non-standard assumptions *in addition to iO*. Non-standard assumptions were required until the work of [CCH+19] which constructed single-input CI hash functions under circular-secure LWE. However, they only managed to do this for a tiny subset of relations

[1] Moreover, given an inverse-subexponential lower bound on the sparsity of the relation, Zhandry's construction is secure under (the more standard) sub-exponential DDH.

that [CCR16, KRR17] achieved. In particular, replicating the results of [KRR17] or even [CCR16] assuming *only iO* (plus standard assumptions) is a challenging open problem.

1.1 Our Results and Techniques

Our main result is a new framework for constructing (single- and) multi-input correlation-intractable (CI) hash functions using a cryptographic primitive called *shift-hiding shiftable functions* (SHSFs) [PS18], a twist on private constrained pseudorandom functions [BW13, BGI14, KPTZ13]. A SHSF family is a PRF family $\{F_{\mathsf{msk}}\}$ that additionally supports the ability to *delegate* a constrained key sk_f that computes the function $F_{\mathsf{msk}}(x)+f(x)$, *without* revealing the "shift function" f. Shift-hiding shiftable functions were originally introduced for the purpose of constructing private pseudorandom functions, but have since found several other applications [PS20, DVW20].

Our main technique is a *lifting theorem* that allows us to construct multi-CI hash functions for complex relations starting from multi-CI hash functions for simpler relations (captured by output intractability). This gives us two new constructions of multi-CI hash functions under different assumptions:

- Our first construction considers the shifted linear relation

$$\mathcal{R}_{\mathsf{lin}} = \{(x_1, \ldots, x_t, y_1, \ldots, y_t) : \sum w_i y_i = \sum w_i f(x_i) \pmod p\}$$

  where p is some large integer, w_i are small weights and f is an arbitrary polynomial-time computable functions. We construct a multi-input CI hash function for $\mathcal{R}_{\mathsf{lin}}$ under standard lattice assumptions, namely learning with errors (LWE) and 1D-short integer solutions (1D-SIS) [BV15]. We note that in the special case $t = 1$, our security notion is equivalent to CI for efficient functions [CCH+19, PS19]; we therefore give an alternative construction to [PS19] for achieving single-input CI (and NIZKs for NP) based on standard lattice assumptions.

- Our second construction considers the shifted general relation

$$\mathcal{R} = \{(x_1, \ldots, x_t, y_1, \ldots, y_t) : \mathcal{R}_0(y_1 - f(x_1), \ldots, y_i - f(x_i)) = 1\}$$

  where $\mathcal{R}_0$ is any polynomial-time decidable relation. We construct a multi-input CI hash function for $\mathcal{R}$ under subexponential iO, subexponential OWFs, and (sufficiently) lossy functions.

Our constructions are rather simple, indeed simple enough that we will be able to describe them in the following introductory overview.

1.1.1 Single-Input CI.

We give an overview of our technical ideas starting from the simplest setting of single-input CI. The interesting feature of our description below is that all the ideas transparently generalize to the multi-input setting, as we detail later.

We start with a simple theorem which states that any SHSF family (for a function class $\mathcal{F}$) satisfying a *very weak* form of correlation intractability can be used to construct a hash family satisfying a stronger form of correlation intractability.

Theorem 1.1 (Informal)**.** *Suppose that* $\mathsf{SHSF} = \{F_{\mathsf{msk}}\}$ *is a family of SHSFs for a function class* $\mathcal{F}$*, and suppose that* F_{msk} *satisfies a* one-wayness *property: given* msk*, it is hard to find (an element in)* $F_{\mathsf{msk}}^{-1}(0)$*. Then,* SHSF *can be used to construct a hash family* $\mathcal{H}$ *that is correlation-intractable for the relation* $R_f(x, y) = 1$ *iff* $y = f(x)$*, for all functions* $f \in \mathcal{F}$*.*

The hash function is extremely simple. Hash keys are shifted keys $\mathsf{sk}_{\mathcal{Z}}$ for the all-zero function $\mathcal{Z}$. Indeed, the hash function evaluation is simply the shifted evaluation using $\mathsf{sk}_{\mathcal{Z}}$ which computes exactly the function F_{msk}. (Philosophically, the CI hash family constructed in this theorem is a form of "obfuscated PRF evaluation" although shift-hiding functions are decidedly more complex to construct than PRFs.) The proof of Theorem 1.1 is also simple.

Proof Sketch. If an adversary $\mathcal{A}$, given a hash key $\mathsf{sk}_{\mathcal{Z}}$, finds an input x such that $F_{\mathsf{sk}_{\mathcal{Z}}}(x) = f(x)$, then by the shift-hiding property of SHSF, $\mathcal{A}$ also produces such an x when given sk_f instead of $\mathsf{sk}_{\mathcal{Z}}$. In that case, $\mathcal{A}$ solves the equation

$$f(x) = F_{\mathsf{sk}_f}(x) = F_{\mathsf{msk}}(x) + f(x),$$

which is equivalent to the equation $F_{\mathsf{msk}}(x) = 0$. This yields a 0-inversion attack on F_{msk}, which is hard by assumption. □

We note that Theorem 1.1 could be proved under a weaker one-wayness assumption, namely, that *it is hard to find an input* x *such that* $F_{\mathsf{msk}}(x) = 0$*, given a shifted key* sk_f *for any pre-specified* f" (as opposed to being given msk in the clear). However, we phrase Theorem 1.1 under the assumption that F_{msk} is one-way (given msk in the clear) because this is a clean, f-independent security property. Moreover, in our instantiations below, we are able to prove the stronger one-wayness property of F_{msk}.

Given Theorem 1.1, it remains to construct an SHSF family satisfying this one-wayness property. We show that a modified variant of the Peikert-Shiehian SHSF [PS18] satisfies this.

Theorem 1.2 (Informal)**.** *Assuming the hardness of standard lattice problems (LWE and 1D-SIS), there exists an SHSF family that is one-way (as above).*

We now sketch our proof, at a very high level and assuming some knowledge of LWE-based cryptography.

Proof Sketch. In the Peikert-Shiehian SHSF construction, $\mathsf{msk} = \mathbf{s}$, an LWE secret, and

$$F_{\mathsf{msk}}(x) = \lfloor \mathbf{s}\mathbf{A}_x \rceil_p$$

where $\mathbf{A}_x$ is a *matrix with small entries* constructed out of matrices $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$ using the gadget homomorphisms from [BGG+14], and $\lfloor \cdot \rceil_p$ denotes the rounding operation that, roughly speaking, keeps the top $\log p$ bits of the argument and discards the rest.

If the adversary finds an x such that $F_{\mathsf{msk}}(x) = 0$, there are two cases; the easier case is when $\mathbf{A}_x$ is non-zero. In that case, we have an (approximate) subset sum (SIS) solution w.r.t. the instance $\mathbf{s}$, that is, $\mathbf{s}\mathbf{A}_x \approx \mathbf{0}$. This is as hard as SIS and therefore worst-case lattice problems.

The harder case is when the adversary finds an x such that $\mathbf{A}_x = 0$. We show that the adversary cannot make this happen without violating SIS (again!) Roughly speaking, we use the fact that if we *program* the matrices $\mathbf{A}_i = \mathbf{A}\mathbf{R}_i + h_i\mathbf{G}$ where $\mathbf{R}_i$ are matrices with small entries, $\mathbf{G}$ is the gadget matrix, and h is the description of some function, the following equation holds due to the gadget homomorphisms of Boneh et al. [BGG+14]:

$$\mathbf{G}\mathbf{A}_x = \mathbf{A}\mathbf{R}_x + h(x)\mathbf{G}$$

for some $\mathbf{R}_x$ that is a function of $\mathbf{R}_1, \ldots, \mathbf{R}_\ell$. We know by assumption that $\mathbf{G}\mathbf{A}_x = 0$. We argue that if h is "random-enough" function, it is statistically unlikely that $h(x) = 0$ as the programming hides h. (For the random enough function, in this case, a shift by a random vector suffices; further down in the intro, in the t-input setting, we will need to use a t-wise independent function). This means that the adversary found a solution $\mathbf{R}_x$ to the (inhomogenous) SIS problem w.r.t. $\mathbf{A}$, which is hard assuming that worst-case lattice problems are hard. This finishes the proof of one-wayness. □

Combining Theorem 1.2 with Theorem 1.1, we already recover a similar result to [PS19]. That is, assuming the hardness of LWE (and 1D-SIS), there exists a hash family that is correlation-intractable for all bounded-size functions. By appealing to [CCH+19], this also gives a lattice-based NIZK argument system for NP. Our construction is very different from and appears to be conceptually simpler than that of [PS19]. Furthermore, as we describe below, this technique directly generalizes to the multi-input setting, giving us *new* feasibility results.

1.1.2 Multi-Input CI.

One consequence of our shift-hiding technique is a collection of feasibility results for multi-input correlation intractability based on standard assumptions. We obtain two flavors of results: constructions from standard (lattice) assumptions, and constructions from indistinguishability obfuscation.

For any output-only relation $\mathcal{R}_0$, we say that a hash family $\mathcal{H}$ is $\mathcal{R}_0$-output intractable if it is hard (given h) to find distinct[2] inputs $x_1, \ldots, x_t$ such that $(y_1, \ldots, y_t) \in \mathcal{R}_0$ for $y_i = h(x_i)$. Output intractability as a standalone property (like collision-resistance) is known to be instantiable based on standard cryptographic assumptions (e.g., lossy functions [PW08]) as we discuss in Section 1.2. We show that *SHSFs that are output-intractable* lead to interesting new CI constructions.

Theorem 1.3. *Suppose that* SHSF *is a shift-hiding shiftable function family. Assume that it is hard, given* msk*, to find distinct* $x_1, \ldots, x_t$ *such that* $\mathcal{R}_0(y_1, \ldots, y_t) = 1$ *where* $y_i = F_{\mathsf{msk}}(x_i)$ *and* $\mathcal{R}_0$ *is some polynomial-time computable relation. Then, there is a CI hash family for the shifted output relation*

$$\mathcal{R} = \{(x_1, \ldots, x_t, y_1, \ldots, y_t) : \mathcal{R}_0(y_1 - f(x_1), \ldots, y_i - f(x_i)) = 1\}$$

The proof of Theorem 1.3 follows from that of the single-input CI case, namely Theorem 1.1, *mutatis mutandis*. All that remains is to construct SHSFs that are *output-intractable*. We show two constructions.

Instantiation from LWE. To obtain a construction of multi-input CI from LWE, we combine Theorem 1.3 with a generalization of Theorem 1.2:

[2] For the relation $\sum_i w_i y_i = 0$ implicitly described above, it is enough to assume that the inputs x_i are not all equal for the relation to be sparse. We elaborate on this weakening of output intractability as compared to [Zha16, HL18] in Section 2.

Theorem 1.4. *Under standard lattice assumptions (LWE and 1D-SIS), there exists a SHSF family* SHSF *satisfying the following form of correlation intractability: for every vector $w \in \{-1,0,1\}^t$, it is hard (given* msk*) to find t distinct inputs $x_1,\ldots,x_t$ such that*

$$\sum_i w_i \cdot F_{\mathsf{msk}}(x_i) = 0,$$

where the sum is computed modulo some (large enough) integer p.

Our modification of the Peikert-Shiehian [PS18] construction satisfies this more general form of output intractability (for small linear equations). Note that this is a strict generalization of both single-input CI for functions (where $t = 1, w = 1$) and collision-resistance (where $t = 2, w = (-1,1)$ and f is the constant function). Previously, this form of correlation intractability was only known assuming iO and (extremely hard) one-way product functions [HL18].

Instantiation from IO. Our second construction achieves correlation intractability for shifted $\mathcal{R}_0$-output relations for a large class of $\mathcal{R}_0$ simultaneously (as opposed to linear $\mathcal{R}_0$ as in the LWE case above). It can be thought of as a (non-black-box) combination of our approach with a construction due to Zhandry [Zha16] of output-intractable hash functions.

