Indistinguishability Obfuscation from Circular Security

Romain Gay
Cornell Tech
romain.rgay@gmail.com

Rafael Pass*
Cornell Tech
rafael@cs.cornell.edu

August 21, 2020

Abstract

We show the existence of indistinguishability obfuscators ($i\mathcal{O}$) for general circuits assuming subexponential security of:

- the Learning with Error (LWE) assumption (with subexponential modulus-to-noise ratio);
- the Decisional Composite Residuosity (DCR) assumption; and,
- a *circular security conjecture* regarding the Gentry-Sahai-Water's (GSW) and the Damgård-Jurik (DJ) encryption schemes.

More precisely, the circular security conjecture states that a notion of leakage-resilient security (which we refer to as "shielded randomness leakage security") satisfied by GSW (assuming LWE) is retained in the presence of a key-cycle w.r.t. GSW and DJ.

Our work thus places $i\mathcal{O}$ on qualitatively similar assumptions as (unlevelled) FHE, for which known constructions also rely on a circular security conjecture.

*Supported in part by NSF Award SATC-1704788, NSF Award RI-1703846, AFOSR Award FA9550-18-1-0267, DARPA SIEVE award HR00110C0086, and a JP Morgan Faculty Award. This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via 2019-19-020700006. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

1 Introduction

The goal of *program obfuscation* is to "scramble" a computer program, hiding its implementation details (making it hard to "reverse-engineer"), while preserving its functionality (i.e, its input/output behavior). In recent years, the notion of *indistinguishability obfuscation ($i\mathcal{O}$)* [BGI+01, GGH+13b] has emerged as the central notion of obfuscation in the cryptographic literature: roughly speaking, this notion requires that obfuscations $i\mathcal{O}(\Pi_1)$, $i\mathcal{O}(\Pi_2)$ of any two *functionally equivalent* circuits Π_1 and Π_2 (i.e., whose outputs agree on all inputs) from some class $\mathcal{C}$ (of circuits of some bounded size) are computationally indistinguishable.

On the one hand, this notion of obfuscation is strong enough for a plethora of amazing applications (see e.g., [SW14, BCP14, BZ14, GGHR14, KNY14, KMN+14, BGL+15, CHJV14, KLW15, CLP15, BPR15, BPW16, BP15].) On the other hand, it may also plausibly exist, whereas stronger notion of obfuscations have run into strong impossibility results, even in idealized models (see e.g., [BGI+01, GK05, CKP15, Ps16, MMN15, LPST16]) Since the original breakthrough candidate constuction of an $i\mathcal{O}$ due to Garg, Gentry, Halevi, Raykova, Sahai, and Waters [GGH+13b], there has been an intensive effort toward obtaining a construction of $i\mathcal{O}$ based on some form of well-studied/nice assumptions. The original work [GGH+13b] provided a *candidate* construction based on high-degree multilinear maps (MLMs) [GGH13a, CLT13, GGH15, CLT15]; there was no proof of security based on an intractability assumption. [PST14] provided the first construction with a reduction-based proof of security, based on a strong notion of security for MLMs, similar to a sort of "Uber assumption". [GLSW14] provided a construction based on a more concrete assumption relying on composite-order MLMs. Unfortunately, both assumptions have been broken for specific candidate constructions of MLMs [CHL+15, MF15].

iO from FE or XiO. Subsequently, several works have been constructing $i\mathcal{O}$ from seemingly weaker primitives, such as Functional Encryption (FE) [AJ15, BV15] or $Xi\mathcal{O}$ [LPST16], while only using standard assumptions, such as Learning with Error (LWE). For both constructions, we actually need to rely on *subexponentially-secure* constructions of either FE or $Xi\mathcal{O}$, as well as subexponential security of LWE. Let us recall the notion of $Xi\mathcal{O}$ as it will be useful to us: roughly speaking, an $Xi\mathcal{O}$ is an $i\mathcal{O}$ with a very weak "exponential" efficiency requirement: the obfuscator is allowed to run in polynomial time in the size of the truth table of the function to be obfuscated, and it is only required that its outputs a program that "slightly" compresses the truth table (technically, it is sublinear in its size).

A breakthrough result by Lin [Lin16] showed how to obtain $i\mathcal{O}$ from just constant-degree MLMs (plus standard assumptions), overcoming the black-box barriers in [Ps16, MMN15]. Her construction relies on the connection between FE and $i\mathcal{O}$. Following this result, a sequence of works (see e.g., [LV16, Lin17, LT17, Agr19, AJKS18, JS18, JLMS19, AJL+19, AP20]) reduced the assumptions and the degree of the MLM, relying on certain types of low-degree pseudorandom generators (PRGs) to instantiate either FE or $Xi\mathcal{O}$. This culminated in the recent work of [GJLS20], which (other than standard assumptions) relies an LWE assumption with *binary noise* in the presence of leakage of the LWE noise: the leakage is a *constant-degree* PRG applied to the LWE noise. While this most recent construction overcomes known attacks, many of the earlier low-degree PRG assumptions used to obtain $i\mathcal{O}$ have been broken using semi-definite programming/sum-of-square algorithms [BBKK18, BHJ+18], and the construction in [GJLS20] requires carefully setting the parameters so as to avoid these attacks.

A very recent work by Brakerski et al [BDGM20] presents a new type of candidate construction of $Xi\mathcal{O}$ by combining a fully-homomorphic encryption (FHE) and a linear-hommorphic encryption (LHE) with certain nice properties (which can be instantiated by the Damgård-Jurik (DJ) [DJ01] encryption scheme whose security can be based on the Decisional Composite Residuosity (DCR) assumption), and relying on a random oracle. More precisely, they define a new primitive called "split-FHE" and provide a candidate construction of it based on the above primitives and a random oracle, and next show how split-FHE implies $Xi\mathcal{O}$ (which by earlier work implies $i\mathcal{O}$ under standard assumptions). We highlight that [BDGM20] does not provide any proof of security of the split-FHE construction (even in the random oracle model), but rather informally argue some intuitions, which include a) *circular security* (more on this below) of the FHE and the LHE, and b) a *"correlations conjecture"* that the FHE randomness (after FHE evaluations) does not correlate "too much" with messages being encrypted. The correlation conjecture is not formalized, as the FHE randomness in known construction actually *does* depend on the message, so the authors simply conjecture that this correlation cannot be exploited by an attacker to break security of the $i\mathcal{O}$ (they also provide heuristic methods to weaken the correlations); as such they only get a heuristic construction.

Summarizing the above, while there have been enormous progress on realizing $i\mathcal{O}$, known constructions are either based on assumptions on primitives that are not well understood (high-degree MLMs, or various low-degree PRGs assumptions), or the construction candidates simply do not have proofs of security.

1.1 Our Results

In this work, we provide a new $i\mathcal{O}$ construction assuming subexponential security of (a) the LWE assumption (with subexponential modulus-to-noise ratio), (b) the DCR assumption, and (c) a natural *circular security conjecture* w.r.t the Gentry-Sahai-Water's (GSW) [GSW13] FHE scheme, and the DJ [DJ01] LHE scheme. On a high-level, our approach follows that in [BDGM20], but we show how to remove the heuristic arguments while instead relying on a concrete circular security assumption. We believe this constitutes strong evidence for the existence of $i\mathcal{O}$, and places $i\mathcal{O}$ on a similar footing as *unlevelled* FHE (i.e., an FHE that support an a-priori unbounded polyonomial number of operations), for which known constructions also rely on a circular security conjecture [Gen09].

Circular security. Circular security of encryption schemes [CL01, BRS02] considers a scenario where the attacker gets to see not only encryptions of messages, but also *encrypted key cycles*. A particularly simple type of circular security, referred to as *2-circular security* considers an encrypted key cycle of size 2—the attacker gets to see public keys $\mathsf{pk}_1, \mathsf{pk}_2$, a length 2 encrypted secret key cycle, $\mathsf{Enc}^1_{\mathsf{pk}_1}(\mathsf{sk}_2), \mathsf{Enc}^2_{\mathsf{pk}_2}(\mathsf{sk}_1)$, and we require that security of $\mathsf{Enc}^1_{\mathsf{pk}_1}$ still holds (i.e., for any m_0, m_1, $\mathsf{Enc}^1_{\mathsf{pk}_1}(m_0)$ is indistinguishable from $\mathsf{Enc}^1_{\mathsf{pk}_1}(m_0)$. Such encrypted key cycles commonly arise in applications of encryption scheme such as storage systems such as Bitlocker, anonymous credentials [CL01] and most recently to construct (unlevelled) FHE [Gen09]. We refer to the assumption that:

If Enc^1 *and* Enc^2 *are semantically secure, then 2-circular security holds w.r.t.* $\mathsf{Enc}^1, \mathsf{Enc}^2$

as the *2-circular security conjecture (*$\mathsf{2CIRC}$*) w.r.t* $\mathsf{Enc}^1, \mathsf{Enc}^2$. For our purposes, we will allow the key generation procedure of Enc^1 to get the public-key pk_2 of Enc^2 as an input—for instance, this will allow Enc^1 and Enc^2 to operate over the same field.

At first sight, one may be tempted to hope that the $\mathsf{2CIRC}$ holds w.r.t. *any* two encryption schemes $\mathsf{Enc}^1, \mathsf{Enc}^2$—after all, the attacker never actually gets to see the secret key, but rather encryptions of it, which intuitively should hide it by semantical security of the encryption schemes. Yet, in recent years, counter examples to 2-circular security for public-key encryption schemes have been found—first based on obfuscation [MO14, KRW15], and subsequently also based on more standard assumptions [BHW15, KW16, GKW17]. However, all the counter examples are highly artificial, and require carefully embedding some trapdoor mechanism in the encryption scheme that enable decrypting the cirphertext once you see an encryption of the secret key. As far as we are aware, no "natural" counterexamples to $\mathsf{2CIRC}$ are known. Indeed, a natural heuristic consist of simply assuming that $\mathsf{2CIRC}$ holds for all "natural" encryption schemes—we refer to this as the $\mathsf{2CIRC}$ *heuristics*. We note that the $\mathsf{2CIRC}$ heuristic is very similar to the Random Oracle Heuristic [BR93]—while "contrived" counterexamples are known (see e.g., [CGH98, MRH04]), it is still a commonly used heuristic for the design on practical protocols.

In this work, we will consider a natural strengthening of the $\mathsf{2CIRC}$ heuristic: We consider whether stronger forms of security of Enc^1 are preserved in the presence of a key cycle. More precisely, we will consider a notion of $\mathcal{O}$-leakage resilient security for Enc^1 where $\mathcal{O}$ is some particular *randomness leakage oracle*; this notion enhances the standard semantical security notion by providing the attacker with some leakage $\mathcal{O}(m, r)$ on the randomness r used to encrypt the message m. We say that the $\mathsf{2CIRC}^{\mathcal{O}}$ *conjecture holds w.r.t* $\mathsf{Enc}^1, \mathsf{Enc}^2$ if the following holds:

> If Enc^1 is $\mathcal{O}$-leakage resilient secure and Enc^2 is secure, then $\mathcal{O}$-leakage resilient security of Enc^1 is preserved in the presence of a length 2 key-cycle w.r.t. Enc^1 and Enc^2.

We will also consider a *subexponential* $\mathsf{2CIRC}^{\mathcal{O}}$ conjecture which is identically defined except it considers subexponential (as opposed to polynomial) security of the encryption schemes.

Our main theorem shows that for a natural notion of randomness leakage $\mathcal{O}_{\mathsf{SRL}}$—which will be referred to as "shielded randomness leakage"—$\mathsf{2CIRC}^{\mathcal{O}_{\mathsf{SRL}}}$ w.r.t. two standard encryption schemes (GSW and DJ) together with standard assumptions implies the existence of $i\mathcal{O}$.

Theorem 1.1 (Informally stated)**.** *Assume subexponential security of the LWE (with subexponential modulus-to-noise ratio) and the DCR assumptions, and assume that the subexponential* $\mathsf{2CIRC}^{\mathcal{O}_{\mathsf{SRL}}}$ *conjecture holds w.r.t. GSW and DJ. Then, $i\mathcal{O}$ exists for the class of polynomial-size circuits.*

Or, informally, assuming the "subexponential $\mathsf{2CIRC}^{\mathcal{O}_{\mathsf{SRL}}}$ heuristic", subexponential security of LWE and DCR implies the existence of $i\mathcal{O}$.

Shielded Randomness Leakage (SRL) Security. As mentionned above, we consider a notion of *shielded randomness leakage (SRL)* security for FHE. Roughly speaking, the attacker gets to see an FHE encryption $c = \mathsf{FHE}(m; r)$, and next gets access to a "leakage oracle" $\mathcal{O}_{\mathsf{SRL}}(m, r)$ which upon every invocation sends the attacker an "extra noisy" encryption $c^\star = \mathsf{FHE}(0; r^\star)$ of 0—we will refer to the random string $r^\star$ as the "shield". Next, the attacker can select some functions f and value α such that $f(m) = \alpha$ (i.e., we restrict the attacker to picking functions for which it knows the output when applying the function to the message m; this restriction will soon become clear). Finally, the oracle homomorphically evaluates f on the ciphertext c, letting $c_f = \mathsf{FHE}(f(m); r_f)$ denote the evaluated ciphertext, and returns $r^\star - r_f$. That is, the attacker gets back the randomness r_f of the evaluated ciphertext masked by the "shield" $r^\star$. The reason why the attacker is restricted to picking functions f for which it knows the output α is that for the FHE we consider, given $c^\star$ and c_f, the attacker can compute $c^\star - c_f = \mathsf{FHE}(0 - f(m); r^\star - r_f)$ and thus knowing $r^\star - r_f$ reveals $f(m)$. So, by restricting to attackers that already know $\alpha = f(m)$, intuitively, $r^\star - r_f$ does not reveal anything else. Indeed, we formally prove that under the LWE assumption, the GSW encryption scheme is *SRL-secure*—that is, $\mathcal{O}_{\mathsf{SRL}}$-leakage resilient secure.

Theorem 1.2 (Informally stated)**.** *Assume the LWE (with subexponential modulus-to-noise ratio) assumption holds. Then, the GSW scheme is SRL-secure.*

On a very high-level, the idea behind the proof is that the encryption $c^\star$ is a projection, $h_{\mathbf{A}}(\mathbf{r}^\star) = \mathbf{A}\mathbf{r}^\star \in \mathbb{Z}_N$, where the randomness $\mathbf{r}^\star$ used to produce $c^\star$ is a vector in $\mathbb{Z}_N^m$ and $\mathbf{A}$ is a matrix in $\mathbb{Z}_N^{n\times m}$

where $m \gg n$, that is, the map $h_{\mathbf{A}}$ that describes the encryption is compressing. Therefore, some "components" of the "shield" $\mathbf{r}^\star$ remain information-theoretically hidden. And this enables hiding the same component of $\mathbf{r}_f$; furthemore, the component that is not hidden by $\mathbf{r}^\star$ is actually already revealed by $f(m)$, which the attacker knows (as we require it to output $\alpha = f(m)$). The formal proof of this proceeds by considering a (simplified) variant of the Micciancio-Peikert lattice trapdoor method [MP12] for generating the matrix $\mathbf{A}$ (which is part of the public key for GSW) together with a trapdoor that enables sampling short preimages to $h_{\mathbf{A}}$ (i.e. solving the ISIS problem). Whereas traditional trapdoor preimage sampling methods require the preimage to be sampled according to some specific distribution (typically discrete Gaussian) over preimages, we will consider a somewhat different notion: we require that given a target vector $\mathbf{t}$, the distribution of randomly sampled preimages of $\mathbf{t}$ is statistically close to the distribution obtained by starting with any "short" preimage $\mathbf{w}$ of $\mathbf{t}$ and next adding a randomly sampled preimage of 0. Our proof relies on the fact that randomly sampled preimages can be sufficiently longer than $\mathbf{w}$ to ensure that they "smudge" $\mathbf{w}$—we here rely on the fact that modulus-to-noise ratio is subexponential (to enable the smudging).[1]

1.2 Overview of the $Xi\mathcal{O}$ Construction

We proceed to explain our construction of an $Xi\mathcal{O}$. This construction will only rely on *polynomial* security of LWE, DCR and $\mathsf{2CIRC}^{\mathcal{O}_{\mathsf{SRL}}}$ w.r.t. GSW and DJ; to obtain a *subexponentially-secure* $Xi\mathcal{O}$ (which is required to obtain $i\mathcal{O}$ by [LPST16]), we need to strengthen these assumptions to also require subexponential security.

As mentionned above, on a high-level, our construction follows similar intuitions as the BDGM construction. We combine an FHE (in our case the GSW FHE) with the DJ LHE to implement an $Xi\mathcal{O}$. In fact, in our approach, we do not directly construct an $Xi\mathcal{O}$, but rather construct an $Xi\mathcal{O}$ with *preprocessing*—this notion, which relaxes $Xi\mathcal{O}$ by allowing the obfuscator to have access to some *long* public parameter pp, was actually already considered in [LPST16] and it was noted there that subexponentially-secure $Xi\mathcal{O}$ with preprocessing also suffices to get $i\mathcal{O}$.

Towards explaining our approach, let us first recall the approach of BDGM using a somewhat different langauge that will be useful for us.

The BDGM construction. The high-level idea is quite simple and very elegant. Recall that an $Xi\mathcal{O}$ is only required to work for programs Π with polynomially many inputs $n = \mathsf{poly}(\lambda)$ where λ is the security parameter, and the obfuscators running time is allowed to be polynomial in n; the only restriction is that the obfuscated code should be sublinear in n—we require a "slight" compression of the truth table. More precisely, the obfuscator is allowed to run in time $\mathsf{poly}(n, \lambda)$ (i.e., polynomial time in the size of the truth table), but must output a circuit of size $\mathsf{poly}(\lambda)n^{1-\varepsilon}$ where $\varepsilon > 0$. Assume that we have access to a special "batched" FHE which enables encrypting (and computing on) long messages of length, say m using a *short randomness* of length $\mathsf{poly}(\lambda)\log m$; and furthermore that 1) given the secret key, and a ciphertext c, we can efficiently recover the ciphertext randomness, and 2) given a ciphertext c and its randomness—which will also be referred as a "hint"—one can efficiently decrypt. Given such a special FHE, it is easy to construct an $Xi\mathcal{O}$: simply cut the truth table into "chunks" of length n^ε, FHE encrypt the circuit C_Π that given an index $\mathbf{y} \in \{0,1\}^{\log(n^{1-\varepsilon})}$ computes the $\mathbf{y}$'th "chunk" of the truth table, and next, evaluates the FHE for each index $\mathbf{y}$ and releases the randomness $r_{\mathbf{y}}$ (i.e., the "hint") of the evaluated ciphertexts. These hints enable compressing n^ε bits into $\mathsf{poly}(\lambda)\log(n^\varepsilon)$ bits and thus the $Xi\mathcal{O}$ is compressing.[2]

[1]As we can use smudging, our lattice trapdoor sampling construction and proof also becomes easier than the one one [MP12].

[2]The reason we need to cut the truth table into chunks and don't just directly compute the whole output is that the FHE may have a public key that depends polynomially on the length of the messages to be encrypted, so the final

Unfortunately, none of the known FHE constructions have short randomness. BDGM, however, observes that there are *linear* homomorphic encryptions schemes (LHE), notably the DJ encryption scheme, that satisfy the above requirements. Now, note that many FHEs are batcheable (with "long" randomness), but have "essentially" linear decryption: decryption is a linear operation on the ciphertext and secret key, and next rounding. So, if we start off with such an FHE, and additionally release an LHE of the FHE secret key, we can get an FHE of the desired "batcheable with short randomness" requirement: we first (linear) homomorphically evaluate the decryption of the FHE ciphertext, and simply release the randomness for the evaluated LHE ciphertext (which now is short).

But there are problems with this approach: 1) since FHE decryption requires performing both a linear operation and *rounding*, we are leaking not only $C_\Pi(\mathbf{y})$ but also the FHE noise of the evaluated ciphertext, and 2) the LHE randomness may actually leak more than just the decrypted message (i.e., something about how the ciphertext was obtained). As BDGM shows, both of these problems can be easily overcome if we have access to many fresh LHE encryptions of some "smudging" noise (which is large enough to smudge the FHE noise).[3] So the only remaining problem is to generate these LHE encryptions of smudging noise. It is here that the construction in BDGM becomes heuristic: (1) they propose to use a random oracle to generate a long sequence of randomness; (2) this sequence of randomness can be interpreted as LHE encryptions of uniformly random strings $\{u_\mathbf{y}\}$ (as the DJ LHE has dense ciphertext); (3) next, if we additionally provide an FHE encryption of the LHE secret key $\overline{\mathsf{sk}}$ (note that we now have a circular security issue), we can FHE-homomorphically evaluate a function f that decrypts the LHE ciphertexts produced by the random oracle, and computes (minus) the most significant bits of $u_\mathbf{y}$; and, (4) we finally LHE-evaluate the (partial) decryption of the evaluated FHE ciphertexts (of minus the most significant bits of $u_\mathbf{y}$); the obtained LHE ciphertexts can now be combined with the LHE ciphertexts from the random oracle to get an LHE encryption of noise $u_\mathbf{y} - \mathsf{MSB}(u_\mathbf{y})$ of the appropriate size, i.e. smudging but not uniform.

The problem with this approach, however, is that while we do obtain an LHE encryption of appropriate smudging noise, it is not not actually a fresh ciphertext (with fresh randomness). The issue is that the randomness $r^f_\mathbf{y}$ of the evaluated ciphertext of (minus) the most significant bits may (and actually will) depend on the randomness of the original LHE ciphertext obtained by the RO. Furthermore, since LHE can only compute the first step of an FHE decryption (namely, inner product), the LHE encryption obtained actually encrypts: $u_\mathbf{y} - \mathsf{MSB}(u_\mathbf{y}) + \mathsf{noise}$. As we know, revealing the extra noise is detrimental for security (this is why we are generating LHE encryptions of smudging noises in the first place). Unfortunately, the noise that results from partially decrypting the FHE ciphertext depends on $u_\mathbf{y}$, so the lower-order bits of the latter cannot smudge the former. BDGM here simply assumes that the attacker cannot exploit these correlations, and thus only obtain a heuristic construction.

We shall now see how to obtain the appropriate LHE encryption of smudging noise in a provably secure way, relying on *circular SRL-security* of GSW and DJ—that is, $\mathcal{O}_{\mathsf{SRL}}$-leakage resilient circular security holds w.r.t. GSW and DJ.

Removing the RO. Our first task will be to remove the use of the RO. That will actually be very easy: as we have already observed, it suffices to get an $Xi\mathcal{O}$ with preprocessing to obtain $i\mathcal{O}$, so instead of using a random oracle, we will simply just a long random string as a public parameter, and interpret it as LHE encryptions of random strings.

obfuscation is only compressing when we have a large number of chunks.

[3] They formally prove the security of their scheme in an idealized model where we have access to an oracle that generates fresh LHE encryptions of smudging noise.

Re-encrypting the FHE. The trickier problem will be to deal with the issue of correlations. We will here rely on the fact that we are considering a particular instantiation of the FHE: namely, using (a batched version of) the GSW encryption scheme. On a high-level, the idea for breaking the correlation is to re-encrypt the evaluated FHE ciphertext (of the most significant bits of the random LHE plaintext) to ensure that the randomness is fresh and independent of the evaluations. That also gives a decryption noise than itself is also independent of the evaluated circuit. If we had access to a fresh extra noisy FHE encryption of 0, then it would be easy to re-randomize a GSW ciphertext: simply add the encryption of 0. So, how do we get an encryption of 0? GSW ciphertexts are not dense, so we cannot put them in the public parameters, and even if they were, we still wouldn't be able to get an encryption of 0 (we would have an encryption of a uniformly random plaintext). However, the public key of the GSW encryption scheme actually contains a bunch of encryptions of 0, but fewer than the amount we need (or else we wouldn't get a compressing $Xi\mathcal{O}$). Instead, we use the public key of the GSW encryption to generate extra noisy encryptions of 0, and we include the (many) random coins $\{r^\star_{\mathbf{y}}\}_{\mathbf{y}}$ used to generate these ciphertexts as part of public parameter of the $Xi\mathcal{O}$ (recall that the public parameter can be as long as we want). This method does indeed enable us to get a fresh FHE encryption of (minus) the most significant bits, and thus the correlation has be broken and intuitively, we should be able to get a provably secure construction. But two obstacles remain: (1) we are revealing the randomness used to re-randomize the ciphertexts, and this could hurt security, or render the re-randomization useless and (2) we still have a circular security issue (as we FHE encrypt the LHE secret key, and LHE encrypt the FHE secret key). Roughly speaking, the first issue will be solved by relying on SRL-security of GSW, and the second issue will be solved by our circular security conjecture.

In more detail, we note that the re-randomized evaluated FHE ciphertext of $-\mathsf{MSB}(u_{\mathbf{y}})$ and the public parameters $r^\star_{\mathbf{y}}$ are statistically close to freshly generated extra noisy FHE encryption of $-\mathsf{MSB}(u_{\mathbf{y}})$ using randomness $r^\star_{\mathbf{y}}$, and setting the public parameter to $r^\star_y - r^f_{\mathbf{y}}$, where $r^f_{\mathbf{y}}$ is the randomness of the evaluated ciphertext, before re-randomization. In other words, the re-randomization achieves a notion which we refer to as "weak circuit privacy", where the re-randomized ciphertext is independent of the evaluated function f. Furthermore, noisy GSW encryptions of $-\mathsf{MSB}(u_{\mathbf{y}})$ essentially have the form of a noisy GSW encryption of 0, to which $-\mathsf{MSB}(u_{\mathbf{y}})$ is added. So, other than $\mathsf{MSB}(u_{\mathbf{y}})$, which is truly random, $r^\star_y - r^f_{\mathbf{y}}$ is simply an SRL leakage on a GSW encryption of the LHE secret key $\overline{\mathsf{sk}}$! Thus, intuitively, security should now follows from circular SRL security of GSW and DJ.