Theorem 1.5. *Assume the existence of subexponential iO, subexponential OWFs, and lossy functions with input domain $\{0,1\}^n$ with a range of size $\leq 2^\ell$ in lossy mode. Then, there exists a hash family $\mathcal{H}$ that is CI for all (efficiently decidable) shifted t-ary output relations with sparsity at most $2^{-t\ell}$.*

As a corollary, we conclude that additionally assuming the existence of *extremely lossy functions* [Zha16], there is a hash family $\mathcal{H}$ that is CI for all (efficiently decidable) shifted t-ary output relations with sparsity $2^{-\omega(t)}$. As another corollary, we note that by combining Theorem 1.5 with [CCH+19], we obtain a construction of dual-mode NIZKs for NP based on iO, (injective) lossy functions, and lossy encryption. This closely matches the assumptions used in the work [HU19] but with a simpler construction. The corollary follows because the hash family from Theorem 1.5 satisfies "somewhere statistical correlation intractability."

A Separation between Single-Input and Multi-Input CI. Finally, we show that single-input and multi-input CI hash functions are fundamentally different beasts, by showing a separation between them.

Theorem 1.6. *Assume the existence of subexponentially secure indistinguishability obfuscation, subexponentially secure one-way functions, and a hash family $\mathcal{H}$ such that $\mathcal{H}$ is $\mathcal{R}_0$-output intractable, and for a random input X, $h_k(X)$ is 2^{-n}-indistinguishable from uniform (even given k). Then, there exists a hash family that is CI for shifted $\mathcal{R}_0$-relations.*

This theorem says that assuming subexponential iO and one-way functions, shifted-CI for $\mathcal{R}_0$ can be constructed (semi-)generically from output intractability for $\mathcal{R}_0$. Theorem 1.6 is proved by combining Theorem 1.3 with a construction of an $\mathcal{R}_0$-output intractable SHSF using iO, puncturable PRFs, and an output-intractable hash function satisfying the above statistical requirement.

We note that as a corollary to Theorem 1.6, we obtain a construction of single-input CI for all efficient functions from iO and one-way permutations.[3]

Corollary 1.7. *If subexponential iO, subexponential OWFs, and (polynomially-secure) OWPs exist, then there exists a hash family that is CI for all efficient functions, that is, relations $\mathcal{R}(x, y)$ which is true iff $y = f(x)$.*

This construction is notable in that it separates *single-input* correlation intractability (theoretically) from *two-input* correlation intractability: due to an impossibility result of Asharov-Segev [AS15], it is known that there is no (black-box) construction of CRHFs from iO and one-way permutations (even with exponential security). A similar separation was shown in [HL18], but the "positive result" required assuming *optimally hard* one-way functions along with iO to obtain CI for all efficient functions (and more). In contrast, our construction is based on assumptions in the quantitatively standard regime.

1.2 Related Work

Multi-Input Correlation Intractability We summarize what was previously known regarding multi-input correlation intractability:

- For subexponentially sparse output relations $\mathcal{R}_0$, output intractability for $\mathcal{R}_0$ can be constructed based on lossy functions (following [Zha16], but relying on less extreme forms of lossiness). Based on "extremely lossy functions", Zhandry [Zha16] constructs a hash family that is CI for all sparse (efficiently decidable) output relations.[4]
- Similarly to Zhandry [Zha16], the construction $x \mapsto p(H_k(x))$ (where H_k is a sufficiently shrinking collision-resistant hash function and p is sampled from a t-wise independent hash family) also yields output intractability for subexponentially sparse (and efficiently decidable) output relations.
- Holmgren and Lombardi [HL18] construct output-intractable hash functions for all sparse (even inefficient) R based on "one-way product functions" (OWPFs), OWFs satisfying a quantitiatively extreme assumption about the hardness of inverting many one-way function challenges in parallel. OWPFs (in different parameter regimes) are existentially incomparable to lossy functions and CHRFs. Under sufficiently strong assumptions, these hash families achieve quantitiatively better security than is possible for the previous two constructionss.
- Holmgren and Lombardi [HL18] also construct correlation-intractable hash families for relations $R(\mathbf{x}, \mathbf{y})$ that include all shifted output relations. However, they rely on both indistinguishability obfuscation and OWPFs (as above).

[3] As is common [GR13], one must be careful about which definitions of "one-way permutation" suffice for this result. In our proof (which is a proof of concept), we assume that the one-way permutation has domain $\{0, 1\}^n$. It turns out that the proof can be made to work for discrete log-based one-way permutations, but does *not* appear to work for the (trapdoor) permutations constructed based on iO [BPW16].

[4] This is a special case of Zhandry's actual result; we refer the reader to [Zha16] for more details.

Comparison with Peikert-Shiehian [PS19]. [PS19] constructs single-input CI based on the LWE (or SIS) assumption. Their construction improves upon the construction of [CCH+19] based on circular-secure FHE: by making use of special properties of the [GSW13] (and related) FHE schemes, they can remove the need for a circular ciphertext $\mathsf{Enc}(\mathsf{sk}, \mathsf{sk})$ in a specific GSW-based construction. By comparison, we show that any SHSF that is one-way is also CI for bounded functions, and that (essentially) the [PS18] SHSF is one-way. It does not seem easy to abstract out a simple, generic property of the [PS19] hash function that implies correlation intractability.

Given our generalization to multi-input CI, it is also reasonable to ask whether the [PS19] hash function also satisfies a form of multi-input CI. In fact, it appears likely that it satisfies CI for shifted-sum relations (just like our construction). However, a proof of this fact requires some of our analysis in the security proof of our multi-input CI construction (Theorem 1.4).

Comparison with Brakerski-Koppula-Mour [BKM20]. We also note that our construction shares some conceptual similarity to the recent CI construction of [BKM20]. We highlight the similarity here:

- In [BKM20], they show that a hash function $x \mapsto h_k(x) - r$ (for a random r) is CI for a (low-degree) function f by writing down an indistinguishable key distribution k_f so that $h_{k_f}(x) - f(x)$ lies in some sparse set S_f. Then, $h_{k_f}(x) - f(x) = r$ typically has no (information theoretic) solution.
- In our construction, we show that a hash function $x \mapsto h_k(x) - r$ is CI for f by writing down an indistinguishable key distribution k_f so that $h_{k_f}(x) - f(x)$ is the evaluation of a PRF $\mathsf{PRF}_s(x)$. Then, as long as it is computationally hard to find a PRF inverse $F_s^{-1}(r)$ (i.e. as long as F_s is one-way), we can conclude that the equation $h_{k_f}(x) - f(x) = r$ is computationally hard to solve.

2 Preliminaries

Some of the preliminaries below are adapted from [HL18, CCH+19].

2.1 Hash Functions and Correlation Intractability

Definition 2.1. *For a pair of efficiently computable functions $(\nu(\cdot), \mu(\cdot))$, a* hash family *with input length ν and output length μ is a collection $\mathcal{H} = \{h_\lambda : \{0,1\}^{\kappa(\lambda)} \times \{0,1\}^{\nu(\lambda)} \to \{0,1\}^{\mu(\lambda)}\}_{\lambda \in \mathbb{N}}$ of keyed hash functions, along with a pair of p.p.t. algorithms:*

- $\mathcal{H}.\mathsf{Gen}(1^\lambda)$ *outputs a hash key $k \in \{0,1\}^{\kappa(\lambda)}$ describing a hash function h.*
- $\mathcal{H}.\mathsf{Hash}(k, x)$ *computes the function $h_\lambda(k, x) = h(x)$. We may use the notation $h(x)$ to denote hash evaluation when the hash family is clear from context.*

Following [HL18, CCH+19], we consider the security notion of correlation intractability [CGH98] for multi-input relations.

Definition 2.2 (Multi-Input Correlation Intractability). *For a given relation ensemble $R = \{R_\lambda \subseteq (\{0,1\}^{\nu(\lambda)})^{t(\lambda)} \times (\{0,1\}^{\mu(\lambda)})^{t(\lambda)}\}$, a hash family $\mathcal{H} = \{h_\lambda : \{0,1\}^{\kappa(\lambda)} \times \{0,1\}^{\nu(\lambda)} \to \{0,1\}^{\mu(\lambda)}\}$ is said to be R-*correlation intractable with security *(s, δ) if for every s-size adversary $\mathcal{A} = \{\mathcal{A}_\lambda\}$,*

$$\Pr_{\substack{k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda) \\ \mathbf{x}=(x_1,\ldots,x_t) \leftarrow \mathcal{A}(k)}} \Big[\big(\mathbf{x}, \mathbf{y} = (h(x_1), \ldots, h(x_t))\big) \in R\Big] = O(\delta(\lambda)).$$

*We say that $\mathcal{H}$ is R-*correlation intractable with security *δ if it is (λ^c, δ)-correlation intractable for all $c > 1$. Finally, we say that $\mathcal{H}$ is R-*correlation intractable *if it is $(\lambda^c, \frac{1}{\lambda^c})$-correlation intractable for all $c > 1$.*

A random oracle is correlation intractable for relations that are *sparse*, defined as follows:

Definition 2.3 (Sparsity). *A relation ensemble $R = \{R_\lambda \subseteq (\{0,1\}^{\nu(\lambda)})^{t(\lambda)} \times (\{0,1\}^{\mu(\lambda)})^{t(\lambda)}\}$, is $\rho(\lambda)$-sparse if for every $\mathbf{x} \in (\{0,1\}^{\nu(\lambda)})^{t(\lambda)}$,*

$$\Pr_{\mathbf{y} \leftarrow (\{0,1\}^{\mu(\lambda)})^{t(\lambda)}} [(\mathbf{x}, \mathbf{y}) \in R] \leq \rho(\lambda).$$

We say that R is sparse *if it is* $\mathsf{negl}(\lambda)$*-sparse.*

In this work, we focus on *distinct input relations*, i.e., relations R such that for any $(\mathbf{x}, \mathbf{y}) \in R$, we have that $x_i \neq x_j$ for any pair (i, j).

We now describe some special cases of the above definition. Two of them (CI for efficient functions and Output Intractability) have been discussed in prior works [Zha16, HL18, CCH+19, PS19], while a third – which we call "CI for shifted relations" – we introduce in this work.

Definition 2.4 (Correlation Intractability for Functions). *For a given function ensemble $\mathcal{F} = \{f_\lambda : \{0,1\}^{\nu(\lambda)} \to \{0,1\}^{\mu(\lambda)}\}$, a hash family $\mathcal{H} = \{h_\lambda : \{0,1\}^{\kappa(\lambda)} \times \{0,1\}^{\nu(\lambda)} \to \{0,1\}^{\mu(\lambda)}\}$ is said to be f-*correlation intractable *if it is R-correlation intractable for the single-input relation*

$$R = \Big\{(x, f(x)) : x \in \{0,1\}^*\Big\}.$$

Formally, the requirement is that for every poly-size $\mathcal{A} = \{\mathcal{A}_\lambda\}$,

$$\Pr_{\substack{k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda) \\ x \leftarrow \mathcal{A}(k)}} \Big[h(k, x) = f(x)\Big] = \mathsf{negl}(\lambda).$$

Definition 2.5 (Output Intractability). *For a given relation ensemble $R_{\text{out}} = \{R_{\text{out},\lambda} \subseteq (\{0,1\}^{\mu(\lambda)})^{t(\lambda)}\}$, a hash family $\mathcal{H} = \{h_\lambda : \{0,1\}^{\kappa(\lambda)} \times \{0,1\}^{\nu(\lambda)} \to \{0,1\}^{\mu(\lambda)}\}$ is said to be R_{out}-*output intractable *if it is R-correlation intractable for the relation*

$$R = \Big\{(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in R_{\text{out}} \textit{ and } x_i \neq x_j \textit{ for all } i \neq j\Big\}.$$

Formally, the requirement is that for every poly-size $\mathcal{A} = \{\mathcal{A}_\lambda\}$,

$$\Pr_{\substack{k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda) \\ \mathbf{x}=(x_1,\ldots,x_t) \leftarrow \mathcal{A}(k)}} \Big[x_i \neq x_j \textit{ for all } i \neq j \textit{ and } \big(\mathbf{y} = (h(x_1), \ldots, h(x_t)) \in R_{\text{out}}\Big] = \mathsf{negl}(\lambda).$$

In this work, we also consider a strengthening of R_{out}-output intractability (as defined above) in which the inputs $x_1, \ldots, x_t$ are not required to be distinct; of course, this larger relation must still be sparse in order for correlation intractability to be feasible.

Definition 2.6 (Not-All-Equal (NAE) Output Intractability)**.** *For a given relation ensemble* $R_{\text{out}} = \{R_{\text{out},\lambda} \subseteq (\{0,1\}^{\mu(\lambda)})^{t(\lambda)}\}$*, a hash family* $\mathcal{H} = \{h_\lambda : \{0,1\}^{\kappa(\lambda)} \times \{0,1\}^{\nu(\lambda)} \to \{0,1\}^{\mu(\lambda)}\}$ *is said to be* not-all-equal R_{out}-output intractable *if it is R-correlation intractable for the relation*

$$R = \Big\{(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in R_{\text{out}} \textit{ and } x_1, \ldots, x_t \textit{ are not all equal } \Big\}.$$

When t is a constant, not-all-equal output intractability for a t-output relation R_{out} follows from standard output intractability for $\leq t^t$ different relations defined based on R_{out} (there is one distinct-input relation for each partition of [t]). When t is superconstant it becomes better to prove the security property directly (without incurring a t^t security loss).

Definition 2.7 ((Not-All-Equal) Multi-Input CI for $\mathbb{Z}_p$-Shifted Relations)**.** *Let* $p = p(\lambda)$ *be an efficiently computable function of* λ*.*

For a given function ensemble $\mathcal{F} = \{f_\lambda : \{0,1\}^{\nu(\lambda)} \to \mathbb{Z}_p^{\mu(\lambda)}\}$ *and relation ensemble* $R_{\text{out}} = \{R_{\text{out},\lambda} \subseteq (\mathbb{Z}_p^{\mu(\lambda)})^{t(\lambda)}\}$*, a hash family* $\mathcal{H} = \{h_\lambda : \{0,1\}^{\kappa(\lambda)} \times \{0,1\}^{\nu(\lambda)} \to \mathbb{Z}_p^{\mu(\lambda)}\}$ *is said to be* (R_{out}, f)-correlation intractable *(respectively,*not-all-equal (R_{out}, f)-correlation intractable*) if it is correlation intractable for the* shifted relation

$$R = \Big\{(\mathbf{x}, \mathbf{y}) : x_i \neq x_j \textit{ for all } i \neq j \textit{ and } (y_1 - f(x_1), \ldots, y_t - f(x_t)) \in R_{\text{out}}\Big\},$$

respectively,

$$R_{\text{NAE}} = \Big\{(\mathbf{x}, \mathbf{y}) : x_1, \ldots, x_t \textit{ are not all equal } (y_1 - f(x_1), \ldots, y_t - f(x_t)) \in R_{\text{out}}.\Big\}$$

We note that Theorem 2.7 generalizes both Theorem 2.4 and Theorem 2.5/Theorem 2.6. In particular, when $p(\lambda)$ is a power-of-two, Theorems 2.5 and 2.6 can be recovered (identifying $\mathbb{Z}_p^\mu = \{0,1\}^{\mu \log p}$) by setting f to be the all-zero function, while Theorem 2.4 can be recovered by setting $R_{\text{out}} = \{\mathbf{0}^\mu \in \mathbb{Z}_p^\mu = \{0,1\}^{\mu \log p}\}$.

Finally, we describe an interesting special case of Theorem 2.7 that we securely instantiate under LWE.

Definition 2.8 (Weighted Sum Resistance mod p)**.** *Let* $t = t(\lambda)$*. A hash function family* $\mathcal{H}$ *with output space* $\mathbb{Z}_p^\mu$ *is weighted sum resistant mod p with weights* $w \in \{-1, 0, 1\}^t$ *if it is not-all-equal output intractable for the t-output relation*

$$R_{\text{out}} = \Big\{\mathbf{y} : \sum_{i=1}^{t} w_i y_i = 0^\mu \ \textit{(mod p)}\Big\}.$$

We say that $\mathcal{H}$ is weighted sum resistant if it is sum resistant for all nonzero weight vectors w. As shown in Section 4, our LWE-based hash family satisfies (NAE) multi-input CI for *shifted* weighted sum resistance mod p with $p = \mathsf{poly}(\lambda)$.

2.2 Shift-Hiding Shiftable Functions

We consider a weakening of the original definition of Peikert and Shiehian [PS18] that does not give the adversary oracle access to the SHSF. We also consider a modified definition with exact correctness rather than approximate correctness (this corresponds to the "rounded version" of the [PS18] construction).

Definition 2.9 (Shift-Hiding Shiftable Functions [PS18])**.** *Let $p = p(\lambda)$ be an efficiently computable function of λ. We define a family of shift-hiding shiftable functions with input space $\{0,1\}^{\nu(\lambda)}$ and output space $\mathbb{Z}_p^{\mu(\lambda)} = \{0,1\}^{\mu(\lambda)\log p(\lambda)}$ for arbitrary polynomial functions $(\nu(\lambda), \mu(\lambda))$.*

For a given class $\mathcal{C}$ of function ensembles $\mathcal{F} = \{f_\lambda : \{0,1\}^{\nu(\lambda)} \to \mathbb{Z}_p^{\mu(\lambda)}\}$, a shift-hiding shiftable function family $\mathsf{SHSF} = (\mathsf{Gen}, \mathsf{Shift}, \mathsf{Eval}, \mathsf{SEval})$ *consists of four PPT algorithms:*

- $\mathsf{Gen}(1^\lambda)$ *outputs a master secret key* msk *and public parameters* pp.
- $\mathsf{Shift}(\mathsf{msk}, f)$ *takes as input a secret key* msk *and a function $f \in \mathcal{F}$. It outputs a shifted key* sk_f.
- $\mathsf{Eval}(\mathsf{pp}, \mathsf{msk}, x)$*, given a secret key* msk *and input $x \in \{0,1\}^{\nu(\lambda)}$, outputs an evaluation $y \in \mathbb{Z}_p^{\mu(\lambda)}$.*
- $\mathsf{SEval}(\mathsf{pp}, \mathsf{sk}_f, x)$*, given a shifted key* sk_f *and input $x \in \{0,1\}^{n(\lambda)}$, outputs an evaluation $y \in \mathbb{Z}_p^{\mu(\lambda)}$.*

We will sometimes use the notation $F_{\mathsf{sk}}(x)$ to mean either $\mathsf{Eval}(\mathsf{sk}, x)$ or $\mathsf{SEval}(\mathsf{sk}, x)$ when the context is clear.

We require that SHSF *satisfies the following two properties:*

- ***Computational Correctness****: for any function $f \in \mathcal{F}$, given public parameters* pp *and a shifted key $\mathsf{sk}_f \leftarrow \mathsf{Shift}(\mathsf{msk}, f)$ (for $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Gen}(1^\lambda)$), it is computationally hard to find an input $x \in \{0,1\}^{\nu(\lambda)}$ such that $\mathsf{Eval}(\mathsf{sk}_f, x) \neq \mathsf{Eval}(\mathsf{msk}, x) + f(x)$ (mod p). In other words, the equation*

$$F_{\mathsf{sk}_f}(x) = F_{\mathsf{msk}}(x) + f(x)$$

  holds computationally (mod p).

- ***Shift Hiding****: for any pair of functions $f, g \in \mathcal{F}$,*

$$\mathsf{sk}_f \approx_c \mathsf{sk}_g,$$

  where $\mathsf{sk}_f \leftarrow \mathsf{Shift}(\mathsf{msk}, f)$, $\mathsf{sk}_g \leftarrow \mathsf{Shift}(\mathsf{msk}, g)$, and $\mathsf{msk} \leftarrow \mathsf{Gen}(1^\lambda)$.

2.3 One-Dimensional Short Integer Solution (1D-SIS)

For the definition of the LWE and SIS problems, we refer the reader to Peikert's survey [Pei16].

We make use of the hardness of a special "one-dimensional" SIS problem [BV15]. This is no easier to solve than LWE, but for clarity, as was done in [BV15, PS18], it is convenient to define it separately.

Definition 2.10 (1D-SIS [BV15, PS18])**.** *Let $p \in \mathbb{N}$ and $p_1 < p_2 < \ldots < p_n$ be pairwise coprime and relatively prime to p. Let $Q = p \cdot \prod_{i=1}^{n} p_i$. Then, for positive integers $m \in \mathbb{N}$ and B, the $\mathsf{1D\text{-}SIS}_{m,p,q,B}$ problem is as follows: given a uniformly random vector $\mathbf{v} \in \mathbb{Z}_q^m$, find a vector $\mathbf{z} \in \mathbb{Z}^m$ such that*

- $||\mathbf{z}||_\infty \leq B$*; and*
- $\langle \mathbf{v}, \mathbf{z} \rangle \pmod q \in \frac{q}{p} \cdot \mathbb{Z} + [-B, B]$*.*

For sufficiently large $p_i \geq B \cdot \mathsf{poly}(n, \log q)$, solving 1D-SIS is at least as hard as approximating certain short vector problems on arbitrary n-dimensional lattices to within $B \cdot \mathsf{poly}(n)$ factors [Ajt96, MR04, BV15].

3 Correlation Intractability from Shift-Hiding Shiftable Functions

In this section, we show that shift-hiding shiftable functions (Theorem 2.9) that are *output intractable* (Theorems 2.5 and 2.6) can be used to construct correlation-intractable hash functions for shifted relations (Theorem 2.7). As a special case, this shows that SHSFs that are *hard to invert* yield correlation-intractable hash functions for all circuits (Theorem 2.4) supported by the SHSF function class $\mathcal{C}$. In other words, SHSFs allow us to *lift* a form of output intractability to a more general form of correlation intractability.

Formally, let $\mathsf{SHSF} = (\mathsf{Gen}, \mathsf{Shift}, \mathsf{Eval})$ be a SHSF family that represents functions of the form $F_{\mathsf{sk}} : \{0,1\}^{\nu(\lambda)} \to \mathbb{Z}_p^{\mu(\lambda)}$ and supports shifts for functions $f \in \mathcal{F}$. We then consider two hash functions $\mathcal{H}_{\text{plain}}, \mathcal{H}_{\text{shift}}$:

- $\mathcal{H}_{\text{plain}}$ uses msk as a hash key, and computes the function $h(\mathsf{msk}, x) = F_{\mathsf{msk}}(x)$.
- $\mathcal{H}_{\text{shift}}$ uses sk_Z as a hash key, where $Z : \{0,1\}^\nu \to \mathbb{Z}_p^\mu$ is an identically zero function. It computes the function $h(\mathsf{sk}_Z, x) = F_{\mathsf{sk}_Z}(x)$.

Theorem 3.1. *Let R_{out} be an efficiently decidable output relation. If SHSF is a shift-hiding shiftable function family and $\mathcal{H}_{\text{plain}}$ is R_{out}-output intractable, then $\mathcal{H}_{\text{shift}}$ is (R, f)-correlation intractable for any $f \in \mathcal{F}$.*

Moreover, if $\mathcal{H}_{\text{plain}}$ is NAE-R_{out}-output intractable, then $\mathcal{H}_{\text{shift}}$ is NAE-(R, f)-CI for any $f \in \mathcal{F}$.

Proof. Suppose that a PPT adversary $\mathcal{A}$ breaks the (R, f)-correlation intractability of $\mathcal{H}_{\text{shift}}$, which means that $\mathcal{A}$ wins the following challenger-based security game with non-negligible probability:

1. The challenger samples $\mathsf{msk} \leftarrow \mathsf{Gen}(1^\lambda)$.
2. The challenger samples $\mathsf{sk} = \mathsf{sk}_Z \leftarrow \mathsf{Shift}(\mathsf{msk}, Z)$ and sends sk to $\mathcal{A}$.
3. $\mathcal{A}(\mathsf{sk})$ outputs $\mathbf{x} = (x_1, \ldots, x_t)$.
4. $\mathcal{A}$ wins if (i) the inputs x_i are distinct, and (ii) for $y_i = F_{\mathsf{sk}}(x_i) - f(x_i)$, the relation $R_{\text{out}}(\mathbf{y})$ holds.