The final construction. We summarize the final construction of an XiO with preprocessing. The public parameter pp will be a long *random* string that consists of two parts:

- The first part $\mathsf{FHE.PubCoin}$ will interpreted as a sequence of rerandomization vectors $\mathbf{r}^\star$;
- The second part $\mathsf{LHE.PubCoin}$ will be interpreted as sequence of LHE encryptions;

The obfuscator, given a security parameter λ and a circuit $\Pi : \{0,1\}^{\log n} \to \{0,1\}$, where $n = \mathsf{poly}(\lambda)$ proceeds as follows:

- **Output Public keys for FHE and LHE:** Generate a fresh key-pair $(\overline{\mathsf{pk}}, \overline{\mathsf{sk}})$ for the DJ LHE, and next generate a key-pair $(\mathsf{pk}, \mathsf{sk})$ for the GSW FHE. (To make it easier for the reader to remember which key refers to which encryption scheme, we place a *line* over all keys, ciphertext and algorithms, that correspond to the *linear* homomorphic encryption.) The modulus N of the GSW encryption is set to be the same as that used for the LHE. Additionally, we let N be large enough to enable encrypting messages of size n^ε. Finally, output the public keys $(\mathsf{pk}, \overline{\mathsf{pk}})$.

- **Output an FHE encryption of the circuit:** Let ct_1 be an FHE encryption (w.r.t. pk) of the circuit C_Π that given an index $\mathbf{y} \in \{0,1\}^{\log(n^{1-\varepsilon})}$ computes the $\mathbf{y}$'th length n^ε "chunk" of the truth table of Π.
- **Output encrypted key cycle:** Let ct_2 be an FHE encryption of $\overline{\mathsf{sk}}$, and let $\overline{\mathsf{ct}}$ be an LHE encrytion of sk. Output the key cycle $\mathsf{ct}_2, \overline{\mathsf{ct}}$.
- **Output hints:** For every $\mathbf{y} \in \{0,1\}^{\log(n^{1-\varepsilon})}$, compute a short "hint" $r_\mathbf{y}$ as follows:
  - **Evaluate the circuit:** Use ct_1 to homomorphically evaluate C_Π (which is encrypted in ct_1) on $\mathbf{y}$, and let $\mathsf{ct}_{1,\mathbf{y}}$ denote the FHE encrypted result.
  - **Compute an FHE encryption $\mathsf{ct}_{\mathsf{MSB}}$ of $\mathsf{MSB}(u_\mathbf{y})$:** Consider the function $f(\Pi, \overline{\mathsf{sk}'})$ that ignores the input Π but uses the input $\overline{\mathsf{sk}'}$ to decrypts the LHE ciphertext in the $\mathbf{y}$'th chunk of from $\mathsf{LHE.PubCoin}$ into a plaintext $u_\mathbf{y}$ and outputs $-\mathsf{MSB}(u_\mathbf{y})$ Homomorphically evaluate f on the ciphertexts $\mathsf{ct}_1, \mathsf{ct}_2$ (where, recall, ct_2 is an encryption of $\overline{\mathsf{sk}}$). Let $\mathsf{ct}_{\mathsf{MSB}} = \mathsf{FHE}(-\mathsf{MSB}(u_\mathbf{y}); \mathbf{r}^f_\mathbf{y})$ denote the resulting FHE ciphertext.
  - **Rerandomize $\mathsf{ct}_{\mathsf{MSB}}$ into $\mathsf{ct}'_{\mathsf{MSB}}$:** Use the $\mathbf{y}$'th chunk of $\mathsf{FHE.PubCoin}$ to get the randomness $\mathbf{r}^\star_\mathbf{y}$; generate an extra noisy FHE encryption of 0 using $\mathbf{r}^\star$ and homomorphically add it to $\mathsf{ct}_{\mathsf{MSB}}$. Let $\mathsf{ct}'_{\mathsf{MSB}} = \mathsf{FHE}(\mathsf{MSB}(u_\mathbf{y}); \mathbf{r}^\star_\mathbf{y} + \mathbf{r}^f_\mathbf{y})$ denote the new (re-randomized) ciphertext.
  - **Proxy re-encrypt $\mathsf{ct}^1_\mathbf{y}$ as LHE ciphertext $\overline{\mathsf{ct}}_y$:** Use $\overline{\mathsf{ct}}$ (which, recall, is an LHE encryption of sk) to homomorphically performing the *linear* part of decrypting $\mathsf{ct}_{1,\mathbf{y}}$ into $\widetilde{m}_\mathbf{y}$ and $\mathsf{ct}_{\mathsf{MSB}}$ into $-\mathsf{MSB}(u_\mathbf{y})$, and computing $m_\mathbf{y} = \widetilde{m}_\mathbf{y} + u_\mathbf{y} - \mathsf{MSB}(u_y)$ where $u_\mathbf{y}$ is the value LHE encrypted in the $\mathbf{y}$'th chunk of $\mathsf{LHE.PubCoin}$. Let $\overline{\mathsf{ct}}_y$ denote the resulting LHE ciphertext (encrypting $m_\mathbf{y}$).
  - **Release hint $r_\mathbf{y}$ for LHE ciphertext $\overline{\mathsf{ct}}_\mathbf{y}$:** Use $\overline{\mathsf{sk}}$ to recover the randomness $r_\mathbf{y}$ of $\overline{\mathsf{ct}}_\mathbf{y}$ (recall that the LHE we use has a randomness recoverability property).

To evaluate the obfuscated program on an input $\mathbf{x} \in \{0,1\}^n$, parse $\mathbf{x}$ as $\mathbf{y}||\mathbf{z}$ where $\mathbf{y} \in \{0,1\}^{\log(n^{1-\varepsilon})}$ and $\mathbf{z} \in \{0,1\}^{\log(n^{1-\varepsilon})}$. Compute $\overline{\mathsf{ct}}_\mathbf{y}$ just like the obfuscator does (note that this does not require knowing the secret key, but only information contained in the obfuscated code). Finally, decrypt $\overline{\mathsf{ct}}_\mathbf{y}$ using the randomness $r_\mathbf{y}$ to recover the message m_y (recall that the LHE we use has the property that ciphertexts can be decrypted if you know the randomness). Finally, perform the rounding step of FHE decryption on $m_\mathbf{y}$ to obtain $m'_\mathbf{y}$ and output the $\mathbf{z}$'th bit of $m'_\mathbf{y}$.

We emphasize that the above description oversimplifies the actual construction. One particular issue that needs to be dealt with (just as in BDGM) is that to get $\mathsf{MSB}(u_\mathbf{y})$ to "blend" with the noise under which $C_\Pi(\mathbf{y})$ is encrypted, we make use of the fact that the GSW scheme admits a flexible "scaled" decryption, where the decryption operation is parametrized by a scaling factor ω and enables recovering $2^\omega m + \mathsf{noise}$ (where m is the encrypted message). Normal decryption is obtained by setting $\omega > \log(B)$ where B is a bound on the noise, but if we instead set $\omega = 1$, m can be recovered in a way that blends with the noise, as is required for us to properly perform the "proxy re-encryption".

Outline of the Proof of security. We provide a very brief outline of the proof of security. We will rely on the fact that LHE cirphertexts (of random messages) are dense (in the set of bit strings), and additionally on the fact that both the LHE and the FHE we rely on (i.e., DJ and GSW) satisfy what we refer to as a *weak circuit privacy* notion. This notion, roughly speaking, says that *any* encryption of a message x can be rerandomized into fresh (perhaps extra noisy) encryption of $x+y$, by adding a fresh (perhaps extra noisy) encryption of y.

As usual, the proof proceeds via a hybrid argument. We start from an $Xi\mathcal{O}$ obfuscation of a program Π_0 and transition until we get an $Xi\mathcal{O}$ obfuscation of Π_1, where Π_0 and Π_1 are two functionally equivalent circuits.

- **Hybrid 0: Honest $Xi\mathcal{O}(\Pi_0)$:** The first hybrid is just the honest obfuscation of the circuit Π_0.

- **Hybrid 1: Switch to a freshly encrypted $\mathsf{ct}'_{\mathsf{MSB}}$:** Hybrid 1 proceeds exactly as Hybrid 0 up until the point that $\mathsf{ct}_{\mathsf{MSB}}$ gets re-encrypted into $\mathsf{ct}'_{\mathsf{MSB}}$, with the exception that $\mathsf{FHE.PubCoin}$ are not sampled yet. Next, instead of performing the re-encryption, we sample $\mathsf{ct}'_{\mathsf{MSB}}$ as a *fresh* extra noisy encryption of $-\mathsf{MSB}(u_{\mathbf{y}})$ using randomness $\mathbf{r}^\star$, and setting $\mathsf{FHE.PubCoin}$ to be $\mathbf{r}^\star_{\mathbf{y}} - \mathbf{r}^f_{\mathbf{y}}$). We finally continue the experiment in exactly the same way as in Hybrid 0.

  It follows from the "weak circuit privacy" property of the FHE that Hybrid 0 and Hybrid 1 are statistically close. (Note that the advantage of Hybrid 1 is that for each $\mathbf{y}$, the $\mathbf{y}$'th chunk of $\mathsf{FHE.PubCoin}$ can be thought of as SRL leakage on the fresh encryption $\mathsf{ct}'_{\mathsf{MSB}}$ computed w.r.t. $\mathbf{y}$.

- **Hybrid 2: Switch $\mathsf{LHE.PubCoin}$ to encryptions of random strings:** Hybrid 2 proceeds exactly as Hybrid 1 except that instead of sampling $\mathsf{LHE.PubCoin}$ as a random string, we sample it as fresh LHE encryptions of random strings $u_{\mathbf{y}}$. It follows by the density property of the LHE that Hybrid 2 is statistically close to Hybrid 1.

- **Hybrid 3: Generate $\overline{\mathsf{ct}}_y$ as a fresh encryption:** Hybrid 3 proceeds exactly as Hybrid 2 except that $\overline{\mathsf{ct}}_y$ is generated as a fresh encryption of $m_{\mathbf{y}}$ using fresh randomness $r_{\mathbf{y}}$, and $\mathsf{LHE.PubCoin}$ is instead computed homomorphically by subtracting the LHE encryptions of $\widetilde{m}_{\mathbf{y}}$ and $\mathsf{MSB}(u_y)$ (obtained after homomorphically decrypting $\mathsf{ct}^1_{\mathbf{y}}$ and $\mathsf{ct}'_{\mathsf{MSB}}$) from $\overline{\mathsf{ct}}_{\mathbf{y}}$. (Recall that $m_{\mathbf{y}} = \widetilde{m}_{\mathbf{y}} + u_{\mathbf{y}} - \mathsf{MSB}(u_y)$ so the above way of computing $\mathsf{LHE.PubCoin}$ ensures that it is valid encryption of $u_{\mathbf{y}}$ as in Hybrid 2, but this time without using homomorphically computed randomness.)

  It follows from the weak circuit privacy property of the LHE that Hybrid 3 and 2 are statistically close.

  Note that it was possible to define this hybrid since $\mathsf{ct}'_{\mathsf{MSB}}$ remains exactly the same no matter what $\mathsf{LHE.PubCoin}$ is. This was not true in Hybrid 0, and we introduced Hybrid 1 to break this dependency.

  Note further that in Hybrid 3, we no longer use $\overline{\mathsf{sk}}$ (i.e., the secret key for LHE); previously it was used to recover $r_{\mathbf{y}}$.

- **Hybrid 4: Generate $\overline{\mathsf{ct}}_y$ using fresh smudging noise:** Hybrid 4 proceeds exactly as Hybrid 3 except that $\overline{\mathsf{ct}}_y$ is generated as a fresh encryption of $C_{\Pi_0}(\mathbf{y})$ plus some fresh smudging noise, whereas in Hybrid 3, it was generated as fresh encryption of $m_{\mathbf{y}} = \widetilde{m}_{\mathbf{y}} + u_{\mathbf{y}} - \mathsf{MSB}(u_y)$, where $\widetilde{m}_{\mathbf{y}}$ is a noisy version of $C_{\Pi_0}(\mathbf{y})$, where importantly, the noise is independent of $u_{\mathbf{y}}$, so $u_{\mathbf{y}} - \mathsf{MSB}(u_{\mathbf{y}})$ can be used to "smudge" out the noise in $\widetilde{m}_{\mathbf{y}}$. It follows that Hybrid 4 is statistically close to Hybrid 3.

- **Hybrid 5: Switch to encryption of Π_1:** Hybrid 5 proceeds exactly as Hybrid 4 except that we let ct_1 be an encryption of C_{Π_1} (instead of C_{Π_0} as in all earlier hybrids).

Note that other than the encrypted key cycle, we never use the FHE secret key, and due to Hybrid 3, we no longer use the LHE secret key. So, at first sight, Hybrid 5 ought to be indistinguishable from Hybrid 4 by circular security of the FHE and the LHE. However, recall that $\mathsf{FHE.PubCoin}$ leaks something about the randomness use by the FHE encryption $\mathsf{ct}'_{\mathsf{MSB}}$, but the leakage is exactly an SRL leakage (and note that in the experiment we do know the output $\alpha_{\mathbf{y}}$ of the function f that is applied to the plaintexts encrypted in $\mathsf{ct}_1, \mathsf{ct}_2$—namely, it is $-\mathsf{MSB}(u_{\mathbf{y}})$ where $u_{\mathbf{y}}$ is a random string selected in the experiment, see Hybrid 2). Thus, indistinguishability of Hybrid 5 and Hybrid 4 follows from circular SRL-security of the FHE and the LHE.

- **Hybrids 6-10:** For $i \in [5]$, Hybrid $5+i$ is defined exactly as $5-i$, except that ct_1 be an encryption of C_{Π_1}. Statistical closeness of intermediary hybrids follows just as before.

The above sequence of hybrid allows us to conclude the following theorem.

Theorem 1.3 (Informally stated). *Assume circular SRL-security of the GSW and DJ encryption scheme holds. Then, there exists an $Xi\mathcal{O}$ for polynomial-size circuits taking inputs of length* $\log(\lambda)$ *where λ is the security parameter.*

The proof of Theorem 1.1 is finally concluded by upgrading Theorems 1.2 and 1.3 to apply also in the subexponential regime, relying on the subexponential $\mathsf{2CIRC}^{\mathcal{O}_{\mathsf{SRL}}}$ conjecture, and finally relying on the transformation from subexponentially-secure $Xi\mathcal{O}$ with pre-processing (and subexponential LWE) to $i\mathcal{O}$ [LPST16].

2 Preliminaries and Definitions

In this section, we recall some standard definitions and results. Additionaly, we include a formalization of the circular security assumption that we consider.

Attackers, negligible functions, and subexponential security. Below, for simplicity of exposition, we provide definitions for *polynomial security* of all the primitives we consider. As usual, we model attackers as *non-uniform probabilistic polynomial-time algorithms*, denoted *nuPPT*. We say that a function $\mu(\cdot)$ is *negligible* if for every polynomial $p(\cdot)$, there exists some λ_0 such that $\mu(\lambda) \leq \frac{1}{p(\lambda)}$ for all $\lambda > \lambda_0$. The security definitions we consider will require that for every nuPPT $\mathcal{A}$, there exists some negligible function μ such that for all λ, $\mathcal{A}$ succeeds in "breaking security" w.r.t. the security parameter λ with probability at most $\mu(\lambda)$.

All the definitions that we consider can be extend to consider *subexponential security*; this is done by requiring the existence of some constant ϵ such that for all non-uniform probabilistic algorithms $\mathcal{A}$ with running time $\mathsf{poly}(\lambda) \times 2^{\lambda^\epsilon}$ (as opposed to all nuPPT), there exists some negligible function μ such that for all λ, $\mathcal{A}$ succeeds in "breaking security" w.r.t. the security parameter λ with probability at most $\mu(\lambda) \times 2^{\lambda^\epsilon}$ (as opposed to it being just $\mu(\lambda)$).

2.1 Some Standard Lemmas

We recall some standard definitions and lemmas.

Definition 2.1 (bounded ensemble). *Let $f(\cdot)$ be a function that outputs in $\mathbb{N}$. An ensemble of distributions that outputs in $\mathbb{Z}$, denoted by $\{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$, is said to be f-bounded if there exists a negligible function $\mu(\cdot)$ such that for all $\lambda \in \mathbb{N}$,* $\Pr[|x| > f(\lambda), x \leftarrow \mathcal{D}_\lambda] < \mu(\lambda)$.

We will make use of a special case of the left over hash lemma from [ILL89].

Lemma 2.1 (Left Over Hash lemma)**.** *For all $\lambda, q, d, m \in \mathbb{N}$ such that $m \geq d\lceil \log(q) \rceil + 2\lambda$, the statistical distance between the following distributions is upperbounded by $2^{-\lambda}$:*

$$\Big\{\mathbf{A} \leftarrow_{\mathsf{R}} \mathbb{Z}_q^{d\times m}, \mathbf{r} \leftarrow_{\mathsf{R}} [-1,1]^m : (\mathbf{A}, \mathbf{A}\mathbf{r})\Big\}$$
$$\Big\{\mathbf{A} \leftarrow_{\mathsf{R}} \mathbb{Z}_q^{d\times m}, \mathbf{u} \leftarrow_{\mathsf{R}} \mathbb{Z}_q^{d} : (\mathbf{A}, \mathbf{u})\Big\}.$$

We will also make use of the following standard "smudging" lemma.

Lemma 2.2 (Smudging)**.** *For all $B, B' \in \mathbb{N}$ such that $B < B'$, all $x \in [-B, B]$, the statistical distance between the following distributions is upperbounded by $\frac{B}{B'}$:*

$$\big\{u \leftarrow_{\mathsf{R}} [-B', B'] : u\big\}$$
$$\big\{u \leftarrow_{\mathsf{R}} [-B', B'] : u + x\big\}.$$

2.2 Indistinguishability

We start by recalling the standard definition of computational indistinguishability [GM84].

Definition 2.2 (Computational indistinguishability)**.** *Two ensembles $\{\mathcal{D}_\lambda^0\}_{\lambda\in\mathbb{N}}$ and $\{\mathcal{D}_\lambda^1\}_{\lambda\in\mathbb{N}}$ are said to be* computationally indistinguishable w.r.t. a class $\mathcal{C}$ of attacker *if for every algorithm $A \in \mathcal{C}$, there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$,*

$$\Big|\Pr[A(1^\lambda, \mathcal{D}_\lambda^0) = 1] - \Pr[A(1^\lambda, \mathcal{D}_\lambda^1) = 1]\Big| \leq \mu(\lambda)$$

We simply say that $\{\mathcal{D}_\lambda^0\}_{\lambda\in\mathbb{N}}$ and $\{\mathcal{D}_\lambda^1\}_{\lambda\in\mathbb{N}}$ are computationally indistinguishable *if they are computationally indistinguishable w.r.t. the class of non-uniform Probabilistic Polynomial Time (nuPPT) attackers.*

2.3 Definition of iO

We recall the definition of $i\mathcal{O}$ [BGI+01, GGH+13b]. Given polynomials $n(\cdot), s(\cdot), d(\cdot)$, let $\mathcal{C}_{n,s,d} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ denote the class of circuits such that for all $\lambda \in \mathbb{N}$, $\mathcal{C}_\lambda$ is the set of circuits with input size $n(\lambda)$, size at most $s(\lambda)$ and depth at most $d(\lambda)$. We say that a sequence of circuits $\{\Pi_\lambda\}_{\lambda\in\mathbb{N}}$ is contained in $\{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ (denoted by $\{\Pi_\lambda\}_{\lambda\in\mathbb{N}} \in \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$) if for all $\lambda \in \mathbb{N}$, $\Pi_\lambda \in \mathcal{C}_\lambda$.

Definition 2.3 ($i\mathcal{O}$ for $\mathsf{P/poly}$)**.** *We say a pair of PPT algorithms* $(\mathsf{Obf}, \mathsf{Eval})$ *is an $i\mathcal{O}$ for* $\mathsf{P/poly}$ *if for all polynomials $n(\cdot), s(\cdot), d(\cdot)$, the following holds:*

- **Correctness:** *For all $\{\Pi_\lambda\}_{\lambda\in\mathbb{N}} \in \mathcal{C}_{n,s,d}$, all $\lambda \in \mathbb{N}$, all $\mathbf{x} \in \{0,1\}^{n(\lambda)}$, all $\widetilde{\Pi}$ in the support of* $\mathsf{Obf}(1^\lambda, \Pi_\lambda)$, *we have:*
$$\mathsf{Eval}(1^\lambda, \widetilde{\Pi}, \mathbf{x}) = \Pi(\mathbf{x}).$$

- **IND-security:** *For all sequences $\{\Pi_0^\lambda\}_{\lambda\in\mathbb{N}}, \{\Pi_1^\lambda\}_{\lambda\in\mathbb{N}} \in \mathcal{C}_{n,s,d}$ such that for all $\lambda \in \mathbb{N}$, Π_0^λ and Π_1^λ are functionally equivalent circuits, the following ensembles are computationally indistinguishable:*

$$\Big\{\widetilde{\Pi} \leftarrow \mathsf{Obf}(1^\lambda, \Pi_\lambda^0) : \widetilde{\Pi})\Big\}_{\lambda\in\mathbb{N}}$$
$$\Big\{\widetilde{\Pi} \leftarrow \mathsf{Obf}(1^\lambda, \Pi_\lambda^1) : \widetilde{\Pi})\Big\}_{\lambda\in\mathbb{N}}.$$

2.4 Definition of XiO

We recall the definition of $Xi\mathcal{O}$ with pre-processing [LPST16]. We restrict our attention to circuits with input length $O(\log\lambda)$: Given polynomials $n(\cdot), s(\cdot), d(\cdot)$, let $\mathcal{C}_{\log(n),s,d} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ denote the class of circuits such that for all $\lambda \in \mathbb{N}$, $\mathcal{C}_\lambda$ is the set of circuits with input size $\log(n(\lambda))$, size at most $s(\lambda)$ and depth at most $d(\lambda)$.

Definition 2.4 ($Xi\mathcal{O}$ for $\mathsf{P}^{\log}/\mathsf{poly}$). *We say a tuple of PPT algorithms* $(\mathsf{Gen}, \mathsf{Obf}, \mathsf{Eval})$ *is an* $Xi\mathcal{O}$ *for* $\mathsf{P}^{\log}/\mathsf{poly}$ *if there exists a polynomial* $p(\cdot)$ *and a constant* ε*, such that for all polynomials* $n(\cdot), s(\cdot), d(\cdot)$*, the following holds:*

- **Correctness:** *For all* $\{\Pi_\lambda\}_{\lambda\in\mathbb{N}} \in \mathcal{C}_{\log(n),s,d}$*, all* $\lambda \in \mathbb{N}$*, all* pp *in the support of* $\mathsf{Gen}(1^\lambda, 1^{n(\lambda)}, 1^{s(\lambda)}, 1^{d(\lambda)})$*, all* $\mathbf{x} \in \{0,1\}^{\log(n(\lambda))}$*, all* $\widetilde{\Pi}$ *in the support of* $\mathsf{Obf}(\mathsf{pp}, \Pi_\lambda)$*, we have:*

$$\mathsf{Eval}(\mathsf{pp}, \widetilde{\Pi}, \mathbf{x}) = \Pi(\mathbf{x}).$$

- **Succinctness:** *For all* $\{\Pi_\lambda\}_{\lambda\in\mathbb{N}} \in \mathcal{C}_{\log(n),s,d}$*, all* $\lambda \in \mathbb{N}$*, all* pp *in the support of* $\mathsf{Gen}(1^\lambda, 1^{n(\lambda)}, 1^{s(\lambda)}, 1^{d(\lambda)})$*, all* $\widetilde{\Pi}$ *in the support of* $\mathsf{Obf}(\mathsf{pp}, \Pi_\lambda)$*, we have that* $\left|\widetilde{\Pi}\right| \le n(\lambda)^{1-\varepsilon} \cdot p(\lambda, s(\lambda), d(\lambda))$

- **IND-security:** *For all sequences* $\{\Pi_0^\lambda\}_{\lambda\in\mathbb{N}}, \{\Pi_1^\lambda\}_{\lambda\in\mathbb{N}} \in \mathcal{C}_{\log(n),s,d}$ *such that for all* $\lambda \in \mathbb{N}$*,* Π_0^λ *and* Π_1^λ *are functionally equivalent circuit, the following ensembles are computationally indistinguishable:*

$$\left\{\mathsf{pp} \leftarrow \mathsf{Gen}(1^\lambda, 1^{n(\lambda)}, 1^{s(\lambda)}, 1^{d(\lambda)}), \widetilde{\Pi} \leftarrow \mathsf{Obf}(\mathsf{pp}, \Pi_\lambda^0) : (\mathsf{pp}, \widetilde{\Pi})\right\}_{\lambda\in\mathbb{N}}$$
$$\left\{\mathsf{pp} \leftarrow \mathsf{Gen}(1^\lambda, 1^{n(\lambda)}, 1^{s(\lambda)}, 1^{d(\lambda)}), \widetilde{\Pi} \leftarrow \mathsf{Obf}(\mathsf{pp}, \Pi_\lambda^1) : (\mathsf{pp}, \widetilde{\Pi})\right\}_{\lambda\in\mathbb{N}}$$

The following theorem from [LPST16] connects $Xi\mathcal{O}$ (with pre-processing) with $i\mathcal{O}$ assuming the LWE assumption (we formally define the LWE assumption in Definition 3.4).