Then, $\mathcal{A}$ also wins each of the following modified security games with non-negligible probability.

- Hybrid Hyb_1: same as the honest security game, except that in step (2), we sample

$$\mathsf{sk}_f \leftarrow \mathsf{Shift}(\mathsf{msk}, f)$$

  This is indistinguishable from the original security game by the shift-hiding of SHSF.

- Hybrid Hyb_2: same as Hyb_1, except that in step (4), we change the win condition (ii) so that $\mathcal{A}$ wins if for $y_i = F_{\mathsf{msk}}(x_i)$, the relation $R_{\text{out}}(\mathbf{y})$ holds.

  This is indistinguishable from Hyb_1 by the computational correctness of SHSF.

Finally, we show that $\mathcal{A}$'s success in Hyb_2 leads to an attack $\mathcal{A}'$ on the R_{out}-output intractability of $\mathcal{H}_{\text{plain}}$. The attack works as follows:

1. The challenger samples $\mathsf{msk} \leftarrow \mathsf{Gen}(1^\lambda)$ and sends msk to $\mathcal{A}'$.
2. $\mathcal{A}'(\mathsf{msk})$ samples $\mathsf{sk} = \mathsf{sk}_f \leftarrow \mathsf{Shift}(\mathsf{msk}, f)$.
3. $\mathcal{A}'$ then calls $\mathcal{A}(\mathsf{sk}_f)$ and outputs $\mathbf{x} = (x_1, \ldots, x_\ell)$.
4. By definition, $\mathcal{A}'$ wins if (i) the x_i are distinct, and (ii) for $y_i = F_{\mathsf{msk}}(x_i)$, the relation $R_{\text{out}}(\mathbf{y})$ holds.

By construction, $\mathcal{A}'$ above wins with the same probability that $\mathcal{A}$ wins in Hyb_2, contradicting the R_{out}-output intractability of $\mathcal{H}_{\text{plain}}$.

The same argument as above applies to NAE-CI, with the condition (i) replaced by "the inputs x_i are not all equal." This completes the proof of Theorem 3.1.

□

4 Construction of (Weighted) Sum-Resistant SHSF

We show the (weighted) sum-resistance of a variant of the Peikert-Shiehian construction of shift-hiding shiftable functions [PS18]. We start by describing the ingredients that we use in the construction. The construction itself is described in Section 4.2, the proof of shift-hiding in Section 4.3, the proof of computational correctness in Section 4.4, and the proof of sum-resistance is in Section 4.5.

4.1 The Ingredients

The Gadget Matrix. An important ingredient in many lattice-based constructions is the gadget matrix $\mathbf{G}$ and the operator $\mathbf{G}^{-1}$ associated to it. Let

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

The gadget matrix $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}$ is a block diagonal matrix with copies of $\mathbf{g}$ on the diagonal. In fact, we will extend $\mathbf{G}$ to m columns for any $m \geq n\lceil \log q \rceil$ by appending zero columns.

An important property of $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ is that for every vector $\mathbf{v} \in \mathbb{Z}_q^n$, there is a 0-1 vector $\mathbf{v}' \in \{0,1\}^m$ such that $\mathbf{G}\mathbf{v}' = \mathbf{v} \pmod q$. This leads us to define the operator $\mathbf{G}^{-1} : \mathbb{Z}_q^n \to \{0,1\}^m$ which has the property that

1. $\mathbf{G}^{-1}(\mathbf{v}) \in \{0,1\}^m$ for every vector $\mathbf{v} \in \mathbb{Z}_q^n$; and
2. $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{v}) = \mathbf{v} \pmod q$.

We will extend $\mathbf{G}^{-1}$ to matrices $\mathbf{V}$ by acting on each column of the matrix separately.
We caution the reader that $\mathbf{G}^{-1}$ refers to a (non-linear) operator, and has little to do with matrix inverses.

Gadget Homomorphisms. The key idea in the SHSF construction is the notion of gadget homomorphisms originating from [BGG⁺14]. For LWE matrices $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{Z}_q^{n \times m}$, define the sum and product matrices

$$\mathbf{A}_+ = \mathbf{A}_1 + \mathbf{A}_2 \quad \text{and} \quad \mathbf{A}_\times = -\mathbf{A}_1\mathbf{G}^{-1}(\mathbf{A}_2) \tag{1}$$

where $\mathbf{G}$ is the gadget matrix and $\mathbf{G}^{-1}$ is the bit decomposition operator defined above. The gadget homomorphisms allow us to start from LWE encodings $\mathbf{c}_1 = \mathbf{s}(\mathbf{A}_1 + x_1\mathbf{G})$ and $\mathbf{c}_2 = \mathbf{s}(\mathbf{A}_2 + x_2\mathbf{G})$ w.r.t. an LWE secret $\mathbf{s} \in \mathbb{Z}_q^n$ (where we suppress the LWE errors for clarity) and compute

$$\mathbf{c}_+ = \mathbf{s}(\mathbf{A}_+ + (x_1 + x_2)\mathbf{G}) \quad \text{and} \quad \mathbf{c}_\times = \mathbf{s}(\mathbf{A}_\times + x_1x_2\mathbf{G}) \tag{2}$$

In particular, this is accomplished by setting

$$\mathbf{c}_+ = \mathbf{c}_1 + \mathbf{c}_2 = \mathbf{s}(\mathbf{A}_1 + \mathbf{A}_2 + (x_1 + x_2)\mathbf{G}) = \mathbf{s}(\mathbf{A}_+ + (x_1 + x_2)\mathbf{G})$$

and

$$\begin{aligned} \mathbf{c}_\times &= -\mathbf{c}_1\mathbf{G}^{-1}(\mathbf{A}_2) + x_1\mathbf{c}_2 \\ &= -\mathbf{s}(\mathbf{A}_1 + x_1\mathbf{G}) \cdot \mathbf{G}^{-1}(\mathbf{A}_2) + \mathbf{s}(\mathbf{A}_2 + x_2\mathbf{G}) \cdot x_1 \\ &= \mathbf{s}(-\mathbf{A}_1\mathbf{G}^{-1}(\mathbf{A}_2) + x_1x_2\mathbf{G}) \\ &= \mathbf{s}(\mathbf{A}_\times + x_1x_2\mathbf{G}) \end{aligned}$$

Crucially, this computation does not require the knowledge of either x_1 or x_2 to compute the sum. It does require the knowledge of x_1 (but not x_2) to compute the product. This *asymmetry* will prove invaluable to us down the line.

More generally, we define the following two algorithms.

- $\mathsf{Gadget.MEval}(g, \mathbf{A}_1, \ldots, \mathbf{A}_\nu)$, the matrix homomorphism, takes as input a function $g : \{0,1\}^\ell \to \{0,1\}$ and ν matrices $\mathbf{A}_1, \ldots, \mathbf{A}_\nu$ and outputs the matrix $\mathbf{A}_g$ obtained by composing together the addition and multiplication operations in Equation 1.

- $\mathsf{Gadget.VEval}(g, x, \mathbf{c}_1, \ldots, \mathbf{c}_\nu)$, the vector homomorphism, takes as input a function $g : \{0,1\}^\nu \to \{0,1\}$, an input $x = x_1x_2 \ldots x_\nu$ and LWE encodings

  $$\mathbf{c}_1 = \mathbf{s}(\mathbf{A}_1 + x_1\mathbf{G}) + \mathbf{e}_1, \ldots, \mathbf{c}_\nu = \mathbf{s}(\mathbf{A}_\nu + x_\nu\mathbf{G}) + \mathbf{e}_\nu$$

  of x w.r.t. $\mathbf{A}_1, \ldots, \mathbf{A}_\nu$, and outputs a vector $\mathbf{c}_g$ obtained by composing together the addition and multiplication operations in Equation 2.

Correctness tells us that

$$\mathbf{c}_g \approx \mathbf{s}(\mathbf{A}_g + g(x)\mathbf{G}) \tag{3}$$

where the difference is an LWE error whose magnitude is $O((n \log q)^{O(d_g)})$ where λ is a security parameter and d_g is the depth of the circuit g.

Looking ahead, we make two important observations on these algorithms:

1. First, if the function g is of a special form, namely $g(x_1, x_2) = \langle x_1, f(x_2)\rangle$ for some $x = x_1 || x_2$, then $\mathsf{Gadget.VEval}$ does not require the knowledge of x_1, rather only x_2. This observation is due to [AFV11, GVW15] where it was used construct a predicate encryption scheme.

2. Secondly, if the first coordinate of $\mathbf{s}$ is 1 (which we can set without loss of security) then we have

$$c_g \approx \mathbf{s}\mathsf{a}_g + g(x) \tag{4}$$

where c_g is the first coordinate of $\mathbf{c}_g$ and a_g is the first column of $\mathbf{A}_g$. This is because the first column of $\mathbf{G}$ is the unit vector with 1 in the first coordinate and 0 everywhere else.

FHE with Almost Linear Decryption. We require the existence of a (secret-key) FHE scheme where the secret key fsk is a vector $\mathbf{s} \in \mathbb{Z}_q^{\widehat{n}}$, ciphertexts fct of messages $m \in \mathbb{Z}_p$ are vectors $\mathbf{c} \in \mathbb{Z}_q^{\widehat{n}}$ and decryption proceeds by first doing a linear operation which gives

$$\langle \mathsf{fsk}, \mathsf{fct} \rangle = m \cdot \left\lfloor \frac{q}{p} \right\rceil + e \pmod q \tag{5}$$

where e is a small error. We will let $\mathsf{FHE.Enc}$ denote the encryption algorithm and $\mathsf{FHE.Eval}$ denote the evaluation algorithm.

4.2 The Shift-Hiding Shiftable Function

Let the family of functions $\mathcal{F}$ consist of functions $f : \{0,1\}^\nu \to \mathbb{Z}_p^\mu$ computable by circuits of size at most $s = s(\lambda)$. We require that $p = p(\lambda)$ is a sufficiently large polynomial (in n).

- $\mathsf{Gen}(1^\lambda)$: picks LWE parameters $n = n(\lambda)$, $m = m(\lambda)$ and $q = q(\lambda)$. Generate the public parameters
$$\mathsf{pp} = (\mathbf{A}_1, \ldots, \mathbf{A}_\ell, \mathbf{u}) \leftarrow (\mathbb{Z}_q^{n\times m})^k \times \mathbb{Z}_q^{1\times m}$$
for a certain $\ell = \ell(s, \lambda)$ that will be specified in due course.

  Choose a uniformly random vector $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ whose first coordinate $\mathbf{s}[1] = 1$. Let $\mathsf{msk} = \mathbf{s}$.

- $\mathsf{Eval}(\mathsf{msk}, x)$: Let FHE be a (leveled) fully homomorphic encryption scheme with almost linear decryption (as defined above in Equation 5) with plaintext space $\mathbb{Z}_p$. Construct the functions $g_x^{(i)}$ that, on input a pair $(\mathsf{fsk}, \mathsf{fct})$, output

$$g_x^{(i)}(\mathsf{fsk}, \mathsf{fct}) = \left\langle \mathsf{fsk}, \mathsf{FHE.Eval}(\mathsf{fct}, \mathcal{U}_x^{(i)}) \right\rangle \pmod q$$

where $\mathcal{U}_x^{(i)}$ is a universal circuit that takes as input the description of a circuit f and outputs the i^{th} bit of $f(x)$.