Theorem 2.5. *Assume the existence of a subexponentially secure* $Xi\mathcal{O}$ *for* $\mathsf{P}^{\log}/\mathsf{poly}$*, and assume subexponential security of the LWE assumption. Then there exists an* $i\mathcal{O}$ *for* P/poly*.*

2.5 Definition of Public-Key Encryption

We start by recalling the definition of public key encryption (PKE). For our purposes, we will consider PKE in a Common Reference String (CRS) model, where we first generate a CRS, and next, the key generation algorithm will take the CRS as input. This added generality will be useful to capture scenarios where multiple encryption schemes will be operating over the same field Z_N^*—this field can be specified in the CRS.

Definition 2.6 (Public-Key Encryption). *A Public-Key Encryption (PKE) scheme is a tuple of PPT algorithms* $(\mathsf{CRSgen}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *where:*

- $\mathsf{CRSgen}(1^\lambda)$*: given as input the security parameter* $\lambda \in \mathbb{N}$*, it outputs a common reference string* crs*.*

- $\mathsf{Gen}(\mathsf{crs})$*: given as input* crs*, it outputs the pair* $(\mathsf{pk}, \mathsf{sk})$*.*

- $\mathsf{Enc}_{\mathsf{pk}}(m; r)$*: given as input the public key* pk*, a message* $m \in \{0,1\}^*$ *and some randomness* $r \leftarrow_{\mathsf{R}} \{0,1\}^{\infty}$[4]*, it outputs a ciphertext* ct*.*

[4]As usual, since all algorithms are PPT we really only need to consider a finite prefix of $\{0,1\}^\infty$ to define the uniform distribution.

- $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct})$*: given as input the secret key* sk *and a ciphertext* ct*, it deterministically outputs a plaintext.*

We furthermore require these algorithms to satisfy the following correctness *condition: for all* $\lambda \in \mathbb{N}$*, all* crs *in the support of* $\mathsf{CRSgen}(1^\lambda)$*, all pairs* $(\mathsf{pk}, \mathsf{sk})$ *in the support* $\mathsf{Gen}(\mathsf{crs})$*, all messages* $m \in \{0,1\}^*$*, all ciphertexts* ct *in the support of* $\mathsf{Enc}_{\mathsf{pk}}(m)$*, we have:*

$$\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}) = m.$$

2.6 Definition of Linearly-Homomorphic Encryption

Definition 2.7 (Linearly-Homomorphic Encryption)**.** *For any polynomial* $\ell(\cdot)$*, a PKE scheme* $(\mathsf{CRSgen}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is said to be a Linearly-Homomorphic Encryption (LHE) with plaintext size* $\ell(\cdot)$*, if there exists a PPT algorithm* Add *such that the following holds:*

- *For all* $\lambda \in \mathbb{N}$*, all* crs *in the support of* $\mathsf{CRSgen}(1^\lambda)$*, all* $(\mathsf{pk}, \mathsf{sk})$ *in the support of* $\mathsf{Gen}(\mathsf{crs})$*, the public key* pk *contains a message space* $(\mathbb{A}_{\mathsf{pk}}, +)$*, which is an Abelian group of size* $|\mathbb{A}| > 2^{\ell(\lambda)}$*.*

- *For all* $\lambda \in \mathbb{N}$*, all* crs *in the support of* $\mathsf{CRSgen}(1^\lambda)$*, all* $(\mathsf{pk}, \mathsf{sk})$ *in the support of* $\mathsf{Gen}(\mathsf{crs})$*, all messages* $m_1, m_2 \in \mathbb{A}_{\mathsf{pk}}$*, all ciphertexts* $\mathsf{ct}_1, \mathsf{ct}_2$ *in the support of* $\mathsf{Enc}_{\mathsf{pk}}(m_1), \mathsf{Enc}_{\mathsf{pk}}(m_2)$ *respectively, the algorithm* $\mathsf{Add}(\mathsf{pk}, \mathsf{ct}_1, \mathsf{ct}_2)$ *determinsitically outputs a ciphertext in the support of* $\mathsf{Enc}_{\mathsf{pk}}(m_1 + m_2)$*, where the addition is performed in* $\mathbb{A}_{\mathsf{pk}}$*.*

2.7 Definition of Fully Homomorphic Encryption

Definition 2.8 (Fully-Homomorphic Encryption)**.** *For any polynomial* $d(\cdot)$*, a PKE scheme* $(\mathsf{CRSgen}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is said to be a Fully-Homomorphic Encryption (FHE) scheme for depth-*$d(\cdot)$ *circuits, if there exists a PPT algorithm* Eval *such that for all* $\lambda \in \mathbb{N}$*, all* crs *in the support of* $\mathsf{CRSgen}(1^\lambda)$*, all pairs* $(\mathsf{pk}, \mathsf{sk})$ *in the support of* Gen*, all messages* $m_1, \ldots, m_n \in \{0,1\}$*, all ciphertexts* $\mathsf{ct}_1, \ldots, \mathsf{ct}_n$ *in the support of* $\mathsf{Enc}_{\mathsf{pk}}(m_1), \ldots, \mathsf{Enc}_{\mathsf{pk}}(m_n)$ *respectively, all circuits* $f : \{0,1\}^n \to \{0,1\}$ *of depth* $d(\lambda)$*,* $\mathsf{Eval}(\mathsf{pk}, f, \mathsf{ct}_1, \ldots, \mathsf{ct}_n)$ *deterministically outputs an evaluated ciphertext* ct_f *such that* $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}_f) = f(m_1, \ldots, m_n)$*.*

Note that the arity of the circuit that is homomorphically evaluated is not a priori bounded. The FHE we will be using — namely, from [GSW13] — natively supports arithmetic circuits (with addition and multiplication gates), which capture Boolean circuits.

2.8 Leakage-resilient and Circular Security

We recall the standard notion of (indistinguishability-based) security for encryption schemes; we also consider a stronger form of $\mathcal{O}$-leakage resilient security, where the attacker also gets access to a leakage oracle $\mathcal{O}$ that has access to the message $m^\star$ being encrypted, and the randomness r under which it is encrypted.

Definition 2.9 (Security)**.** *We say that a public-key encryption scheme* $\mathcal{PKE} = (\mathsf{CRSgen}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* secure *if for every sequence of pairs of messages* $\{m_\lambda^0, m_\lambda^1\}_{\lambda\in\mathbb{N}}$ *such that for all* $\lambda \in \mathbb{N}$*:* $|m_\lambda^0| = |m_\lambda^1|$*, the ensembles* $\{\mathcal{D}_\lambda^0\}_{\lambda\in\mathbb{N}}$ *and* $\{\mathcal{D}_\lambda^1\}_{\lambda\in\mathbb{N}}$ *are computationally indistinguishable, where* $\mathcal{D}_\lambda^b$ *is defined as follows:*

$$\left\{\ \mathsf{crs} \leftarrow \mathsf{CRSgen}(1^\lambda), (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{crs}), m^\star = m_\lambda^b, r \leftarrow_{\mathsf{R}} \{0,1\}^\infty, \mathsf{ct} = \mathsf{Enc}_{\mathsf{pk}}(m^\star; r) : (\mathsf{crs}, \mathsf{pk}, \mathsf{ct})\ \right\}$$

We additionally say that $\mathcal{PKE}$ *is* $\mathcal{O}$-leakage resilient secure *if indistinguishability holds w.r.t. all nuPPT distinguishers that get unbounded oracle access to* $\mathcal{O}(m^\star, r)$*, and never make the oracle output* $\perp$*.*

We proceed to defining a notion of 2-circular security w.r.t two encryption schemes $\mathcal{PKE}, \overline{\mathcal{PKE}}$. For our purposes, the key generation algorithm of $\mathcal{PKE}$ will be allowed to depend on the public key of $\overline{\mathcal{PKE}}$ (so they can operate over the same field). To enables this, we make use of the CRS: the CRS of $\mathcal{PKE}$ will be set to the public key for $\overline{\mathcal{PKE}}$. 2-circular security will next require indistinguishability of encryptions using $\mathcal{PKE}$ of any message m^0, m^1 in the presence of encrypted key cycle w.r.t. $\mathcal{PKE}$ and $\overline{\mathcal{PKE}}$. We furthermore generalize this definition to consider leakage-resilient security of $\mathcal{PKE}$.

Definition 2.10 (Circular security)**.** *We say that* 2-circular security *holds w.r.t.* $\mathcal{PKE}, \overline{\mathcal{PKE}}$ *where* $\mathcal{PKE} = (\mathsf{CRSgen}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *and* $\overline{\mathcal{PKE}} = (\overline{\mathsf{CRSgen}}, \overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ *if for every sequence of pairs of messages* $\{m^0_\lambda, m^1_\lambda\}_{\lambda \in \mathbb{N}}$ *such that for all* $\lambda \in \mathbb{N}$*:* $|m^0_\lambda| = |m^1_\lambda|$*, the ensembles* $\{\mathcal{D}^0_\lambda\}_{\lambda\in\mathbb{N}}$ *and* $\{\mathcal{D}^1_\lambda\}_{\lambda\in\mathbb{N}}$ *are computationally indistinguishable, where* $\mathcal{D}^b_\lambda$ *is defined as follows:*

$$\left\{ \begin{array}{l} \overline{\mathsf{crs}} \leftarrow \overline{\mathsf{CRSgen}}(1^\lambda), (\overline{\mathsf{pk}}, \overline{\mathsf{sk}}) \leftarrow \overline{\mathsf{Gen}}(\overline{\mathsf{crs}}), \mathsf{crs} = \overline{\mathsf{pk}}, (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{crs}) \\ \overline{\mathsf{ct}} \leftarrow \overline{\mathsf{Enc}}_{\overline{\mathsf{pk}}}(\mathsf{sk}), m^\star = (\overline{\mathsf{sk}} \| m^b_\lambda), r \leftarrow_{\mathsf{R}} \{0,1\}^\infty, \mathsf{ct} = \mathsf{Enc}_{\mathsf{pk}}(m^\star; r) \quad : \quad (\mathsf{crs}, \overline{\mathsf{crs}}, \mathsf{pk}, \overline{\mathsf{pk}}, \mathsf{ct}, \overline{\mathsf{ct}}) \end{array} \right\}.$$

We additionally say that $\mathcal{O}$-leakage resilient 2-circular security holds w.r.t. $\mathcal{PKE}$ and $\overline{\mathcal{PKE}}$ *if indistinguishability holds w.r.t. all nuPPT distinguishers that get unbounded oracle access to* $\mathcal{O}(m^\star, r)$*, and never make the oracle output* $\perp$*.*

We are finally ready to state the $\mathsf{2CIRC}$ assumption that we will rely on.

Definition 2.11 (2CIRC assumption)**.** *We say that the* $\mathsf{2CIRC}^{\mathcal{O}}$ assumption holds w.r.t. $\mathcal{PKE}$ and $\overline{\mathcal{PKE}}$ *if the following holds: if* $\mathcal{PKE}$ *is* $\mathcal{O}$*-leakage resilient secure, and* $\overline{\mathcal{PKE}}$ *is secure, then* $\mathcal{O}$*-leakage resilient 2-circular security holds w.r.t.* $\mathcal{PKE}$ *and* $\overline{\mathcal{PKE}}$*.*

We will also consider a *subexponential* $\mathsf{2CIRC}^{\mathcal{O}}$ assumption which is identically defined except it considers subexponential (as opposed to polynomial) security of the encryption schemes.

Definition 2.12 (Subexponential 2CIRC assumption)**.** *We say that the* subexponential $\mathsf{2CIRC}^{\mathcal{O}}$ assumption holds w.r.t. $\mathcal{PKE}$ and $\overline{\mathcal{PKE}}$ *if the following holds: if* $\mathcal{PKE}$ *is subexponentially* $\mathcal{O}$*-leakage resilient secure, and* $\overline{\mathcal{PKE}}$ *is subexponentially secure, then subexponential* $\mathcal{O}$*-leakage resilient 2-circular security holds w.r.t.* $\mathcal{PKE}$ *and* $\overline{\mathcal{PKE}}$*.*

3 Shielded Randomness Leakage Security of GSW

In this section, we define our notion of Shielded Randomness Leakage (SRL) security, which corresponds to $\mathcal{O}$-leakage resilience security for a particular leakage oracle $\mathcal{O}$. Then, we prove GSW's FHE is SRL secure under the LWE assumption.

3.1 Definition of Shielded Randomness Leakage Security

To define our notion of SRL security, we focus on FHE schemes that satisfy the following properties.

3.1.1 Batch correctness

This property states that decryption of evaluated ciphertexts solely consists of computing the inner product of the evaluated ciphertext with the secret key (both of which are vectors), then rounding. Also, a single scalar obtained by decryption can encode many output bits of the evaluated function, up to choosing a modulus N that is large enough. Our definition of FHE is flexible with respect to the choice of the modulus N, which we can afford since the LWE assumption holds for essentially any (large enough) modulus. As observed in [Mic19, BDGM19, BDGM20], most existing FHE schemes can fit this framework.

Definition 3.1 (Batch correctness)**.** *For any poynomials $d(\cdot)$ and $\ell(\cdot)$, an FHE scheme* $(\mathsf{CRSgen}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *for depth-$d(\cdot)$ circuits satisfies $\ell(\cdot)$-batch correctness if there exist a PPT* Eval' *and a polynomial $n(\cdot)$ such that following holds:*

- *For all $\lambda \in \mathbb{N}$, all* crs *in the support of* $\mathsf{CRSgen}(1^\lambda)$*, all* $(\mathsf{pk}, \mathsf{sk})$ *in the support of* $\mathsf{Gen}(\mathsf{crs})$*,* pk *contains $N_{\mathsf{pk}}, B_{\mathsf{pk}} \in \mathbb{N}$ such that $N_{\mathsf{pk}} > B_{\mathsf{pk}} \cdot 2^{\ell(\lambda)+\lambda}$. The secret key is of the form:* $\mathsf{sk} \in \mathbb{Z}^{n(\lambda)}$*.*
- *For all $\lambda \in \mathbb{N}$, all* crs *in the support of* $\mathsf{CRSgen}(1^\lambda)$*, all* $(\mathsf{pk}, \mathsf{sk})$ *in the support of* $\mathsf{Gen}(\mathsf{crs})$*, all $\nu \in \mathbb{N}$, all messages $m_1, \ldots, m_\nu \in \{0,1\}$, all depth-$d(\lambda)$ circuits f of arity ν, all ciphertexts* ct_i *in the support of* $\mathsf{Enc}_{\mathsf{pk}}(m_i)$ *for all $i \in [\nu]$, all scaling factors $\omega < \log(N_{\mathsf{pk}})$, the algorithm* $\mathsf{Eval}'(\mathsf{pk}, f, \omega, \mathsf{ct}_1, \ldots, \mathsf{ct}_\nu)$ *deterministically outputs an evaluated ciphertext* $\mathsf{ct}_f \in \mathbb{Z}_{N_{\mathsf{pk}}}^{n(\lambda)}$ *such that:*
$$\mathsf{sk}^\top \mathsf{ct}_f = 2^\omega f(\mathbf{m}) + \mathsf{noise}_f \in \mathbb{Z}_{N_{\mathsf{pk}}},$$
with $|\mathsf{noise}_f| < B_{\mathsf{pk}}$*.*

Note that one can recover the value $f(\mathbf{m})$ when using a scaling factor $\omega > \log(B_{\mathsf{pk}})$. That is, we can define $\mathsf{Eval}(\mathsf{pk}, f, \mathsf{ct}_1, \ldots, \mathsf{ct}_\nu) = \mathsf{Eval}'(\mathsf{pk}, f, \lceil \log(B_{\mathsf{pk}}) \rceil, \mathsf{ct}_1, \ldots, \mathsf{ct}_\nu)$.

3.1.2 Randomness homomorphism

This property states that it is possible to homomorphically evaluates a circuit f not only on the ciphertexts, but also the randomness used by the ciphertexts. The resulting evaluated randomness r_f belongs to a noisy randomness space $\mathcal{R}^\star$ — typically the fresh randomness comprises noises, and the evaluated randomness consists of larger-magnitude noises. The encryption algorithm $\mathsf{Enc}^\star$ is essentially the same as Enc except it operates on the evaluated (noisier) randomness. The ciphertext obtained by first evaluating the randomness, then using the noisy encryption algorithm $\mathsf{Enc}^\star$ is the same as obtained by directly evaluating the original ciphertexts.

Definition 3.2 (Randomness homomorphism)**.** *An FHE scheme $\mathcal{FHE}$ =* $(\mathsf{CRSgen}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *for depth-$d(\cdot)$ circuits that satisfies $\ell(\cdot)$-batch correctness (defined above) also satisfies randomness homomorphism if there exists a sequence of noisy randomness spaces $\{\mathcal{R}^\star_\lambda\}_{\lambda \in \mathbb{N}}$, and the following additional PPT algorithms:*

- $\mathsf{Eval}_{\mathsf{rand}}(\mathsf{pk}, f, \mathbf{r}, \mathbf{m})$*: given as input the public key* pk*, a depth-$d(\lambda)$ circuit f of arity ν, random coins $\mathbf{r} = (r_1, \ldots, r_\nu)$ where for all $i \in [\nu]$, $r_i \in \{0,1\}^\infty$, and messages $\mathbf{m} \in \{0,1\}^\nu$, it deterministically outputs an evaluated randomness $r_f \in \mathcal{R}^\star$.*
- $\mathsf{Enc}^\star_{\mathsf{pk}}(m; r^\star)$*: given as input the public key* pk*, a message $m \in \mathbb{Z}$ and the randomness $r^\star \in \mathcal{R}^\star$, it outputs a noisy ciphertext* $\mathsf{ct}^\star$*.*

We furthermore require these algorithms to satisfy the following condition: for every $\lambda \in \mathbb{N}$, all crs *in the support of* $\mathsf{CRSgen}(1^\lambda)$*, all pairs* $(\mathsf{pk}, \mathsf{sk})$ *in the support of* $\mathsf{Gen}(\mathsf{crs})$*, all $\nu \in \mathbb{N}$, all depth-$d(\lambda)$ circuits f of arity n, all messages $m_i \in \{0,1\}$, $r_i \in \{0,1\}^\infty$* $\mathsf{ct}_i = \mathsf{Enc}_{\mathsf{pk}}(m_i; r_i)$ *for all $i \in [\nu]$, denoting* $r_f = \mathsf{Eval}_{\mathsf{rand}}(\mathsf{pk}, f, \mathbf{r}, \mathbf{m})$*, we have:*

$$\mathsf{Eval}'(\mathsf{pk}, f, 1, \mathsf{ct}_1, \ldots, \mathsf{ct}_\nu) = \mathsf{Enc}^\star_{\mathsf{pk}}(f(\mathbf{m}); r_f).$$

3.1.3 Shielded Randomness-Leakage security

To define the following Shielded Randomness-Leakage oracle oracle, we restrict ourselves to FHE where the noisy randomness consists of integer vectors. That is, there exists a polynomial $t(\cdot)$ such that the sequence $\{\mathcal{R}^\star_\lambda\}_{\lambda\in\mathbb{N}}$ is such that for all $\lambda \in \mathbb{N}$, $\mathcal{R}^\star_\lambda \subseteq \mathbb{Z}^{t(\lambda)}$. Henceforth, we denote by $r_1 + r_2 \in \mathcal{R}^\star_\lambda$ and $r_1 - r_2 \in \mathcal{R}^\star_\lambda$ the addition and subtraction in $\mathbb{Z}^{t(\lambda)}$. We denote $\mathcal{R}^\star_\lambda$ by $\mathcal{R}^\star$ for simplicity.

Definition 3.3 (SRL security)**.** *An FHE scheme $\mathcal{FHE}$ for depth $d(\cdot)$ circuits satisfying randomness homomorphism is said to be SRL-secure if it is $\mathcal{O}^{\mathcal{FHE}}_{\mathsf{SRL}}$-leakage resilient secure for the following oracle $\mathcal{O}^{\mathcal{FHE}}_{\mathsf{SRL}}$, where* $\mathsf{Eval}_{\mathsf{rand}}$ *and* $\mathsf{Enc}^\star$ *are the algorithms guaranteed to exists by the definition of randomness homomorphism. Similarly, for any PKE scheme $\mathcal{PKE}$, we say 2-circular SRL security holds with respect to $\mathcal{FHE}$ and $\mathcal{PKE}$ if the $\mathcal{O}^{\mathcal{FHE}}_{\mathsf{SRL}}$-leakage resilient 2-circular security holds with respect to $\mathcal{FHE}$ and $\mathcal{PKE}$.*

```
$\underline{\mathcal{O}^{\mathcal{FHE}}_{\mathsf{SRL}}(\mathbf{m}^\star, \mathbf{r})}$:
$r^\star \leftarrow_{\mathsf{R}} \mathcal{R}^\star$, $\mathsf{ct}^\star = \mathsf{Enc}^\star_{\mathsf{pk}}(0; r^\star)$
$(f, \alpha) \leftarrow \mathcal{A}(\mathsf{ct}^\star)$
$r_f = \mathsf{Eval}_{\mathsf{rand}}(\mathsf{pk}, f, \mathbf{r}, \mathbf{m}^\star)$.
If $f(\mathbf{m}^\star) = \alpha$ and $f$ is of depth at most $d$, then $\mathsf{leak} = r^\star - r_f \in \mathcal{R}^\star$.
Otherwise, $\mathsf{leak} = \perp$. Return $\mathsf{leak}$.
```

Roughly speaking, given a message $\mathbf{m}^\star$ and randomness $\mathbf{r}$, the oracle $\mathcal{O}^{\mathcal{FHE}}_{\mathsf{SRL}}$ samples fresh random coins $r^\star$ from which it generates a noisy encryption of zero, that is sent to the adversary. The adcersary next chooses a circuit f and a value $\alpha \in \mathbb{Z}$. The oracle then checks that $f(\mathbf{m}^\star) = \alpha$, upon which it returns the evaluated randomness "shielded" with the randomness $r^\star$.

In the concrete FHE we consider from [GSW13], the randomness leakage corresponds to the randomness obtained from homomorphically subtracting the encryption $\mathsf{Enc}^\star_{\mathsf{pk}}(0; r^\star)$ by the evaluated challenge ciphertext. Revealing such leakage allows the adversary to decrypt and recover the value $0 - f(\mathbf{m}^\star)$. By enforcing $f(\mathbf{m}^\star) = \alpha$, we make sure the adversary does not learn anything more than what she already knew (recalling that the notion of Ø-leakage resilient security only requires indistinguishability w.r.t. adversarys that do not make the oracle output $\perp$).

Whenever the scheme $\mathcal{FHE}$ is clear from context, we simply write $\mathcal{O}_{\mathsf{SRL}}$ to denote $\mathcal{O}^{\mathcal{FHE}}_{\mathsf{SRL}}$.

3.2 SRL Security of GSW's FHE from LWE

We now recall the FHE scheme from [GSW13], whose security relies on the LWE assumption. The variant we present uses a large modulus to permit batching many output bits in a single scalar. We prove the GSW scheme is SRL-secure (as per Definition 3.3) under the LWE assumption.

3.2.1 Learning With Error Assumption

We recall the Learning with Error (LWE) assumption [Reg05] with subexponential modulus-to-noise ratio.

Definition 3.4 (Learning With Error Assumption)**.** *There exists a constant $c \in (0, 1)$ such that for all polynomials $n(\cdot), m(\cdot)$, all sequences $\{q_\lambda\}_{\lambda\in\mathbb{N}}$, such that for every $\lambda \in \mathbb{N}$, the bit size $|q_\lambda|$ is polynomially bounded in λ and $q_\lambda \cdot 2^{-n(\lambda)^c} \geq 2\sqrt{n(\lambda)\log(\lambda)}$, there exists a $q_\lambda \cdot 2^{-n(\lambda)^c}$-bounded* *ensemble of distributions $\{\chi_\lambda\}_{\lambda\in\mathbb{N}}$ such that for all $\lambda \in \mathbb{N}$, χ_λ is an efficiently sampleable distribution over $\mathbb{Z}$ and such that the following ensembles are computationally indistinguishable:*

$$\{\mathcal{D}^0_\lambda\}_{\lambda\in\mathbb{N}} = \left\{\mathbf{A} \leftarrow_{\mathsf{R}} \mathbb{Z}_{q_\lambda}^{m(\lambda)\times n(\lambda)}, \mathbf{s} \leftarrow \chi_\lambda^{n(\lambda)}, \mathbf{e} \leftarrow \chi_\lambda^{m(\lambda)}, \mathbf{z} = \mathbf{A}\mathbf{s} + \mathbf{e} \in \mathbb{Z}_{q_\lambda}^{m(\lambda)} : (\mathbf{A}, \mathbf{z})\right\}_{\lambda\in\mathbb{N}}.$$

$$\{\mathcal{D}^1_\lambda\}_{\lambda\in\mathbb{N}} = \left\{\mathbf{A} \leftarrow_{\mathsf{R}} \mathbb{Z}_{q_\lambda}^{m(\lambda)\times n(\lambda)}, \mathbf{z} \leftarrow_{\mathsf{R}} \mathbb{Z}_{q_\lambda}^{m(\lambda)} : (\mathbf{A}, \mathbf{z})\right\}_{\lambda\in\mathbb{N}}.$$

In [Reg05], Regev showed that solving the LWE problem with modulus q, dimension n, arbitrary number of samples m, and discrete Gaussian distribution χ of standard deviation $\sigma = \alpha q \geq 2\sqrt{n}$ (this is the distribution over $\mathbb{Z}$ that follows the normal distribution of standard deviation σ, which is $\sigma \cdot \omega(\sqrt{\log(\lambda)})$-bounded) is at least as hard as quantumly approximating the shortest independent vector problem (SIVP) to within an approximation factor $\gamma = \widetilde{\mathcal{O}}(n/\alpha)$ in the *worst case* n-dimensional lattices. His result only applied to every modulus q that is a prime power, or a product of small (poly-size) distinct primes. Later, in [PRS17], the result was generalized to any modulus q.

As typical, we obtain Definition 3.4 by choosing a noise-to-modulus ratio $\alpha = 2^{-n^c}$ for a constant $c \in (0, 1)$, which corresponds to the SIVP problem with an approximation factor $\gamma = \widetilde{\mathcal{O}}(n \cdot 2^{n^c})$, which is believed to be intractable for c small enough.

3.2.2 The GSW scheme

We present an FHE for depth-$d(\cdot)$ circuits and $\ell(\cdot)$-batch correctness from [GSW13]. For concreteness (and because this is what we will rely on later on), we focus on an instantiation of GSW when the modulus is selected as in the DJ scheme (i.e., as an RSA modulus), but nothing that we prove in this section depends on this choice of the modulus—any sufficiently large modulus would have worked.

• $\underline{\mathsf{CRSgen}(1^\lambda)}$:
Generate $\overline{\mathsf{pk}} = (N, \zeta)$ as done by the key generation algorithm of the DJ's LHE parameterized by the polynomial $p(\lambda) = d(\lambda) \cdot \lambda + \ell(\lambda) + \lambda$, described Section 4.1. We have $p(\lambda) \leq \log(N) < p(\lambda) + 2\lambda$. It outputs $\mathsf{crs} = \overline{\mathsf{pk}}$.

• $\underline{\mathsf{Gen}(\mathsf{crs})}$:
It sets $n = (d(\lambda) \cdot \lambda + \ell(\lambda))^{1/c}$, where $c \in (0, 1)$ is the constant from Definition 3.4 (LWE). For this (simple, but not tight) choice of parameters n and $p(\lambda)$, denoting $B' = N 2^{-n^c}$, we have $B' \geq 2\sqrt{n \log(\lambda)}$. Thus, by the LWE assumption (as per Definition 3.4), there exists an efficiently computable distribution χ over $\mathbb{Z}$ that is B'-bounded, such that LWE holds with respect to N, n, χ. We write $w = (n+1)\lceil\log(N)\rceil$, $m = 2(n+1)\lceil\log(N)\rceil + 2\lambda$, $\widetilde{B} = (w+1)^d\lceil\log(N)\rceil$ and $B = \widetilde{B}B'm$. It samples $\mathbf{A} \leftarrow_{\mathsf{R}} \mathbb{Z}_N^{n\times m}$, $\mathbf{s} \leftarrow \chi^n$, $\mathbf{e} \leftarrow \chi^m$, $\mathbf{g} = (1, 2, \ldots, 2^{\lceil\log(N)\rceil-1}) \in \mathbb{Z}_N^{\lceil\log(N)\rceil}$, $\mathbf{G} = \mathrm{Id} \otimes \mathbf{g}^\top \in \mathbb{Z}_N^{(n+1)\times w}$ where $\mathrm{Id} \in \mathbb{Z}_N^{(n+1)\times(n+1)}$ denotes the identity matrix, $\mathbf{U} = \begin{pmatrix} \mathbf{A} \\ \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top \end{pmatrix} \in \mathbb{Z}_N^{(n+1)\times m}$. It sets $\mathsf{pk} = (N, B, \mathbf{U}, \mathbf{G})$, and $\mathsf{sk} = (-\mathbf{s}, 1) \in \mathbb{Z}_N^{n+1}$. The parameters define the noisy randomness space $\mathcal{R}^\star = [-2^\lambda\widetilde{B}, 2^\lambda\widetilde{B}]^m$. It outputs $(\mathsf{pk}, \mathsf{sk})$.

• $\underline{\mathsf{Enc}(\mathsf{pk}, m)}$:
Given the public pk, a message $m \in \{0, 1\}$, it samples the randomness $\mathbf{R} \leftarrow_{\mathsf{R}} [-1, 1]^{m\times w}$ and outputs the ciphertext $\mathsf{ct} = \mathbf{U}\mathbf{R} + m\mathbf{G} \in \mathbb{Z}_N^{(n+1)\times w}$. For any $\mathbf{m} \in \{0, 1\}^n$, we denote by $\mathsf{Enc}_{\mathsf{pk}}(\mathbf{m}; \mathbf{r})$ the concatenation of the encryptions $\mathsf{Enc}_{\mathsf{pk}}(m_1; \mathbf{R}_1), \ldots, \mathsf{Enc}_{\mathsf{pk}}(m_n; \mathbf{R}_n)$.

- $\underline{\mathsf{Eval}(\mathsf{pk}, f, \mathsf{ct}_1, \dots, \mathsf{ct}_\nu)}$:
Given the public key pk, a depth-$d(\lambda)$ arithmetic $f : \{0,1\}^\nu \to \{0,1\}$, ciphertexts $\mathsf{ct}_1, \dots, \mathsf{ct}_\nu$, it runs $\mathsf{ct}_f \leftarrow \mathsf{Eval}'(\mathsf{pk}, f, \omega, \mathsf{ct}_1, \dots, \mathsf{ct}_\nu)$ with scaling factor $\omega = 2^{2\lambda}B$, where the algorithm Eval' is described below, for the batch correctness property.

- $\underline{\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct})}$: The decryption multiplies ct by $\mathsf{sk} \in \mathbb{Z}_N^{n+1}$, and rounds the result.

We demonstrate that the GSW FHE satisfies the batch correctness property.

Proposition 1 (Batch correctness)**.** *The GSW FHE described above for depth-$d(\cdot)$ circuits satisfies $\ell(\cdot)$-batch correctness, as per Definition 3.1.*

Proof: We present the following PPT algorithm:

- $\underline{\mathsf{Eval}'(\mathsf{pk}, f, \omega, \mathsf{ct}_1, \dots, \mathsf{ct}_\nu)}$:
Given the public pk, a depth-d arithmetic circuit $f : \{0,1\}^\nu \to \mathbb{Z}_N$, a scaling factor $\omega < \log(N)$, ciphertexts $\mathsf{ct}_1, \dots, \mathsf{ct}_n$, it evaluates the circuit gate by gate as follows.

  - Addition gate between ct_i and ct_j: return $\mathsf{ct}_i + \mathsf{ct}_j \in \mathbb{Z}_N^{(n+1)\times w}$.

  - Multiplication gate between ct_i and ct_j: return $\mathsf{ct}_i \cdot \mathsf{BD}(\mathsf{ct}_j) \in \mathbb{Z}_N^{(n+1)\times w}$, where $\mathsf{BD}(\mathsf{ct}_j) \in \{0,1\}^{w\times w}$ denotes the binary decomposition of $\mathsf{ct}_j \in \mathbb{Z}_N^{(n+1)\times w}$.

By recursively applying the above operations, one can turn the ciphertexts $\mathsf{ct}_1, \dots, \mathsf{ct}_n$ into $\mathbf{C}_f^i = \mathbf{U}\mathbf{R}_f^i + f_i(\mathbf{m})\mathbf{G} \in \mathbb{Z}_N^{(n+1)\times w}$, where $f_i(\mathbf{m}) \in \{0,1\}$ denotes the i'th bit of the binary decomposition of $f(\mathbf{m}) \in \mathbb{Z}_N$, that is, $f(\mathbf{m}) = \sum_{i=0}^{\lceil\log(N)\rceil-1} 2^i f_i(\mathbf{m})$. For all $i \in [0, \lceil\log(N)\rceil - 1]$, we have $\|\mathbf{R}_f^i\|_\infty \leq (w+1)^d$. By definition of the matrix $\mathbf{G}$, choosing the $n \cdot \lceil\log(N)\rceil + i + \omega + 1$'th column of $\mathbf{C}_f^i$ yields:

$$\mathbf{c}_f^i = \Big(\mathbf{A}\mathbf{r}_f^i, (\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top)\mathbf{r}_f^i + 2^{\omega+i} f_i(\mathbf{m})\Big) \in \mathbb{Z}_N^{n+1}.$$

Summing up for all $i \in [0, \lceil\log(N)\rceil - 1]$, we get: $\mathsf{ct}_f = \big(\mathbf{A}\mathbf{r}_f, (\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top)\mathbf{r}_f + 2^\omega f(\mathbf{m})\big) \in \mathbb{Z}_N^{n+1}$, where $\mathbf{r}_f = \sum_{i=0}^{\lceil\log(N)\rceil-1} \mathbf{r}_f^i$ of norm $\|\mathbf{r}_f\|_\infty \leq (w+1)^d\lceil\log(N)\rceil = \widetilde{B}$. It outputs the evaluated ciphertext ct_f.

The evaluated ciphertext ct_f is such that:

$$\mathsf{sk}^\top\mathsf{ct}_f = -\mathbf{s}^\top\mathbf{A}\mathbf{r}_f + (\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top)\mathbf{r}_f + 2^\omega f(\mathbf{m}) = 2^\omega f(\mathbf{m}) + \mathsf{noise}_f \in \mathbb{Z}_N,$$

where $\mathsf{noise}_f = \mathbf{e}^\top\mathbf{r}_f$. We have $|\mathsf{noise}_f| < \widetilde{B}B'm = B$. Moreover, parameters N and B are set such that $N \geq 2^{\ell(\lambda)+2\lambda}B$. □

We turn to proving that it also satisfies the randomness homomorphism property.

Proposition 2 (Randomness homomorphism)**.** *The GSW FHE for depth-$d(\cdot)$ circuits, satisfying $\ell(\cdot)$-bath correctness presented above satisfies the randomness homomorphism property as per Definition 3.2.*

Proof: We present the following PPT algorithms:

- $\underline{\mathsf{Enc}^\star(\mathsf{pk}, m; \mathbf{r}^\star)}$:

Given the public pk, a message $m \in \mathbb{Z}$, the randomness $\mathbf{r}^\star \in [-2^\lambda\widetilde{B}, 2^\lambda\widetilde{B}]^m$ it outputs the ciphertext $\mathsf{ct} = \big(\mathbf{A}\mathbf{r}^\star, (\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top)\mathbf{r}^\star + m\big) \in \mathbb{Z}_N^{n+1}$.

- $\underline{\mathsf{Eval}_{\mathsf{rand}}(\mathsf{pk}, f, (\mathbf{R}_i)_{i\in[\nu]}, (m_i)_{i\in[\nu]})}$:

This algorithm is similar to the ciphertext evaluation algorithm. Namely, given the public pk, a depth-d arithmetic circuit $f : \{0,1\}^\nu \to \mathbb{Z}_N$, randomness $\mathbf{R}_1, \ldots, \mathbf{R}_\nu \in [-1,1]^{m\times w}$, it evaluates the circuit gate by gate as follows.

- Addition gate between $\mathbf{R}_i$ and $\mathbf{R}_j$: return $\mathbf{R}_i + \mathbf{R}_j \in \mathbb{Z}_N^{m\times w}$.
- Multiplication gate between $\mathbf{R}_i$ and $\mathbf{R}_j$: compute $\mathsf{ct}_j = \mathsf{Enc}_{\mathsf{pk}}(m_j; \mathbf{R}_j)$, return $\mathbf{R}_i \mathsf{BD}(\mathsf{ct}_j) + m_i \mathbf{R}_j \in \mathbb{Z}_N^{m\times w}$, where $\mathsf{BD}(\mathsf{ct}_j) \in \{0,1\}^{w\times w}$ denotes the binary decomposition of $\mathsf{ct}_j \in \mathbb{Z}_N^{(n+1)\times w}$.

By recursively applying the above operations, one can turn the randomness $\mathbf{R}_1, \ldots, \mathbf{R}_n$ into $\mathbf{R}_f^i \in \mathbb{Z}_N^{m\times w}$ such that: $\mathbf{C}_f^i = \mathbf{U}\mathbf{R}_f^i + f_i(\mathbf{m})\mathbf{G} \in \mathbb{Z}_N^{(n+1)\times w}$, for all $i \in [0, \lceil\log(N)\rceil - 1]$; and $\|\mathbf{R}_f^i\|_\infty \leq (w+1)^d$. By definition of the matrix $\mathbf{G}$, choosing the $n\lceil\log(N)\rceil + i + 1$'th column of $\mathbf{R}_f^i$ yields $\mathbf{r}_f^i \in \mathbb{Z}_N^m$ such that: $\mathbf{c}_f^i = \left(\mathbf{A}\mathbf{r}_f^i, (\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top)\mathbf{r}_f^i + 2^i f_i(\mathbf{m})\right) \in \mathbb{Z}_N^{n+1}$, and $\|\mathbf{r}_f^i\|_\infty \leq (w+1)^d$. Summing up for all $i \in [0, \lceil\log(N)\rceil - 1]$, we get: $\mathbf{r}_f = \sum_{i=0}^{\lceil\log(N)\rceil-1} \mathbf{r}_f^i \in \mathcal{R}^\star$ such that $\mathsf{ct}_f = \left(\mathbf{A}\mathbf{r}_f, (\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top)\mathbf{r}_f + f(\mathbf{m})\right) \in \mathbb{Z}_N^{n+1}$. It outputs the evaluated randomness $\mathbf{r}_f$. $\square$

3.2.3 SRL Security

Before proving the SRL security of GSW under the LWE assumption, we describe new trapdoor generation and pre-image sampling algorithms that are inspired by those from [MP12]. As in prior works, the trapdoor generation algorithm generates a matrix $\mathbf{U} \in \mathbb{Z}_N^{d\times m}$ that is statistically close to uniformly random over $\mathbb{Z}_N^{d\times m}$, together with an associated trapdoor $T_{\mathbf{U}}$. The pre-image sampling algorithm, given a target vector $\mathbf{t} \in \mathbb{Z}_N^d$, produces a short pre-image, that is, a short vector $\mathbf{r} \in \mathbb{Z}_N^m$ such that $\mathbf{U}\mathbf{r} = \mathbf{t}$. In these works, the distribution of these short pre-images is independent of the trapdoor — typically they follow a discrete (spherical) Gaussian distribution. Our requirements are slightly different: a pre-image produced by our sampling algorithm when given as input a target vector $\mathbf{t} \in \mathbb{Z}_N^d$ should be statistically close to a pre-image produced by our sampling algorithm when given as input the vector $\mathbf{0} \in \mathbb{Z}_N^d$, shifted by a much smaller pre-image of $\mathbf{t}$. That is, if a very short pre-image is given, adding a somewhat short pre-image of $\mathbf{0}$ (produced by the sampling algorithm) to it will produce a pre-image that looks like a fresh output of the sampling algorithm on input $\mathbf{t}$. This inherently requires smudging size noises, which implies the use of an exponential-size modulus q. In fact this property is not known to hold for existing trapdoor generation and pre-image sampling algorithms using polynomial-size modulus.

We prove this property for the concrete algorithms provided in [MP12], which we simplify since we can afford to use smudging-size noises. We provide a self-contained description of the scheme and its proofs here.

Lattice trapdoors.

- $\underline{\mathsf{TrapGen}(1^\lambda, q, d)}$:

Given as input the security parameter $\lambda \in \mathbb{N}$, a modulus $q \in \mathbb{N}$, a dimension $d \in \mathbb{N}$, it sets $\widetilde{m} = d\lceil\log(q)\rceil + 2\lambda$, $w = d\lceil\log(q)\rceil$, $m = \widetilde{m} + w$, computes the gadget matrix $\mathbf{G} = \mathrm{Id} \otimes \mathbf{g}^\top$ where $\mathrm{Id} \in \mathbb{Z}_q^{d\times d}$ denotes the identity matrix and $\mathbf{g} = (1, 2, \ldots, 2^{\lceil\log(q)\rceil-1}) \in \mathbb{Z}_q^{\lceil\log(q)\rceil}$, $\widetilde{\mathbf{U}} \leftarrow_{\mathsf{R}} \mathbb{Z}_q^{d\times\widetilde{m}}$, $\mathbf{R} \leftarrow_{\mathsf{R}} [-1,1]^{\widetilde{m}\times w}$, $\mathbf{U} = (\widetilde{\mathbf{U}} \| -\widetilde{\mathbf{U}}\mathbf{R} + \mathbf{G}) \in \mathbb{Z}_q^{d\times m}$, $T_{\mathbf{U}} = \mathbf{R}$. It outputs $(\mathbf{U}, T_{\mathbf{U}})$.

- $\underline{\mathsf{PreImSamp}(\mathbf{U}, T_{\mathbf{U}}, \mathbf{t}, B)}$:

Given as input the matrix $\mathbf{U}$, the trapdoor $T_{\mathbf{U}}$, a target vector $\mathbf{t} \in \mathbb{Z}_q^d$ and a bound $B \in \mathbb{N}$, it samples $\mathbf{v} \leftarrow_{\mathsf{R}} [-B2^{\lambda/2}, B2^{\lambda/2}]^m$, sets $\mathbf{b} = \mathsf{BD}\left(\mathbf{U}\mathbf{v} + \mathbf{t}\right) \in \{0,1\}^{w \times w}$ which denotes the binary decomposition of $\mathbf{U}\mathbf{v} + \mathbf{t} \in \mathbb{Z}_q^d$. It outputs $\begin{pmatrix} \mathbf{R}\mathbf{b} \\ \mathbf{b} \end{pmatrix} - \mathbf{v} \in \mathbb{Z}_q^m$.

We show the following properties hold.

Proposition 3 (Correctness of $\mathsf{TrapGen}$)**.** *For all λ, q, d, writing $m = 2d\lceil \log(N) \rceil + 2\lambda$, the following distributions have statistical distance at most $2^{-\lambda}$:*

$$\left\{\mathbf{U} \leftarrow_{\mathsf{R}} \mathbb{Z}_N^{d \times m} : \mathbf{U}\right\}$$
$$\left\{(\mathbf{U}, T_{\mathbf{U}}) \leftarrow \mathsf{TrapGen}(1^\lambda, N, d) : \mathbf{U}\right\}.$$

Proof: The proposition follows readily from Lemma 2.1 (left over hash lemma). □

Proposition 4 (Correctness of $\mathsf{PreImSamp}$)**.** *For all $\lambda, q, d, B \in \mathbb{N}$, all $(\mathbf{U}, T_{\mathbf{U}})$ in the support of $\mathsf{TrapGen}(1^\lambda, q, d)$, all $\mathbf{t} \in \mathbb{Z}_N^d$, all $\mathbf{r} \in \mathbb{Z}_N^m$ in the support of $\mathsf{PreImSamp}(\mathbf{U}, T_{\mathbf{U}}, \mathbf{t}, B)$, we have $\mathbf{U}\mathbf{r} = \mathbf{t}$ and $\|\mathbf{r}\|_\infty < B2^{\lambda/2} + w$.*

Proof: Straightforward. □

Proposition 5 (Security)**.** *For all $\lambda, q, d, B \in \mathbb{N}$, writing $m = 2d\lceil \log(N) \rceil + 2\lambda$, for all $\mathbf{w} \in \mathbb{Z}_N^m$ such that $\|\mathbf{w}\|_\infty < B$, the the statistical distance of the two following distributions is upper-bounded by $2^{-\lambda/2}$:*

$$\{(\mathbf{U}, T_{\mathbf{U}}) \leftarrow \mathsf{TrapGen}(1^\lambda, q, d), \widetilde{\mathbf{r}_0} \leftarrow_{\mathsf{R}} \mathsf{PreImSamp}(\mathbf{U}, T_{\mathbf{U}}, \mathbf{0}, B) : \widetilde{\mathbf{r}_0} + \mathbf{w} \in \mathbb{Z}_N^m\}$$
$$\{(\mathbf{U}, T_{\mathbf{U}}) \leftarrow \mathsf{TrapGen}(1^\lambda, q, d), \widetilde{\mathbf{r}} \leftarrow \mathsf{PreImSamp}(\mathbf{U}, T_{\mathbf{U}}, \mathbf{U}\mathbf{w}, B) : \widetilde{\mathbf{r}}\}$$

Proof: By definition of $\mathsf{PreImSamp}$ we have: $\widetilde{\mathbf{r}_0} = \begin{pmatrix} \mathbf{R}\mathbf{b} \\ \mathbf{b} \end{pmatrix} - \mathbf{v}$ where $\mathbf{v} \leftarrow_{\mathsf{R}} [-B2^{\lambda/2}, B2^{\lambda/2}]^m$ and $\mathbf{b} = \mathsf{BD}(\mathbf{U}\mathbf{v}) \in \{0,1\}^w$. Since $\|\mathbf{w}\|_\infty < B$, by Lemma 2.2 (smudging), we have: $\mathbf{v} \approx_s \mathbf{v} + \mathbf{w}$ with statistical distance $2^{-\lambda/2}$. This implies $\widetilde{\mathbf{r}_0} + \mathbf{w} \approx_s \begin{pmatrix} \mathbf{R}\mathbf{b}' \\ \mathbf{b}' \end{pmatrix} - \mathbf{v}$, where $\mathbf{b}' = \mathsf{BD}(\mathbf{U}\mathbf{v} + \mathbf{U}\mathbf{w})$. The latter is identically distributed to $\mathsf{PreImSamp}(\mathbf{U}, T_{\mathbf{U}}, \mathbf{U}\mathbf{w}, B)$. □

We also prove the following lemma, that will prove useful later.

Lemma 3.1. *For all $\lambda, q, d \in \mathbb{N}$, the following distributions have statistical distance upper-bounded by $2^{-\lambda}$:*

$$\left\{(\mathbf{U}, T_{\mathbf{U}}) \leftarrow_{\mathsf{R}} \mathsf{TrapGen}(1^\lambda, q, d), \mathbf{r} \leftarrow_{\mathsf{R}} [-1,1]^m : (\mathbf{U}, T_{\mathbf{U}}, \mathbf{U}\mathbf{r})\right\}.$$
$$\left\{(\mathbf{U}, T_{\mathbf{U}}) \leftarrow_{\mathsf{R}} \mathsf{TrapGen}(1^\lambda, q, d), \mathbf{v} \leftarrow_{\mathsf{R}} \mathbb{Z}_N^m : (\mathbf{U}, T_{\mathbf{U}}, \mathbf{v})\right\}.$$

Proof: By Lemma 2.1 (left over hash lemma), the following distribution have statistical distance at most $2^{-\lambda}$:

$$(\widetilde{\mathbf{U}}, \widetilde{\mathbf{U}}\mathbf{r}_1) \text{ and } (\widetilde{\mathbf{U}}, \mathbf{v}),$$

where $\mathbf{v} \leftarrow_{\mathsf{R}} \mathbb{Z}_N^{\widetilde{m}}$ and $\widetilde{m} = d\lceil \log(N) \rceil + 2\lambda$. Thus, the following distributions have statistical distance at most $2^{-\lambda}$:

$$\left(\mathbf{U}, T_{\mathbf{U}}, \widetilde{\mathbf{U}}\mathbf{r}_1 + (-\widetilde{\mathbf{U}}\mathbf{R} + \mathbf{G})\mathbf{r}_2\right) \quad \text{and} \quad (\mathbf{U}, T_{\mathbf{U}}, \mathbf{v}),$$

where the left one corresponds to top distribution in the lemma, and the right one corresponds to the bottom one. □

Theorem 3.5 (SRL security)**.** *For any polynomials $d(\cdot), \ell(\cdot)$, the GSW FHE for depth d circuits with ℓ-batch correctness presented above is SRL-secure, under the LWE assumption.*

Proof: We proceed via a hybrid argument using the following ensembles for all $b \in \{0,1\}$.

- $\underline{\mathcal{D}_\lambda^b}$: is the ensemble given in Definition 3.3.