Define

$$\mathbf{A}_x^{(i)} = \mathsf{Gadget.MEval}(g_x^{(i)}, \mathbf{A}_1, \ldots, \mathbf{A}_\ell) \in \mathbb{Z}_q^n \ ,$$

let $\mathsf{a}_x^{(i)}$ denote the first column of $\mathbf{A}_x^{(i)}$ and let

$$\mathbf{A}_x := [\mathsf{a}_x^{(1)} || \mathsf{a}_x^{(2)} || \ldots || \mathsf{a}_x^{(\mu)}] \in \mathbb{Z}_q^{n \times \mu}$$

denote the concatenation of $\mathsf{a}_x^{(i)}$. The output is

$$\lfloor \mathbf{s}\mathbf{A}_x + \mathbf{u}\mathbf{G}^{-1}(\mathbf{A}_x) \rceil_p := \left\lfloor \frac{p}{q} \cdot (\mathbf{s}\mathbf{A}_x + \mathbf{u}\mathbf{G}^{-1}(\mathbf{A}_x)) \right\rceil \in \mathbb{Z}_p^{1 \times \mu}$$

- $\mathsf{Shift}(\mathsf{msk}, f)$: Choose an FHE secret key $\mathsf{fsk} \in \mathbb{Z}_q^{\widehat{n}}$, encrypt the description of f into an FHE ciphertext fct, let $\phi := \mathsf{fct} || \mathsf{fsk}$, and let

  $$\mathbf{A}_f := [\mathbf{A}_1 + \phi_1 \mathbf{G} || \ldots || \mathbf{A}_\ell + \phi_\ell \mathbf{G}]$$

  Output as the shift key

  $$\mathsf{sk}_f := (\mathsf{fct}, \mathbf{s}\mathbf{A}_f + \mathbf{e})$$

  where $\mathbf{e}$ is drawn from the LWE noise distribution.

  Note that ℓ is the bit-length of $\mathsf{fsk} || \mathsf{fct}$ and is $\mathsf{poly}(s, \lambda)$.

- $\mathsf{SEval}(\mathsf{sk}_f, x)$: Let the circuits $g_x^{(i)}$ be as in the definition of Eval.

  $$\mathbf{c}_x^{(i)} = \mathsf{Gadget.VEval}(g_x^{(i)}, \mathsf{fct}, \mathbf{c}_1, \ldots, \mathbf{c}_\ell) \in \mathbb{Z}_q^n$$

  where $\mathbf{c}_i = \mathbf{s}[\mathbf{A}_i + \phi_i \mathbf{G}]$. Note that crucially, $\mathsf{Gadget.VEval}$ does not require fsk as input because, by observation (1) above, $g_x^{(i)}$ only linearly depends on it. Let $c_x^{(i)}$ denote the first element of $\mathbf{c}_x^{(i)}$ and let $\mathbf{c}_x$ be the concatenation of all $c_x^{(i)}$.

  Output

  $$\lfloor \mathbf{c}_x + \mathbf{u}\mathbf{G}^{-1}(\mathbf{A}_x) \rceil_p$$

  as the shifted evaluation.

Basic Correctness. We first (informally) show correctness of SEval for any fixed x. By the correctness of the gadget homomorphisms (equation 4), we know that

$$\begin{aligned} c_x^{(i)} &\approx \mathsf{s}\mathsf{a}_x^{(i)} + g_x^{(i)}(\mathsf{fsk}, \mathsf{fct}) \\ &= \mathsf{s}\mathsf{a}_x^{(i)} + \langle \mathsf{fsk}, \mathsf{FHE.Eval}(\mathsf{fct}, \mathcal{U}_x^{(i)}) \rangle \\ &\approx \mathsf{s}\mathsf{a}_x^{(i)} + \mathcal{U}_x^{(i)} \cdot \left\lfloor \frac{q}{p} \right\rceil \\ &= \mathsf{s}\mathsf{a}_x^{(i)} + f^{(i)}(x) \cdot \left\lfloor \frac{q}{p} \right\rceil \end{aligned} \tag{6}$$

where the second equation is by the definition of $g_x^{(i)}$, the third (approximate) equation is by the correctness of FHE decryption (equation 5), and the fourth equation is by the definition of the universal circuit $\mathcal{U}$. The approximation error is proportional to the SHSF evaluation error plus the FHE decryption error which is

$$O((\hat{n}\log q)^{O(d)} + (n\log q)^{O(d')}) = O(\lambda^{\mathsf{poly}(d,\log\lambda)})$$

where d is the depth of the circuit $\mathcal{U}_x^{(i)}$ and $d' = O(d\cdot\log(n\log q))$ is the depth of the circuit $g_x^{(i)}$ that homomorphically evaluates $\mathcal{U}_x^{(i)}$ and decrypts.

Now, as long as $c_x^{(i)}$ does not fall too close to the boundaries of multiples of q/p, we have

$$\begin{aligned}\mathsf{SEval}(\mathsf{sk}_f, x) &= \lfloor \mathbf{c}_x + \mathbf{u}\mathbf{G}^{-1}(\mathbf{A}_x)\rceil_p \\ &= \lfloor \mathbf{s}\mathbf{A}_x + \mathbf{u}\mathbf{G}^{-1}(\mathbf{A}_x)\rceil_p + f(x) = \mathsf{Eval}(\mathsf{msk}, x) + f(x) \pmod p \end{aligned} \tag{7}$$

It turns out that for any fixed x, the boundary event happens with a negligible probability. In fact, we will show a morally stronger statement below: adapting arguments from [BV15, PS18], we will show in Section 4.4 that it is computationally hard to find an x for which correctness (that is, equation 7) fails. (This is stronger in that it holds for any adaptively chosen x, and weaker because the guarantee is computational, but necessarily so.)

4.3 Proof of Shift-Hiding

We wish to show that for any two functions $f_0, f_1 \in \mathcal{F}$,

$$(\mathsf{Shift}(\mathsf{msk}, f_0), \mathsf{pp}) \approx_c (\mathsf{Shift}(\mathsf{msk}, f_1), \mathsf{pp})$$

where $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$. Shift-hiding follows by the following sequence of hybrids.

Hybrid 0**.** This is the distribution generated by picking $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and outputting pp together with

$$\mathsf{sk}_{f_0} \leftarrow \mathsf{Shift}(\mathsf{msk}, f_0)$$

That is,

$$\mathsf{sk}_{f_0} := (\mathsf{fct}, \mathbf{s}\mathbf{A}_{f_0} + \mathbf{e})$$

where fct is an FHE encryption of f_0 under an FHE secret key fsk, and

$$\mathbf{A}_{f_0} = [\mathbf{A}_1 + \phi_1\mathbf{G}||\ldots||\mathbf{A}_\ell + \phi_\ell\mathbf{G}]$$

where $\phi = \mathsf{fsk}||\mathsf{fct}$ and the matrices $\mathbf{A}_i$ live in the public parameters.

Hybrid 1**.** Generate $\mathsf{fct} = \mathsf{FHE.Enc}(\mathsf{fsk}, f_0)$ as above, and let $\phi = \mathsf{fsk}||\mathsf{fct}$. Choose

$$\mathbf{A}_{f_0} = [\mathbf{A}'_1||\ldots||\mathbf{A}'_\ell]$$

to be a truly random LWE matrix of the appropriate dimensions, and program $\mathbf{A}_i$ in the public parameters to be $\mathbf{A}'_i - \phi_i\mathbf{G}$. Hybrid 1 is distributed identically to that in Hybrid 0.

Hybrid 2**.** Replace $\mathbf{s}\mathbf{A}_{f_0} + \mathbf{e}$ in Hybrid 1 with a uniformly random vector. This is computationally indistinguishable from Hybrid 1 by an application of LWE w.r.t. the uniformly random matrix $\mathbf{A}_f$.

Hybrid 3. Replace the public parameters by uniformly random matrices $\mathbf{A}_i$. This hybrid is distributed identically to Hybrid 2. Note that the distribution in this hybrid is independent of the FHE secret key fsk.

Hybrid 4. Replace fct in Hybrid 3 with an encryption of f_1 instead of f_0. This is computationally indistinguishable from Hybrid 3 by an application of FHE semantic security.

The remaining hybrids backtrack through hybrids 2 back to 0 using f_1 instead of f_0.

Hybrid 5–7. This is identical to Hybrid 2–0, except that fct is an encryption of f_1.

Putting together, we have that given the public parameters pp, the shift keys for f_0 and f_1 are computationally indistinguishable.

4.4 Proof of Computational Correctness

Computational correctness follows from an essentially identical argument in [PS18]. We sketch it here for completeness.

By the calculation in equation 6, we know that for each $i \in [\mu]$,

$$c_x^{(i)} = \mathsf{sa}_x^{(i)} + f^{(i)}(x) \cdot \left\lfloor \frac{q}{p} \right\rceil + e_i$$

where $|e_i| \leq B = O(\lambda^{\mathsf{poly}(d, \log \lambda)})$.

Assume that there is an adversary that, given the shift key $\mathsf{sk}_f \leftarrow \mathsf{Shift}(\mathsf{msk}, f)$ for some f of his choice, produces an x such that

$$\mathsf{SEval}(\mathsf{sk}_f, x) \neq \mathsf{Eval}(\mathsf{msk}, x)$$

meaning that they differ in some coordinate, say i.

Then, by the expressions for SEval and Eval, we have

$$\begin{aligned}
\mathsf{SEval}(\mathsf{sk}_f, x)|_i = \left\lfloor \frac{p}{q} c_x^{(i)} \right\rceil &= \left\lfloor \frac{p}{q} \cdot (\mathsf{sa}_x^{(i)} + f^{(i)}(x) \cdot \left\lfloor \frac{q}{p} \right\rceil + e_i) \right\rceil \\
&= \left\lfloor \frac{p}{q} \cdot (\mathsf{sa}_x^{(i)} + f^{(i)}(x) \cdot \frac{q}{p} + e_i') \right\rceil \\
&\neq \left\lfloor \frac{p}{q} \cdot (\mathsf{sa}_x^{(i)} + f^{(i)}(x) \cdot \frac{q}{p}) \right\rceil \\
&= \left\lfloor \frac{p}{q} \cdot \mathsf{sa}_x^{(i)} \right\rceil + + f^{(i)}(x) = \mathsf{Eval}(\mathsf{msk}, x)|_i
\end{aligned}$$

This can only happen when

$$c_x^{(i)} \in \frac{q}{p}\mathbb{Z} + [-(B+p), B+p]$$

where B comes from the magnitude of e_i and the additional p comes from the difference between $\lfloor q/p \rceil$ and q/p.

Now, observe that

$$c_x^{(i)} = [\mathbf{c}_1 || \dots || \mathbf{c}_\ell] \cdot \mathbf{h}^{(i)}$$

for some vector $\mathbf{h}^{(i)}$ of low norm $B = O(\lambda^{\mathsf{poly}(d, \log \lambda)})$. Since $\mathbf{c}_i$ are pseudorandom, this gives us a solution to the $\mathsf{1D\text{-}SIS}_{\ell,p,q,\approx B}$ problem.

4.5 Proof of Sum-Resistance

Assume that an adversary $\mathcal{A}$, given msk and pp, comes up with weights $w_1, \ldots, w_t \in \{-1, 0, 1\}^t \backslash \{0^t\}$ and not-all-equal inputs $x_1, \ldots, x_t$ such that

$$\sum_{i=1}^{t} w_i \cdot \mathsf{Eval}(\mathsf{msk}, x_i) = 0 \pmod p$$

That is,

$$\sum_{i=1}^{t} w_i \cdot \lfloor \mathbf{s}\mathbf{A}_{x_i} + \mathbf{u}\mathbf{G}^{-1}(\mathbf{A}_{x_i}) \rceil_p = 0 \pmod p$$

Rewriting this, we have

$$\sum_{i=1}^{t} w_i \cdot \lfloor (\mathbf{s}\mathbf{G} + \mathbf{u})\mathbf{G}^{-1}(\mathbf{A}_{x_i}) \rceil_p = \sum_{i=1}^{t} w_i \cdot \left\lfloor \frac{p}{q} \cdot (\mathbf{s}\mathbf{G} + \mathbf{u})\mathbf{G}^{-1}(\mathbf{A}_{x_i}) \right\rceil = 0 \pmod p$$

Writing $\mathbf{v}$ for $\mathbf{s}\mathbf{G} + \mathbf{u}$, and isolating the rounding errors ϵ_i, we have

$$\frac{p}{q} \cdot \mathbf{v} \cdot \sum_{i=1}^{t} w_i \cdot \mathbf{G}^{-1}(\mathbf{A}_{x_i}) = \sum_{i=1}^{t} w_i \epsilon_i \pmod p$$

Note that $|| \sum_{i=1}^{k} w_i \epsilon_i || \leq t$. Multiplying both sides by q/p,

$$\mathbf{v} \cdot \sum_{i=1}^{t} w_i \cdot \mathbf{G}^{-1}(\mathbf{A}_{x_i}) = \frac{q}{p} \cdot \sum_{i=1}^{t} w_i \epsilon_i := \epsilon \pmod q$$

where $||\epsilon||_\infty \leq qt/p$. Now, we have two possibilities:

Case 1. $\sum_{i=1}^{t} w_i \cdot \mathbf{G}^{-1}(\mathbf{A}_{x_i}) \neq 0 \pmod q$. In this case, it forms a SIS solution w.r.t. $\mathbf{v}$. (Technically, this is a one-dimensional version of SIS that is at least as hard as "short vector" problems on lattices [Ajt96, MR04]).

Case 2. $\sum_{i=1}^{t} w_i \cdot \mathbf{G}^{-1}(\mathbf{A}_{x_i}) = 0 \pmod q$. In this case, we know that

$$\mathbf{G} \cdot \sum_{i=1}^{t} w_i \cdot \mathbf{G}^{-1}(\mathbf{A}_{x_i}) = \sum_{i=1}^{t} w_i \mathbf{A}_{x_i} = 0 \pmod q$$

We now show how to use this to break SIS.

Let $h = h_1 \ldots h_\ell$ be the description of a random function chosen from a t-wise independent hash family. Moreover, let $x_1, \ldots x_t$ denote the inputs returned by any fixed execution of $\mathcal{A}$. Then, let

$$y = \sum_{i=1}^{t} w_i h(x_i) \pmod q.$$

We note that if $x_1, \ldots, x_t$ are not-all-equal, then with high probability over the choice of h we have $y \neq 0$. This follows directly from the t-wise independence of h: if the x_i are distinct, then indeed

$\sum_{i=1}^{t} w_i h(x_i)$ always has min-entropy (since there exists a term $\sum_{i\in S} w_i h(x_i)$ corresponding to one "super-variable" where $\sum_{i\in S} w_i \neq 0$). Therefore, we conclude that with non-negligible probability over the randomness of $\mathcal{A}, \mathsf{msk}, h$, $\mathcal{A}$ outputs $\mathbf{x}$ such that $\sum_{i=1}^{t} w_i \mathbf{G}^{-1}(\mathbf{A}_{x_i}) = 0$ and $y \neq 0$.

Now, imagine the experiment where $\mathbf{A}_j$ is picked as $\mathbf{A}\mathbf{R}_j + h_j\mathbf{G}$. Here,

$$\mathbf{A} = \begin{bmatrix} \mathbf{a} \\ \underline{\mathbf{A}} \end{bmatrix}$$

where $\underline{\mathbf{A}}$ is an SIS challenge matrix and $\mathbf{a}$ is uniformly random. This is statistically indistinguishable from above, so the same claimed property holds. Now,

$$\mathbf{A}_x^{(i)} = \mathsf{Gadget.MEval}(\mathcal{U}_x^{(i)}, \mathbf{A}_1, \ldots, \mathbf{A}_\ell) = \mathbf{A}\mathbf{R}_{x,i} + h(x)|_i \mathbf{G}$$

and

$$\mathsf{a}_x^{(i)} = \mathbf{A}\mathbf{r}_{x,i} + h(x)|_i \mathbf{u}_i$$

where $\mathbf{u}_i$ is the first unit vector. (Technically, $\mathbf{A}_x^{(i)}$ is computed by doing a homomorphic evaluation of h and then decrypting. However, this complication does not make a significant difference to our argument below.)

We know that for each $i \in [\mu]$,

$$\sum_{j=1}^{t} w_j \mathsf{a}_{x_j}^{(i)} = 0 \pmod q$$

In other words,

$$\mathbf{A} \cdot \underbrace{\sum_{j=1}^{t} w_j \mathbf{R}_{x_j}}_{:=\mathbf{R}} + \underbrace{\sum_{j=1}^{t} w_j h(x_j)}_{:=y} \mathbf{u}_1 = 0 \pmod q$$

Whenever $y \neq 0 \pmod q$, it follows that $\mathbf{R}$ is not zero. Now, we have $\underline{\mathbf{A}}\mathbf{R} = 0 \pmod q$ and $\mathbf{R} \neq 0$ giving us a SIS solution w.r.t. $\underline{\mathbf{A}}$. This finishes the proof of weighted t-sum-resistance.

4.6 Putting it Together: Weighted Sum-Resistant SHSFs

Combining the results of Section 4.3, Section 4.4, and Section 4.5, we obtain the following theorem.

Theorem 4.1. *Assume that there is some $\epsilon > 0$ for which it is hard to approximate short vector problems in worst case n-dimensional lattices to within 2^{n^ϵ} factor. Let $\mathsf{SHSF} = (\mathsf{Gen}, \mathsf{Shift}, \mathsf{Eval})$ be the SHSF family constructed above. Then, the hash function family $\mathcal{H}_{\text{plain}}$ that uses $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Gen}(1^\lambda)$ as a hash key, and computes the function*

$$h((\mathsf{pp}, \mathsf{msk}), x) = \mathsf{Eval}(\mathsf{pp}, \mathsf{msk}, x)$$

is t-weighted-sum-resistant for every $t = \mathsf{poly}(\lambda)$.

Combining Theorem 4.1 and Theorem 3.1 (the CI lifting theorem), we get a hash family that is CI for shifted (weighted) sum relations.

Theorem 4.2. *Under the same assumption as in Theorem 4.1, there is a hash function family $\mathcal{H}$ that is (R_{out}, f)-correlation intractable (as in Definition 2.7), where R_{out} is the weighted sum relation as in Definition 2.8 and f is any efficiently computable function. That is, $\mathcal{H}$ is correlation-intractable for shifted (weighted) sum relations.*

5 Output-Intractable SHSFs from iO

In this section, we present constructions of Output-Intractable SHSFs from iO (Theorem 1.6 and Theorem 1.5). For simplicity, we set the shift modulus $p = 2$ for SHSFs in the remainder of this section.

5.1 IO-Related Preliminaries

5.1.1 Indistinguishability Obfuscation

An *obfuscator for all circuits* is a PPT algorithm $\mathcal{O}$ such that for every circuit C, $\mathcal{O}(C)$ is with probability 1 a circuit $\tilde{C}$ with the same functionality as C.

Definition 5.1 (Indistinguishability Obfuscation [BGI+01]). *$\mathcal{O}$ is a (s, δ)-secure* indistinguishability obfuscator *(iO) if for all pairs of functionally equivalent circuits C_0 and C_1 of size $|C_0| = |C_1| = \lambda$, and all circuits $\mathcal{A}$ of size $s(\lambda)$, it holds that*

$$\Pr[\mathcal{A}(\mathcal{O}(C_0)) = 1] - \Pr[\mathcal{A}(\mathcal{O}(C_1)) = 1] \leq O(\delta(\lambda)).$$

5.1.2 Puncturable PRFs

Definition 5.2 (Puncturable PRF [BW13, BGI14, KPTZ13, SW14]). *A puncturable PRF family is a family of functions*

$$\mathcal{F} = \left\{F_{\lambda,s} : \{0,1\}^{\nu(\lambda)} \to \{0,1\}^{\mu(\lambda)}\right\}_{\lambda\in\mathbb{N}, s\in\{0,1\}^{\ell(\lambda)}}$$

with associated (deterministic) polynomial-time algorithms $(\mathcal{F}.\mathsf{Eval}, \mathcal{F}.\mathsf{Puncture}, \mathcal{F}.\mathsf{PuncEval})$ *satisfying*

- *For all $x \in \{0,1\}^{\nu(\lambda)}$ and all $s \in \{0,1\}^{\ell(\lambda)}$, $\mathcal{F}.\mathsf{Eval}(s, x) = F_{\lambda,s}(x)$.*
- *For all distinct $x, x' \in \{0,1\}^{\nu(\lambda)}$ and all $s \in \{0,1\}^{\ell(\lambda)}$,*

$$\mathcal{F}.\mathsf{PuncEval}(\mathcal{F}.\mathsf{Puncture}(s, x), x') = \mathcal{F}.\mathsf{Eval}(s, x')$$

For ease of notation, we write $F_s(x)$ and $\mathcal{F}.\mathsf{Eval}(s, x)$ interchangeably, and we write $s\{x\}$ to denote $\mathcal{F}.\mathsf{Puncture}(s, x)$.

$\mathcal{F}$ is said to be (s, δ)-secure if for every $\{x^{(\lambda)} \in \{0,1\}^{\nu(\lambda)}\}_{\lambda\in\mathbb{N}}$, the following two distribution ensembles (indexed by λ) are $\delta(\lambda)$-indistinguishable to circuits of size $s(\lambda)$:

$$(S\{x^{(\lambda)}\}, F_S(x^{(\lambda)})) \text{ where } S \leftarrow \{0,1\}^{\ell(\lambda)}$$

and

$$(S\{x^{(\lambda)}\}, U) \text{ where } S \leftarrow \{0,1\}^{\ell(\lambda)},\ U \leftarrow \{0,1\}^{\mu(\lambda)}.$$

Theorem 5.3 ([GGM84, KPTZ13, BW13, BGI14, SW14]). *If {polynomially secure, subexponentially secure} one-way functions exist, then for all functions $\mu : \mathbb{N} \to \mathbb{N}$ (with $1^{\mu(\nu)}$ polynomial-time computable from 1^{ν}), and all $\delta : \mathbb{N} \to [0,1]$ with $\delta(\nu) \geq 2^{-\mathsf{poly}(\nu)}$, there are polynomials $\ell(\lambda), \nu(\lambda)$ and a {polynomially secure, $(\frac{1}{\delta(\nu(\lambda))}, \delta(\nu(\lambda)))$-secure} puncturable PRF family*

$$\mathcal{F}_\mu = \left\{F_{\lambda,s} : \{0,1\}^{\nu(\lambda)} \to \{0,1\}^{\mu(\nu(\lambda))}\}_{\lambda\in\mathbb{N}, s\in\{0,1\}^{\ell(\lambda)}}\right\}.$$

5.1.3 Lossy Functions

Definition 5.4 (Lossy Functions [PW08])**.** *A* lossy function family $\mathsf{LF} = (\mathsf{LF.Gen}, \mathsf{LF.Eval})$ *consists of two PPT algorithms:*

- $\mathsf{LF.Gen}(1^\lambda, \text{injective/lossy})$ *outputs an evaluation key* ek *either in "injective mode" or "lossy mode."*
- $\mathsf{LF.Eval}(\mathsf{ek}, x)$ *takes an evaluation key* ek *as well as an input* $x \in \{0,1\}^{\nu(\lambda)}$*. It returns a* deterministic *output* $y \in \{0,1\}^{N(\lambda)}$*.*

We require that LF *satisfies three properties:*

- ***Injectivity**: With probability* $1 - \mathsf{negl}(\lambda)$ *over the randomness of* $\mathsf{ek} \leftarrow \mathsf{LF.Gen}(1^\lambda, \text{injective})$*, the function* $\mathsf{LF.Eval}(\mathsf{ek}, \cdot)$ *is injective.*
- ***Lossiness** (with parameter* $\ell(\lambda)$*): With probability* $1 - \mathsf{negl}(\lambda)$ *over the randomness of* $\mathsf{ek} \leftarrow \mathsf{LF.Gen}(1^\lambda, \text{lossy})$*, the range of the function* $\mathsf{LF.Eval}(\mathsf{ek}, \cdot)$ *has size at most* $2^{\ell(\lambda)}$*.*
- ***Key Indistinguishability**: randomly sampled injective and lossy keys are computationally indistinguishable.*

5.2 Output-Intractable SHSFs from iO + Output-Intractable Puncturable PRFs

In this section, we note that the natural construction of SHSFs from (subexponential) iO and puncturable PRFs (following the [BLW17] construction of private constrained PRFs from iO) also yields output-intractable SHSFs from iO along with output-intractable puncturable PRFs. This fact will be used in all of our iO-based constructions.