- $\underline{\mathcal{H}_\lambda^{b.1}}$: is as $\mathcal{D}_\lambda^b$ except the LWE sample $\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top$ from the public key is switched to a uniformly random vector using the LWE assumption. That is, the public key is computed as follows: $\mathbf{A} \leftarrow_{\mathsf{R}} \mathbb{Z}_N^{n\times m}$, $\mathbf{v} \leftarrow_{\mathsf{R}} \mathbb{Z}_N^m$, $\mathbf{U} = \begin{pmatrix}\mathbf{A} \\ \mathbf{v}^\top\end{pmatrix}$; the gadget matrix $\mathbf{G}$ is computed as in $\mathcal{D}_\lambda^b$, and $\mathsf{pk} = (N, B, \mathbf{U}, \mathbf{G})$. The secret key is also computed as in $\mathcal{D}_\lambda^b$ (but now it is uncorrelated with pk), namely: $\mathsf{s} \leftarrow_{\mathsf{R}} \chi^n$, $\mathsf{sk} = (-\mathbf{s}, 1) \in \mathbb{Z}_N^{n+1}$. The challenge ciphertext is computed as $\mathsf{ct} = \mathsf{Enc}_{\mathsf{pk}}(\mathbf{m}^\star; \mathbf{r})$ and the oracle $\mathcal{O}_{\mathsf{SRL}}(\mathbf{m}^\star, \mathbf{r})$ behaves as in $\mathcal{D}_\lambda^b$. From the LWE assumption, we have:

$$\mathcal{D}_\lambda^b \approx_c \mathcal{H}_\lambda^{b.1}.$$

- $\underline{\mathcal{H}_\lambda^{b.2}}$: is as $\mathcal{H}_\lambda^{b.1}$ except the matrix $\mathbf{U}$ from the public key is sampled from $(\mathbf{U}, T_{\mathbf{U}}) \leftarrow \mathsf{TrapGen}(1^\lambda, N, n+1)$. By Property 3, this is statistically close to generating a uniformly random $\mathbf{U} \leftarrow_{\mathsf{R}} \mathbb{Z}_N^{(n+1)\times m}$ as done in $\mathcal{H}_\lambda^{b.1}$. The rest of the adversary view can be generated from $\mathbf{U}$, thus, we have:

$$\mathcal{H}_\lambda^{b.1} \approx_s \mathcal{H}_\lambda^{b.2}.$$

- $\underline{\mathcal{H}_\lambda^{b.3}}$: is as $\mathcal{H}_\lambda^{b.2}$ except we use the oracle $\widetilde{\mathcal{O}}_{\mathsf{SRL}}$ instead of $\mathcal{O}_{\mathsf{SRL}}$:

```
$\underline{\widetilde{\mathcal{O}}_{\mathsf{SRL}}(\mathbf{m}^\star, \mathsf{ct})}$:
$\mathbf{r}^\star \leftarrow_{\mathsf{R}} \mathcal{R}^\star$, $\mathsf{ct}^\star = \mathsf{Enc}^\star_{\mathsf{pk}}(0; \mathbf{r}^\star)$
$(f, \alpha) \leftarrow \mathcal{A}(\mathsf{ct}^\star)$
$\mathsf{ct}_f = \mathsf{Eval}'(\mathsf{pk}, f, 1, \mathsf{ct})$. Parse $\mathsf{ct}_f = \left(\mathbf{A}\mathbf{r}_f, \mathbf{v}^\top\mathbf{r}_f + f(\mathbf{m}^\star)\right) \in \mathbb{Z}_N^{n+1}$.
Compute $\mathbf{t}_f = (\mathbf{A}\mathbf{r}_f, \mathbf{v}^\top\mathbf{r}_f) \in \mathbb{Z}_N^{n+1}$, and $\widetilde{\mathbf{r}}_f \leftarrow \mathsf{PreImSamp}(\mathbf{U}, T_{\mathbf{U}}, \mathbf{t}_f, \widetilde{B})$.
If $f(\mathbf{m}^\star) = \alpha$, and $f$ is of depth $d$, then $\mathsf{leak} = \mathbf{r}^\star - \widetilde{\mathbf{r}}_f \in \mathcal{R}^\star$.
Otherwise, $\mathsf{leak} = \perp$. Return $\mathsf{leak}$.
```

Note that the oracle $\widetilde{\mathcal{O}}_{\mathsf{SRL}}$ only takes as input the message $\mathbf{m}^\star \in \{0,1\}^*$ and the challenge ciphertext $\mathsf{ct} = \mathsf{Enc}_{\mathsf{pk}}(\mathbf{m}^\star; \mathbf{r})$, but not the randomness $\mathbf{r}$ itself. Instead of computing the evaluated randomness $\mathbf{r}_f = \mathsf{Eval}_{\mathsf{rand}}(\mathsf{pk}, f, \mathbf{m}^\star, \mathbf{r})$, it computes a small $\widetilde{\mathbf{r}}_f$ that is consistent with ct_f, that is, such that $\mathsf{ct}_f = (\mathbf{A}\widetilde{\mathbf{r}}_f, \mathbf{v}^\top\widetilde{\mathbf{r}}_f + f(\mathbf{m}))$. Clearly, the distributions: $(\mathsf{ct}, \mathbf{r}_f)$, which corresponds to $\mathcal{H}_\lambda^{b.2}$ and $(\mathsf{ct}, \widetilde{\mathbf{r}}_f)$, which corresponds to $\mathcal{H}_\lambda^{b.3}$ are distinct — for one thing, the first distribution has less entropy than the second distribution where $\widetilde{\mathbf{r}}_f$ is sampled freshly. However, the value $\widetilde{\mathbf{r}}_f$ is shielded by the noisy randomness $\mathbf{r}^\star \leftarrow_{\mathsf{R}} \mathcal{R}^\star$. Because it is of much larger magnitude than $\mathbf{r}_f$ and $\widetilde{\mathbf{r}}_f$, the latter can smudge the difference $\delta_f = \mathbf{r}_f - \widetilde{\mathbf{r}}_f$, which would successfully transition from $\mathcal{H}_\lambda^{b.2}$ to $\mathcal{H}_\lambda^{b.3}$. To effectively hide δ_f, we need to make sure $\mathbf{r}^\star \in \mathbb{Z}^m$ itself is hidden. Partial information is revealed in

$\mathsf{ct}^\star = \mathsf{Enc}^\star_{\mathsf{pk}}(0; \mathbf{r}^\star)$, of the form $\mathbf{U}\mathbf{r}^\star \in \mathbb{Z}_N^{n+1}$. Intuitively, the component of $\mathbf{r}^\star$ along $\mathbf{U}$ is revealed by $\mathsf{ct}^\star$, but the remaining entropy of $\mathbf{r}^\star$ is hidden; in particular, its component along $\mathbf{U}^\perp$, the orthogonal space of $\mathbf{U}$, is hidden. Because $\widetilde{\mathbf{r}}_f$ is consistent with ct_f, we have $\mathbf{U}\delta_f = \mathbf{0}$; that is, δ_f is orthogonal to $\mathbf{U}$. The orthogonal component of $\mathbf{r}^\star$ can simply smudge δ_f. This argument is formalized in Lemma 3.2. Overall, we have:

$$\mathcal{H}^{b.2}_\lambda \approx_s \mathcal{H}^{b.3}_\lambda.$$

To complete the proof of Theorem 3.5, we show that $\mathcal{H}^{0.3}_\lambda \approx_s \mathcal{H}^{1.3}_\lambda$, which gives overall:

$$\mathcal{D}^0_\lambda \approx_c \mathcal{H}^{0.1}_\lambda \approx_s \mathcal{H}^{0.2}_\lambda \approx_s \mathcal{H}^{0.3}_\lambda \approx_s \mathcal{H}^{1.3}_\lambda \approx_s \mathcal{H}^{1.2}_\lambda \approx_s \mathcal{H}^{1.1}_\lambda \approx_c \mathcal{D}^1_\lambda.$$

We look at the challenge ciphertext in $\mathcal{H}^{0.3}_\lambda$. It of the form $\mathsf{ct} = (\mathsf{ct}_i)_{i \in [|\mathbf{m}^\star|]}$ where for all $i \in [|\mathbf{m}^\star|]$, $\mathsf{ct}_i = \mathbf{U}\mathbf{R}_i + m_i^\star \mathbf{G} \in \mathbb{Z}_N^{(n+1)\times w}$, where the randomness $\mathbf{R}_i \leftarrow_{\mathsf{R}} [-1,1]^{m \times w}$, and $m_i^\star$ denotes the i'th bit of the message $\mathbf{m}^\star$. The only information revealed about $\mathbf{R}_i$ is $\mathbf{U}\mathbf{R}_i$ — in particular the oracle $\widetilde{\mathcal{O}}_{\mathsf{SRL}}$ does not require to know $\mathbf{R}_i$. Thus, we can use Lemma 3.1, which directly implies that the following two distributions have statistical distance at most $2^{-\lambda}$:

$$\Big\{(\mathbf{U}, T_{\mathbf{U}}) \leftarrow \mathsf{TrapGen}(1^\lambda, N, n+1), \mathbf{R}_i \leftarrow_{\mathsf{R}} [-1,1]^{m \times w} : \Big(\mathbf{U}, T_{\mathbf{U}}, \big(\mathbf{U}\mathbf{R}_i + m_i^0 \mathbf{G}\big)_{i \in [s]}\Big)\Big\}$$
$$\Big\{(\mathbf{U}, T_{\mathbf{U}}) \leftarrow \mathsf{TrapGen}(1^\lambda, N, n+1), \mathbf{R}_i \leftarrow_{\mathsf{R}} [-1,1]^{m \times w} : \Big(\mathbf{U}, T_{\mathbf{U}}, \big(\mathbf{U}\mathbf{R}_i + m_i^1 \mathbf{G}\big)_{i \in [s]}\Big)\Big\},$$

where $s = |\mathbf{m}^0| = |\mathbf{m}^1|$. The first distribution corresponds to $\mathcal{H}^{3.0}_\lambda$, whereas the second corresponds to $\mathcal{H}^{3.1}_\lambda$. □

Lemma 3.2. *The following ensembles are statistically close:* $\mathcal{H}^{b.2}_\lambda \approx_s \mathcal{H}^{b.3}_\lambda$.

Proof: We introduce intermediate ensembles $\{\mathcal{H}^{b.2.i}_\lambda\}_{\lambda \in \mathbb{N}}$ for $i = 1, 2$ which are defined as follows.

Ensemble $\mathcal{H}^{b.2.1}_\lambda$ is as $\mathcal{H}^{b.2}_\lambda$ except is uses the following oracle $\mathcal{O}^1_{\mathsf{SRL}}$ instead of $\mathcal{O}_{\mathsf{SRL}}$.

```
$\underline{\mathcal{O}^1_{\mathsf{SRL}}(\mathbf{m}^\star, \mathbf{r})}$:
$\mathbf{r}^\star \leftarrow_{\mathsf{R}} \mathcal{R}^\star$, $\widetilde{\mathbf{r}}_0 \leftarrow \mathsf{PreImSamp}(\mathbf{U}, T_{\mathbf{U}}, \mathbf{0}, \widetilde{B})$, $\mathsf{ct}^\star = \mathsf{Enc}^\star_{\mathsf{pk}}(0; \mathbf{r}^\star - \widetilde{\mathbf{r}}_0)$
$(f, \alpha) \leftarrow \mathcal{A}(\mathsf{ct}^\star)$
$\mathbf{r}_f = \mathsf{Eval}_{\mathsf{rand}}(\mathsf{pk}, f, \mathbf{r}, \mathbf{m}^\star)$.
If $f(\mathbf{m}^\star) = \alpha$ and $f$ is of depth $d$, then $\mathsf{leak} = \mathbf{r}^\star - \widetilde{\mathbf{r}}_0 - \mathbf{r}_f \in \mathcal{R}^\star$.
Otherwise, $\mathsf{leak} = \perp$. Return $\mathsf{leak}$.
```

We first prove that:

$$\mathcal{H}^{b.2}_\lambda \approx_s \mathcal{H}^{b.2.1}_\lambda.$$

The only difference between these ensembles is that $\mathcal{O}^1_{\mathsf{SRL}}$ adds a pre-image of $\mathbf{0} \in \mathbb{Z}_N^m$ to the shielded randomness, that is, it uses $\mathbf{r}^\star - \widetilde{\mathbf{r}}_0$ with $\mathbf{r}^\star \leftarrow_{\mathsf{R}} \mathcal{R}^\star$ and $\widetilde{\mathbf{r}}_0 \leftarrow \mathsf{PreImSamp}(\mathbf{U}, T_{\mathbf{U}}, \mathbf{0}, \widetilde{B})$ instead of $\mathbf{r}^\star$.

By Property 4, $\widetilde{\mathbf{r}}_0 \in \mathbb{Z}_N^m$ is such that $\|\widetilde{\mathbf{r}}_0\|_\infty < \widetilde{B}2^{\lambda/2}$. Thus, by Lemma 2.2 (smudging), the following distributions have statistical distance at most $2^{-\lambda/2}$:

$$\{\mathbf{r}^\star \leftarrow_{\mathsf{R}} [-\widetilde{B}2^\lambda, \widetilde{B}2^\lambda]^m : \mathbf{r}^\star\} \quad \text{and} \quad \{\mathbf{r}^\star \leftarrow_{\mathsf{R}} [-\widetilde{B}2^\lambda, \widetilde{B}2^\lambda]^m : \mathbf{r}^\star - \widetilde{\mathbf{r}}_0\}.$$

The leftmost ensemble corresponds to $\mathcal{H}^{b.2}_\lambda$, whereas the rightmost ensemble corresponds to $\mathcal{H}^{b.2.1}_\lambda$. This completes the proof that $\mathcal{H}^{b.2}_\lambda \approx_s \mathcal{H}^{b.2.1}_\lambda$.

Now, we introduce another intermediate ensemble, $\mathcal{H}^{b.2.2}_\lambda$, which is defined as $\mathcal{H}^{b.2.1}_\lambda$ except is uses the following oracle $\mathcal{O}^2_{\mathsf{SRL}}$ instead of $\mathcal{O}^1_{\mathsf{SRL}}$.

$\underline{\mathcal{O}^2_{\mathsf{SRL}}(\mathbf{m}^\star, \mathbf{r})}$:

$\mathbf{r}^\star \leftarrow_{\mathsf{R}} \mathcal{R}^\star$, $\mathsf{ct}^\star = \mathsf{Enc}^\star_{\mathsf{pk}}(0; \mathbf{r}^\star)$
$(f, \alpha) \leftarrow \mathcal{A}(\mathsf{ct}^\star)$
$\mathbf{r}_f = \mathsf{Eval}_{\mathsf{rand}}(\mathsf{pk}, f, \mathbf{r}, \mathbf{m}^\star)$, $\widetilde{\mathbf{r}_0} \leftarrow \mathsf{PreImSamp}(\mathbf{U}, T_{\mathbf{U}}, \mathbf{0}, \widetilde{B})$.
If $f(\mathbf{m}^\star) = \alpha$ and f is of depth d, then $\mathsf{leak} = \mathbf{r}^\star - \widetilde{\mathbf{r}_0} - \mathbf{r}_f \in \mathcal{R}^\star$.
Otherwise, $\mathsf{leak} = \perp$. Return leak.

by Property 4, we have, $\mathbf{U}\widetilde{\mathbf{r}_0} = \mathbf{0}$. This implies $\mathsf{Enc}^\star_{\mathsf{pk}}(0; \mathbf{r}^\star - \widetilde{\mathbf{r}_0}) = \mathsf{Enc}^\star_{\mathsf{pk}}(0; \mathbf{r}^\star)$. Thus, we have:

$$\mathcal{H}^{b.2.1}_\lambda = \mathcal{H}^{b.2.2}_\lambda.$$

To conclude the proof of this lemma, we now prove that:

$$\mathcal{H}^{b.2.2}_\lambda \approx_s \mathcal{H}^{b.3}_\lambda.$$

To do so, we note that $\mathbf{r}_f \in \mathbb{Z}_N^m$ is such that $\|\mathbf{r}_f\|_\infty < \widetilde{B}$. Moreover, it is independent of the vector $\widetilde{\mathbf{r}_0} \leftarrow \mathsf{PreImSamp}(\mathbf{U}, T_{\mathbf{U}}, \mathbf{0}, \widetilde{B})$. Therefore, we can use Property 5, which states that the following distributions have statistical distance at most $2^{-\lambda/2}$:

$$\{\widetilde{\mathbf{r}_0} \leftarrow_{\mathsf{R}} \mathsf{PreImSamp}(\mathbf{U}, T_{\mathbf{U}}, \mathbf{0}, \widetilde{B}) : \widetilde{\mathbf{r}_0} + \mathbf{r}_f \in \mathbb{Z}_N^m\} \text{ and } \{\widetilde{\mathbf{r}_f} \leftarrow \mathsf{PreImSamp}(\mathbf{U}, T_{\mathbf{U}}, \mathbf{U}\mathbf{r}_f, \widetilde{B}) : \widetilde{\mathbf{r}_f}\}.$$

The leftmost distribution corresponds to $\mathcal{H}^{b.2.2}_\lambda$, whereas the rightmost distribution corresponds to $\mathcal{H}^{b.3}_\lambda$. □

4 Constructing XiO for $\mathsf{P}^{\log}/\mathsf{poly}$

We present an XiO for the class of circuits $\mathcal{C}_{\log(n),s,d}$ for polynomials $n(\cdot), s(\cdot), d(\cdot)$, from the following building blocks:

- DJ's Linearly-Homomorphic Encryption scheme $\mathcal{LHE} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}}, \overline{\mathsf{Add}}, \overline{\mathsf{Subtract}}, \overline{\mathsf{Ext}}, \overline{\mathsf{Rec}})$, recalled in Section 4.1,
- GSW's Fully-Homomorphic Encryption scheme $\mathcal{FHE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Enc}^\star, \mathsf{Eval})$, recalled in Section 3.2.2.

4.1 Building Block 1: DJ's LHE

We recall the Damgård Jurik Linearly Homomorphic Encryption scheme from [DJ01], which generalizes Paillier's encryption scheme [Pai99] to larger message spaces, whose security relies on the Decisional Composite Residuosity (DCR) assumption, recalled here.

Beyond linear homomorphism, we show DJ's LHE has the additional proposition that the randomness is extractable. That is, given the secret key, one can recover the random coins used to produce a ciphertext. This randomness can be used with the ciphertext to recover the plaintext. We show that the randomness is succinct, that is, its size is independent of the plaintext size. We also prove a weak circuit privacy proposition that is similar to that achieved by GSW's FHE in Section 4.2. These properties are similar to those used in [BDGM20].

Definition 4.1 (Decisional Composite Residuosity (DCR) assumption [Pai99])**.** *There exists a PPT algorithm* $\mathsf{RSAsample}$ *that on input a security parameter* λ*, outputs a pair* $(N, \phi(N))$ *where* N *is a* 2λ*-bits integer,* ϕ *denotes Euler's totient function; such that* $\gcd(\phi(N), N) = 1$ *and such that for all polynomial* $\zeta(\cdot)$*, the following ensembles are computationally indistinguishable:*

$$\{\mathcal{D}^0_\lambda\}_{\lambda\in\mathbb{N}} = \left\{(N, \phi(N)) \leftarrow \mathsf{RSAsample}(1^\lambda); r \leftarrow_{\mathsf{R}} \mathbb{Z}_N : r^{N^{\zeta(\lambda)}} \in \mathbb{Z}_{N^{\zeta(\lambda)+1}}\right\}_{\lambda\in\mathbb{N}}$$

$$\{\mathcal{D}^0_\lambda\}_{\lambda\in\mathbb{N}} = \left\{(N, \phi(N)) \leftarrow \mathsf{RSAsample}(1^\lambda); u \leftarrow_{\mathsf{R}} \mathbb{Z}_{N^{\zeta(\lambda)+1}} : u \in \mathbb{Z}_{N^{\zeta(\lambda)+1}}\right\}_{\lambda\in\mathbb{N}}$$

As explained in [DJ01] in further details, the algorithm $\mathsf{RSAsample}(1^\lambda)$ samples two safe primes p, q of λ bits each, and compute the RSA modulus $N = pq$.

4.1.1 The DJ scheme

For any polynomial $\ell(\cdot)$, we present the randomness-extractable, randomness-succinct, weakly circuit private LHE of plaintext size $\ell(\cdot)$ from [DJ01]. The scheme consists of the PPT algorithms $(\mathsf{CRSgen}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Add})$ presented here. We also present in Section 4.1.2 additional PPT algorithms that will be useful for us to define extra properties we will rely on to build XiO.

• $\underline{\mathsf{CRSgen}(1^\lambda)}$:
Given as input the security parameter $\lambda \in \mathbb{N}$, it outputs $\mathsf{crs} = 1^\lambda$. That is, the scheme does not require an actual CRS.

• $\underline{\mathsf{Gen}(\mathsf{crs})}$:
Given $\mathsf{crs} = 1^\lambda$, it uses the sampling algorithm from Definition 4.1, $(N', \phi(N')) \leftarrow \mathsf{RSAsample}(1^\lambda)$, where $N' \in \mathbb{N}$ is a 2λ-bit modulus, ϕ denotes Euler's totient function, and we have $\gcd(\phi(N'), N') = 1$. Then it chooses a polynomial $\zeta(\cdot)$ such that $2^{\ell(\lambda)+2\lambda} > N \geq 2^{\ell(\lambda)}$, where $N = N'^{\zeta(\lambda)}$. It sets $\mathsf{pk} = (N, \zeta)$, $\mathsf{sk} = \phi(N')$ and outputs $(\mathsf{pk}, \mathsf{sk})$. The plaintext space is $\mathbb{Z}_{N^\zeta}$. The randomness space for Enc is $\mathbb{Z}^*_N$, and the ciphertext space is $\mathbb{Z}^*_{N^\zeta}$.

• $\underline{\mathsf{Enc}(\mathsf{pk}, m; r)}$:
Given the public pk, a message $m \in \mathbb{Z}_{N^\zeta}$ and the randomness $r \leftarrow_{\mathsf{R}} \mathbb{Z}^*_N$, it outputs the ciphertext $\mathsf{ct} = r^{N^\zeta} \cdot (1 + N)^m \in \mathbb{Z}^*_{N^{\zeta+1}}$. For any $\nu \in \mathbb{N}$, $\mathbf{m} = (m_1, \ldots, m_\nu) \in \mathbb{Z}^\nu_{N^\zeta}$, we denote by $\mathsf{Enc}_{\mathsf{pk}}(\mathbf{m})$ the concatenation of the encryptions $\mathsf{Enc}_{\mathsf{pk}}(m_1), \ldots, \mathsf{Enc}_{\mathsf{pk}}(m_\nu)$.

• $\underline{\mathsf{Add}(\mathsf{pk}, \mathsf{ct}_1, \mathsf{ct}_2)}$:
Given the public pk and two ciphertexts ct_1, ct_2, it outputs the evaluated ciphertext $\mathsf{ct} = \mathsf{ct}_1 \cdot \mathsf{ct}_2 \in \mathbb{Z}^*_{N^{\zeta+1}}$, where $\cdot$ denotes the integer multiplication in $\mathbb{Z}^*_{N^{\zeta+1}}$.

• $\underline{\mathsf{Dec}(\mathsf{pk}, \mathsf{ct})}$:
Given the secret key sk, a ciphertext ct, it runs $\mathsf{Ext}(\mathsf{sk}, \mathsf{ct})$ to recover the randomness $r \in \mathbb{Z}^*_N$, then runs $\mathsf{Rec}(\mathsf{pk}, \mathsf{ct}, r)$ to recover the plaintext, where the PPT algorithms Ext and Rec are defined below.

The correctness of the scheme is implied by the properties 11 and 12 proven below. Recall that an LHE must fulfill the following linear homomorphism proposition.

Proposition 6 (Linear homomorphism)**.** *For all* $\lambda \in \mathbb{N}$*, all* $(\mathsf{pk}, \mathsf{sk})$ *in the support of* $\mathsf{Gen}(1^\lambda)$ *that defines the message space* $\mathbb{Z}_{N^\zeta}$*, all messages* $m_1, m_2 \in \mathbb{Z}_{N^\zeta}$*, all* ct_1*,* ct_2 *in the support of*

$\mathsf{Enc}_{\mathsf{pk}}(m_1)$ *and* $\mathsf{Enc}_{\mathsf{pk}}(m_2)$*, respectively, the evaluated ciphertext* $\mathsf{ct} = \mathsf{Add}(\mathsf{pk}, \mathsf{ct}_1, \mathsf{ct}_2)$ *is in the support of* $\mathsf{Enc}_{\mathsf{pk}}(m_1 + m_2)$*.*

Proof: For any ciphertexts ct_1, ct_2 in the support of $\mathsf{Enc}(\mathsf{pk}, m_1)$, $\mathsf{Enc}(\mathsf{pk}, m_2)$, we have $\mathsf{ct}_1 \cdot \mathsf{ct}_2 = r_1^{N^\zeta} \cdot (1+N)^{m_1} \cdot r_2^{N^\zeta} \cdot (1+N)^{m_2} = (r_1 \cdot r_2)^{N^\zeta} \cdot (1+N)^{m_1+m_2} \in \mathbb{Z}^*_{N^{\zeta+1}}$, which is in the support of $\mathsf{Enc}(\mathsf{pk}, m_1 + m_2)$. Similarly, $\mathsf{ct}_1 \cdot \mathsf{ct}_2^{-1} = (r_1 \cdot r_2^{-1})^{N^\zeta} \cdot (1+N)^{m_1-m_2} \in \mathbb{Z}^*_{N^{\zeta+1}}$, where r_2^{-1} denotes the inverse of r_2 in $\mathbb{Z}^*_N$. $\square$

Theorem 4.2 (Security [DJ01])**.** *Assuming the DCR assumption (see Definition 4.1), the DJ scheme is secure.*

4.1.2 Additional Properties

We present several additional PPT algorithms satisfying the properties listed below, that will be useful for us in the context of building an XiO scheme.

- $\underline{\mathsf{Subtract}(\mathsf{pk}, \mathsf{ct}_1, \mathsf{ct}_2)}$:
Given the public pk and two ciphertexts ct_1, ct_2, it outputs the evaluated ciphertext $\mathsf{ct} = \mathsf{ct}_1 \cdot \mathsf{ct}_2^{-1} \in \mathbb{Z}^*_{N^{\zeta+1}}$, where ct_2^{-1} denotes the inverse of ct_2 in the multiplicative group $\mathbb{Z}^*_{N^{\zeta+1}}$.

- $\underline{\mathsf{InProd}(\mathsf{pk}, (\mathsf{ct}_1, \ldots, \mathsf{ct}_\nu), \mathbf{y})}$:
Given the public pk, ν ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_\nu$ and a vector $\mathbf{y} = (y_1, \ldots, y_\nu) \in \mathbb{Z}^\nu_{n^\zeta}$, it outputs the evaluated ciphertext $\mathsf{ct} = \prod_{i \in [\nu]} \mathsf{ct}_i^{y_i} \in \mathbb{Z}^*_{N^{\zeta+1}}$.

- $\underline{\mathsf{Ext}(\mathsf{sk}, \mathsf{ct})}$:
Given the secret key sk and a ciphertext ct, it computes $d = \mathsf{ct} \mod N$. Since $\gcd(N^\zeta, \phi(n)) = 1$, it can compute $N^{-\zeta} \in \mathbb{Z}$ such that $N^\zeta \cdot N^{-\zeta} = 1 \mod \phi(N)$. It outputs $d^{N^{-\zeta}} \in \mathbb{Z}^*_N$.

- $\underline{\mathsf{Rec}(\mathsf{pk}, \mathsf{ct}, r)}$:
Given the public key pk, a ciphertext ct, and randomness $r \in \mathbb{Z}^*_N$. It computes $d = \mathsf{ct} \cdot r^{-N^\zeta} \in \mathbb{Z}^*_{N^{\zeta+1}}$, where r^{-N^ζ} is the inverse of r^{N^ζ} in $\mathbb{Z}^*_{N^{\zeta+1}}$.

We now show the following properties are fulfilled.

Proposition 7 (Randomness succinctness.)**.** *There exists a polynomial $s(\cdot)$ such that for all $\lambda \in \mathbb{N}$, all* $(\mathsf{pk}, \mathsf{sk})$ *in the support of* $\mathsf{Gen}(1^\lambda)$ *that defines a randomness space $\mathcal{R}$ for* Enc*, all elements $r \in \mathcal{R}$ has bit size at most $|r| \le s(\lambda)$ (independent of $\ell(\lambda)$).*

Proof: For all $\lambda \in \mathbb{N}$, all $(\mathsf{pk}, \mathsf{sk})$ in the support of $\mathsf{Gen}(1^\lambda)$ defines a randomness space $\mathbb{Z}^*_N$ for an RSA modulus N of 2λ bits. In particular the size of N is independent of $\ell(\lambda)$. $\square$

Proposition 8 (Inner products)**.** *For all $\lambda \in \mathbb{N}$, all* $(\mathsf{pk}, \mathsf{sk})$ *in the support of* $\mathsf{Gen}(1^\lambda)$ *that defines the message space $\mathbb{Z}_{N^\zeta}$, all messages $m_1, \ldots, m_\nu \in \mathbb{Z}_{N^\zeta}$, all* $\mathsf{ct}_1, \ldots, \mathsf{ct}_\nu$ *in the support of* $\mathsf{Enc}_{\mathsf{pk}}(m_1), \ldots, \mathsf{Enc}_{\mathsf{pk}}(m_\nu)$*, respectively, all vectors $\mathbf{y} \in \mathbb{Z}^\nu_{N^\zeta}$, the evaluated ciphertext* $\mathsf{ct} = \mathsf{InProd}(\mathsf{pk}, (\mathsf{ct}_1, \ldots, \mathsf{ct}_\nu), \mathbf{y})$ *is in the support of* $\mathsf{Enc}_{\mathsf{pk}}(\mathbf{m}^\top \mathbf{y})$*.*

Proof: For any ciphertexts ct_i in the support of $\mathsf{Enc}(\mathsf{pk}, m_i)$ we have $\mathsf{ct}_i^{y_i} = (r_i^{y_i})^{N^\zeta} \cdot (1+N)^{y_i m_i} \in \mathbb{Z}^*_{N^{\zeta+1}}$, which is in the support of $\mathsf{Enc}(\mathsf{pk}, m_i \cdot y_i)$. Then, by multiplying ciphertexts in $\mathbb{Z}_{N^{\zeta+1}}$, we obtain a ciphertext in the support of $\mathsf{Enc}_{\mathsf{pk}}(\mathbf{m}^\top \mathbf{y})$. $\square$

Proposition 9 (Weak circuit privacy)**.** *For all $\lambda \in \mathbb{N}$, all $(\mathsf{pk}, \mathsf{sk})$ in the support of $\mathsf{Gen}(1^\lambda)$ that defines the message space $\mathbb{Z}_{N^\zeta}$, all messages $m_1, m_2 \in \mathbb{Z}_{N^\zeta}$, all ct_1 in the support of $\mathsf{Enc}_{\mathsf{pk}}(m_1)$, the following distributions are identical:*

$$\{\mathsf{ct}_2 \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_2); \mathsf{ct}_3 \leftarrow \mathsf{Add}(\mathsf{pk}, \mathsf{ct}_1, \mathsf{ct}_2) : (\mathsf{ct}_2, \mathsf{ct}_3)\}.$$
$$\{\mathsf{ct}_3 \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_1 + m_2); \mathsf{ct}_2 \leftarrow \mathsf{Subtract}(\mathsf{pk}, \mathsf{ct}_3, \mathsf{ct}_1) : (\mathsf{ct}_2, \mathsf{ct}_3)\}.$$

Proof: For any message $m_1, m_2 \in \mathbb{Z}_{N^\zeta}$, any ciphertext ct_1 in the support of $\mathsf{Enc}(\mathsf{pk}, m_1)$, we aim at proving that following distributions are identical:

$$\mathcal{D}_0 : \left\{r_2 \leftarrow_{\mathsf{R}} \mathbb{Z}_N^*, \mathsf{ct}_2 = r_2^{N^\zeta} \cdot (1+N)^{m_2}, \mathsf{ct}_3 = \mathsf{ct}_1 \cdot \mathsf{ct}_2 \in \mathbb{Z}_{N^{\zeta+1}}^* : (\mathsf{ct}_2, \mathsf{ct}_3)\right\}.$$
$$\mathcal{D}_1 : \left\{r_3 \leftarrow_{\mathsf{R}} \mathbb{Z}_N^*, \mathsf{ct}_3 = r_3^{N^\zeta} \cdot (1+N)^{m_1+m_2}, \mathsf{ct}_2 = \mathsf{ct}_3 \cdot \mathsf{ct}_1^{-1} \in \mathbb{Z}_{N^{\zeta+1}}^* : (\mathsf{ct}_2, \mathsf{ct}_3)\right\}.$$

To do so, we use the fact that the following distributions are identical (where operations are performed in $\mathbb{Z}_N^*$):

$$\mathcal{D}'_0 : \{r_1, r_2 \leftarrow_{\mathsf{R}} \mathbb{Z}_N^* : (r_1, r_2, r_1 \cdot r_2)\}.$$
$$\mathcal{D}'_1 : \{r_1, r_3 \leftarrow_{\mathsf{R}} \mathbb{Z}_N^* : (r_1, r_3 \cdot r_1^{-1}, r_3)\}.$$

The distribution $\mathcal{D}'_0$ corresponds to the randomness used in the ciphertext distribution $\mathcal{D}_0$, whereas the distribution $\mathcal{D}'_1$ corresponds to the randomness used in the ciphertext distribution $\mathcal{D}_1$. The distributions $\mathcal{D}'_0$ and $\mathcal{D}'_1$ are identical since for all $r_1 \in \mathbb{Z}_N^*$, the following distributions are identical:

$$\mathcal{D}''_0 : \{r_3 \leftarrow_{\mathsf{R}} \mathbb{Z}_N^* : r_3\}.$$
$$\mathcal{D}''_1 : \{r_3 \leftarrow_{\mathsf{R}} \mathbb{Z}_N^* : r_3 \cdot r_1\}.$$

The distribution $\mathcal{D}''_0$ corresponds to $\mathcal{D}'_0$, whereas the distribution $\mathcal{D}''_1$ corresponds $\mathcal{D}'_1$. $\square$

Proposition 10 (Density of the ciphertext space)**.** *Let $s(\cdot)$ be a polynomial and $\mathsf{CTsample}$ be a deterministic poly-time algorithm such that for all $\lambda \in \mathbb{N}$, all $(\mathsf{pk}, \mathsf{sk})$ in the support of $\mathsf{Gen}(1^\lambda)$ that defines the message space $\mathbb{Z}_{N^\zeta}$, all messages $m \in \mathbb{Z}_{N^\zeta}$, all ciphertexts ct in the support of $\mathsf{Enc}_{\mathsf{pk}}(m)$ have bit size $|\mathsf{ct}| \leq s(\lambda)$, and the following ensembles are statistically close:*

$$\left\{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda), \mathbf{r} \leftarrow_{\mathsf{R}} \{0,1\}^{s(\lambda)} : \mathsf{CTsample}(\mathbf{r})\right\}_{\lambda \in \mathbb{N}}.$$
$$\left\{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda) \textit{ with } \mathcal{M}_\lambda = \mathbb{Z}_{N^\zeta}, m \leftarrow_{\mathsf{R}} \mathbb{Z}_{N^\zeta}; \mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m) : \mathsf{ct}\right\}_{\lambda \in \mathbb{N}}.$$

Proof: One can sample a uniform random value $u \leftarrow_{\mathsf{R}} \mathbb{Z}_{N^{\zeta+1}}$ from $\lceil \log(N^{\zeta+1}) \rceil$ random bits. The random value $u \in \mathbb{Z}_{N^{\zeta+1}}$ can be written $u = r^{N^\zeta} \cdot (1+N)^m$ where $m \in \mathbb{Z}_{N^\zeta}$ and $r \in \mathbb{Z}_N^*$ with probability $\frac{\phi(N)}{N} < \frac{3}{2^\lambda}$ over the choice of $u \leftarrow_{\mathsf{R}} \mathbb{Z}_{N^{\zeta+1}}$. $\square$

Proposition 11 (Randomness recoverability.)**.** *For all $\lambda \in \mathbb{N}$, all $(\mathsf{pk}, \mathsf{sk})$ in the support of $\mathsf{Gen}(1^\lambda)$ that defines a message space $\mathbb{Z}_{N^\zeta}$, all $m \in \mathbb{Z}_{N^\zeta}$, all $r \in \mathbb{Z}_N^*$, we have: $\mathsf{Ext}(\mathsf{sk}, \mathsf{Enc}_{\mathsf{pk}}(m; r)) = r$.*

Proof: Let $\mathsf{ct} = r^{N^\zeta} \cdot (1+N)^m \in \mathbb{Z}_{N^{\zeta+1}}^*$, where $m \in \mathbb{Z}_{N^\zeta}$ and $r \in \mathbb{Z}_N^*$. We have $\mathsf{ct} \mod N = r^{N^\zeta \mod \phi(N)} \in \mathbb{Z}_N^*$. Since $\gcd(\phi(N), N^\zeta) = 1$, there is $N^{-\zeta} \in \mathbb{Z}$ such that $N^\zeta \cdot N^{-\zeta} = 1 \mod \phi(N)$. Thus, the algorithm $\mathsf{Ext}(\mathsf{sk}, \mathsf{ct})$ obtains $r \in \mathbb{Z}_N^*$. $\square$

Proposition 12 (Randomness decryptability)**.** *For all* $\lambda \in \mathbb{N}$*, all* $(\mathsf{pk}, \mathsf{sk})$ *in the support of* $\mathsf{Gen}(1^\lambda)$ *that defines a message space* $\mathbb{Z}_{N^\zeta}$*, all* $m \in \mathbb{Z}_{N^\zeta}$*, all* $r \in \mathbb{Z}_N^*$*, we have:* $\mathsf{Rec}(\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(m; r), r) = m$*.*

Proof: The algorithm $\mathsf{Rec}(\mathsf{pk}, \mathsf{ct}, r)$ computes $d = \mathsf{ct} \cdot r^{-N^\zeta} = (1+N)^m \in \mathbb{Z}^*_{N^{\zeta+1}}$ where r^{-N^ζ} is the inverse of r^{N^ζ} in $\mathbb{Z}^*_{N^{\zeta+1}}$. Then it applies Paillier's decryption recursively to obtain $m \in \mathbb{Z}^*_{N^\zeta}$. $\square$

4.2 Building Block 2: GSW's FHE

The construction is described in Section 3.2.2. We show that GSW satisfies a property that involves a public-key algorithm that re-randomizes evaluated ciphertext so that they look like fresh ciphertexts from the support of the noise encryption algorithm $\mathsf{Enc}^\star$. Namely, we show that there exists a PPT algorithm ReRand that takes as input the public key pk, an evaluated ciphertext ct, some random coins $\mathbf{r}^\star \in \mathcal{R}^\star$, and outputs an evaluated ciphertext ct computed as described below.

- $\underline{\mathsf{ReRand}(\mathsf{pk}, \mathsf{ct}; \mathbf{r}^\star)}$:

Given $\mathsf{pk} = (\mathbf{U}, \mathbf{G})$, $\mathsf{ct} \in \mathbb{Z}_N^{n+1}$ and $\mathbf{r}^\star \leftarrow_{\mathsf{R}} [-\widetilde{B}2^\lambda, \widetilde{B}2^\lambda]^m$, it outputs the re-randomized ciphertext $\mathsf{ct} + \mathbf{U}\mathbf{r}^\star \in \mathbb{Z}_N^{n+1}$.

Theorem 4.3 (weak circuit privacy)**.** *For all* $\lambda \in \mathbb{N}$*, all* $\mathsf{crs} = (N, \zeta)$ *in the support of* $\mathsf{CRSgen}(1^\lambda)$*, all* $\nu \in \mathbb{N}$*, all messages* $m_1, \ldots, m_\nu \in \{0,1\}^\nu$*, all depth-d circuits* $f : \{0,1\}^\nu \to \mathbb{Z}_N$*, all* $(\mathsf{pk}, \mathsf{sk})$ *in the support of* $\mathsf{Gen}(\mathsf{crs})$*, the following distributions have statistical distance at most* $2^{-\lambda}$*:*

$$\mathcal{D}_0 : \left\{ \begin{array}{l} \forall i \in [\nu] : \mathbf{R}_i \leftarrow_{\mathsf{R}} [-1,1]^{m \times w}, \mathsf{ct}_i \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_i; \mathbf{R}_i), \mathsf{ct}_f = \mathsf{Eval}'(\mathsf{pk}, f, 1, \mathsf{ct}_1, \ldots, \mathsf{ct}_\nu) \\ \mathbf{r}^\star \leftarrow_{\mathsf{R}} \mathcal{R}^\star, \mathsf{ct}^\star_f = \mathsf{ReRand}(\mathsf{pk}, \mathsf{ct}_f; \mathbf{r}^\star) : \left(\mathsf{pk}, \mathsf{sk}, \{\mathsf{ct}_i, \mathbf{R}_i\}_{i \in [\nu]}, \mathbf{r}^\star, \mathsf{ct}^\star_f\right). \end{array} \right\}$$

$$\mathcal{D}_1 : \left\{ \begin{array}{l} \forall i \in [\nu] : \mathbf{R}_i \leftarrow_{\mathsf{R}} [-1,1]^{m \times w}, \mathsf{ct}_i \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_i; \mathbf{R}_i), \mathbf{r}^\star \leftarrow_{\mathsf{R}} \mathcal{R}^\star, \mathsf{ct}^\star_f = \mathsf{Enc}^\star_{\mathsf{pk}}(f(\mathbf{m}); \mathbf{r}^\star) \\ \mathbf{r}_f = \mathsf{Eval}_{\mathsf{rand}}\left(\mathsf{pk}, f, (\mathbf{R}_i)_{i \in [\nu]}, (m_i)_{i \in [\nu]}\right) : \left(\mathsf{pk}, \mathsf{sk}, \{\mathsf{ct}_i, \mathbf{R}_i\}_{i \in [\nu]}, \mathbf{r}^\star - \mathbf{r}_f \in \mathbb{Z}_N^m, \mathsf{ct}^\star_f\right) \end{array} \right\}$$

Proof: By batch correctness of the scheme, the evaluated ciphertext is of the form $\mathsf{ct}_f = \left(\mathbf{A}\mathbf{r}_f, (\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top)\mathbf{r}_f + f(\mathbf{m})\right) \in \mathbb{Z}_N^{n+1}$, and the re-randomized ciphertext is of the form $\mathsf{ct}^\star_f = \left(\mathbf{A}(\mathbf{r}_f + \mathbf{r}^\star), (\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top)(\mathbf{r}_f + \mathbf{r}^\star) + f(\mathbf{m})\right) \in \mathbb{Z}_N^{n+1}$, where $\|\mathbf{r}_f\|\infty < \widetilde{B}$ and $\mathbf{r}^\star \leftarrow_{\mathsf{R}} [-\widetilde{B}2^\lambda, \widetilde{B}2^\lambda]^m$. By Lemma 2.2 (smudging), the following distributions have statistical distance at most $2^{-\lambda}$:

$$\mathbf{r}^\star \approx_s \mathbf{r}^\star - \mathbf{r}_f.$$

The leftmost distribution corresponds to $\mathcal{D}_0$, whereas the rightmost distribution corresponds to $\mathcal{D}_1$. $\square$

We also show that a fresh noisy encryption satisfies the following weak correctness notion.

Proposition 13 (weak noisy correctness)**.** *For all* $\lambda \in \mathbb{N}$*, all* $\mathsf{crs} = (N, \zeta)$ *in the support of* $\mathsf{CRSgen}(1^\lambda)$*, all* $(\mathsf{pk}, \mathsf{sk})$ *in the support of* $\mathsf{Gen}(\mathsf{crs})$*, all messages* $\mu \in \mathbb{Z}$*, all* $\mathbf{r}^\star \in \mathcal{R}^\star$*, writing* $\mathsf{ct} = \mathsf{Enc}^\star_{\mathsf{pk}}(\mu; \mathbf{r}^\star)$*, we have:*

$$\mathsf{sk}^\top\mathsf{ct} = \mu + \mathsf{noise} \in \mathbb{Z}_N,$$

with $|\mathsf{noise}| < B2^\lambda$*. Moreover,* noise *is independent of* μ*; that is, it is a deterministic function of* $\mathbf{r}^\star$ *and the random coins used to generate* crs *and* $(\mathsf{pk}, \mathsf{sk})$ *only.*

Proof: For every message $\mu \in \mathbb{Z}$ and randomness $\mathbf{r}^\star \in [-\widetilde{B}2^\lambda, \widetilde{B}2^\lambda]^m$, we have $\mathsf{Enc}^\star_{\mathsf{pk}}(\mu; \mathbf{r}^\star) = \left(\mathbf{A}\mathbf{r}^\star, (\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top)\mathbf{r}^\star + \mu\right)$, where the matrix $\mathbf{U} = \begin{pmatrix} \mathbf{A} \\ \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top \end{pmatrix}$ is part pk, and $\|\mathbf{e}\|_\infty < B'$. We also have $\mathsf{sk} = (-\mathbf{s}, 1)$. Thus, denoting $\mathsf{ct} = \mathsf{Enc}^\star_{\mathsf{pk}}(\mu; \mathbf{r}^\star)$, we have: $\mathsf{sk}^\top \mathsf{ct} = \underbrace{\mathsf{noise}}_{=\mathbf{e}^\top \mathbf{r}^\star} + \mu \in \mathbb{Z}_N$, where $\|\mathsf{noise}\|_\infty < 2^\lambda \widetilde{B} B' m = 2^\lambda B$. Note that noise only depends on $\mathbf{e}$ and $\mathbf{r}^\star$, not the message μ. $\square$

4.3 XiO Construction

We directly dive into the formal description of the construction, see the introduction for a detailed overview.

• $\underline{\mathsf{Gen}(1^\lambda, 1^{n(\lambda)}, 1^{s(\lambda)}, 1^{d(\lambda)})}$:
Set the parameters:

- Choose a constant $0 < \varepsilon < 1$ that is small enough so as to ensure succinctness of the scheme (see paragraph succinctness below).
- Let $\mathcal{LHE} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}}, \overline{\mathsf{Add}}, \overline{\mathsf{Subtract}}, \overline{\mathsf{InProd}}, \overline{\mathsf{Ext}}, \overline{\mathsf{Rec}})$ be DJ's LHE parameterized by the plaintext size $p(\cdot)$ where for all $\lambda \in \mathbb{N}$, $p(\lambda) = 2(\delta(\lambda) \cdot \lambda + n(\lambda)^\varepsilon)$, and $\mathcal{FHE} = (\mathsf{CRSgen}, \mathsf{Gen}, \mathsf{Eval}', \mathsf{Dec}, \mathsf{Enc}^\star, \mathsf{Eval}_{\mathsf{rand}}, \mathsf{ReRand})$ be GSW's FHE parameterized by depth $\delta(\cdot)$ and that satisfies $n^\varepsilon(\cdot)$-batch correctness, where the polynomial $\delta(\cdot)$ is such that for all λ, all $(\overline{\mathsf{pk}}, \overline{\mathsf{sk}})$ in the support of $\overline{\mathsf{Gen}}(1^\lambda)$ that defines the message space $\mathbb{Z}_N$, all $(\mathsf{pk}, \mathsf{sk})$ in the support of $\mathsf{Gen}(\overline{\mathsf{pk}})$ where pk contains the bound $B \in \mathbb{N}$, the following circuits have depth at most $\delta(\lambda, n(\lambda)^\varepsilon)$:
  - circuit C_Π: on input $\mathbf{y} \in \{0,1\}^{\log(n(\lambda)^{1-\varepsilon})}$, it outputs $C_\Pi(\mathbf{y}) \in [0, n(\lambda)^\varepsilon - 1]$, whose binary decomposition is the $\mathbf{y}$'th chunk of the truth table of Π, that is: $(\Pi(\mathbf{y}, \mathbf{z}'))_{\mathbf{z}' \in \{0,1\}^{\log(n(\lambda)^\varepsilon)}} \in \{0,1\}^{\log(n(\lambda)^\varepsilon)}$.
  - the MSB circuits $f_\mathbf{y} : \{0,1\}^{|\overline{\mathsf{sk}}|} \to \mathbb{Z}_N$ for all $\mathbf{y} \in \{0,1\}^{n(\lambda)^{1-\varepsilon}}$, defined as follows. It takes as input a bit string $\mathbf{a} \in \{0,1\}^{|\overline{\mathsf{sk}}|}$, which it uses as an LHE secret key to compute $u_\mathbf{y} = \overline{\mathsf{Dec}}_\mathbf{a}(\mathsf{LHE.PubCoin}_\mathbf{y})$, where $\mathsf{LHE.PubCoin}_\mathbf{y}$ is interpreted as an LHE ciphertext $\overline{\mathsf{Enc}}_{\overline{\mathsf{pk}}}(u_\mathbf{y})$, with $u_\mathbf{y} \in \mathbb{Z}_N$, by density of the LHE ciphertext space. Then it outputs (minus) the most significant bits of $u_\mathbf{y}$, of the form: $-\mathsf{MSB}(u_\mathbf{y}) = -u_\mathbf{y} + \mathsf{LSB}(u_\mathbf{y}) \in \mathbb{Z}_N$, where the (shifted) least significant bits are of the form: $\mathsf{LSB}(u_\mathbf{y}) = u_\mathbf{y} \mod 2^{2\lambda} B - 2^{2\lambda} B/2 \in \mathbb{Z}_N$.

  Note that the ciphertext size $|\overline{\mathsf{ct}}|$ depends on $\delta(\lambda, n^\varepsilon(\lambda))$, which may it turn depend on the ciphertext size. The circular issue is avoided since the depth of the circuit $C_{\mathsf{LHE.PubCoin}_\mathbf{y}}(\cdot)$ that on input $\mathbf{a} \in \{0,1\}^{|\overline{\mathsf{sk}}|}$ outputs $\overline{\mathsf{Dec}}_\mathbf{a}(\mathsf{LHE.PubCoin}_\mathbf{y})$ is a fixed polynomial in λ.
- Let $s_{|\overline{\mathsf{ct}}|}(\cdot)$ be a polynomial such that for every $\lambda \in \mathbb{N}$, every $(\overline{\mathsf{pk}}, \overline{\mathsf{sk}})$ in the support of $\overline{\mathsf{Gen}}(1^\lambda)$ that defines the message space $\mathbb{Z}_N$, every message $m \in \mathbb{Z}_N$, every ciphertext in the support of $\overline{\mathsf{Enc}}_{\overline{\mathsf{pk}}}(m)$ has a bit size at most $s_{|\overline{\mathsf{ct}}|}(\lambda, n^\varepsilon(\lambda), d(\lambda))$.
- Let $s_{|\mathbf{r}^\star|}(\cdot)$ be a polynomial such that for every $\lambda \in \mathbb{N}$, every $(\overline{\mathsf{pk}}, \overline{\mathsf{sk}})$ in the support of $\overline{\mathsf{Gen}}(1^\lambda)$, every $(\mathsf{pk}, \mathsf{sk})$ in the support of $\mathsf{Gen}(\overline{\mathsf{pk}})$ that defines a noisy randomness space $\mathcal{R}^\star$, every $\mathbf{r}^\star \in \mathcal{R}^\star$ has bit size at most $s_{|\mathbf{r}^\star|}(\lambda, n^\varepsilon(\lambda), d(\lambda))$.
- $\mathsf{FHE.PubCoin} \leftarrow_{\mathsf{R}} \{0,1\}^{n(\lambda)^{1-\varepsilon} \cdot s_{|\mathbf{r}^\star|}(\lambda, n(\lambda)^\varepsilon, d(\lambda))}$.
- $\mathsf{LHE.PubCoin} \leftarrow_{\mathsf{R}} \{0,1\}^{n(\lambda)^{1-\varepsilon} \cdot s_{|\overline{\mathsf{ct}}|}(\lambda, n(\lambda)^\varepsilon, d(\lambda))}$.

Return $\mathsf{pp} = (\mathsf{FHE.PubCoin}, \mathsf{LHE.PubCoin})$.

- $\underline{\mathsf{Obf}(\mathsf{pp}, 1^{n(\lambda)}, \Pi)}$:

Sample the following parameters:

- $\left(\overline{\mathsf{pk}} = (N, \zeta), \overline{\mathsf{sk}}\right) \leftarrow \overline{\mathsf{Gen}}(1^\lambda)$.
- $\mathsf{crs} = \overline{\mathsf{pk}}$, $(\mathsf{pk} = (N, B, \mathbf{U}, \mathbf{G}), \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{crs})$ that defines the noisy randomness space $\mathcal{R}^\star$.

Compute the following ciphertexts:

- $\mathsf{ct}_1 \leftarrow \mathsf{Enc}_{\mathsf{pk}}\left(C_\Pi\right)$ where the circuit C_Π is defined above.
- $\mathsf{ct}_2 \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\overline{\mathsf{sk}})$.
- $\overline{\mathsf{ct}} \leftarrow \overline{\mathsf{Enc}}_{\overline{\mathsf{pk}}}(\mathsf{sk})$.

For all $\mathbf{y} \in \{0,1\}^{\log(n(\lambda)^{1-\varepsilon})}$, compute the following:

- $\mathsf{ct}_{1,\mathbf{y}} = \mathsf{Eval}(\mathsf{pk}, C_\mathbf{y}, 2^{2\lambda}B, \mathsf{ct}_1) \in \mathbb{Z}_N^{n+1}$, where the circuit $C_\mathbf{y}$ takes as input a circuit C_Π and outputs $C_\Pi(\mathbf{y}) \in \{0,1\}^{n(\lambda)^\varepsilon}$. The circuit $C_\mathbf{y}$ is evaluated homomorphically with scaling factor $2^{2\lambda}B$.
- $\mathsf{ct}_{\mathsf{MSB},\mathbf{y}} = \mathsf{Eval}(\mathsf{pk}, f_\mathbf{y}, 1, \mathsf{ct}_2) \in \mathbb{Z}_N^{n+1}$, where $f_\mathbf{y}$ is the MSB circuit described above. The latter is evaluated homomorphically with scaling factor 1.
- Parse $\mathsf{FHE.PubCoin}_\mathbf{y} = \mathbf{r}^\star_\mathbf{y} \in \mathcal{R}^\star$ and compute $\mathsf{ct}'_{\mathsf{MSB},\mathbf{y}} = \mathsf{ReRand}(\mathsf{pk}, \mathsf{ct}_{\mathsf{MSB},\mathbf{y}}; \mathbf{r}^\star_\mathbf{y})$.
- Compute $\overline{\mathsf{ct}}_{1,\mathbf{y}} = \overline{\mathsf{InProd}}\left(\overline{\mathsf{pk}}, \overline{\mathsf{ct}}, \mathsf{ct}_{1,\mathbf{y}}\right)$, $\overline{\mathsf{ct}}_{\mathsf{MSB},\mathbf{y}} = \overline{\mathsf{InProd}}\left(\overline{\mathsf{pk}}, \overline{\mathsf{ct}}, \mathsf{ct}_{\mathsf{MSB},\mathbf{y}}\right)$
- Then, it computes: $\overline{\mathsf{ct}}_\mathbf{y} = \overline{\mathsf{Add}}(\overline{\mathsf{pk}}, \overline{\mathsf{ct}}_{1,\mathbf{y}}, \overline{\mathsf{ct}}_{\mathsf{MSB},\mathbf{y}}, \mathsf{LHE.PubCoin}_\mathbf{y})$.
- Compute $r_\mathbf{y} \leftarrow \overline{\mathsf{Ext}}(\overline{\mathsf{sk}}, \overline{\mathsf{ct}}_\mathbf{y})$.

Return $\widetilde{\Pi} = \left(\mathsf{pk}, \overline{\mathsf{pk}}, \mathsf{ct}_1, \mathsf{ct}_2, \overline{\mathsf{ct}}, \{r_\mathbf{y}\}_{\mathbf{y} \in \{0,1\}^{\log(n(\lambda)^{1-\varepsilon})}}\right)$.

- $\underline{\mathsf{Eval}(\mathsf{pp}, \widetilde{\Pi}, \mathbf{x})}$:

  - Parse $\mathbf{x} \in \{0,1\}^{\log(n(\lambda))}$ as $\mathbf{x} = (\mathbf{y}, \mathbf{z})$ where $\mathbf{y} \in \{0,1\}^{\log(n(\lambda)^{1-\varepsilon})}$ and $\mathbf{z} \in \{0,1\}^{\log(n(\lambda)^\varepsilon)}$.
  - Compute $\overline{\mathsf{ct}}_\mathbf{y}$ as described above.
  - Recover $m_\mathbf{y} \leftarrow \overline{\mathsf{Rec}}(\overline{\mathsf{pk}}, \overline{\mathsf{ct}}_\mathbf{y}, r_\mathbf{y})$.
  - Compute $m'_\mathbf{y} = \lfloor 2^{-2\lambda}/B \cdot m_\mathbf{y} \rceil$.
  - Parse $m'_\mathbf{y}$ as $\left(\Pi(\mathbf{y}, \mathbf{z}')_{\mathbf{z}' \in \{0,1\}^{\log(n(\lambda)^\varepsilon)}}\right)$, and return $\Pi(\mathbf{y}, \mathbf{z})$.

Succinctness. By succinctness of $\mathcal{LHE}$ (Proposition 7), for all $\mathbf{y} \in \{0,1\}^{n(\lambda)^{1-\varepsilon}}$, we have $|r_\mathbf{y}| \leq 2\lambda$. In particular, it is independent of $n^\varepsilon(\lambda)$. The rest of the obfuscated circuit $\widetilde{\Pi}$ is of size $\mathsf{poly}(\lambda, n^\varepsilon(\lambda), d(\lambda))$. Overall, there exists a constant $c \in \mathbb{N}$ (independent of ε), polynomials $p_1(\cdot)$ and $p_2(\cdot)$ such that $\left|\widetilde{\Pi}\right| \in n^{c\varepsilon}(\lambda) \cdot p_1(\lambda) + n^{1-\varepsilon}(\lambda) \cdot p_2(\lambda)$. For succinctness, we pick an appropriately small $0 < \varepsilon < 1/c$.

Correctness.

- By the batch correctness of the GSW scheme $\mathcal{FHE}$ (Proposition 1), the ciphertext $\mathsf{ct}_{1,\mathbf{y}}$ is of the form:
$$\mathsf{ct}_{1,\mathbf{y}} = \Big(\mathbf{A}\mathbf{r}_{C_\mathbf{y}}, (\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top)\mathbf{r}_{C_\mathbf{y}} + 2^{2\lambda}B \cdot C_\Pi(\mathbf{y})\Big) \in \mathbb{Z}_N^{n+1}.$$
Thus, we have:
$$\mathsf{sk}^\top\mathsf{ct}_{1,\mathbf{y}} = 2^{2\lambda}B \cdot C_\Pi(\mathbf{y}) + \mathsf{noise}_{1,\mathbf{y}} \in \mathbb{Z}_N,$$
where $\mathsf{noise}_{1,\mathbf{y}} = \mathbf{e}^\top\mathbf{r}_{C_\mathbf{y}} \in \mathbb{Z}_N$. We have $|\mathbf{e}^\top\mathbf{r}_{C_\mathbf{y}}| < B$.

- By the density of the ciphertext space of $\mathcal{LHE}$ (Proposition 10), for all $\mathbf{y} \in \{0,1\}^{n(\lambda)^{1-\varepsilon}}$, we have $\mathsf{LHE.PubCoin}_\mathbf{y} = \overline{\mathsf{Enc}}_{\overline{\mathsf{pk}}}(u_\mathbf{y})$ with $u_\mathbf{y} \leftarrow_\mathsf{R} \mathbb{Z}_N$.

- By the noisy correctness of $\mathcal{FHE}$ (Proposition 13), the ciphertext $\mathsf{ct}'_{\mathsf{MSB},\mathbf{y}}$ is of the form:
$$\mathsf{ct}'_{\mathsf{MSB},\mathbf{y}} = \Big(\mathbf{A}(\mathbf{r}_{f_\mathbf{y}} + \mathbf{r}^\star_\mathbf{y}), (\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top)(\mathbf{r}_{f_\mathbf{y}} + \mathbf{r}^\star_\mathbf{y}) - u_\mathbf{y} + \mathsf{LSB}(u_\mathbf{y})\Big) \in \mathbb{Z}_N^{n+1},$$
where $\mathsf{LSB}(u_\mathbf{y}) = u_\mathbf{y} \mod 2^{2\lambda}B - 2^\lambda B/2 \in \mathbb{Z}_N$. Thus, we have:
$$\mathsf{sk}^\top\mathsf{ct}'_{\mathsf{MSB},\mathbf{y}} = -u_\mathbf{y} + \mathsf{LSB}(u_\mathbf{y}) + \mathsf{noise}_{\mathsf{MSB},\mathbf{y}} \in \mathbb{Z}_N,$$
where $\mathsf{noise}_{\mathsf{MSB},\mathbf{y}} = \mathbf{e}^\top(\mathbf{r}_{f_\mathbf{y}} + \mathbf{r}^\star_\mathbf{y}) \in \mathbb{Z}_N$. We have $|\mathbf{e}^\top\mathbf{r}^\star_\mathbf{y}| < B + 2^\lambda B$.

- By linear homomorphism of $\mathcal{LHE}$ (Proposition 6), the ciphertext $\overline{\mathsf{ct}}_\mathbf{y}$ is in the support of $\overline{\mathsf{Enc}}_{\overline{\mathsf{pk}}}(m_\mathbf{y})$, with:
$$m_\mathbf{y} = 2^{2\lambda}B \cdot C_\Pi(\mathbf{y}) + \mathsf{LSB}(u_\mathbf{y}) + \mathsf{noise}_{1,\mathbf{y}} + \mathsf{noise}_{\mathsf{MSB},\mathbf{y}} \in \mathbb{Z}_N.$$

- By randomness recoverability and randomness decryptability of $\mathcal{LHE}$ (Proposition 11 and 12), the evaluator of the obfuscated circuit recovers the message $m_\mathbf{y} \in \mathbb{Z}_N$.

- With probability $1 - \mathsf{negl}(\lambda)$ over the choice of $u_\mathbf{y} \leftarrow_\mathsf{R} \mathbb{Z}_N$, we have $|\mathsf{LSB}(u_\mathbf{y}) + \mathsf{noise}_{1,\mathbf{y}} + \mathsf{noise}_{\mathsf{MSB},\mathbf{y}}| < 2^{2\lambda}B/2$. Thus, $m'_\mathbf{y} = \lfloor m_\mathbf{y} 2^{-2\lambda}/B \rceil = C_\Pi(\mathbf{y})$, and the evaluator outputs $\Pi(\mathbf{y},\mathbf{z})$.

Theorem 4.4 (IND-security)**.** *The XiO scheme presented in Section 4.3 is IND-secure, provided the underlying $\mathcal{FHE}$ and $\mathcal{LHE}$ are 2-circular SRL secure (as per Definition 3.3).*

Proof: We proceed via a hybrid argument using the following ensembles for all $b \in \{0,1\}$.

- $\underline{\mathcal{D}^b_\lambda}$: this is the ensemble from Definition 2.4. For completeness, we describe it here.

  - Generation of pp: for all $\mathbf{y} \in \{0,1\}^{n^{1-\varepsilon}}$, $\mathsf{LHE.PubCoin}_\mathbf{y} \leftarrow_\mathsf{R} \{0,1\}^{s_{|\overline{\mathsf{ct}}|}}$ where $s_{|\overline{\mathsf{ct}}|}(\lambda, n(\lambda)^\varepsilon, d(\lambda))$ is the bit-size of the ciphertexts of $\mathcal{LHE}$; $\mathsf{FHE.PubCoin}_\mathbf{y} \leftarrow_\mathsf{R} \mathcal{R}^\star$, where $\mathcal{R}^\star$ denotes the noisy randomness space of $\mathcal{FHE}$. Return $\mathsf{pp} = \Big((\mathsf{LHE.PubCoin}_\mathbf{y})_{\mathbf{y}\in\{0,1\}^{n^{1-\varepsilon}}}, (\mathsf{FHE.PubCoin}_\mathbf{y})_{\mathbf{y}\in\{0,1\}^{n^{1-\varepsilon}}}\Big)$.

  - Generation of $\widetilde{\Pi_b}$: $(\overline{\mathsf{pk}}, \overline{\mathsf{sk}}) \leftarrow \overline{\mathsf{Gen}}(1^\lambda)$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\overline{\mathsf{pk}})$, $\mathsf{ct}_1 \leftarrow \mathsf{Enc}_\mathsf{pk}(C_\Pi)$, $\mathsf{ct}_2 \leftarrow \mathsf{Enc}_\mathsf{pk}(\overline{\mathsf{sk}})$, $\overline{\mathsf{ct}} \leftarrow \overline{\mathsf{Enc}}_{\overline{\mathsf{pk}}}(\mathsf{sk})$. For all $\mathbf{y} \in \{0,1\}^{n^{1-\varepsilon}}$ compute the following:

    - $\mathsf{ct}_{1,\mathbf{y}} = \mathsf{Eval}(\mathsf{pk}, C_\mathbf{y}, 2^{2\lambda}B, \mathsf{ct}_1) \in \mathbb{Z}_N^{n+1}$;
    - $\mathsf{ct}_{\mathsf{MSB},\mathbf{y}} = \mathsf{Eval}(\mathsf{pk}, f_\mathbf{y}, 1, \mathsf{ct}_2) \in \mathbb{Z}_N^{n+1}$;

– Parse $\mathsf{FHE.PubCoin}_{\mathbf{y}} = \mathbf{r}^\star_{\mathbf{y}} \in \mathcal{R}^\star$ and compute $\mathsf{ct}'_{\mathsf{MSB},\mathbf{y}} = \mathsf{ReRand}(\mathsf{pk}, \mathsf{ct}_{\mathsf{MSB},\mathbf{y}}; \mathbf{r}^\star_{\mathbf{y}})$.

– $\overline{\mathsf{ct}}_{1,\mathbf{y}} = \overline{\mathsf{InProd}}(\overline{\mathsf{pk}}, \overline{\mathsf{ct}}, \mathsf{ct}_{1,\mathbf{y}})$, $\overline{\mathsf{ct}}_{\mathsf{MSB},\mathbf{y}} = \overline{\mathsf{InProd}}(\overline{\mathsf{pk}}, \overline{\mathsf{ct}}, \mathsf{ct}_{\mathsf{MSB},\mathbf{y}})$;

– $\overline{\mathsf{ct}}_{\mathbf{y}} = \overline{\mathsf{Add}}(\overline{\mathsf{pk}}, \overline{\mathsf{ct}}_{1,\mathbf{y}}, \overline{\mathsf{ct}}_{\mathsf{MSB},\mathbf{y}}, \mathsf{LHE.PubCoin}_{\mathbf{y}})$;

– $r_{\mathbf{y}} \leftarrow \overline{\mathsf{Ext}}(\overline{\mathsf{sk}}, \overline{\mathsf{ct}}_{\mathbf{y}})$.

Return $\widetilde{\Pi}_b = (\mathsf{pk}, \overline{\mathsf{pk}}, \mathsf{ct}_1, \mathsf{ct}_2, \overline{\mathsf{ct}}, (r_{\mathbf{y}})_{\mathbf{y}\in\{0,1\}^{n^\varepsilon}})$.

• $\underline{\mathcal{H}^{b.1}_\lambda}$: the ensemble samples $\mathsf{LHE.PubCoin}$ as in $\mathcal{D}^b_\lambda$, but does not sample $\mathsf{FHE.PubCoin}$ just yet; it then generates $\widetilde{\Pi}_b$ as in $\mathcal{D}^b_\lambda$ up until the point that $\mathsf{ct}_{\mathsf{MSB}}$ gets re-randomized into $\mathsf{ct}'_{\mathsf{MSB}}$ via ReRand. Next, instead of performing the re-randomization, it samples $\mathsf{ct}'_{\mathsf{MSB}}$ as a *fresh* extra noisy encryption of $-\mathsf{MSB}(u_{\mathbf{y}})$ using randomness $\mathbf{r}^\star_{\mathbf{y}} \leftarrow_{\mathsf{R}} \mathcal{R}^\star$, and setting $\mathsf{FHE.PubCoin}_{\mathbf{y}}$ to be $\mathbf{r}^\star_{\mathbf{y}} - \mathbf{r}_{f_{\mathbf{y}}}$, where $\mathbf{r}_{f_{\mathbf{y}}}$ denotes the evaluated randomness computed via $\mathsf{Eval}_{\mathsf{rand}}$. Afterwards, experiment continues exactly the same way as in $\mathcal{D}^b_\lambda$.

By the weak circuit privacy of $\mathcal{FHE}$ (Theorem 4.3) we have:

$$\mathcal{D}^b_\lambda \approx_s \mathcal{H}^{b.1}_\lambda.$$

The latter states that these two distributions have statistical distance at most $2^{-\lambda}$:

$$\mathcal{D}_0 : \left\{ \begin{array}{l} \forall i \in [|\overline{\mathsf{sk}}|] : \mathbf{R}_i \leftarrow_{\mathsf{R}} [-1,1]^{m\times w}, \mathsf{ct}_i \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\overline{\mathsf{sk}}_i; \mathbf{R}_i), \mathsf{ct}_{f_{\mathbf{y}}} = \mathsf{Eval}'(\mathsf{pk}, f_{\mathbf{y}}, 1, \mathsf{ct}_1, \dots, \mathsf{ct}_{|\overline{\mathsf{sk}}|}) \\ \mathbf{r}^\star_{\mathbf{y}} \leftarrow_{\mathsf{R}} \mathcal{R}^\star, \mathsf{ct}^\star_{f_{\mathbf{y}}} = \mathsf{ReRand}(\mathsf{pk}, \mathsf{ct}_{f_{\mathbf{y}}}; \mathbf{r}^\star_{\mathbf{y}}) : \left(\mathsf{pk}, \mathsf{sk}, \{\mathsf{ct}_i, \mathbf{R}_i\}_{i\in[|\overline{\mathsf{sk}}|]}, \mathbf{r}^\star, \mathsf{ct}^\star_{f_{\mathbf{y}}}\right). \end{array} \right\}$$

$$\mathcal{D}_1 : \left\{ \begin{array}{l} \forall i \in [|\overline{\mathsf{sk}}|] : \mathbf{R}_i \leftarrow_{\mathsf{R}} [-1,1]^{m\times w}, \mathsf{ct}_i \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\overline{\mathsf{sk}}_i; \mathbf{R}_i), \mathbf{r}^\star_{\mathbf{y}} \leftarrow_{\mathsf{R}} \mathcal{R}^\star, \mathsf{ct}^\star_{f_{\mathbf{y}}} = \mathsf{Enc}^\star_{\mathsf{pk}}(-\mathsf{MSB}(u_{\mathbf{y}}); \mathbf{r}^\star_{\mathbf{y}}) \\ \mathbf{r}_{f_{\mathbf{y}}} = \mathsf{Eval}_{\mathsf{rand}}\left(\mathsf{pk}, f_{\mathbf{y}}, (\mathbf{R}_i)_{i\in[|\overline{\mathsf{sk}}|]}, (\overline{\mathsf{sk}}_i)_{i\in[|\overline{\mathsf{sk}}|]}\right) : \left(\mathsf{pk}, \mathsf{sk}, \{\mathsf{ct}_i, \mathbf{R}_i\}_{i\in[|\overline{\mathsf{sk}}|]}, \mathbf{r}^\star_{\mathbf{y}} - \mathbf{r}_{f_{\mathbf{y}}} \in \mathbb{Z}_N^m, \mathsf{ct}^\star_{f_{\mathbf{y}}}\right) \end{array} \right\}$$

There is a distinguisher $\mathcal{D}$ such that when fed with a tuple coming from distribution $\mathcal{D}_0$ or $\mathcal{D}_1$, samples $\mathsf{LHE.PubCoin}_{\mathbf{y}} \leftarrow_{\mathsf{R}} \{0,1\}^{s_{|\overline{\mathsf{ct}}|}}$ for all $\mathbf{y} \in \{0,1\}^{n^\varepsilon}$, $(\overline{\mathsf{pk}}, \overline{\mathsf{sk}}) \leftarrow \overline{\mathsf{Gen}}(1^\lambda)$, upon which it can simulate pp and $\widetilde{\Pi}_b$ straightforwardly.

• $\underline{\mathcal{H}^{b.2}_\lambda}$: this ensemble is the same as $\mathcal{H}^{b.1}_\lambda$, except that instead of sampling $\mathsf{LHE.PubCoin}_{\mathbf{y}}$ as random strings, they are sampled as fresh LHE ciphertext of random plaintext, that is, of the form $\mathsf{LHE.PubCoin}_{\mathbf{y}} \leftarrow \overline{\mathsf{Enc}}_{\overline{\mathsf{pk}}}(u_{\mathbf{y}})$ for $u_{\mathbf{y}} \leftarrow_{\mathsf{R}} \mathbb{Z}_N$. By density of the ciphertexts of $\mathcal{LHE}$ Property 10, we have:

$$\mathcal{H}^{b.1}_\lambda \approx_s \mathcal{H}^{b.2}_\lambda.$$

• $\underline{\mathcal{H}^{b.3}_\lambda}$: this ensemble is the same as $\mathcal{H}^{b.2}_\lambda$, except it generates $\overline{\mathsf{ct}}_y$ as a fresh LHE encryption of $m_{\mathbf{y}} = \mathsf{sk}^\top(\mathsf{ct}_{1,\mathbf{y}} + \mathsf{ct}'_{\mathsf{MSB},\mathbf{y}}) + u_{\mathbf{y}} \in \mathbb{Z}_N$, using fresh randomness $r_{\mathbf{y}}$, and $\mathsf{LHE.PubCoin}_{\mathbf{y}}$ is instead computed homomorphically by subtracting the LHE encryption $\mathsf{InProd}(\overline{\mathsf{pk}}, \overline{\mathsf{ct}}, \mathsf{ct}_{1,\mathbf{y}} + \mathsf{ct}'_{\mathsf{MSB},\mathbf{y}})$ of $\widetilde{m}_{\mathbf{y}} = \mathsf{sk}^\top(\mathsf{ct}_{1,\mathbf{y}} + \mathsf{ct}'_{\mathsf{MSB},\mathbf{y}}) \in \mathbb{Z}_N$ from the fresh encryption of $m_{\mathbf{y}}$. That is, $\mathsf{LHE.PubCoin} = \overline{\mathsf{Subtract}}\left(\overline{\mathsf{pk}}, \overline{\mathsf{Enc}}_{\overline{\mathsf{pk}}}(m_{\mathbf{y}}; r_{\mathbf{y}}), \mathsf{InProd}(\overline{\mathsf{pk}}, \overline{\mathsf{ct}}, \mathsf{ct}_{1,\mathbf{y}} + \mathsf{ct}'_{\mathsf{MSB},\mathbf{y}})\right)$.

Note that it is possible to define this hybrid since $\mathsf{ct}'_{\mathsf{MSB},\mathbf{y}}$ remains exactly the same no matter what $\mathsf{LHE.PubCoin}_{\mathbf{y}}$ is. This was not true in $\mathcal{D}^b_\lambda$, and we introduced $\mathcal{H}^{b.1}_\lambda$ to break this dependency.

By the weak circuit privacy property of $\mathcal{LHE}$ (Proposition 9), we have:

$$\mathcal{H}^{b.2}_\lambda \approx_s \mathcal{H}^{b.3}_\lambda.$$

The latter states that the following distributions are identical, for all $u_{\mathbf{y}} \in \mathbb{Z}_N$, for $\widetilde{\mathsf{ct}}_{\mathbf{y}} = \mathsf{InProd}(\overline{\mathsf{pk}}, \overline{\mathsf{ct}}, \mathsf{ct}_{1,\mathbf{y}} + \mathsf{ct}'_{\mathsf{MSB},\mathbf{y}})$:

$$\mathcal{D}_0 = \left\{\mathsf{LHE.PubCoin}_{\mathbf{y}} \leftarrow \mathsf{Enc}_{\mathsf{pk}}(u_{\mathbf{y}}); \overline{\mathsf{ct}}_{\mathbf{y}} \leftarrow \overline{\mathsf{Add}}(\overline{\mathsf{pk}}, \widetilde{\mathsf{ct}}_{\mathbf{y}}, \mathsf{LHE.PubCoin}_{\mathbf{y}}) : (\mathsf{LHE.PubCoin}_{\mathbf{y}}, \widetilde{\mathsf{ct}}_{\mathbf{y}})\right\}.$$
$$\mathcal{D}_1 = \left\{\overline{\mathsf{ct}}_{\mathbf{y}} \leftarrow \overline{\mathsf{Enc}}_{\overline{\mathsf{pk}}}(m_{\mathbf{y}}); \mathsf{LHE.PubCoin}_{\mathbf{y}} \leftarrow \overline{\mathsf{Subtract}}(\overline{\mathsf{pk}}, \overline{\mathsf{ct}}_{\mathbf{y}}, \widetilde{\mathsf{ct}}_{\mathbf{y}}) : (\mathsf{LHE.PubCoin}_{\mathbf{y}}, \overline{\mathsf{ct}}_{\mathbf{y}})\right\}.$$

Distribution $\mathcal{D}_0$ corresponds to $\mathcal{H}_\lambda^{b.2}$, whereas distribution $\mathcal{D}_1$ corresponds to $\mathcal{H}_\lambda^{b.3}$.

Note further that in $\mathcal{H}_\lambda^{b.3}$, we no longer use the LHE secret key $\overline{\mathsf{sk}}$; previously it was used to recover the randomness $r_{\mathbf{y}}$.

• $\underline{\mathcal{H}_\lambda^{b.4}}$: it is the same ensemble as $\mathcal{H}_\lambda^{b.3}$, except that $\overline{\mathsf{ct}}_{\mathbf{y}}$ is generated as a fresh encryption of $2^{2\lambda}B \cdot C_{\Pi_b}(\mathbf{y}) + \mathsf{LSB}(u_{\mathbf{y}}) \in \mathbb{Z}_N$, where $\mathsf{LSB}(u_{\mathbf{y}}) = u_{\mathbf{y}} \mod 2^{2\lambda}B - 2^{2\lambda}B/2 \in \mathbb{Z}_N$ instead of a fresh encryption of $\mathsf{sk}^\top(\mathsf{ct}_{1,\mathbf{y}} + \mathsf{ct}'_{\mathsf{MSB},\mathbf{y}}) + u_{\mathbf{y}} = 2^{2\lambda}B \cdot C_{\Pi_b}(\mathbf{y}) + \mathsf{noise}_{1,\mathbf{y}} + \mathsf{noise}_{\mathsf{MSB},\mathbf{y}} + \mathsf{LSB}(u_{\mathbf{y}}) \in \mathbb{Z}_N$, where $\mathsf{noise}_{1,\mathbf{y}} = \mathbf{e}^\top \mathbf{r}_{C_{\mathbf{y}}}$, $\mathbf{r}_{C_{\mathbf{y}}}$ is the randomness obtained when evaluating the circuit $C_{\mathbf{y}}$ on the FHE ciphertext ct_1, and the noise $\mathbf{e} \leftarrow \chi^m$ is part of the public key pk, which contains a matrix $\mathbf{U} = \begin{pmatrix} \mathbf{A} \\ \mathbf{s}^\top\mathbf{A} = \mathbf{e}^\top \end{pmatrix}$; $\mathsf{noise}_{\mathsf{MSB},\mathbf{y}} = \mathbf{e}^\top \mathbf{r}^\star_{\mathbf{y}}$, with $\mathbf{r}^\star_{\mathbf{y}} \leftarrow_{\mathsf{R}} [-\widetilde{B}2^\lambda, \widetilde{B}2^\lambda]^m$. Note that these noises are small, indeed $|\mathsf{noise}_{1,\mathbf{y}}| = |\mathbf{e}^\top \mathbf{r}_{C_{\mathbf{y}}}| < B$ and $|\mathsf{noise}_{\mathsf{MSB},\mathbf{y}}| = |\mathbf{e}^\top \mathbf{r}^\star_{\mathbf{y}}| < B2^\lambda$ and independent of the value $u_{\mathbf{y}} \leftarrow_{\mathsf{R}} \mathbb{Z}_N$. Thus, we can use the value $\mathsf{LSB}(u_{\mathbf{y}})$, which is uniformly random over $[-2^{2\lambda}B/2, 2^{2\lambda}B/2]$ to smudge the noise $\mathsf{noise}_{1,\mathbf{y}} + \mathsf{noise}_{\mathsf{MSB},\mathbf{y}}$, thereby transitioning from $\mathcal{H}_\lambda^{b.3}$ to $\mathcal{H}_\lambda^{b.4}$. Note that the value $\mathsf{LSB}(u_{\mathbf{y}})$ is only used to compute the fresh LHE ciphertext $\overline{\mathsf{ct}}_{\mathbf{y}}$, thus, the statistical change does not affect the way the rest of $\widetilde{\Pi}_b$ or pp are computed. By Lemma 2.2 (smudging), we have:

$$\mathcal{H}_\lambda^{b.3} \approx_s \mathcal{H}_\lambda^{b.4}.$$

• $\underline{\mathcal{H}_\lambda^{0.4} \approx_c \mathcal{H}_\lambda^{1.4}}$: To complete the proof, we show that $\mathcal{H}_\lambda^{0.4}$ is computationally indistinguishable from $\mathcal{H}_\lambda^{1.4}$. These two ensembles are the same except the former obfuscates the program Π_0, whereas the latter obfuscates Π_1. Note that other than the encrypted key cycle, we never use the FHE secret key, and due to Hybrid $\mathcal{H}_\lambda^{b.3}$, we no longer use the LHE secret key. The coins $\mathsf{FHE.PubCoin}$ exactly correspond to an SRL leakage on the FHE ciphertext ct_2 (and note that in the experiment we do know the output $\alpha_{\mathbf{y}}$ of the function $f_{\mathbf{y}}$ that is applied to the plaintexts encrypted in $\mathsf{ct}_1, \mathsf{ct}_2$—namely, it is $-\mathsf{MSB}(u_{\mathbf{y}})$ where $u_{\mathbf{y}}$ is a random element of $\mathbb{Z}_N$ selected in the experiment, see Hybrid $\mathcal{H}_\lambda^{b.2}$). Thus, indistinguishability of $\mathcal{H}_\lambda^{0.4}$ and $\mathcal{H}_\lambda^{1.4}$ follows from 2-circular SRL-security of $\mathcal{FHE}$ and $\mathcal{LHE}$.

Namely, we provide a reduction from the 2-circular SRL security with respect to $\mathcal{FHE}$ and $\mathcal{LHE}$ (as per Definition 3.3) to distinguishing $\mathcal{H}_\lambda^{0.4}$ and $\mathcal{H}_\lambda^{1.4}$. Given the public keys pk, $\overline{\mathsf{pk}}$ and the ciphertext ct, $\overline{\mathsf{ct}}$ provided by the 2-circular security experiment, where ct encrypts $(\overline{\mathsf{sk}}\|\Pi_0)$ or $(\overline{\mathsf{sk}}\|\Pi_1)$, and $\overline{\mathsf{ct}}$ encrypts sk, the reduction samples $u_{\mathbf{y}} \leftarrow_{\mathsf{R}} \mathbb{Z}_N$ for all $\mathbf{y} \in \{0,1\}^{n^\varepsilon}$, and generates the following:

- Generation of pp:
  - To generate $\mathsf{LHE.PubCoin}_{\mathbf{y}}$:
    - It samples $r_{\mathbf{y}} \leftarrow_{\mathsf{R}} \overline{\mathcal{R}}$, where $\overline{\mathcal{R}}$ denotes the randomness space of $\overline{\mathsf{Enc}}$, and computes $\overline{\mathsf{ct}}_{\mathbf{y}} = \overline{\mathsf{Enc}}_{\overline{\mathsf{pk}}}(m_{\mathbf{y}}; r_{\mathbf{y}})$, where $m_{\mathbf{y}} = 2^{2\lambda}B \cdot C_{\Pi_b}(\mathbf{y}) + \mathsf{LSB}(u_{\mathbf{y}}) \in \mathbb{Z}_N$. Note that this does not require to know the bit b, since $C_{\Pi_0}(\mathbf{y}) = C_{\Pi_1}(\mathbf{y})$ for all $\mathbf{y} \in \{0,1\}^{n^\varepsilon}$, because the program Π_0 and Π_1 are functionally equivalent.
    - Then, it computes $\mathsf{ct}_{1,\mathbf{y}} = \mathsf{Eval}'(\mathsf{pk}, C_{\mathbf{y}}, 2^{2\lambda}B, \mathsf{ct})$, where $C_{\mathbf{y}}$ takes as input $(\overline{\mathsf{sk}}\|\Pi_b)$, ignores the first part of the input, namely $\overline{\mathsf{sk}}$, and outputs the $\mathbf{y}$'th chunk of the truth table of Π_b.

  * Then, it queries its $\mathcal{O}_{\mathsf{SRL}}$ oracle, to obtain a fresh, extra noisy encryption $\mathsf{Enc}^{\star}_{\mathsf{pk}}(0; \mathbf{r}^{\star}_{\mathbf{y}})$. It leaves the oracle $\mathcal{O}_{\mathsf{SRL}}$ pending.
  * Then, it computes $\mathsf{ct}'_{\mathsf{MSB},\mathbf{y}}$ by simply adding the vector $(\mathbf{0}, -\mathsf{MSB}(u_{\mathbf{y}})) \in \mathbb{Z}_N^{n+1}$ to $\mathsf{Enc}^{\star}_{\mathsf{pk}}(0; \mathbf{r}^{\star}_{\mathbf{y}})$, which yields $\mathsf{ct}'_{\mathsf{MSB},\mathbf{y}} = \mathsf{Enc}^{\star}_{\mathsf{pk}}(-\mathsf{MSB}(u_{\mathbf{y}}); \mathbf{r}^{\star}_{\mathbf{y}})$.
  * Then, it computes $\widetilde{\mathsf{ct}}_{\mathbf{y}} = \overline{\mathsf{InProd}}(\overline{\mathsf{pk}}, \overline{\mathsf{ct}}, \mathsf{ct}_{1,\mathbf{y}} + \mathsf{ct}'_{\mathsf{MSB},\mathbf{y}})$.
  * Finally, it computes $\mathsf{LHE.PubCoin}_{\mathbf{y}} = \overline{\mathsf{Subtract}}(\overline{\mathsf{pk}}, \overline{\mathsf{ct}}_{\mathbf{y}}, \widetilde{\mathsf{ct}}_{\mathbf{y}})$.

- To generate $\mathsf{FHE.PubCoin}_{\mathbf{y}}$: it answers the pending oracle $\mathcal{O}_{\mathsf{SRL}}$ with the function $f_{\mathbf{y}}$ and the value $\alpha = -\mathsf{MSB}(u_{\mathbf{y}})$, where $f_{\mathbf{y}}$ takes as input $(\overline{\mathsf{sk}} \| \Pi_b)$, ignores the second part of the input, namely Π_b, decrypts the LHE ciphertext $\mathsf{LHE.PubCoin}_{\mathbf{y}}$ defined above using $\overline{\mathsf{sk}}$ to obtain a plaintext $v_{\mathbf{y}} \in \mathbb{Z}_N$, and outputs $-\mathsf{MSB}(v_{\mathbf{y}})$. By definition of $\mathsf{LHE.PubCoin}_{\mathbf{y}}$, we know that $-\mathsf{MSB}(v_{\mathbf{y}}) = -\mathsf{MSB}(u_{\mathbf{y}})$. The oracle $\mathcal{O}_{\mathsf{SRL}}$ returns the leakage $\mathbf{r}^{\star}_{\mathbf{y}} - \mathbf{r}_{f_{\mathbf{y}}} \in \mathcal{R}^{\star}$. The reduction sets $\mathsf{FHE.PubCoin}_{\mathbf{y}} = \mathbf{r}^{\star}_{\mathbf{y}} - \mathbf{r}_{f_{\mathbf{y}}}$.

It returns $\mathsf{pp} = (\mathsf{LHE.PubCoin}, \mathsf{FHE.PubCoin})$.

- Generation of $\widetilde{\Pi}_b$: it returns $(\mathsf{pk}, \overline{\mathsf{pk}}, \overline{\mathsf{ct}}, \mathsf{ct}, (r_{\mathbf{y}})_{\mathbf{y} \in \{0,1\}^{n^{\varepsilon}}})$ computed as described above.

When ct encrypts $(\overline{\mathsf{sk}} \| \Pi_0)$, the reduction simulates $\mathcal{H}^{0.4}_{\lambda}$, whereas it simulates $\mathcal{H}^{1.4}_{\lambda}$ when ct encrypts $(\overline{\mathsf{sk}} \| \Pi_1)$. □

5 Concluding the Main Theorem

Let $\mathcal{FHE}$ be GSW's FHE for depth-$\delta(\cdot)$ circuits and $n^{\varepsilon}(\cdot)$-batch correctness, presented in Section 3.2.2, and $\mathcal{LHE}$ be DJ's LHE with plaintext size parameterized by the polynomial $p(\cdot)$, presented in Section 4.1, where δ, n^{ε} and p are described in the XiO construction, Section 4.3.

By combining Theorem 4.2 (i.e. security of $\mathcal{LHE}$ under DCR) with Theorem 3.5 (i.e. SRL-security of $\mathcal{FHE}$ under LWE), and noting that both those results directly upgrade to subexponential security if we assume subexponential security of the underlying assumptions, we get:

Lemma 5.1. *Assume the (subexponential) DCR and (subexponential) LWE assumptions hold. Then the (subexponential)* $\mathsf{2CIRC}^{\mathcal{O}_{\mathsf{SRL}}}$ *assumption w.r.t.* $\mathcal{FHE}$ *and* $\mathcal{LHE}$ *implies that (subexponential) 2-circular SRL security holds w.r.t.* $\mathcal{FHE}$ *and* $\mathcal{LHE}$*.*

By combing Lemma 5.1 with Theorem 4.4, and noting that Theorem 4.4 yields a subexponentially-secure $Xi\mathcal{O}$ assuming the subexponential security of the underlying building block (i.e., subexponential 2-circular SRL security of $\mathcal{FHE}$ and $\mathcal{LHE}$), we get:

Theorem 5.1. *Assume the (subexponential) DCR and (subexponential) LWE assumptions hold. Then, the (subexponential)* $\mathsf{2CIRC}^{\mathcal{O}_{\mathsf{SRL}}}$ *assumption w.r.t.* $\mathcal{FHE}$ *and* $\mathcal{LHE}$ *implies the existence of* $Xi\mathcal{O}$ *for* $\mathsf{P}^{\log}/\mathsf{poly}$*.*

Finally, combining Theorem 5.1 with Theorem 2.5 (i.e., $i\mathcal{O}$ from $Xi\mathcal{O}$ and LWE) yields out main theorem:

Theorem 5.2. *Assume the subexponential DCR and subexponential LWE assumptions hold. Then, the subexponential* $\mathsf{2CIRC}^{\mathcal{O}_{\mathsf{SRL}}}$ *assumption w.r.t.* $\mathcal{FHE}$ *and* $\mathcal{LHE}$ *implies the existence of* $i\mathcal{O}$ *for* P/poly*.*

References

[Agr19] S. Agrawal. Indistinguishability obfuscation without multilinear maps: New methods for bootstrapping and instantiation. In *EUROCRYPT 2019, Part I*, *LNCS* 11476, pages 191–225. Springer, Heidelberg, May 2019.

[AJ15] P. Ananth and A. Jain. Indistinguishability obfuscation from compact functional encryption. In *CRYPTO 2015, Part I*, *LNCS* 9215, pages 308–326. Springer, Heidelberg, August 2015.

[AJKS18] P. Ananth, A. Jain, D. Khurana, and A. Sahai. Indistinguishability obfuscation without multilinear maps: iO from LWE, bilinear maps, and weak pseudorandomness. Cryptology ePrint Archive, Report 2018/615, 2018. https://eprint.iacr.org/2018/615.

[AJL$^+$19] P. Ananth, A. Jain, H. Lin, C. Matt, and A. Sahai. Indistinguishability obfuscation without multilinear maps: New paradigms via low degree weak pseudorandomness and security amplification. Cryptology ePrint Archive, Report 2019/643, 2019. https://eprint.iacr.org/2019/643.

[AP20] S. Agrawal and A. Pellet-Mary. Indistinguishability obfuscation without maps: Attacks and fixes for noisy linear FE. In *EUROCRYPT 2020, Part I*, LNCS, pages 110–140. Springer, Heidelberg, May 2020.

[BBKK18] B. Barak, Z. Brakerski, I. Komargodski, and P. K. Kothari. Limits on low-degree pseudorandom generators (or: Sum-of-squares meets program obfuscation). In *EUROCRYPT 2018, Part II*, *LNCS* 10821, pages 649–679. Springer, Heidelberg, April / May 2018.

[BCP14] E. Boyle, K.-M. Chung, and R. Pass. On extractability obfuscation. In *TCC*, pages 52–73, 2014.

[BDGM19] Z. Brakerski, N. Döttling, S. Garg, and G. Malavolta. Leveraging linear decryption: Rate-1 fully-homomorphic encryption and time-lock puzzles. In *TCC 2019, Part II*, LNCS, pages 407–437. Springer, Heidelberg, March 2019.

[BDGM20] Z. Brakerski, N. Döttling, S. Garg, and G. Malavolta. Candidate iO from homomorphic encryption schemes. In *EUROCRYPT 2020, Part I*, LNCS, pages 79–109. Springer, Heidelberg, May 2020.

[BGI$^+$01] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In *CRYPTO 2001*, *LNCS* 2139, pages 1–18. Springer, Heidelberg, August 2001.

[BGL$^+$15] N. Bitansky, S. Garg, H. Lin, R. Pass, and S. Telang. Succinct randomized encodings and their applications. *IACR Cryptology ePrint Archive*, 2015:356, 2015.

[BHJ$^+$18] B. Barak, S. B. Hopkins, A. Jain, P. Kothari, and A. Sahai. Sum-of-squares meets program obfuscation, revisited. Cryptology ePrint Archive, Report 2018/1237, 2018. https://eprint.iacr.org/2018/1237.

[BHW15] A. Bishop, S. Hohenberger, and B. Waters. New circular security counterexamples from decision linear and learning with errors. In *ASIACRYPT 2015, Part II*, *LNCS* 9453, pages 776–800. Springer, Heidelberg, November / December 2015.

[BP15] N. Bitansky and O. Paneth. ZAPs and non-interactive witness indistinguishability from indistinguishability obfuscation. In *TCC 2015, Part II*, *LNCS* 9015, pages 401–427. Springer, Heidelberg, March 2015.

[BPR15] N. Bitansky, O. Paneth, and A. Rosen. On the cryptographic hardness of finding a Nash equilibrium. In *56th FOCS*, pages 1480–1498. IEEE Computer Society Press, October 2015.

[BPW16] N. Bitansky, O. Paneth, and D. Wichs. Perfect structure on the edge of chaos - trapdoor permutations from indistinguishability obfuscation. In *TCC 2016-A, Part I*, *LNCS* 9562, pages 474–502. Springer, Heidelberg, January 2016.

[BR93] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS 93*, pages 62–73. ACM Press, November 1993.

[BRS02] J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. Cryptology ePrint Archive, Report 2002/100, 2002. http://eprint.iacr.org/2002/100.

[BV15] N. Bitansky and V. Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In *56th FOCS*, pages 171–190. IEEE Computer Society Press, October 2015.

[BZ14] D. Boneh and M. Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 480–499, 2014.

[CGH98] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218. ACM Press, May 1998.

[CHJV14] R. Canetti, J. Holmgren, A. Jain, and V. Vaikuntanathan. Indistinguishability obfuscation of iterated circuits and RAM programs. Cryptology ePrint Archive, Report 2014/769, 2014. http://eprint.iacr.org/2014/769.

[CHL$^+$15] J. H. Cheon, K. Han, C. Lee, H. Ryu, and D. Stehlé. Cryptanalysis of the multilinear map over the integers. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, pages 3–12, 2015.

[CKP15] R. Canetti, Y. T. Kalai, and O. Paneth. On obfuscation with random oracles. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, pages 456–467, 2015.

[CL01] J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT 2001*, *LNCS* 2045, pages 93–118. Springer, Heidelberg, May 2001.

[CLP15] K.-M. Chung, H. Lin, and R. Pass. Constant-round concurrent zero-knowledge from indistinguishability obfuscation. In *CRYPTO 2015, Part I*, *LNCS* 9215, pages 287–307. Springer, Heidelberg, August 2015.

[CLT13] J.-S. Coron, T. Lepoint, and M. Tibouchi. Practical multilinear maps over the integers. In *CRYPTO 2013, Part I*, *LNCS* 8042, pages 476–493. Springer, Heidelberg, August 2013.

[CLT15] J.-S. Coron, T. Lepoint, and M. Tibouchi. New multilinear maps over the integers. In *CRYPTO 2015, Part I*, *LNCS* 9215, pages 267–286. Springer, Heidelberg, August 2015.

[DJ01] I. Damgård and M. Jurik. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In *PKC 2001*, *LNCS* 1992, pages 119–136. Springer, Heidelberg, February 2001.

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

[GGH13a] S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. In *EUROCRYPT 2013*, *LNCS* 7881, pages 1–17. Springer, Heidelberg, May 2013.

[GGH$^+$13b] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.

[GGH15] C. Gentry, S. Gorbunov, and S. Halevi. Graph-induced multilinear maps from lattices. In *TCC 2015, Part II*, *LNCS* 9015, pages 498–527. Springer, Heidelberg, March 2015.

[GGHR14] S. Garg, C. Gentry, S. Halevi, and M. Raykova. Two-round secure MPC from indistinguishability obfuscation. In *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, pages 74–94, 2014.

[GJLS20] R. Gay, A. Jain, H. Lin, and A. Sahai. Indistinguishability obfuscation from simple-to-state hard problems: New assumptions, new techniques, and simplification. Technical report, Cryptology ePrint Archive, Report 2020/764, 2020. https://eprint.iacr.org/2020/764, 2020.

[GK05] S. Goldwasser and Y. T. Kalai. On the impossibility of obfuscation with auxiliary input. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 553–562, 2005.

[GKW17] R. Goyal, V. Koppula, and B. Waters. Separating semantic and circular security for symmetric-key bit encryption from the learning with errors assumption. In *EUROCRYPT 2017, Part II*, *LNCS* 10211, pages 528–557. Springer, Heidelberg, April / May 2017.

[GLSW14] C. Gentry, A. Lewko, A. Sahai, and B. Waters. Indistinguishability obfuscation from the multilinear subgroup elimination assumption. Cryptology ePrint Archive, Report 2014/309, 2014.

[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[GSW13] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO 2013, Part I*, *LNCS* 8042, pages 75–92. Springer, Heidelberg, August 2013.

[ILL89] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions (extended abstracts). In *21st ACM STOC*, pages 12–24. ACM Press, May 1989.

[JLMS19] A. Jain, H. Lin, C. Matt, and A. Sahai. How to leverage hardness of constant-degree expanding polynomials overa $\mathbb{R}$ to build $i\mathcal{O}$. In *EUROCRYPT 2019, Part I*, *LNCS* 11476, pages 251–281. Springer, Heidelberg, May 2019.

[JS18] A. Jain and A. Sahai. How to leverage hardness of constant-degree expanding polynomials over $\mathbb{R}$ to build iO. Cryptology ePrint Archive, Report 2018/973, 2018. https://eprint.iacr.org/2018/973.

[KLW15] V. Koppula, A. B. Lewko, and B. Waters. Indistinguishability obfuscation for turing machines with unbounded memory. In *47th ACM STOC*, pages 419–428. ACM Press, June 2015.

[KMN+14] I. Komargodski, T. Moran, M. Naor, R. Pass, A. Rosen, and E. Yogev. One-way functions and (im)perfect obfuscation. In *55th FOCS*, pages 374–383. IEEE Computer Society Press, October 2014.

[KNY14] I. Komargodski, M. Naor, and E. Yogev. Secret-sharing for NP. In *ASIACRYPT 2014, Part II*, *LNCS* 8874, pages 254–273. Springer, Heidelberg, December 2014.

[KRW15] V. Koppula, K. Ramchen, and B. Waters. Separations in circular security for arbitrary length key cycles. In *TCC 2015, Part II*, *LNCS* 9015, pages 378–400. Springer, Heidelberg, March 2015.

[KW16] V. Koppula and B. Waters. Circular security separations for arbitrary length cycles from LWE. In *CRYPTO 2016, Part II*, *LNCS* 9815, pages 681–700. Springer, Heidelberg, August 2016.

[Lin16] H. Lin. Indistinguishability obfuscation from constant-degree graded encoding schemes. In *EUROCRYPT 2016, Part I*, *LNCS* 9665, pages 28–57. Springer, Heidelberg, May 2016.

[Lin17] H. Lin. Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGs. In *CRYPTO 2017, Part I*, *LNCS* 10401, pages 599–629. Springer, Heidelberg, August 2017.

[LPST16] H. Lin, R. Pass, K. Seth, and S. Telang. Indistinguishability obfuscation with non-trivial efficiency. In *PKC 2016, Part II*, *LNCS* 9615, pages 447–462. Springer, Heidelberg, March 2016.

[LT17] H. Lin and S. Tessaro. Indistinguishability obfuscation from trilinear maps and block-wise local PRGs. In *CRYPTO 2017, Part I*, *LNCS* 10401, pages 630–660. Springer, Heidelberg, August 2017.

[LV16] H. Lin and V. Vaikuntanathan. Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings. In *57th FOCS*, pages 11–20. IEEE Computer Society Press, October 2016.

[MF15] B. Minaud and P.-A. Fouque. Cryptanalysis of the new multilinear map over the integers. Cryptology ePrint Archive, Report 2015/941, 2015. http://eprint.iacr.org/.

[Mic19] D. Micciancio. From linear functions to fully homomorphic encryption. https://bacrypto.github.io/presentations/2018.11.30-micciancio-fhe.pdf. Technical report, 2019.

[MMN15] M. Mahmoody, A. Mohammed, and S. Nematihaji. More on impossibility of virtual black-box obfuscation in idealized models. *IACR Cryptology ePrint Archive*, 2015:632, 2015.

[MO14] A. Marcedone and C. Orlandi. Obfuscation $\Rightarrow$ (IND-CPA security $\not\Rightarrow$ circular security). In *SCN 14*, *LNCS* 8642, pages 77–90. Springer, Heidelberg, September 2014.

[MP12] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT 2012*, *LNCS* 7237, pages 700–718. Springer, Heidelberg, April 2012.

[MRH04] U. M. Maurer, R. Renner, and C. Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In *TCC 2004*, *LNCS* 2951, pages 21–39. Springer, Heidelberg, February 2004.

[Pai99] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT'99*, *LNCS* 1592, pages 223–238. Springer, Heidelberg, May 1999.

[PRS17] C. Peikert, O. Regev, and N. Stephens-Davidowitz. Pseudorandomness of ring-LWE for any ring and modulus. In *49th ACM STOC*, pages 461–473. ACM Press, June 2017.

[Ps16] R. Pass and a. shelat. Impossibility of VBB obfuscation with ideal constant-degree graded encodings. In *TCC 2016-A, Part I*, *LNCS* 9562, pages 3–17. Springer, Heidelberg, January 2016.

[PST14] R. Pass, K. Seth, and S. Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. In *CRYPTO 2014, Part I*, *LNCS* 8616, pages 500–517. Springer, Heidelberg, August 2014.

[Reg05] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

[SW14] A. Sahai and B. Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014.