Construction 5.5 (SHSF from IO)**.** Let $\mathsf{PRF} = \{F_s : \{0,1\}^{\nu(\lambda)} \to \{0,1\}^{\mu(\lambda)}\}$ denote a (puncturable) PRF family and let $\mathcal{O}$ denote an indistinguishability obfuscator. Then, PRF can be augmented with the algorithm Shift, defined as follows:

$$\mathsf{Shift}(s, f) = \mathcal{O}\Big(x \mapsto \mathsf{PRF}_s(x) + f(x)\Big).$$

Moreover, a shifted key $\mathsf{sk}_f \leftarrow \mathsf{Shift}(s, f)$ can be evaluated on an input x simply by interpreting sk_f as a program and evaluating $\mathsf{sk}_f(x)$.

Lemma 5.6. *Suppose that* PRF *is a* $2^{-\nu(\lambda)} \cdot \mathsf{negl}(\lambda)$*-secure puncturable PRF (Theorem 5.2, and* $\mathcal{O}$ *is a* $2^{-\nu(\lambda)} \cdot \mathsf{negl}(\lambda)$ *secure indistinguishability obfuscator (Theorem 5.1).*

Then, $(\mathsf{PRF}, \mathsf{Shift})$ *is a SHSF for bounded-size shift functions. Moreover, if the hash family* $\mathcal{H}_{\text{plain}}(\mathsf{msk}, x) = \mathsf{PRF}_{\mathsf{msk}}(x)$ *is output-intractable (or NAE-output-intractable) for a relation* R_{out}*, then the same is true for* SHSF*.*

Proof. For the first claim, it suffices to show that $(\mathsf{PRF}, \mathsf{Shift})$ satisfies correctness and shift-hiding. Correctness follows immediately from the correctness of $\mathcal{O}$.

To see that $(\mathsf{PRF}, \mathsf{Shift})$ is shift-hiding – namely, that $\mathsf{sk}_f \approx_c \mathsf{sk}_g$ for any pair of (bounded-size) circuits (f, g), we closely follows the CHCPRF security proof in [BLW17]. Namely, we appeal to a hybrid argument with $2^\nu + 2$ hybrid distributions on keys sk, defined as follows:

- Hyb_{-1}: $\mathsf{sk} = \mathsf{sk}_f \leftarrow \mathsf{Shift}(s, f) = \mathcal{O}\left(x \mapsto \mathsf{PRF}_s(x) + f(x)\right)$.

- For every $0 \leq x^* \leq 2^\nu - 1$ (interpreting x^* as both an integer and a string $\mathsf{Hyb}_{x^*} = \mathsf{sk} \leftarrow \mathcal{O}\left(x \mapsto \mathsf{PRF}_s(x) + g(x) \text{ if } x < x^*, x \mapsto \mathsf{PRF}_s(x) + f(x) \text{ if } x \geq x^*\right)$

- Hyb_{2^ν}: $\mathsf{sk} = \mathsf{sk}_g \leftarrow \mathsf{Shift}(s, g) = \mathcal{O}\left(x \mapsto \mathsf{PRF}_s(x) + g(x)\right)$.

We note that $\mathsf{Hyb}_{-1} \approx_{c,2^{-\nu}\mathsf{negl}(\lambda)} \mathsf{Hyb}_0$ and $\mathsf{Hyb}_{2^\nu-1} \approx_{c,2^{-\nu}\mathsf{negl}(\lambda)} \mathsf{Hyb}_{2^\nu}$ by the $2^{-\nu} \cdot \mathsf{negl}(\lambda)$-security of $\mathcal{O}$. Additionally, we note that $\mathsf{Hyb}_{x^*} \approx_{c,O(2^{-\nu}\cdot\mathsf{negl}(\lambda))} \mathsf{Hyb}_{x^*+1}$ for every $0 \leq x^* \leq 2^\nu - 2$ by a standard puncturing argument. This relies on the $2^{-\nu} \cdot \mathsf{negl}(\lambda)$-security of both the obfuscator and the puncturable PRF. This completes the proof of shift-hiding.

Finally, since the honest evaluation of the SHSF in Theorem 5.5 is identical to a puncturable PRF evaluation (with the same secret key), we note that the SHSF SHSF is (NAE) output-intractable for a relation R_{out} if and only if PRF is (NAE) output-intractable for the same relation R_{out}. Thus, by Theorem 3.1, in order to obtain correlation-intractable hash functions based on IO, we have reduced the problem to constructing output-intractable $2^{-\nu}$-secure puncturable PRFs. □

We now present two constructions of $2^{-\nu}$-secure puncturable PRFs, based on different assumptions.

5.3 Construction 1: Postcomposition with an Output-Intractable Hash

Construction 5.7. Let PRF denote a puncturable PRF family mapping $\{0,1\}^{\nu(\lambda)} \to \{0,1\}^{N(\lambda)}$. Let $\mathcal{H}$ denote an R_{out}-output intractable hash family mapping $\{0,1\}^{N(\lambda)} \to \{0,1\}^{\mu(\lambda)}$. Then, we define the PRF family $\mathsf{PRF}_{\mathcal{H}} = \mathcal{H} \circ \mathsf{PRF}$ as follows:

- A secret key for $\mathsf{PRF}_{\mathcal{H}}$ is a pair (k, sk) with $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)$ and $\mathsf{sk} \leftarrow \mathsf{PRF}.\mathsf{Gen}(1^\lambda)$.

- Evaluation is defined to be

$$\mathsf{PRF}_{\mathcal{H}}(k, \mathsf{sk}, x) = h(k, \mathsf{PRF}_{\mathsf{sk}}(x)).$$

Lemma 5.8. *Suppose that* PRF *is a* $2^{-\nu} \cdot \mathsf{negl}(\lambda)$*-secure puncturable PRF family that is* injective *with high probability,* $\mathcal{H}$ *is* R_{out}*-output intractable (or NAE-*R_{out}*-output intractable), and* $\mathcal{H}$ *has a* nearly uniform output distribution*, meaning that*

$$\left\{k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda), x \leftarrow \{0,1\}^{N(\lambda)} : (k, h(x))\right\}$$

$$\approx_{c,2^{-\nu}\cdot\mathsf{negl}(\lambda)} \left\{k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda), y \leftarrow \{0,1\}^{\mu(\lambda)} : (k, y)\right\}.$$

Then, $\mathsf{PRF}_{\mathcal{H}}$ *is a* $2^{-\nu} \cdot \mathsf{negl}(\lambda)$*-secure puncturable PRF family that is also* R_{out}*-output intractable (or NAE-*R_{out}*-output intractable).*

Proof. We first show output intractability. If an adversary $\mathcal{A}(k, \mathsf{sk})$ finds distinct (respectively, not-all-equal) inputs $(x_1, \ldots, x_t)$ such that $(h_k(\mathsf{PRF}_{\mathsf{sk}}(x_1), \ldots, h_k(\mathsf{PRF}_{\mathsf{sk}}(x_t))) \in R_{\text{out}}$ with non-negligible probability, then we claim that this violates the R_{out}-output intractability of $\mathcal{H}$. This holds because with all but negligible probability, $\mathsf{PRF}_{\mathsf{sk}}$ is an injective function, in which case the inputs $\mathsf{PRF}_{\mathsf{sk}}(x_1), \ldots, \mathsf{PRF}_{\mathsf{sk}}(x_t)$ to h_k are distinct (respectively, not-all-equal) as long as $x_1, \ldots, x_t$ are distinct (respectively, not-all-equal). This gives an attack on the R_{out}-output intractability of $\mathcal{H}$: given a key k, an adversary $\mathcal{A}'$ can sample sk, call $(x_1, \ldots, x_t) \leftarrow \mathcal{A}(k, \mathsf{sk})$, and output $(\mathsf{PRF}_{\mathsf{sk}}(x_1), \ldots, \mathsf{PRF}_{\mathsf{sk}}(x_t))$.

Next, we show that $\mathsf{PRF}_{\mathcal{H}}$ is a $2^{-\nu}\mathsf{negl}(\lambda)$-secure puncturable PRF family. To do so, we define a puncturing algorithm:

$$\mathsf{PRF}_{\mathcal{H}}.\mathsf{Puncture}(k, \mathsf{sk}, x^*) = (k, \mathsf{sk}\{x^*\}).$$

One can then verify that for $x \neq x^*$

$$\mathsf{PuncEval}((k, \mathsf{sk})\{x^*\}, x) = \mathsf{PRF}_{\mathcal{H}}(k, \mathsf{sk}, x).$$

Finally, $2^{-\nu} \cdot \mathsf{negl}(\lambda)$-pseudorandomness at punctured points follows from the analogous property for PRF along with the fact that $\mathcal{H}$ has a nearly uniform output distribution.

□

5.4 Construction 2: Precomposition with a Lossy Function

Construction 5.9. Let PRF denote a puncturable PRF family mapping $\{0,1\}^{N(\lambda)} \to \{0,1\}^{\mu(\lambda)}$. Let $\mathsf{LF} = (\mathsf{LF.Gen}, \mathsf{LF.Eval})$ denote a lossy function family mapping $\{0,1\}^{\nu(\lambda)} \to \{0,1\}^{N(\lambda)}$ and lossiness parameter $\ell(\lambda)$. Then, we define the PRF family $\mathsf{PRF}_{\mathsf{LF}} = \mathsf{PRF} \circ \mathsf{LF}$ as follows:

- A secret key for $\mathsf{PRF}_{\mathsf{LF}}$ is a pair $(\mathsf{sk}, \mathsf{ek})$ with $\mathsf{ek} \leftarrow \mathsf{LF.Gen}(1^\lambda, \text{injective})$ and $\mathsf{sk} \leftarrow \mathsf{PRF.Gen}(1^\lambda)$.
- Evaluation is defined to be

$$\mathsf{PRF}_{\mathsf{LF}}(\mathsf{sk}, \mathsf{ek}, x) = \mathsf{PRF}(\mathsf{sk}, \mathsf{LF.Eval}(\mathsf{ek}, x)).$$

Lemma 5.10. *Suppose that* PRF *is a* $\left(2^{N(\lambda)+\ell(\lambda)t(\lambda)}, 2^{-\nu(\lambda)} \cdot \mathsf{negl}(\lambda)\right)$*-secure puncturable PRF family, and suppose that* LF *is a lossy function family with lossiness parameter* $\tau(\lambda)$*.*
Then, for any relation R_{out} *with sparsity at most* $2^{-t(\lambda)\ell(\lambda)} \cdot \mathsf{negl}(\lambda)$*,* $\mathsf{PRF}_{\mathsf{LF}}$ *is a* $2^{-\nu} \cdot \mathsf{negl}(\lambda)$*-secure puncturable PRF family that is also* R_{out}*-output intractable.*

Moreover, if R_{out} *is also sparse whenever the inputs* $x_1, \ldots, x_t$ *are not-all-equal, then the PRF family satisfies NAE-*R_{out}*-output intractability.*

Proof. We first show puncturing-pseudorandomness. To do so, we define a puncturing algorithm

$$\mathsf{PRF}_{\mathcal{H}}.\mathsf{Puncture}(\mathsf{sk}, \mathsf{ek}, x^*) = (k, \mathsf{sk}\{\mathsf{LF.Eval}(x^*)\}).$$

Punctured evaluation correctness (with all but negligible probability over the sampling of $(\mathsf{sk}, \mathsf{ek})$) follows from the fact that ek is sampled in injective mode. Pseudorandomness follows directly from the pseudorandomness of PRF.

We next show output intractability. If an adversary $\mathcal{A}(\mathsf{sk}, \mathsf{ek})$ finds distinct (respectively, not-all-equal) inputs $(x_1, \ldots, x_t)$ such that

$$(\mathsf{PRF}_{\mathsf{sk}}(\mathsf{LF.Eval}(\mathsf{ek}, x_1)), \ldots, \mathsf{PRF}_{\mathsf{sk}}(\mathsf{LF.Eval}(\mathsf{ek}, x_t))) \in R_{\text{out}}$$

with non-negligible probability ϵ, then since ek is sampled in injective mode, the same claim holds where $(\mathsf{LF.Eval}(\mathsf{ek}, x_1), \ldots, \mathsf{LF.Eval}(\mathsf{ek}, x_t))$ are distinct (respectively, not-all-equal).

Then, by the security of LF, we also know that when $\mathsf{ek} \leftarrow \mathsf{LF.Gen}(1^\lambda, \text{lossy})$ is sampled from the *lossy* distribution, we have that

$$(x_1, \ldots, x_t) \leftarrow \mathcal{A}(\mathsf{sk}, \mathsf{ek}) : (\mathsf{LF.Eval}(\mathsf{ek}, x_1), \ldots, \mathsf{LF.Eval}(\mathsf{ek}, x_t)) \text{ are distinct}$$
$$\text{and } (\mathsf{PRF}_{\mathsf{LF}}(\mathsf{sk}, \mathsf{ek}, x_1), \ldots, \mathsf{PRF}_{\mathsf{LF}}(\mathsf{sk}, \mathsf{ek}, x_t)) \in R_{\text{out}} \geq \epsilon - \mathsf{negl}(\lambda).$$

Finally, we claim that in reality, with high probability over $(\mathsf{sk}, \mathsf{ek})$, *there do not exist such input tuples.* This follows from the pseudorandomness of PRF: for any fixed set S of size $2^{\ell(\lambda)}$, the probability that a random function F has an t-tuple of distinct (respectively, not-all-equal) inputs $z_1, \ldots, z_t$ from S such that $(F(z_1), \ldots, F(z_t)) \in R_{\text{out}}$ is at most $|S|^t \cdot \beta$ if R_{out} has sparsity β, which is negligible under our hypotheses. Picking $S = \mathrm{Im}(\mathsf{LF}(\mathsf{ek}, \cdot))$, we conclude that the same holds for the PRF family $\mathsf{PRF}_{\mathsf{sk}}$, as this condition can be tested in time $2^{N(\lambda)+\ell(\lambda)t(\lambda)}$ by enumeration. Thus, we obtain a contradiction, completing the proof of Theorem 5.10.

□

5.5 Putting it Together

Combining Theorem 3.1 and Theorem 5.6 with Theorem 5.8 and Theorem 5.10, respectively, we obtain our final constructions of correlation intractable hash families based on obfuscation. We restate the results (Theorem 1.6 and Theorem 1.5) from the introduction for completeness.

Theorem 5.11 (Theorem 1.6, restated)**.** *Assume the existence of*

1. *Subexponentially secure indistinguishability obfuscation,*
2. *Subexponentially secure one-way functions, and*
3. *A hash family $\mathcal{H}$ such that (i) $\mathcal{H}$ is R_{out}-output intractable, and (ii) for a random input X, $h_k(X)$ is $2^{-\nu} \cdot \mathsf{negl}(\lambda)$-indistinguishable from uniform (even given k).*

Then, there exists a hash family that is CI for shifted R_{out}-relations.

This follows by combining Theorem 3.1, Theorem 5.6, and Theorem 5.8.

Theorem 5.12 (Theorem 1.5, restated)**.** *Assume the existence of*

1. *Subexponential IO,*
2. *Subexponential OWFs, and*
3. *Lossy functions with input domain $\{0,1\}^\nu$ with a range of size $\leq 2^\ell$ in lossy mode.*

Then, there exists a hash family $\mathcal{H}$ that is CI for all (efficiently decidable) shifted t-ary output relations with sparsity at most $2^{-t\ell}$.

This follows by combining Theorem 3.1, Theorem 5.6, and Theorem 5.10.

Acknowledgements

We thank an anonymous reviewer for pointing out that the [PS19] hash function can likely also be shown to satisfy multi-input CI for shifted sum relations.

References

[AFV11] Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 21–40, Seoul, South Korea, December 4–8, 2011. Springer, Heidelberg, Germany.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th Annual ACM Symposium on Theory of Computing*, pages 99–108, Philadephia, PA, USA, May 22–24, 1996. ACM Press.

[AS15] Gilad Asharov and Gil Segev. Limits on the power of indistinguishability obfuscation and functional encryption. In Venkatesan Guruswami, editor, *56th Annual Symposium on Foundations of Computer Science*, pages 191–209, Berkeley, CA, USA, October 17–20, 2015. IEEE Computer Society Press.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B: 14th Theory of Cryptography Conference, Part II*, volume 9986 of *Lecture Notes in Computer Science*, pages 31–60, Beijing, China, October 31 – November 3, 2016. Springer, Heidelberg, Germany.

[BDGM20] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Factoring and pairings are not necessary for io: Circular-secure lwe suffices. *IACR Cryptology ePrint Archive*, 2020:1024, 2020.

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 533–556, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany.

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Heidelberg, Germany.

[BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In Hugo Krawczyk, editor, *PKC 2014: 17th International Conference on Theory and Practice of Public Key Cryptography*, volume 8383 of *Lecture Notes in Computer Science*, pages 501–519, Buenos Aires, Argentina, March 26–28, 2014. Springer, Heidelberg, Germany.

[BKM20] Zvika Brakerski, Venkata Koppula, and Tamer Mour. NIZK from LPN and trapdoor hash via correlation intractability for approximable relations. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 738–767, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Heidelberg, Germany.

[BLW17] Dan Boneh, Kevin Lewi, and David J. Wu. Constraining pseudorandom functions privately. In Serge Fehr, editor, *PKC 2017: 20th International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 10175 of *Lecture Notes in Computer Science*, pages 494–524, Amsterdam, The Netherlands, March 28–31, 2017. Springer, Heidelberg, Germany.

[BPW16] Nir Bitansky, Omer Paneth, and Daniel Wichs. Perfect structure on the edge of chaos - trapdoor permutations from indistinguishability obfuscation. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A: 13th Theory of Cryptography Conference, Part I*, volume 9562 of *Lecture Notes in Computer Science*, pages 474–502, Tel Aviv, Israel, January 10–13, 2016. Springer, Heidelberg, Germany.

[BR94] Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 232–249, Santa Barbara, CA, USA, August 22–26, 1994. Springer, Heidelberg, Germany.

[BV15] Zvika Brakerski and Vinod Vaikuntanathan. Constrained key-homomorphic PRFs from standard lattice assumptions - or: How to secretly embed a circuit in your PRF. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015: 12th Theory of Cryptography Conference, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 1–30, Warsaw, Poland, March 23–25, 2015. Springer, Heidelberg, Germany.

[BW13] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part II*, volume 8270 of *Lecture Notes in Computer Science*, pages 280–300, Bengalore, India, December 1–5, 2013. Springer, Heidelberg, Germany.

[CCH+19] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-Shamir: from practice to theory. In Moses Charikar and Edith Cohen, editors, *51st Annual ACM Symposium on Theory of Computing*, pages 1082–1090, Phoenix, AZ, USA, June 23–26, 2019. ACM Press.

[CCR16] Ran Canetti, Yilei Chen, and Leonid Reyzin. On the correlation intractability of obfuscated pseudorandom functions. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A: 13th Theory of Cryptography Conference, Part I*, volume 9562 of *Lecture Notes in Computer Science*, pages 389–415, Tel Aviv, Israel, January 10–13, 2016. Springer, Heidelberg, Germany.

[CCRR18] Ran Canetti, Yilei Chen, Leonid Reyzin, and Ron D. Rothblum. Fiat-Shamir and correlation intractability from strong KDM-secure encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 91–122, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Heidelberg, Germany.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th Annual ACM Symposium on Theory of Computing*, pages 209–218, Dallas, TX, USA, May 23–26, 1998. ACM Press.

[CMR98] Ran Canetti, Daniele Micciancio, and Omer Reingold. Perfectly one-way probabilistic hash functions (preliminary version). In *30th Annual ACM Symposium on Theory of Computing*, pages 131–140, Dallas, TX, USA, May 23–26, 1998. ACM Press.

[DNRS99] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. In *40th Annual Symposium on Foundations of Computer Science*, pages 523–534, New York, NY, USA, October 17–19, 1999. IEEE Computer Society Press.

[DVW20] Yevgeniy Dodis, Vinod Vaikuntanathan, and Daniel Wichs. Extracting randomness from extractor-dependent sources. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 313–342, Zagreb, Croatia, May 10–14, 2020. Springer, Heidelberg, Germany.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Santa Barbara, CA, USA, August 1987. Springer, Heidelberg, Germany.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *25th Annual Symposium on Foundations of Computer Science*, pages 464–479, Singer Island, Florida, October 24–26, 1984. IEEE Computer Society Press.

[GP20] Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. Cryptology ePrint Archive, Report 2020/1010, 2020. https://eprint.iacr.org/2020/1010.

[GR13] Oded Goldreich and Ron D. Rothblum. Enhancements of trapdoor permutations. *Journal of Cryptology*, 26(3):484–512, July 2013.

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

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 503–523, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.

[HL18] Justin Holmgren and Alex Lombardi. Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). In Mikkel Thorup, editor, *59th Annual Symposium on Foundations of Computer Science*, pages 850–858, Paris, France, October 7–9, 2018. IEEE Computer Society Press.

[HU19] Dennis Hofheinz and Bogdan Ursu. Dual-mode NIZKs from obfuscation. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019, Part I*, volume 11921 of *Lecture Notes in Computer Science*, pages 311–341, Kobe, Japan, December 8–12, 2019. Springer, Heidelberg, Germany.

[JLS20] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. Cryptology ePrint Archive, Report 2020/1003, 2020. https://eprint.iacr.org/2020/1003.

[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, *ACM CCS 2013: 20th Conference on Computer and Communications Security*, pages 669–684, Berlin, Germany, November 4–8, 2013. ACM Press.

[KRR17] Yael Tauman Kalai, Guy N. Rothblum, and Ron D. Rothblum. From obfuscation to the security of Fiat-Shamir for proofs. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 224–251, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany.

[LV20] Alex Lombardi and Vinod Vaikuntanathan. Fiat-shamir for repeated squaring with applications to PPAD-hardness and VDFs. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 632–651, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Heidelberg, Germany.

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th Annual Symposium on Foundations of Computer Science*, pages 372–381, Rome, Italy, October 17–19, 2004. IEEE Computer Society Press.

[Nak08] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Technical report, Manubot, 2008. https://git.dhimmel.com/bitcoin-whitepaper/.

[Pei16] Chris Peikert. A decade of lattice cryptography. *Foundations and Trends in Theoretical Computer Science*, 10(4):283–424, 2016.

[PS96] David Pointcheval and Jacques Stern. Security proofs for signature schemes. In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 387–398, Saragossa, Spain, May 12–16, 1996. Springer, Heidelberg, Germany.

[PS18] Chris Peikert and Sina Shiehian. Privately constraining and programming PRFs, the LWE way. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018: 21st International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 10770 of *Lecture Notes in Computer Science*, pages 675–701, Rio de Janeiro, Brazil, March 25–29, 2018. Springer, Heidelberg, Germany.

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 89–114, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

[PS20] Chris Peikert and Sina Shiehian. Constraining and watermarking PRFs from milder assumptions. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020: 23rd International Conference on Theory and Practice of Public Key Cryptography, Part I*, volume 12110 of *Lecture Notes in Computer Science*, pages 431–461, Edinburgh, UK, May 4–7, 2020. Springer, Heidelberg, Germany.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 187–196, Victoria, BC, Canada, May 17–20, 2008. ACM Press.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th Annual ACM Symposium on Theory of Computing*, pages 475–484, New York, NY, USA, May 31 – June 3, 2014. ACM Press.

[Zha16] Mark Zhandry. The magic of ELFs. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 479–508, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany.