Indistinguishability Obfuscation from Circular Security

Romain Gay
IBM Research Zurich
Switzerland
romain.rgay@gmail.com

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

ABSTRACT

We show the existence of indistinguishability obfuscators (iO) for general circuits assuming subexponential security of: (a) the Learning with Errors (LWE) assumption (with subexponential modulus-to-noise ratio); (b) a *circular security conjecture* regarding the Gentry-Sahai-Waters' (GSW) encryption scheme and a Packed version of Regev's encryption scheme. The circular security conjecture states that a notion of leakage-resilient security, that we prove is satisfied by GSW assuming LWE, is retained in the presence of an encrypted key-cycle involving GSW and Packed Regev.

CCS CONCEPTS

• **Security and privacy → Cryptography**.

KEYWORDS

obfuscation, LWE

ACM Reference Format:
Romain Gay and Rafael Pass. 2021. Indistinguishability Obfuscation from Circular Security. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC '21), June 21–25, 2021, Virtual, Italy*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3406325.3451070

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 (iO)* [9, 36] 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. [13–16, 19, 20, 26, 30, 35, 50–52, 72]). 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. [9, 27, 44, 57, 60, 67]). Since the breakthrough of Garg, Gentry, Halevi, Raykova, Sahai and Waters [36] that presented the first $i\mathcal{O}$ candidate, 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 [36] provided a *candidate* construction based on high-degree multilinear maps (MLMs) [31, 32, 34, 39]; there was no proof of security based on an intractability assumption. [66] 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". [41] 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 [29, 65].

$i\mathcal{O}$ from FE or $X i\mathcal{O}$. Subsequently, several works have been constructing $i\mathcal{O}$ from seemingly weaker primitives, such as Functional Encryption (FE) [6, 17] or $X i\mathcal{O}$ [57], while only using standard assumptions, such as Learning with Errors (LWE)[1]. For both constructions, we actually need to rely on *subexponentially-secure* constructions of either FE or $X i\mathcal{O}$, as well as subexponential security of LWE. Let us recall the notion of $X i\mathcal{O}$ as it will be useful to us: roughly speaking, an $X i\mathcal{O}$ 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 to output a program that "slightly" compresses the truth table (technically, it is sublinear in its size).

A breakthrough result by Lin [55] showed how to obtain $i\mathcal{O}$ from *constant-degree* MLMs (plus standard assumptions), overcoming the black-box barriers in [60, 67]. Her construction relies on the connection between FE and $i\mathcal{O}$. Following this result, a sequence of works (see e.g. [7, 8, 37, 47, 49, 56, 58, 59]) reduced the assumptions and the degree of the MLM — all the way down to 2-linear maps a.k.a. pairings— relying on certain types of low-degree pseudorandom generators (PRGs) to build FE. This culminated in the work of [37], whose security rely on the LWE assumption with binary errors in the presence of some PRG leakage, which despite being quite elegant, is new to their work, and as such, has not been significantly crypt-analyzed. Another line of work [2, 3] replace the use of 2-linear maps used by the aforementioned works by a noisy linear FE for inner products. While being plausibly post-quantum, these constructions are heuristic and do not provide a security reduction to a simple assumption.

A recent work by Brakerski et al [22] presents a new type of candidate construction of $X i\mathcal{O}$ by combining a fully-homomorphic encryption (FHE) and a linear-homomorphic encryption (LHE) with

STOC '21, June 21–25, 2021, Virtual, Italy

ACM ISBN 978-1-4503-8053-9/21/06…$15.00
https://doi.org/10.1145/3406325.3451070

[1] Note that we are omitting some works that build $i\mathcal{O}$ without going through FE or $X i\mathcal{O}$, such as [40] that gives a direct heuristic construction of $i\mathcal{O}$ from tensor products, or [10] that describes a candidate from Affine Determinant Programs. None of these provide a security proof.

certain nice properties (which can be instantiated by the Damgård-Jurik (DJ) [33] encryption scheme whose security relies 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 XiO (which by earlier work implies iO under standard assumptions). We highlight that [22] 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 *"correlation conjecture"* that the FHE randomness (after FHE evaluations) does not correlate "too much" with the 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 iO (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 iO, known constructions are either based on assumptions that are not well understood (high-degree MLMs, various low-degree PRGs assumptions and LWE with leakage type of assumptions), or the construction candidates simply do not have proofs of security.

1.1 Our Results

In this work, we provide a new iO construction assuming subexponential security of (a) the LWE assumption (with subexponential modulus-to-noise ratio), and (b) an (in our eyes) natural *circular security assumption* w.r.t the Gentry-Sahai-Waters' (GSW) [43] FHE scheme and the DJ [33] LHE scheme. Alternatively, assumption (b) can be replaced by a circular security conjecture regarding the GSW encryption scheme and a "packed" variant of Regev's encryption scheme [69, 70].

On a high-level, our approach follows that in [22], 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 iO, and places iO on a qualitatively similar footing as *unlevelled* FHE (i.e. an FHE that support an a-priori unbounded polynomial number of operations), for which known constructions also rely on a circular security conjecture [38]. We emphasize that the type of circular security conjecture that we rely on is stronger and more complex than the "plain" circular security conjecture used for unlevelled FHE. Yet on a philosophical level, we do not see any concrete evidence for why the plain circular security is more believable.[2]

Circular security. Circular security of encryption schemes [18, 24] considers a scenario where the attacker gets to see not only encryptions of messages, but also *encrypted key cycles*. The simplest form of circular security, referred to as *1-circular security*, requires that security holds even if the attacker gets to see not only the public key pk and an encryption $\mathsf{Enc}_{\mathsf{pk}}(\mathbf{m})$ of a message $\mathbf{m}$ (to be secured), but also an encryption $\mathsf{Enc}_{\mathsf{pk}}(\mathsf{sk})$ of the secret key sk.

[2] See Section 1.4 for an extended comparison.

A slightly more complex type of circular security, referred to as *2-circular security*, considers an encrypted key cycle of size 2 where 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 $\mathsf{Enc}^1_{\mathsf{pk}_1}(m_1)$). Encrypted key cycles commonly arise in applications of encryption scheme such as storage systems (e.g. BitLocker disk encryption utility), anonymous credentials [24] and most recently to construct (unlevelled) FHE [38].

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 (*2CIRC*) w.r.t* $\mathsf{Enc}^1, \mathsf{Enc}^2$. (We may also consider a subexponential version of this conjecture which is identically defined except that "security" is replaced by "subexponential security".) 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 circular security holds w.r.t. *all* secure encryption schemes—$\mathsf{Enc}^1, \mathsf{Enc}^2$— after all, the attacker never actually gets to see the secret key, but rather an encryption of it, which intuitively should hide it by semantic security of the encryption schemes. Yet, in recent years, counter examples to 2-circular security have been found. While for 1-circular security of *string encryption* schemes, it is easy to come up with a counter example—simply take any encryption scheme and modify it so that an encryption of $\mathbf{m}$ outputs $\mathbf{m}$ iff $\mathbf{m}$ is a valid secret key, and otherwise proceeds just as before—coming up with counterexamples for 1-circular security of *bit encryption*, or 2-circular security for either string or bit encryption, is a lot harder (see e.g. [1, 12, 28, 45, 46, 53, 54, 61, 71, 75]). In fact, all known counter examples are highly artificial, and require carefully embedding some trapdoor mechanism in the encryption scheme that enables decrypting the ciphertext once you see an encryption of the secret key. As far as we are aware, no "natural" counterexamples are known. Indeed, a common heuristic consists of simply assuming that 2-circular security holds for all "natural" encryption schemes that are secure; that is, 2CIRC holds for all "natural" encryption schemes —we refer to this as the 2CIRC heuristic. It is similar to the Random Oracle Heuristic [11]: while "contrived" counterexamples are known (see e.g., [25, 62]), it is still commonly used for the design of practical protocols.

Leakage-resilient Circular Security. In this work, we rely on the assumption that stronger forms of security are preserved in the presence of a key cycle. More precisely, we consider a notion of $\mathcal{O}$-leakage resilient security where $\mathcal{O}$ is some particular *randomness leakage oracle*; this notion enhances the standard semantic security notion by providing the attacker with access to an oracle $\mathcal{O}(\mathsf{pk}, \mathbf{m}, \mathbf{r})$ that is parameterized by the public key pk, the message $\mathbf{m}$ being encrypted and the randomness $\mathbf{r}$ under which it is encrypted, while restricting the attacker to making only "valid" leakage queries (that do not trivially leak information about the message—this is formalized by letting the oracle output $\perp$ whenever a query is invalid, and saying that the attacker fails whenever

this happens). We next define the notion of *2-circular $\mathcal{O}$-leakage resilient security* analogously to 2-circular security, and also define a $\mathsf{2CIRC}^{\mathcal{O}}$ conjecture (resp. a subexponential $\mathsf{2CIRC}^{\mathcal{O}}$ conjecture) in the same way as the 2CIRC conjecture except that "security" is replaced by "$\mathcal{O}$-leakage resilient security"; that is, 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.

Note that we cannot hope that $\mathsf{2CIRC}^{\mathcal{O}}$ security holds for *all* oracles $\mathcal{O}$, even with respect to "natural" encryption schemes: simply consider an oracle $\mathcal{O}(\mathsf{pk}, \mathbf{m}, \mathbf{r})$ that outputs the message $\mathbf{m}$ iff $\mathbf{m}$ is a valid secret key (just as in the counterexample to "plain" 1-circular security for string encryption). Thus, for $\mathsf{2CIRC}^{\mathcal{O}}$ to be meaningful, we need to restrict not only to "natural" encryption schemes, but also to "natural" oracles $\mathcal{O}$.

Our first theorem shows that for a natural leakage oracle $\mathcal{O}_{\mathsf{SRL}}$—which will be referred to as the "shielded randomness leakage (SRL) oracle"—$\mathsf{2CIRC}^{\mathcal{O}_{\mathsf{SRL}}}$ 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 the subexponential security of the LWE assumption (with subexponential modulus-to-noise ratio) and the DCR assumption, and the subexponential* $\mathsf{2CIRC}^{\mathcal{O}_{\mathsf{SRL}}}$ *conjecture w.r.t. GSW and and DJ. Then, $i\mathcal{O}$ exists for the class of polynomial-size circuits.*

Alternatively, we can replace the DJ encryption scheme with a "packed" variant of Regev's encryption scheme [70], which we refer to as Packed Regev. (We note that our Packed Regev is very similar to, but actually different from, the Packed Regev in [69].) This construction only relies on LWE and the 2-circular security conjecture.

Theorem 1.2 (Informally stated). *Assume the subexponential security of the LWE (with subexponential modulus-to-noise ratio) assumption, and assume that the subexponential* $\mathsf{2CIRC}^{\mathcal{O}_{\mathsf{SRL}}}$ *conjecture holds w.r.t. GSW and Packed Regev. Then, $i\mathcal{O}$ exists for the class of polynomial-size circuits.*

In the sequel, we refer to $\mathcal{O}_{\mathsf{SRL}}$-leakage resilient security (resp. 2-circular $\mathcal{O}_{\mathsf{SRL}}$-leakage resilient security) as SRL-security (resp 2-circular SRL security). We proceed to explain the notion of SRL security and how the above theorems are proven.

1.2 Shielded Randomness Leakage (SRL) Security

As mentioned above, we consider a notion of *shielded randomness leakage (SRL)* security for FHE. Roughly speaking, given two messages $\mathbf{m}^0, \mathbf{m}^1$, the attacker gets to see an FHE encryption $\mathbf{c} = \mathsf{FHE}(\mathbf{m}^b; \mathbf{r})$ of $\mathbf{m}^b$ for a randomly selected $b \in \{0,1\}$, and next gets access to a "leakage oracle" $\mathcal{O}_{\mathsf{SRL}}(\mathbf{m}^b, \mathbf{r})$ which upon every invocation sends the attacker an "extra noisy" encryption $\mathbf{c}^\star = \mathsf{FHE}(0; \mathbf{r}^\star)$ of 0—we will refer to the random string $\mathbf{r}^\star$ as the "shield". Next, the attacker can select some functions f and values α such that $f(\mathbf{m}^b) = \alpha$—that is, we restrict the attacker to picking functions for which it *knows the output* when applying the function to the message $\mathbf{m}^b$; if $f(\mathbf{m}^b) \neq \alpha$, the attacker directly fails in the game. (The reason why we add this restriction on the attacker will soon become clear). Finally, the oracle homomorphically evaluates f on the ciphertext $\mathbf{c}$, letting $\mathbf{c}_f = \mathsf{FHE}(f(\mathbf{m}); \mathbf{r}_f)$ denote the evaluated ciphertext, and returns $\mathbf{r}^\star - \mathbf{r}_f$. That is, the attacker gets back the randomness $\mathbf{r}_f$ of the evaluated ciphertext masked by the "shield" $\mathbf{r}^\star$, and as usual, the attacker's goal is to guess the bit b. 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 $\mathbf{c}^\star$ and $\mathbf{c}_f$, the attacker can compute $\mathbf{c}^\star - \mathbf{c}_f = \mathsf{FHE}(0 - f(\mathbf{m}^b); \mathbf{r}^\star - \mathbf{r}_f)$ and thus knowing $\mathbf{r}^\star - \mathbf{r}_f$ reveals $f(\mathbf{m}^b)$. So, by restricting to attackers that already know $\alpha = f(\mathbf{m}^b)$, intuitively, $\mathbf{r}^\star - \mathbf{r}_f$ does not reveal anything else. Indeed, we formally prove that under the LWE assumption, the GSW encryption scheme is SRL-secure (i.e. $\mathcal{O}_{\mathsf{SRL}}$-leakage resilient secure).

Theorem 1.3 (Informally stated). *Assume the LWE assumption holds (with subexponential modulus-to-noise ratio). Then, the GSW scheme is SRL-secure.*

On a very high-level, the idea behind the proof is that the encryption $\mathbf{c}^\star$ is a projection, $h_{\mathbf{A}}(\mathbf{r}^\star) = \mathbf{A}\mathbf{r}^\star \in \mathbb{Z}_N^n$, where the randomness $\mathbf{r}^\star$ used to produce $\mathbf{c}^\star$ is a vector in $\mathbb{Z}_N^\ell$ and $\mathbf{A}$ is a matrix in $\mathbb{Z}_N^{n \times \ell}$ where $\ell \gg n$, 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 components of $\mathbf{r}_f$; furthermore, the components that are not hidden by $\mathbf{r}^\star$ are actually already revealed by $f(\mathbf{m}^b)$, which the attacker knows (as we require it to output $\alpha = f(\mathbf{m}^b)$). The formal proof of this proceeds by considering a (simplified) variant of the Micciancio-Peikert lattice trapdoor method [64] for generating the matrix $\mathbf{A}$ (which is part of the public key for GSW) together with a trapdoor that enables sampling short preimages of $h_{\mathbf{A}}$ (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 larger than $\mathbf{w}$ to ensure that they "smudge" $\mathbf{w}$—we here rely on the fact that modulus-to-noise ratio is subexponential (which we need anyway for the security of our construction) to enable the smudging[3].

2-Circular SRL Security. As mentioned, we define 2-circular SRL security as 2-circular $\mathcal{O}_{\mathsf{SRL}}$-leakage resilient security; we emphasize that this security game is identically defined to the "plain" SRL security game (described above), with the only exception being that the challenge message encrypted (using $\mathsf{Enc}^1_{\mathsf{pk}_1}$) has the form $\mathsf{sk}_2 || \mathbf{m}^b$ (as opposed to just being $\mathbf{m}^b$), and that the attacker also gets to see an encryption of sk_1 (using $\mathsf{Enc}^2_{\mathsf{pk}_2}$).

[3] Another consequence of using smudging is that our lattice trapdoor mechanism and its proof become simpler than [64], which uses a polynomial-size modulus instead, for a better efficiency.

1.3 Overview of the XiO Construction

We present a construction that makes a modular use of any LHE satisfying certain properties, and whose security relies the 2-circular SRL-security w.r.t. GSW and the LHE (i.e., that SRL security of GSW holds in the presence of a encrypted key cycle of length 2 using GSW and the LHE). To obtain a *subexponentially-secure* XiO (which is required to obtain iO by [57]), we need to strengthen the assumptions to also require subexponential security. Next, we note that the DJ LHE satisfies the desired properties. We prove that a packed version of Regev's encryption scheme [70] that is similar to, but actually different from, the packed construction from [69], does so as well. We refer to our LHE simply as Packed Regev LHE.

Let us start with the construction assuming 2-circular SRL-security w.r.t. GSW and any LHE satisfying the desired properties. As mentioned, on a high-level, our construction follows similar intuitions as the BDGM construction. We combine an FHE (in our case the GSW FHE) with a (special-purpose) LHE to implement an XiO. In fact, in our approach, we do not directly construct an XiO, but rather construct an XiO with *preprocessing*—this notion, which relaxes XiO by allowing the obfuscator to have access to some *long* public parameter pp, was actually already considered in [57] and it was noted there that subexponentially-secure XiO with preprocessing also suffices to get iO.

Towards explaining our approach, let us first recall the approach of BDGM—which relies on the DJ LHE—using a somewhat different language that will be useful for us.

The BDGM construction. The high-level idea is quite simple and very elegant. Recall that an XiO 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 $\mathbf{c}$, we can efficiently recover the ciphertext randomness 2) given a ciphertext $\mathbf{c}$ and its randomness—which will also be referred to as a "hint"—one can efficiently decrypt. Given such a special FHE, it is easy to construct an XiO: simply cut the truth table into "chunks" of length n^ϵ, FHE encrypt the program Π, then, homomorphically evalute circuits C_i for indices $i \in [n^{1-\epsilon}]$ such that given the program Π as input, C_i outputs the i'th "chunk" of the truth table, which we denote by Π_i; finally, release the randomness $\mathbf{r}_i$ (i.e. the "hint") of the evaluated ciphertexts. These hints enable compressing n^ϵ bits into $\mathsf{poly}(\lambda)\log(n^\epsilon)$ bits and thus the XiO is compressing.[4]

Unfortunately, none of the known FHE constructions have short randomness. BDGM, however, observes that there are *linear* homomorphic encryptions schemes (LHE), notably the DJ LHE, that satisfy the above requirements. Moreover, many FHEs are batcheable (with "long" randomness) and have "essentially" linear decryption: decryption is an inner product of the ciphertext with the secret key, then rounding. That is, the linear operations yield the plaintext with some additional small decryption noises, that are removing when rounding. So if we start off with such an FHE and additionally release an LHE encryption of the FHE secret key, we can get an FHE with the desired "batcheable with short randomness" requirement: we first homomorphically evaluate the inner product of the FHE ciphertext with the encrypted FHE secret key, then 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 Π_i but also the decryption noises, which is detrimental for the security of the FHE (2) the LHE randomness may actually leak more than just the decrypted LHE plaintext (i.e. something about how the LHE 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 decryption noises)[5]. Therefore, the only remaining problem is to generate these LHE encryptions of smudging noises. This is where 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 a sequence of LHE encryptions of uniformly random strings u_i for $i = 1, \ldots, n^{1-\varepsilon}$, since the DJ LHE has dense ciphertext (3) they additionally provide an FHE encryption of the LHE secret key $\overline{\mathsf{sk}}$ (note that there is now a circular security issue), on which they FHE-homomorphically evaluate a function f_i that decrypts the i'th LHE ciphertext produced by the random oracle, and computes $\mathsf{MSB}(u_i)$, the most significant bits of u_i (4) finally they LHE-evaluate the (partial) decryption of the evaluated FHE ciphertext (which encrypts $\mathsf{MSB}(u_i)$); the obtained LHE ciphertext can now be subtracted from the LHE ciphertexts generated by the random oracle, to get an LHE encryption of $u_i - \mathsf{MSB}(u_i)$, which is a noise of the appropriate size, i.e. smudging but not uniform.

One problem with this approach, however, is that while we do obtain an LHE encryption of appropriate smudging noise, it is not actually a fresh ciphertext (with fresh randomness). The issue is that the randomness $\mathbf{r}_{f_i}$ of the evaluated FHE ciphertext of $\mathsf{MSB}(u_i)$ may (and actually will) depend on the randomness of the original LHE ciphertext obtained by the RO. Another problem is that LHE can only compute the first step of an FHE decryption (namely, the linear operations), the LHE encryption obtained actually encrypts a message of the form: $u_i - \mathsf{MSB}(u_i) + \mathsf{noise}_i$. 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 extra noise that results from partially decrypting the FHE ciphertext depends on u_i, so the lower-order bits of the

[4] The reason we need to cut the truth table into chunks instead of directly computing the whole output is that for existing FHE schemes, the size of the FHE public key and ciphertexts grow polynomially with the length of the output of the homomorphic evaluation i.e. the size of the plaintext encrypted in an evaluated ciphertext, also referred to as "batching capacity". So the obfuscation is only compressing when we have a large number of chunks.

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

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 noises in a provably secure way, relying on *2-circular SRL-security* of GSW and DJ—that is, O_{SRL}-leakage resilient circular security of 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 XiO with preprocessing to obtain iO, so instead of using a random oracle, we will simply use a long random string as a public parameter, and interpret it as LHE encryptions of random strings.

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 "refresh" or re-encrypt the evaluated FHE ciphertext (which encrypts $\mathsf{MSB}(u_i)$) to ensure that the randomness is fresh and independent of the evaluations. This way, the decryption noise itself is independent of the evaluated circuit. GSW ciphertexts can be re-randomized simply by adding a fresh extra noisy FHE encryption of 0. How do we get such enecryptions? 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). 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 XiO). Instead, we use the public key of the GSW encryption to generate extra noisy encryptions of 0, and we include the (many) random coins $(\mathbf{r}_i^\star)_{i\in[n^{1-\varepsilon}]}$ used to generate these ciphertexts as part of the public parameters of the XiO (recall that the public parameters can be as long as we want). This method does indeed enable us to get a fresh FHE encryption of 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 $\mathsf{MSB}(u_i)$ and the public parameters $\mathbf{r}_i^\star$ are statistically close to freshly generated extra noisy FHE encryption of $\mathsf{MSB}(u_i)$ using randomness $\mathbf{r}_i^\star$, and setting the public parameter to $\mathbf{r}_i^\star - \mathbf{r}_{f_i}$, where $\mathbf{r}_{f_i}$ 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_i. Furthermore, noisy GSW encryptions of $\mathsf{MSB}(u_i)$ essentially have the form of a noisy GSW encryption of 0, to which $\mathsf{MSB}(u_i)$ is added. So, other than $\mathsf{MSB}(u_i)$, which is truly random, $\mathbf{r}_i^\star - \mathbf{r}_{f_i}$ is simply an SRL leakage on a GSW encryption of the LHE secret key $\overline{\mathsf{sk}}$! Thus, intuitively, security should now follow from circular SRL security of GSW and the LHE.

The final construction. We summarize our final XiO construction with preprocessing. The public parameter pp is a long *random* string that consists of two parts:

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

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

- **Output the public keys of the FHE and LHE:** The obfuscator generates a fresh key-pair $(\overline{\mathsf{pk}}, \overline{\mathsf{sk}})$ for the 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, ciphertexts and algorithms, that correspond to the *linear* homomorphic encryption.) The modulus N of the GSW encryption is set to be the same that the modulus that defines the message space $\mathbb{Z}_N$ of the LHE scheme. Additionally, it chooses N large enough to enable encrypting messages of size n^ε. Finally, it outputs the public keys $(\mathsf{pk}, \overline{\mathsf{pk}})$.
- **Output an FHE encryption of the circuit:** It outputs an FHE encryption (w.r.t. pk) of the program Π, which we denote by ct_1.
- **Output encrypted key cycle:** It computes ct_2, an FHE encryption of $\overline{\mathsf{sk}}$, and $\overline{\mathsf{ct}}$, an LHE encrytion of sk. It outputs the key cycle $\mathsf{ct}_2, \overline{\mathsf{ct}}$.
- **Output hints:** For every $i \in [n^{1-\varepsilon}]$, it outputs a short "hint" $\mathbf{r}_i$ computed as follows:
  - **Evaluate the circuit:** Homomorphically evaluate the circuit C_i on ct_1 and let ct_i denote the resulting evaluated FHE ciphertext — recall that ct_1 encrypts a program Π, and the circuit C_i takes a input a program Π and outputs the i'th chunk of its truth table.
  - **Compute an FHE encryption $\mathsf{ct}_{\mathsf{MSB},i}$ of $\mathsf{MSB}(u_i)$:** Consider the function $f_i(\Pi, \overline{\mathsf{sk}})$ that ignores the input Π but uses the input $\overline{\mathsf{sk}}$ to decrypt the i'th LHE ciphertext from LHE.PubCoin into a plaintext u_i and outputs $\mathsf{MSB}(u_i)$. The obfuscator homomorphically evaluates f_i on the ciphertexts $\mathsf{ct}_1, \mathsf{ct}_2$ (where, recall, ct_2 is an encryption of $\overline{\mathsf{sk}}$). Let $\mathsf{ct}_{\mathsf{MSB},i} = \mathsf{FHE}(\mathsf{MSB}(u_i); \mathbf{r}_{f_i})$ denote the resulting evaluated FHE ciphertext.
  - **Rerandomize $\mathsf{ct}_{\mathsf{MSB},i}$ into $\mathsf{ct}'_{\mathsf{MSB},i}$:** It uses the i'th chunk of FHE.PubCoin to get the randomness $\mathbf{r}_i^\star$; generates an extra noisy FHE encryption of 0 using $\mathbf{r}_i^\star$ and homomorphically adds it to $\mathsf{ct}_{\mathsf{MSB},i}$. Let $\mathsf{ct}'_{\mathsf{MSB},i} = \mathsf{FHE}(\mathsf{MSB}(u_i); \mathbf{r}_i^\star + \mathbf{r}_{f_i})$ denote the new (re-randomized) ciphertext.
  - **Proxy re-encrypt ct_i as an LHE ciphertext $\overline{\mathsf{ct}}_i$:** It uses $\overline{\mathsf{ct}}$ (which, recall, is an LHE encryption of sk) to homomorphically compute the *linear* part of the FHE decryption of ct_i, which yields an LHE encryption of the value $2^\omega \cdot \Pi_i + \mathsf{noise}_i$ where noise_i is an FHE decryption noise,

and 2^{ω} is taken large enough so that the plaintext Π_i can be recovered by rounding.
Similarly, it homomorphically computes the partial FHE decryption of $\mathsf{ct}'_{\mathsf{MSB},i}$, which yields an LHE encryption of the value $2^{\omega'} \cdot \mathsf{MSB}(u_i) + \mathsf{noise}_{\mathsf{MSB},i}$, where once again $\mathsf{noise}_{\mathsf{MSB},i}$ denotes an FHE decryption noise, and $2^{\omega'} = 1$ for reasons that will become clear later. We rely on the fact that GSW FHE (and many other FHE schemes) admits a flexible "scaled" evaluation algorithm, that can choose which integer 2^{ω} to use when performing the homomorphic evaluation (this was used also in prior works, including [22]). The resulting LHE ciphertext is subtracted from $\mathsf{LHE}(2^{\omega} \cdot \Pi_i + \mathsf{noise}_i)$, and therefore yields $\mathsf{LHE}(2^{\omega} \cdot \Pi_i + \mathsf{noise}_i - \mathsf{MSB}(u_i) - \mathsf{noise}_{\mathsf{MSB},i})$.
Finally, it homomorphically adds the LHE encryption of u_i that is part of the LHE public coins, to obtain $\overline{\mathsf{ct}}_i = \mathsf{LHE}(m_i)$, where $m_i = 2^{\omega} \cdot \Pi_i + \mathsf{noise}_i - \mathsf{MSB}(u_i) - \mathsf{noise}_{\mathsf{MSB},i} + u_i = 2^{\omega} \cdot \Pi_i + \mathsf{noise}_i + \mathsf{noise}_{\mathsf{MSB},i} + \mathsf{LSB}(u_i)$, where $\mathsf{LSB}(u_i)$ denotes the least significant bits of u_i.
The integer ω' is chosen to be equal to 0 so that the smudging noise $\mathsf{LSB}(u_i)$ is directly added to the FHE noises $\mathsf{noise}_i - \mathsf{noise}_{\mathsf{MSB},i}$. As opposed to the value Π_i that we place in the higher-order bits of the plaintext, we need the smudging noise to be at the same level than the FHE noises, so they "blend" together.

– **Release hint $\mathbf{r}_i$ for LHE ciphertext $\overline{\mathsf{ct}}_i$:** It uses $\overline{\mathsf{sk}}$ to recover the randomness $\mathbf{r}_i$ of $\overline{\mathsf{ct}}_i$ (recall that the LHE we use has a randomness recoverability property), and outputs $\mathbf{r}_i$.

To evaluate the obfuscated program on an input $\mathbf{x} \in \{0,1\}^n$, that pertains to the i'th chunk of the truth table of Π for some $i \in [n^{1-\varepsilon}]$, we compute $\overline{\mathsf{ct}}_i$ just like the obfuscator did (note that this does not require knowing the secret key, but only information contained in the obfuscated code). Finally, we decrypt $\overline{\mathsf{ct}}_i$ using the hint $\mathbf{r}_i$ to recover the message m_i described above (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_i to obtain Π_i, which contains $\Pi(\mathbf{x})$.

Outline of the security proof. We provide a very brief outline of the security proof. We will rely on the fact that LHE ciphertexts (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 XiO obfuscation of a program Π^0 and transition until we get an XiO obfuscation of Π^1, where Π^0 and Π^1 are two functionally equivalent circuits of the same size.

- **Hybrid 0: Honest $XiO(\Pi^0)$.** The first hybrid is just the honest obfuscation of the circuit Π^0.
- **Hybrid 1: Switch to freshly encrypted $\mathsf{ct}'_{\mathsf{MSB},i}$.** Hybrid 1 proceeds exactly as Hybrid 0 up until the point that the ciphertexts $\mathsf{ct}_{\mathsf{MSB},i}$ get re-encrypted into $\mathsf{ct}'_{\mathsf{MSB},i}$, with the exception that FHE.PubCoin are not sampled yet. Next, instead of performing the re-encryption, we sample $\mathsf{ct}'_{\mathsf{MSB},i}$ as a *fresh* extra noisy encryption of $\mathsf{MSB}(u_i)$ using randomness $\mathbf{r}_i^{\star}$, and setting FHE.PubCoin to be $\mathbf{r}_i^{\star} - \mathbf{r}_{f_i}$ (recall that $\mathbf{r}_{f_i}$ is the randomness obtained when homomorphically evaluating f_i on the FHE encryption of $\overline{\mathsf{sk}}$). 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 in Hybrid 1, for each $i \in [n^{1-\varepsilon}]$, the i'th chunk of FHE.PubCoin can be thought of as SRL leakage on the fresh encryption $\mathsf{ct}'_{\mathsf{MSB},i}$ computed w.r.t. function f_i, which will be useful for us later.
- **Hybrid 2: Switch LHE.PubCoin to encryptions of random strings.** Hybrid 2 proceeds exactly as Hybrid 1 except that instead of sampling LHE.PubCoin as a random string, we sample it as fresh LHE encryptions of random strings u_i, for $i = 1, \ldots, n^{1-\varepsilon}$. It follows by the density property of the LHE that Hybrid 2 is statistically close to Hybrid 1.
- **Hybrid 3: Generate $\overline{\mathsf{ct}}_i$ as a fresh encryption.** Hybrid 3 proceeds exactly as Hybrid 2 except that $\overline{\mathsf{ct}}_i$ is generated as a fresh encryption of m_i using fresh randomness $\mathbf{r}_i$, and the i'th chunk of LHE.PubCoin is instead computed homomorphically by subtracting the LHE encryption of $\mathsf{sk}^{\top}(\mathsf{ct}_i - \mathsf{ct}_{\mathsf{MSB},i})$ (obtained after homomorphically decrypting ct_i and $\mathsf{ct}'_{\mathsf{MSB},i}$ using $\overline{\mathsf{ct}}$) from the LHE ciphertext $\overline{\mathsf{ct}}_i$. Recall that $m_i = \mathsf{sk}^{\top}(\mathsf{ct}_i - \mathsf{ct}_{\mathsf{MSB},i}) + u_i$ so the above way of computing the i'th chunk of LHE.PubCoin ensures that it is valid encryption of u_i as in Hybrid 2, but this time with non-fresh, homomorphically evaluated 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 $\mathsf{ct}'_{\mathsf{MSB},i}$ remains exactly the same no matter what the LHE.PubCoin are. 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 $\mathbf{r}_i$.
- **Hybrid 4: Generate $\overline{\mathsf{ct}}_i$ without FHE noises.** Hybrid 4 proceeds exactly as Hybrid 3 except that $\overline{\mathsf{ct}}_i$ is generated as a fresh encryption of $m_i = 2^{\omega} \cdot \Pi_i^0 + \mathsf{LSB}(u_i)$, whereas in Hybrid 3, it was generated as fresh encryption of $m_i = 2^{\omega} \cdot \Pi_i^0 + \mathsf{LSB}(u_i) + \mathsf{noise}_i - \mathsf{noise}_{\mathsf{MSB},i}$. That is, we use $\mathsf{LSB}(u_i)$ as a smudging noise to hide the extra noise $\mathsf{noise}_i - \mathsf{noise}_{\mathsf{MSB},i}$. We can do so since (1) the extra FHE noise is small and independent of $\mathsf{LSB}(u_i)$ (2) the rest of the obfuscated code can be generated from the value $\mathsf{LSB}(u_i) + \mathsf{noise}_i - \mathsf{noise}_{\mathsf{MSB},i}$ only (in particular it does not require to know $\mathsf{LSB}(u_i)$ itself). 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 ct_1 is an encryption of Π^1 (instead of Π^0 in prior 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. Recall that FHE.PubCoin leaks something about the randomness used by the FHE encryption $\mathsf{ct}'_{\mathsf{MSB},i}$, but the leakage is exactly an SRL leakage (and note that in the experiment we do know the output α_i of the function f_i that is applied to the plaintexts encrypted in $\mathsf{ct}_1, \mathsf{ct}_2$—namely, it is $\mathsf{MSB}(u_i)$ where u_i is a random string selected in the experiment, see Hybrid 2). Thus, indistinguishability of Hybrid 5 and Hybrid 4 follows from 2-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 Π^1. Statistical closeness of intermediary hybrids follows just as before.

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

Theorem 1.4 (Informally stated). *Assume the 2-circular SRL-security of the GSW and DJ encryption schemes. Then, there exists an XiO for polynomial-size circuits taking inputs of length $\log(\lambda)$ where λ is the security parameter.*

An alternative LHE based on Packed Regev. We remark that we can obtain an alternative construction of an LHE with the desired properties by considering an *packed* version of the Regev encryption scheme. Our construction is slightly different, but similar in spirit, to the Packed Regev from [69]. Recall that a (plain) Regev public key consist of a pair $\mathbf{A}, \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top$, where $\mathbf{A} \leftarrow_R \mathbb{Z}_q^{m\times n}$ with $m \geq n\log(q)$, the vector $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$ is the secret key, and $\mathbf{e} \in \mathbb{Z}_q^m$ is some small "noise" vector. An encryption of a message μ has the form $\mathbf{Ar}, (\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top)\mathbf{r} + B\cdot\mu$ where $\mathbf{r} \leftarrow_R \{0,1\}^m$ is the encryption randomness and B is an upper bound on the size of noise (so as to enable decryption). This scheme is linearly homomorphic, but for security, the size of the randomness $|\mathbf{r}|$ needs to be greater than $n\log(q)$, which is more than that size of the message: the randomness is too long for our purposes.

To get succinct decryption hints, we simply reuse the same randomness $\mathbf{r}$ for many encryptions using different secret keys $\mathbf{s}_1, \mathbf{s}_2, \ldots \mathbf{s}_\ell$ and different noises $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_\ell$. The secret key is now a matrix $\mathbf{S} \in \mathbb{Z}_q^{\ell\times n}$, and the public key becomes $(\mathbf{A}, \mathbf{SA} + \mathbf{E})$ where $\mathbf{E} \in \mathbb{Z}_q^{\ell\times m}$ is a noise matrix. The encryption of a vector of messages $\mu = (\mu_1, \ldots, \mu_\ell)$ is then $(\mathbf{Ar}, (\mathbf{SA} + \mathbf{E})\mathbf{r} + B\mu)$. This is the scheme from [69]. Despite the fact that this encryption is still linearly homomorphic, and has the advantage of having rate-1 ciphertext size, its randomness is not short: to carry on the proof of security, we need to rely on the fact that $\mathbf{r}$ contains enough bits of entropy even when the information $\mathbf{Ar}$ (which is short) and $\mathbf{Er}$ (that is long) is leaked. The can only be true when the dimension of $\mathbf{r}$, m, grows with the number of bits that are batched, ℓ.

Thus, we depart from the scheme in [69] by adding a smudging noise[6] in the ciphertext, to hide the information $\mathbf{Er}$. The ciphertext is of the form: $(\mathbf{Ar}, (\mathbf{SA} + \mathbf{E})\mathbf{r} + \mathbf{e}' + B\cdot\mu)$, where $\mathbf{e}'$ is the extra smudging noise that hides the error term $\mathbf{Er}$, ensuring that we only have the short $\mathbf{Ar}$ leakage and the usual proof can again be applied.

This scheme is still linearly homomorphic, but the encryption randomness is still large, as even though we reuse $\mathbf{r}$, the added noise terms $\mathbf{e}'$ are large. However, we rely on the fact that knowing $\mathbf{e}'$ is not needed for decrypting. Indeed, to decrypt, we just need to know a small vector $\widetilde{\mathbf{r}} \in \mathbb{Z}_q^m$ such that $\mathbf{A}\widetilde{\mathbf{r}} = \mathbf{Ar}$. That can be used to remove the term $\mathbf{SAr}$ from the ciphertext, and recover $B\cdot\nu$ plus some small noise. To sample such vector, we use a standard trapdoor sampling mechanism as in prior works [4, 5, 42, 64]. This makes the scheme hintable with succinct hints.

We still have two (minor) obstacles, though. This scheme (as well as Regev's original scheme or the scheme from [69]) does not satisfy two of the other properties needed for our XiO construction: (1) density, and (2) weak circuit privacy. But it almost does. *Extra noisy* ciphertexts, where the noise reaches the bound B are actually dense, and for extra noisy ciphertext, weak circuit privacy also holds (just as it did for GSW). So, we can directly instantiate the LHE in our XiO construction with this Packed Regev construction, as long as we slightly relax the notion of an LHE to just require density when considering extra noisy ciphertexts.

Thus we can conclude:

Theorem 1.5 (Informally stated). *Assume 2-circular SRL security of the GSW and the Packed Regev encryption schemes holds. Then, there exists an XiO for polynomial-size circuits taking inputs of length $\log(\lambda)$ where λ is the security parameter.*

The proof of Theorems 1.1, 1.2 is finally concluded by upgrading Theorems 1.3, 1.4 and 1.5 to apply also in the subexponential regime, relying on the subexponential $\mathsf{2CIRC}^{O_{\mathsf{SRL}}}$ conjecture, and finally relying on the transformation from subexponentially-secure XiO with pre-processing (and subexponential LWE) to iO [57].

1.4 Comparing Circular SRL-security to "Plain" Circular Security

Let us make a few remarks on the 2-circular SRL-security assumption w.r.t GSW and some LHE (e.g. Dåmgard Jurik or Packed-Regev). Clearly, this assumption is stronger than the 2-circular assumption w.r.t GSW and the LHE—simply consider an attacker that does not request any leakage. Additionally, we wish to highlight a few qualitative differences between "plain" circular security and circular SRL-security:

- "Plain" circular security is a simple non-interactive falsifiable assumption. Circular SRL-security is also a (relatively simple) falsifiable assumption, but the security game is now *interactive*; for the type of SRL security needed for our application, a single "parallel" SRL query suffices and such a notion of SRL security can be specified as a 5-round security game.
  As we explain in more detail in the full version of this paper, for our application, one could define a *non-interactive* falsifiable variant of SRL security—roughly speaking, where the messages and the leakage-selection algorithm are randomly selected—such that the subexponential hardness of this circular "random-SRL" security notion suffices[7], but in our eyes, this non-interactive security game is less natural than the interactive one (and thus does not add much insight).

[6] Note that using a carefully crafted noise that needs not be of smudging size, as done in [64], we can "unskew" the noise $\mathbf{Er}$ and hide the information of $\mathbf{r}$. We favor clarify of the exposition over efficiency and resort to using smudging noises.

[7] This follows from a union bound as the length of both the messages $\mathbf{m}^0, \mathbf{m}^1$ and the description of the leakage-selection algorithm are "short".

- It is also worth noting that for the notion of "plain" circular security, an alternative way of defining circular security is to require that $\mathsf{Enc}^1_{\mathsf{pk}_1}(\mathsf{sk}_2), \mathsf{Enc}^2_{\mathsf{pk}_2}(\mathsf{sk}_1)$ is computationally indistinguishable from $\mathsf{Enc}^1_{\mathsf{pk}_1}(0^{|\mathsf{sk}_2|}), \mathsf{Enc}^2_{\mathsf{pk}_2}(\mathsf{sk}_1)$ (here Enc^1 denotes GSW and Enc^2 denotes LHE); this notion (together with non-circular security) implies circular security the way we have defined it (i.e. indistinguishability of encryptions of two messages in the presence of an encrypted key cycle). However, this implication no longer holds in the context of SRL security. And this is why we are directly defining circular security as indistinguishability of encryptions of messages in the presence of an encrypted key cycle.

The above two points indicate that the circular SRL-security w.r.t. GSW and LHE is both (a-priori) stronger, and also different from a qualitative point of view than the "plain" circular security assumption. Yet, some of the justifications for believing the latter holds true are also valid for circular SRL-security:

- In both cases (plain and SRL), security holds in a non-circular setting, assuming LWE.
- In both cases (plain and SRL), the security game being considered captures a simple and natural process (albeit for the case of SRL security, it is more complex).
- Finally, just as for the notion of plain circular security, it does not appear simple to even just come up with any *bit*-encryption scheme (such as GSW) that is SRL secure, but not circular SRL secure.[8]

1.5 Concurrent and Subsequent Work

A concurrent and independent breakthrough result by Jain, Lin and Sahai [48] presents a construction of $i\mathcal{O}$ based on subexponential security of well-founded assumptions: (1) the SXDH assumption on asymmetric bilinear groups, (2) the LWE assumption with subexponential modulus-to-noise ration, (3) a Boolean PRG in NC^0, and (4) an LPN assumption over a *large field* and with a small error rate $\frac{1}{\ell^\delta}$ where $\delta > 0$ and ℓ is the dimension of the LPN secret. Assumptions (1) and (2) have widespread use and are considered standard. (3) has also been well-studied in recent years. (4) is a very natural coding problem, but the range of parameters used in (4) differs from most prior works in the cryptographic literature, a majority of which focus on a less sparse error rate (typically a constant) and/or use the field $\mathbb{F}_2$.

A concurrent and independent work by Wee and Wichs [74] presents a new elegant heuristic instantiation of the BDGM paradigm based only on lattice-based primitives. Similarly to us, their construction proceeds by implementing $Xi\mathcal{O}$ with pre-processing. They also state a new security assumption with a circular security flavor (involving a PRF and LWE samples) under which they can prove the security of their construction: Roughly speaking, their construction proceed by reducing $Xi\mathcal{O}$ with pre-processing to the task of "oblivious LWE sampling", and next they provide a heuristic instantiation of a protocol for performing oblivious LWE sampling. Their security assumption is essentially that their protocol is a secure oblivious LWE sampler. It is worth noting that even though they also rely on the BDGM approach to implement $Xi\mathcal{O}$, they manage to directly construct an FHE with short randomness, relying on a "dual" variant of the GSW encryption scheme, thereby completely removing the use of any LHE (whereas we obtain short decryption hints by combining GSW with our Packed Regev).

The initial version of our paper did not contain the LWE-based instantiation of the LHE using Packed Regev (we just had the DJ-based instantiation). Following up on the initial posting of our paper, but concurrently and independently from our LWE-based construction, a preprint by Brakerski et al [23] also provides an LWE-based way to instantiate the LHE within our framework. Differently from our construction, however, they rely on a variant of the "Dual Regev" encryption scheme, whereas we rely on regular Regev.

Attacks on SRL security. Following up our work, the recent beautiful work [73] provides counter-examples to a generalization of the $\mathsf{2CIRC}^{\mathcal{O}_{\mathsf{SRL}}}$ conjecture where the SRL leakage can be obtained for circuits with algebraic gates that depends on the structure of the encryption scheme itself, as opposed to Boolean circuits (call this "extended SRL security"). Namely, they rely on the fact that the GSW FHE can also evaluate multiplication by a constant mod N gates, where N is the modulus used by the scheme itself. They present a variant of the GSW encryption scheme (GSW with even noise) which is both semantically and extended-SRL secure under LWE, yet circular extended-SRL security fails to hold—they present a concrete attack that cleverly exploits the above-mentioned homomorphic property. Alternatively, [73] can re-interpret their attack as an attack of "plain" (as opposed to "extended") circular SRL-security but w.r.t to a more artificial variant of the GSW encryption scheme. In this new encryption scheme, homomorphic multiplication is done in a more complicated way to ensure that the randomness of the evaluated ciphertext is correlated with one of the plaintexts. (Roughly speaking, the new homomorphic multiplication operation is performing some extra multiplication mod N computations, "under the hood" to achieve the same effect as in an extended SRL attack).

Their result thus highlights that circular SRL-security indeed is a qualitatively stronger notion than "plain" circular security (for which non-trivial attacks are not known for "natural" bit encryption schemes), and that more research is needed to understand the interplay between the class of leakage functions allowed in the definition of SRL security and the underlying FHE scheme, and notably, how the homomorphic operations are done. In particular, it appears important to use an FHE where homomorphic operations do not enable "biasing" the randomness of evaluated ciphertexts in a substantial way. Formalizing such a property is left as an important problem for future research.

2 DEFINITIONS

2.1 Definition of PKE and FHE

We start by recalling the definition of public key encryption (PKE) and fully-homomorphic encryption (FHE). For our purposes, we will consider the Common Reference String (CRS) model, where

[8] Wichs and Zirdelis [75] show that any public-key bit encryption scheme can modified in a way that preserves security yet violates circular security (using a special form of obfuscation that can be satisfied under LWE). The same method can be used to obtain an SRL-secure encryption scheme (by modifying GSW as in [75]) that is not 1-circular SRL secure.

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 ring $\mathbb{Z}_N$—this ring can be specified in the CRS.

Definition 2.1 (Public-Key Encryption). A Public-Key Encryption (PKE) scheme is a tuple of PPT algorithms (CRSgen, Gen, Enc, 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 $r \leftarrow_R \{0,1\}^\infty$[9], it outputs a ciphertext ct.
- $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct})$: 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:

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

Definition 2.2 (Fully-Homomorphic Encryption). A PKE scheme (CRSgen, Gen, Enc, Dec) is said to be a Fully-Homomorphic Encryption (FHE) scheme for depth $\delta(\cdot)$ 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 $n \in \mathbb{N}$, 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 Boolean circuits $f : \{0,1\}^n \to \{0,1\}$ of depth at most $\delta(\lambda)$, $\mathsf{Eval}(\mathsf{pk}, f, \mathsf{ct}_1, \ldots, \mathsf{ct}_n)$ 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 depth of the Boolean circuits that can be homomorphically evaluated is a priori bounded by $\delta(\lambda)$ for a polynomial δ (that is, we consider the case of *leveled* FHE). We consider Boolean circuits, that is, directed acyclic graphs where each vertex corresponds to an input or a logical (NOT, OR, AND) gate.

2.2 Leakage-Resilient and Circular Security

We recall the standard definition of CPA-security for encryption schemes; we furthermore generalize it to a notion of $\mathcal{O}$-leakage resilient security, which extends the standard definition by also providing the attacker with access to a leakage oracle $\mathcal{O}$ receiving the public key pk, the message $\mathbf{m}^\star$ being encrypted, and the randomness $\mathbf{r}$ under which it is encrypted. Our notion of $\mathcal{O}$ leakage-resilience restricts to attackers that only make "valid" leakage queries, where a query is said to be valid if the oracle does not return $\perp$ in response to it. In more detail, to "win" in the security game, the attacker $\mathcal{A}$ must (a) correctly guess which among two message $\mathbf{m}^0, \mathbf{m}^1$ is being encrypted, while (b) not having made any queries to $\mathcal{O}$ on which $\mathcal{O}$ returns $\perp$.

[9] 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.

Definition 2.3 ($\mathcal{O}$-leakage resilient security). We say that a public-key encryption scheme $\mathcal{PKE} = (\mathsf{CRSgen}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is $\mathcal{O}$-leakage resilient secure if for all stateful nuPPT adversaries $\mathcal{A}$, there exists some negligible function $\mu(\cdot)$ such that for all $\lambda \in \mathbb{N}$, $\Pr[\mathsf{Exp}^{\mathcal{PKE}}_{\lambda,\mathcal{A}} = 1] \leq 1/2 + \mu(\lambda)$, where the experiment $\mathsf{Exp}^{\mathcal{PKE}}_{\lambda,\mathcal{A}}$ is defined as follows:

$$\mathsf{Exp}^{\mathcal{PKE}}_{\lambda,\mathcal{A}} = \left\{ \begin{array}{l} \mathsf{crs} \leftarrow \mathsf{CRSgen}(1^\lambda), (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{crs}) \\ (\mathbf{m}^0, \mathbf{m}^1) \leftarrow \mathcal{A}(\mathsf{pk}), b \leftarrow \{0,1\} \\ \mathbf{m}^\star = \mathbf{m}^b, \mathbf{r} \leftarrow_R \{0,1\}^\infty \\ \mathsf{ct} = \mathsf{Enc}_{\mathsf{pk}}(\mathbf{m}^\star; \mathbf{r}), b' \leftarrow \mathcal{A}^{\mathcal{O}(\mathsf{pk}, \mathbf{m}^\star, \mathbf{r})}(\mathsf{ct}) \\ \text{Return 1 if } |\mathbf{m}^0| = |\mathbf{m}^1|,\ b' = b \text{ and} \\ \mathcal{O} \text{ did not return } \perp; \text{ 0 otherwise.} \end{array} \right\}$$

We say that $\mathcal{PKE}$ is simply *secure* if the above holds when we do not give $\mathcal{A}$ access to an oracle.

We will also consider a 2-circular secure variant of $\mathcal{O}$-leakage resilient security, which is similarly defined except we require indistinguishability of $\mathbf{m}^0$ and $\mathbf{m}^1$ in the presence not only of some randomness leakage, but also of an encrypted key cycle w.r.t. two public-key encryption schemes $\mathcal{PKE}$ and $\overline{\mathcal{PKE}}$. Note that we set the CRS of $\mathcal{PKE}$ to be the public key of $\overline{\mathcal{PKE}}$; this is to ensure compatibility between the schemes, i.e. for them to operate on the same ring.

Definition 2.4 ($\mathcal{O}$-leakage resilient 2-circular security). We say that public-key encryption schemes $\mathcal{PKE} = (\mathsf{CRSgen}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and $\overline{\mathcal{PKE}} = (\overline{\mathsf{CRSgen}}, \overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ are $\mathcal{O}$-leakage resilient 2-circular secure if for all stateful nuPPT adversaries $\mathcal{A}$, there exists some negligible function $\mu(\cdot)$ such that for all $\lambda \in \mathbb{N}$,

$$\Pr[\mathsf{Exp}^{\mathcal{PKE}, \overline{\mathcal{PKE}}}_{\lambda,\mathcal{A}} = 1] \leq 1/2 + \mu(\lambda),$$

where the experiment $\mathsf{Exp}^{\mathcal{PKE}, \overline{\mathcal{PKE}}}_{\lambda,\mathcal{A}}$ is defined as follows:

$$\mathsf{Exp}^{\mathcal{PKE}, \overline{\mathcal{PKE}}}_{\lambda,\mathcal{A}} = \left\{ \begin{array}{l} \overline{\mathsf{crs}} \leftarrow \overline{\mathsf{CRSgen}}(1^\lambda), (\overline{\mathsf{pk}}, \overline{\mathsf{sk}}) \leftarrow \overline{\mathsf{Gen}}(\overline{\mathsf{crs}}) \\ (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\overline{\mathsf{pk}}) \\ (\mathbf{m}^0, \mathbf{m}^1) \leftarrow \mathcal{A}(\overline{\mathsf{pk}}, \mathsf{pk}) \\ b \leftarrow \{0,1\}, \mathbf{m}^\star = \overline{\mathsf{sk}} \| \mathbf{m}^b \\ \mathbf{r} \leftarrow_R \{0,1\}^\infty, \mathsf{ct} = \mathsf{Enc}_{\mathsf{pk}}(\mathbf{m}^\star; \mathbf{r}) \\ \overline{\mathsf{ct}} \leftarrow \overline{\mathsf{Enc}}_{\overline{\mathsf{pk}}}(\mathsf{sk}), b' \leftarrow \mathcal{A}^{\mathcal{O}(\mathsf{pk}, \mathbf{m}^\star, \mathbf{r})}(\mathsf{ct}, \overline{\mathsf{ct}}) \\ \text{Return 1 if } |\mathbf{m}^0| = |\mathbf{m}^1|,\ b' = b \text{ and} \\ \mathcal{O} \text{ did not return } \perp; \text{ 0 otherwise.} \end{array} \right\}$$

We finally state the 2CIRC assumption that we will rely in our main theorem.

Definition 2.5 (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 $\overline{\mathcal{PKE}}$.

2.3 Definition of Shielded Randomness Leakage Security

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

2.3.1 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. That is, we consider FHE scheme where the crs contains a modulus N_{crs} such that decryption of an evaluated ciphertext yields a scalar in $\mathbb{Z}_{N_{\mathsf{crs}}}$. Our definition of FHE is flexible w.r.t. the choice of the modulus N_{crs}, which we can afford since the LWE assumption holds for essentially any (large enough) modulus. As observed in [21, 22, 63], most existing FHE schemes can fit this framework.

Definition 2.6 (Batch correctness). For all poynomials δ, an FHE scheme $(\mathsf{CRSgen}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ for depth-δ circuits satisfies batch correctness if there exist a PPT Eval' and a polynomial σ such that following holds:

- For all $\lambda \in \mathbb{N}$, all crs in the support of $\mathsf{CRSgen}(1^\lambda)$ contain a modulus $N_{\mathsf{crs}} \in \mathbb{N}$; for all $(\mathsf{pk}, \mathsf{sk})$ in the support of $\mathsf{Gen}(\mathsf{crs})$, we have: pk contains $B_{\mathsf{pk}} \in \mathbb{N}$ such that $N_{\mathsf{crs}} \geq 2^\lambda B_{\mathsf{pk}}$; the secret key is of the form: $\mathsf{sk} \in \mathbb{Z}^{\sigma(\lambda)}$.
- 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 input length $n \in \mathbb{N}$, all output length $\nu \in \mathbb{N}$ s.t. $\nu < \log(N_{\mathsf{crs}})$, all messages $m_1, \ldots, m_n \in \{0,1\}$, all depth-$\delta(\lambda)$ Boolean circuits $f : \{0,1\}^n \to \{0,1\}^\nu$, all ciphertexts ct_i in the support of $\mathsf{Enc}_{\mathsf{pk}}(m_i)$ for all $i \in [m]$, all scaling factors $\omega < \log(N_{\mathsf{crs}})$, the algorithm $\mathsf{Eval}'(\mathsf{pk}, f, \omega, \mathsf{ct}_1, \ldots, \mathsf{ct}_\nu)$ deterministically outputs an evaluated ciphertext $\mathsf{ct}_f \in \mathbb{Z}_{N_{\mathsf{crs}}}^{\sigma(\lambda)}$ such that:
$$\mathsf{sk}^\top \mathsf{ct}_f = 2^\omega f(\mathbf{m}) + \mathsf{noise}_f \in \mathbb{Z}_{N_{\mathsf{crs}}},$$
with $|\mathsf{noise}_f| < B_{\mathsf{pk}}$.

Here, $\mathbf{m} = (m_1, \ldots, m_n) \in \{0,1\}^N$, and $f(\mathbf{m}) = \sum_{i=1}^{\nu} 2^{i-1} f_i(\mathbf{m}) \in \mathbb{Z}_{N_{\mathsf{crs}}}$, where for all $i \in [\nu]$, $f_i(\mathbf{m}) \in \{0,1\}$ denotes the i'th output bit of the Boolean circuit f. Note that if $\nu < \log(N_{\mathsf{crs}}/B_{\mathsf{pk}})$, one can recover the value $f(\mathbf{m})$ when using any scaling factor $\omega > \log(B_{\mathsf{pk}})$. That is, we can define $\mathsf{Eval}(\mathsf{pk}, f, \mathsf{ct}_1, \ldots, \mathsf{ct}_\nu) = \mathsf{Eval}'(\mathsf{pk}, f, \lceil \log(B_{\mathsf{pk}}) \rceil + 1, \mathsf{ct}_1, \ldots, \mathsf{ct}_\nu)$.

2.3.2 Randomness Homomorphism. This property states that it is possible to homomorphically evaluates a Boolean circuit f not only on the ciphertexts, but also the randomness used by the ciphertexts. The resulting evaluated randomness $\mathbf{r}_f$ 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 2.7 (Randomness homomorphism). An FHE scheme $\mathcal{FHE} = (\mathsf{CRSgen}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ for depth-δ Boolean circuits that satisfies batch correctness (defined above) also satisfies randomness homomorphism if there exists a sequence of noisy randomness spaces $\{\mathcal{R}_\lambda^\star\}_{\lambda \in \mathbb{N}}$, 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-$\delta(\lambda)$ Boolean circuit $f : \{0,1\}^n \to \{0,1\}^\nu$, random coins $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_n)$ where for all $i \in [n]$, $\mathbf{r}_i \in \{0,1\}^\infty$, and messages $\mathbf{m} \in \{0,1\}^n$, it deterministically outputs an evaluated randomness $\mathbf{r}_f \in \mathcal{R}_\lambda^\star$.
- $\mathsf{Enc}_{\mathsf{pk}}^\star(\mu; \mathbf{r}^\star)$: given as input the public key pk, a message $\mu \in \mathbb{Z}_{N_{\mathsf{crs}}}$ and the randomness $\mathbf{r}^\star \in \mathcal{R}^\star$, 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 $n \in \mathbb{N}$, $\nu \in \mathbb{N}$ s.t. $\nu < \log(N_{\mathsf{crs}})$, all depth-$\delta(\lambda)$ Boolean circuits $f : \{0,1\}^n \to \{0,1\}^\nu$, all messages $m_1, \ldots, m_n \in \{0,1\}$, all randomness $\mathbf{r}_1, \ldots, \mathbf{r}_n \in \{0,1\}^\infty$, denoting $\mathsf{ct}_i = \mathsf{Enc}_{\mathsf{pk}}(m_i; \mathbf{r}_i)$ for all $i \in [n]$ and $\mathbf{r}_f = \mathsf{Eval}_{\mathsf{rand}}(\mathsf{pk}, f, \mathbf{r}, \mathbf{m})$, we have $\mathbf{r}_f \in \mathcal{R}_\lambda^\star$ and:
$$\mathsf{Eval}'(\mathsf{pk}, f, 0, \mathsf{ct}_1, \ldots, \mathsf{ct}_\nu) = \mathsf{Enc}_{\mathsf{pk}}^\star(f(\mathbf{m}); \mathbf{r}_f),$$
where $f(\mathbf{m}) = \sum_{i=1}^{\nu} 2^{i-1} f_i(\mathbf{m}) \in \mathbb{Z}_{N_{\mathsf{crs}}}$ and for all $i \in [\nu]$, $f_i(\mathbf{m}) \in \{0,1\}$ denotes the i'th output bit of the Boolean circuit f.

2.3.3 Shielded Randomness-Leakage Security. We proceed to formally define shielded randomness leakage (SRL) security for randomness homomorphic FHEs with batch correctness. SRL security will be defined as O-leakage resilient security for a particular leakage oracle O_{SRL} that given the public key pk, a message $\mathbf{m}^\star$ and randomness $\mathbf{r}$, allows the attacker $\mathcal{A}$ to ask to see a "shielded" version of the homomorphically evaluated randomness $\mathbf{r}_f$ for any Boolean circuit f *for which* $\mathcal{A}$ *knows the output* $f(\mathbf{m}^\star)$. To make sure the attacker can only query the oracle with Boolean circuits on which it knows the output, we require the attacker to also provide the output α, and the oracle outputs $\perp$ if $f(\mathbf{m}^\star) \neq \alpha$ (and thus, by the definition of O-leakage resilient security, the attacker fails if it ever picks a function for which it does not know the output).

To formalize the SRL 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 $\{\mathcal{R}_\lambda^\star\}_{\lambda \in \mathbb{N}}$ is such that for all $\lambda \in \mathbb{N}$, $\mathcal{R}_\lambda^\star \subseteq \mathbb{Z}^{t(\lambda)}$. Henceforth, we denote by $\mathbf{r}_1 + \mathbf{r}_2 \in \mathcal{R}_\lambda^\star$ and $\mathbf{r}_1 - \mathbf{r}_2 \in \mathcal{R}_\lambda^\star$ the addition and subtraction in $\mathbb{Z}^{t(\lambda)}$. We denote $\mathcal{R}_\lambda^\star$ by $\mathcal{R}^\star$ for simplicity.

Definition 2.8 (SRL security). An FHE scheme $\mathcal{FHE}$ for depth δ Boolean circuits satisfying randomness homomorphism is said to be SRL-secure if it is $O_{\mathsf{SRL}}^{\mathcal{FHE}}$-leakage resilient secure for the following oracle $O_{\mathsf{SRL}}^{\mathcal{FHE}}$, where $\mathsf{Eval}_{\mathsf{rand}}$ and $\mathsf{Enc}^\star$ are the algorithms guaranteed to exist by the definition of randomness homomorphism. Similarly, for any public-key encryption scheme $\overline{\mathcal{PKE}}$, we say 2-circular SRL security holds w.r.t. $\mathcal{FHE}$ and $\overline{\mathcal{PKE}}$ if $O_{\mathsf{SRL}}^{\mathcal{FHE}}$-leakage resilient 2-circular security holds w.r.t. $\mathcal{FHE}$ and $\overline{\mathcal{PKE}}$.

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

Roughly speaking, given a message $\mathbf{m}^\star$ and randomness $\mathbf{r}$, the oracle $\mathcal{O}_{\mathsf{SRL}}^{\mathcal{FHE}}$ samples fresh random coins $\mathbf{r}^\star$ from which it generates a noisy encryption of zero, that is sent to the adversary. The adversary next chooses a Boolean circuit f and a value $\alpha \in \{0,1\}^*$. The oracle then checks that $f(\mathbf{m}^\star) = \alpha$, upon which it returns the evaluated randomness "shielded" with the randomness $\mathbf{r}^\star$; otherwise, it outputs $\perp$ and in this case, the attacker fails.

In the concrete FHE we consider from [43], the randomness leakage corresponds to the randomness obtained from homomorphically subtracting the evaluated challenge ciphertext from $\mathsf{Enc}_{\mathsf{pk}}^\star(0; \mathbf{r}^\star)$. Revealing such leakage allows the adversary to decrypt and recover the value $0 - f(\mathbf{m}^\star)$. This is why we only allow the attacker to request leakage for Boolean circuits f for which it knows the output $f(\mathbf{m}^\star) \in \{0,1\}^*$.

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

2.4 Definition of iO

We recall the definition of $i\mathcal{O}$ [9, 36]. Given polynomials $n(\cdot), s(\cdot), d(\cdot)$, let $\mathcal{C}_{n,s,d} = \{C_\lambda\}_{\lambda\in\mathbb{N}}$ denote the class of circuits such that for all $\lambda \in \mathbb{N}$, C_λ 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 $\{C_\lambda\}_{\lambda\in\mathbb{N}}$ (denoted by $\{\Pi_\lambda\}_{\lambda\in\mathbb{N}} \in \{C_\lambda\}_{\lambda\in\mathbb{N}}$) if for all $\lambda \in \mathbb{N}$, $\Pi_\lambda \in C_\lambda$.

Definition 2.9 ($i\mathcal{O}$ for P/poly). We say that $i\mathcal{O}$ exists for P/poly if for all polynomials $n(\cdot), s(\cdot), d(\cdot)$, there exists a tuple of PPT algorithms (Obf, Eval) such that the following holds:

- **Correctness:** For all $\{\Pi_\lambda\}_{\lambda\in\mathbb{N}} \in \mathcal{C}_{n,s,d}$, there exists a negligible function μ such that for all $\lambda \in \mathbb{N}$, all $\mathbf{x} \in \{0,1\}^{n(\lambda)}$,

$$\Pr[\widetilde{\Pi} \leftarrow \mathsf{Obf}(1^\lambda, \Pi_\lambda) : \mathsf{Eval}(1^\lambda, \widetilde{\Pi}, \mathbf{x}) = \Pi(\mathbf{x})] \geq 1 - \mu(n)$$

- **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:

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

2.5 Learning with Errors Assumption

Definition 2.10 (LWE assumption [68, 70]). For all sequences $q \in 2^{\mathsf{poly}(\kappa)}$, all ensembles χ of efficiently sampleable distributions over $\mathbb{Z}$, we say that (subexponential) security of the LWE assumption holds w.r.t. the sequence q and the ensemble χ if for all polynomials $m(\cdot)$, the following ensembles are (subexponentially) computationally indistinguishable:

$$\left\{\mathbf{A} \leftarrow_R \mathbb{Z}_{q_\kappa}^{m(\kappa)\times\kappa}, \mathbf{s} \leftarrow \chi_\kappa^\kappa, \mathbf{e} \leftarrow \chi_\kappa^{m(\kappa)}, \mathbf{z} = \mathbf{A}\mathbf{s} + \mathbf{e} \in \mathbb{Z}_{q_\kappa}^{m(\kappa)} : (\mathbf{A}, \mathbf{z})\right\}_{\kappa\in\mathbb{N}}.$$
$$\left\{\mathbf{A} \leftarrow_R \mathbb{Z}_{q_\kappa}^{m(\kappa)\times\kappa}, \mathbf{z} \leftarrow_R \mathbb{Z}_{q_\kappa}^{m(\kappa)} : (\mathbf{A}, \mathbf{z})\right\}_{\kappa\in\mathbb{N}}.$$

We say the (subexponential) security of the LWE assumption holds if there exists a constant $c \in (0,1)$ such that for all sequences $q \in 2^{\mathsf{poly}(\kappa)}$ and all polynomials B such that for all $\kappa \in \mathbb{N}$, the following holds:

- $B(\kappa) \geq 2\sqrt{\kappa \log(\kappa)}$
- $B(\kappa) \geq q_\kappa 2^{-\kappa^c}$

the LWE assumption holds w.r.t. q, χ, where $\chi = \{\chi_\kappa\}_{\kappa\in\mathbb{N}}$ is the ensemble of distributions where for all $\kappa \in \mathbb{N}$, χ_κ is the uniformly random distribution over $[-B(\kappa), B(\kappa)]$.

3 OUR IO CONSTRUCTION

Our construction relies on the GSW FHE and the Packed-Regev PKE, decribed below.

3.1 The GSW FHE Scheme

We recall the FHE from [43]. We present the leveled variant (without bootstrapping), which is parameterized by a polynomial δ that bounds the depth of the Boolean circuits that can be homomorphically evaluated. Its security relies on the LWE assumption with a subexponential modulus-to-noise ratio. We denote the scheme by GSW_δ.

For all polynomials δ, we denote by b_δ a polynomial (aribtrarily chosen) such that for all polynomials B_χ, κ, all $\lambda \in \mathbb{N}$, the noise obtained from homomorphically evaluating circuits of depth at most $\delta(\lambda)$ on GSW ciphertexts generated with an LWE noise distribution uniformly random over $[-B_\chi(\lambda,), B_\chi(\lambda)]$ and an LWE secret of dimension $\kappa(\lambda)$, is upper bounded by $2^{b_\delta(\lambda)}$.

Boolean circuits are encoded naturally with addition, subtraction, multiplication, and addition by a constant gates over the integers. Namely, for any bit $a, b \in \{0,1\}$, $\mathsf{NOT}(a)$ is implemented by $1 - a$, $\mathsf{AND}(a,b)$ is implemented by $a \cdot b$, and $\mathsf{OR}(a,b)$ is implemented by $a + b - a \cdot b$, where the addition $+$ and multiplication $\cdot$ are performed in $\mathbb{Z}$. Note that this encoding only incurs a constant mutiplicative blow-up in the circuit size, and no increase in the circuit depth.

• Gen(crs):
Given as input crs which contains a modulus $N \geq 2^{2\lambda + b_\delta(\lambda)}$, it chooses a sequence $\{q_n\}_{n\in\mathbb{N}}$ and polynomials B_χ , κ such that LWE holds w.r.t. q and χ the B_χ-bounded ensemble of uniformly random distributions, and $q_{\kappa(\lambda)} = N$ (by the LWE assumption, given in Definition 2.10, we know such parameters exist). We abuse notations and write $\kappa = \kappa(\lambda)$, $\chi = \chi_{\kappa(\lambda)}$ and $B_\chi = B_\chi(\kappa(\lambda))$ from here on. The algorithm sets $w = (\kappa+1)\lceil\log(N)\rceil$, $m = 2(\kappa+1)\lceil\log(N)\rceil + 2\lambda$, $B^\star = 2^\lambda(w+1)^\delta\lceil\log(N)\rceil$ and $B = B_\chi(w+1)^\delta \log(N)m$. Note that we have $N \geq 2^{2\lambda}B$.

It samples $\mathbf{A} \leftarrow_R \mathbb{Z}_N^{\kappa\times m}$, $\mathbf{s} \leftarrow \chi^\kappa$, $\mathbf{e} \leftarrow \chi^m$, $\mathbf{g} = (1, 2, \ldots, 2^{\lceil\log(N)\rceil - 1}) \in \mathbb{Z}_N^{\lceil\log(N)\rceil}$, $\mathbf{G} = \mathbf{g}^\top \otimes \mathrm{Id} \in \mathbb{Z}_N^{(\kappa+1)\times w}$ where $\mathrm{Id} \in \mathbb{Z}_N^{(\kappa+1)\times(\kappa+1)}$ denotes the identity matrix, $\mathbf{U} = \begin{pmatrix}\mathbf{A}\\ \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top\end{pmatrix} \in \mathbb{Z}_N^{(\kappa+1)\times m}$.

It sets $\mathsf{pk} = (B, \mathbf{U}, \mathbf{G})$, and $\mathsf{sk} = (-\mathbf{s}, 1) \otimes \mathbf{g} \in \mathbb{Z}_N^w$. The parameters define the noisy randomness space $\mathcal{R}^\star = [-B^\star, B^\star]^m$. 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 $\mathbf{R} \leftarrow_R [-1,1]^{m \times w}$ and outputs the ciphertext $\mathsf{ct} = \mathbf{UR} + m\mathbf{G} \in \mathbb{Z}_N^{(\kappa+1)\times w}$. For any $\mathbf{m} \in \{0,1\}^n$, we denote by $\mathsf{Enc}_{\mathsf{pk}}(\mathbf{m}; \mathbf{r})$ the concatenation of the encryptions $\mathsf{Enc}_{\mathsf{pk}}(m_1; \mathbf{R}_1), \ldots, \mathsf{Enc}_{\mathsf{pk}}(m_n; \mathbf{R}_n)$.

• $\underline{\mathsf{Eval}(\mathsf{pk}, f, \mathsf{ct}_1, \ldots, \mathsf{ct}_\nu)}$:
Given the public key pk, a depth-$\delta(\lambda)$ Boolean $f : \{0,1\}^n \to \{0,1\}^\nu$, ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_n$, it runs $\mathsf{ct}_f \leftarrow \mathsf{Eval}'(\mathsf{pk}, f, \omega, \mathsf{ct}_1, \ldots, \mathsf{ct}_n)$ with scaling factor $\omega = \lceil \log(B) \rceil + 1$, where the algorithm Eval' is described below, for the batch correctness property.

Theorem 3.1 (SRL security). *Assume the (subexponential) LWE assumption holds. Then, for all polynomials δ, GSW_δ is (subexponentially) SRL secure.*

The proof of this theorem is given in the full version of this paper.

3.2 The Packed-Regev PKE Scheme

We present a packed version of Regev encryption scheme [70], which is parameterized by polynomials ℓ_1 and ℓ_2. We denote the scheme by $\text{P-Regev}_{\ell_1,\ell_2}$. These parameters are used to define some special properties of the encryption scheme (e.g. some "batching" properties), which are defined and proven in the full version of this paper. The scheme relies on trapdoor sampling, whereby a matrix $\mathbf{A}$ is sampled together with an associated trapdoor $T_{\mathbf{A}}$ that permits to sample short pre-image of the map $\mathbf{x} \to \mathbf{Ax}$. Again, we defer to the full version of this paper for further details.

• $\underline{\mathsf{CRSgen}(1^\lambda)}$:
It simply outputs $\mathsf{crs} = 1^\lambda$, i.e. there is no proper crs for that scheme.

• $\underline{\mathsf{Gen}(\mathsf{crs})}$:
Given as input $\mathsf{crs} = 1^\lambda$, it chooses $q = \{q_\kappa\}_{\kappa \in \mathbb{N}}$ with $q_\kappa = 2^{\kappa^c}$, $B_\chi(\kappa) = \kappa$ and $\kappa(\lambda) = \ell_1(\lambda)^{1/c}$, where $c \in (0,1)$ is the constant from Definition 2.10, and $N = q_{\kappa(\lambda)}$. We abuse notations and write $\kappa = \kappa(\lambda)$, $\chi = \chi_{\kappa(\lambda)}$ and $B_\chi = B_\chi(\kappa(\lambda))$ from here on.

Then, the algorithm samples $(\mathbf{A}, T_{\mathbf{A}}) \leftarrow \mathsf{TrapGen}(1^\lambda, N, \kappa)$, $\mathbf{S} \leftarrow \chi^{\ell_2 \times \kappa}$, $\mathbf{E} \leftarrow \chi^{\ell_2 \times m}$, and sets $\mathsf{pk} = (N, \mathbf{A}, \mathbf{SA} + \mathbf{E}) \in \mathbb{N} \times \mathbb{Z}_N^{\kappa \times m} \times \mathbb{Z}_N^{\ell_2 \times m}$, $\mathsf{sk} = \mathbf{S}$. It outputs $(\mathsf{pk}, \mathsf{sk})$.

• $\underline{\mathsf{Enc}_{\mathsf{pk}}(\mathbf{x} \in \mathbb{Z}_N^\nu)}$:
Given the public pk, a vector $\mathbf{x} \in \mathbb{Z}_N^\nu$, it samples $\mathbf{R} \leftarrow [-1,1]^{m \times \nu\ell_2}$, $\mathbf{E}' \leftarrow_R [-2^{\lambda/2}, 2^{\lambda/2}]^{\ell_2 \times \nu\ell_2}$ and outputs the ciphertext $\mathsf{ct} = \left(\mathbf{AR}, (\mathbf{SA}+\mathbf{E})\mathbf{R} + \mathbf{E}' + \mathbf{x}^\top \otimes \mathrm{Id}_{\ell_2}\right) \in \mathbb{Z}_N^{(\kappa+\ell_2)\times\nu\ell_2}$, where $\mathrm{Id}_{\ell_2} \in \mathbb{Z}_N^{\ell_2 \times \ell_2}$ denotes the identity matrix, and $\mathbf{x}^\top \otimes \mathrm{Id}_{\ell_2} \in \mathbb{Z}_N^{\ell_2 \times \nu\ell_2}$.

• $\underline{\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct})}$:
Given as input the secret key sk and a ciphertext $\mathsf{ct} = (\mathbf{t}, \mathbf{z})$ with $\mathbf{t} \in \mathbb{Z}_N^\kappa$, $\mathbf{z} \in \mathbb{Z}_N^{\ell_2}$, it outputs $\mathbf{d} = \mathbf{z} - \mathbf{St} \in \mathbb{Z}_N^{\ell_2}$.

Theorem 3.2 (security). *Assume the (subexponential) LWE assumption holds. Then, for all polynomials ℓ_1, ℓ_2, $\text{P-Regev}_{\ell_1,\ell_2}$ is (subexponentially) secure.*

The proof of this theorem is given in the full version of this paper.

3.3 Our Main Result

Theorem 3.3. *Assume the subexponential LWE assumption holds. Assume further that for all polynomials δ, ℓ_1 and ℓ_2, the subexponential $2\mathrm{CIRC}^{\mathcal{O}_{\mathsf{SRL}}}$ conjecture holds w.r.t. GSW_δ and $\text{P-Regev}_{\ell_1,\ell_2}$ (described above). Then subexponentially-secure $i\mathcal{O}$ for P/poly exists.*

The proof of this theorem is given in the full version of this paper.

ACKNOWLEDGMENTS

Romain Gay performed this work while partially at Cornell Tech.

Rafael Pass is 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.

We wish to thank Hoeteck Wee and Daniel Wichs for their insightful feedback on a previous eprint version of this paper. In particular, they encouraged us to present a game-based definition of SRL security (as opposed to the indistinguishability based definition we presented in our earlier draft), and to clarify differences between circular SRL-security and "plain" circular security. We are also very grateful to Sam Hopkins, Aayush Jain and Rachel Lin for sharing and disussing their new paper on counter-examples for (extended) SRL circular security and for insightful feedback on this paper.

REFERENCES

[1] Tolga Acar, Mira Belenkiy, Mihir Bellare, and David Cash. 2010. Cryptographic Agility and Its Relation to Circular Encryption. 403–422. https://doi.org/10.1007/978-3-642-13190-5_21

[2] Shweta Agrawal. 2019. Indistinguishability Obfuscation Without Multilinear Maps: New Methods for Bootstrapping and Instantiation. 191–225. https://doi.org/10.1007/978-3-030-17653-2_7

[3] Shweta Agrawal and Alice Pellet-Mary. 2020. Indistinguishability Obfuscation Without Maps: Attacks and Fixes for Noisy Linear FE. 110–140. https://doi.org/10.1007/978-3-030-45721-1_5

[4] Miklós Ajtai. 1996. Generating Hard Instances of Lattice Problems (Extended Abstract). 99–108. https://doi.org/10.1145/237814.237838

[5] Joël Alwen and Chris Peikert. 2009. Generating Shorter Bases for Hard Random Lattices. In *26th International Symposium on Theoretical Aspects of Computer Science STACS 2009 (Proceedings of the 26th Annual Symposium on the Theoretical Aspects of Computer Science)*, Susanne Albers and Jean-Yves Marion (Eds.). IBFI Schloss Dagstuhl, Freiburg, Germany, 75–86.

[6] Prabhanjan Ananth and Abhishek Jain. 2015. Indistinguishability Obfuscation from Compact Functional Encryption. 308–326. https://doi.org/10.1007/978-3-662-47989-6_15

[7] Prabhanjan Ananth, Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. 2019. Indistinguishability Obfuscation Without Multilinear Maps: New Paradigms via Low Degree Weak Pseudorandomness and Security Amplification. Cryptology ePrint Archive, Report 2019/643. https://eprint.iacr.org/2019/643.

[8] Prabhanjan Ananth, Aayush Jain, and Amit Sahai. 2018. *Indistinguishability Obfuscation Without Multilinear Maps: iO from LWE, Bilinear Maps, and Weak Pseudorandomness.* Technical Report. Cryptology ePrint Archive, Report 2020/764, 2020. https://eprint.iacr.org/2018/615.

[9] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. 2001. On the (Im)possibility of Obfuscating Programs. 1–18. https://doi.org/10.1007/3-540-44647-8_1

[10] James Bartusek, Yuval Ishai, Aayush Jain, Fermi Ma, Amit Sahai, and Mark Zhandry. 2020. Affine Determinant Programs: A Framework for Obfuscation and Witness Encryption. 82:1–82:39. https://doi.org/10.4230/LIPIcs.ITCS.2020.82

[11] Mihir Bellare and Phillip Rogaway. 1993. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. 62–73. https://doi.org/10.1145/168588.168596

[12] Allison Bishop, Susan Hohenberger, and Brent Waters. 2015. New Circular Security Counterexamples from Decision Linear and Learning with Errors. 776–800. https://doi.org/10.1007/978-3-662-48800-3_32

[13] Nir Bitansky, Sanjam Garg, Huijia Lin, Rafael Pass, and Sidharth Telang. 2015. Succinct Randomized Encodings and their Applications. *IACR Cryptology ePrint Archive* 2015 (2015), 356. http://eprint.iacr.org/2015/356

[14] Nir Bitansky and Omer Paneth. 2015. ZAPs and Non-Interactive Witness Indistinguishability from Indistinguishability Obfuscation. 401–427. https://doi.org/10.1007/978-3-662-46497-7_16

[15] Nir Bitansky, Omer Paneth, and Alon Rosen. 2015. On the Cryptographic Hardness of Finding a Nash Equilibrium. 1480–1498. https://doi.org/10.1109/FOCS.2015.94

[16] Nir Bitansky, Omer Paneth, and Daniel Wichs. 2016. Perfect Structure on the Edge of Chaos - Trapdoor Permutations from Indistinguishability Obfuscation. 474–502. https://doi.org/10.1007/978-3-662-49096-9_20

[17] Nir Bitansky and Vinod Vaikuntanathan. 2015. Indistinguishability Obfuscation from Functional Encryption. 171–190. https://doi.org/10.1109/FOCS.2015.20

[18] J. Black, P. Rogaway, and T. Shrimpton. 2002. Encryption-Scheme Security in the Presence of Key-Dependent Messages. Cryptology ePrint Archive, Report 2002/100. https://eprint.iacr.org/2002/100.

[19] Dan Boneh and Mark Zhandry. 2014. 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.* 480–499. https://doi.org/10.1007/978-3-662-44371-2_27

[20] Elette Boyle, Kai-Min Chung, and Rafael Pass. 2014. On Extractability Obfuscation. In *TCC*. 52–73.

[21] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. 2019. Leveraging Linear Decryption: Rate-1 Fully-Homomorphic Encryption and Time-Lock Puzzles. 407–437. https://doi.org/10.1007/978-3-030-36033-7_16

[22] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. 2020. Candidate iO from Homomorphic Encryption Schemes. 79–109. https://doi.org/10.1007/978-3-030-45721-1_4

[23] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. 2020. Factoring and Pairings are not Necessary for iO: Circular-Secure LWE Suffices. Cryptology ePrint Archive, Report 2020/1024. https://eprint.iacr.org/2020/1024.

[24] Jan Camenisch and Anna Lysyanskaya. 2001. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. 93–118. https://doi.org/10.1007/3-540-44987-6_7

[25] Ran Canetti, Oded Goldreich, and Shai Halevi. 1998. The Random Oracle Methodology, Revisited (Preliminary Version). 209–218. https://doi.org/10.1145/276698.276741

[26] Ran Canetti, Justin Holmgren, Abhishek Jain, and Vinod Vaikuntanathan. 2014. Indistinguishability Obfuscation of Iterated Circuits and RAM Programs. Cryptology ePrint Archive, Report 2014/769. https://eprint.iacr.org/2014/769.

[27] Ran Canetti, Yael Tauman Kalai, and Omer Paneth. 2015. 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.* 456–467. https://doi.org/10.1007/978-3-662-46497-7_18

[28] David Cash, Matthew Green, and Susan Hohenberger. 2012. New Definitions and Separations for Circular Security. 540–557. https://doi.org/10.1007/978-3-642-30057-8_32

[29] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. 2015. 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.* 3–12. https://doi.org/10.1007/978-3-662-46800-5_1

[30] Kai-Min Chung, Huijia Lin, and Rafael Pass. 2015. Constant-Round Concurrent Zero-Knowledge from Indistinguishability Obfuscation. 287–307. https://doi.org/10.1007/978-3-662-47989-6_14

[31] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. 2013. Practical Multilinear Maps over the Integers. 476–493. https://doi.org/10.1007/978-3-642-40041-4_26

[32] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. 2015. New Multilinear Maps Over the Integers. 267–286. https://doi.org/10.1007/978-3-662-47989-6_13

[33] Ivan Damgård and Mats Jurik. 2001. A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. 119–136. https://doi.org/10.1007/3-540-44586-2_9

[34] Sanjam Garg, Craig Gentry, and Shai Halevi. 2013. Candidate Multilinear Maps from Ideal Lattices. 1–17. https://doi.org/10.1007/978-3-642-38348-9_1

[35] Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. 2014. 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.* 74–94. https://doi.org/10.1007/978-3-642-54242-8_4

[36] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. 2013. Candidate Indistinguishability Obfuscation and Functional Encryption for all Circuits. 40–49. https://doi.org/10.1109/FOCS.2013.13

[37] Romain Gay, Aayush Jain, Huijia Lin, and Amit Sahai. 2020. Indistinguishability Obfuscation from Simple-to-State Hard Problems: New Assumptions, New Techniques, and Simplification. Cryptology ePrint Archive, Report 2020/764. https://eprint.iacr.org/2020/764.

[38] Craig Gentry. 2009. Fully homomorphic encryption using ideal lattices. 169–178. https://doi.org/10.1145/1536414.1536440

[39] Craig Gentry, Sergey Gorbunov, and Shai Halevi. 2015. Graph-Induced Multilinear Maps from Lattices. 498–527. https://doi.org/10.1007/978-3-662-46497-7_20

[40] Craig Gentry, Charanjit S Jutla, and Daniel Kane. 2018. Obfuscation Using Tensor Products.. In *Electronic Colloquium on Computational Complexity (ECCC)*, Vol. 25. 149.

[41] Craig Gentry, Allison Lewko, Amit Sahai, and Brent Waters. 2014. Indistinguishability Obfuscation from the Multilinear Subgroup Elimination Assumption. Cryptology ePrint Archive, Report 2014/309.

[42] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. 2008. Trapdoors for hard lattices and new cryptographic constructions. 197–206. https://doi.org/10.1145/1374376.1374407

[43] Craig Gentry, Amit Sahai, and Brent Waters. 2013. Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. 75–92. https://doi.org/10.1007/978-3-642-40041-4_5

[44] Shafi Goldwasser and Yael Tauman Kalai. 2005. 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.* 553–562. https://doi.org/10.1109/SFCS.2005.60

[45] Rishab Goyal, Venkata Koppula, and Brent Waters. 2017. Separating Semantic and Circular Security for Symmetric-Key Bit Encryption from the Learning with Errors Assumption. 528–557. https://doi.org/10.1007/978-3-319-56614-6_18

[46] Matthew Green and Susan Hohenberger. 2010. CPA and CCA-Secure Encryption Systems that are not 2-Circular Secure. http://eprint.iacr.org/2010/144 matthewdgreen@gmail.com 14686 received 16 Mar 2010, last revised 18 Mar 2010.

[47] Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. 2019. How to Leverage Hardness of Constant-Degree Expanding Polynomials overa $\mathbb{R}$ to build $i\mathcal{O}$. 251–281. https://doi.org/10.1007/978-3-030-17653-2_9

[48] Aayush Jain, Huijia Lin, and Amit Sahai. 2020. Indistinguishability Obfuscation from Well-Founded Assumptions. Cryptology ePrint Archive, Report 2020/1003. https://eprint.iacr.org/2020/1003.

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

[50] Ilan Komargodski, Tal Moran, Moni Naor, Rafael Pass, Alon Rosen, and Eylon Yogev. 2014. One-Way Functions and (Im)Perfect Obfuscation. 374–383. https://doi.org/10.1109/FOCS.2014.47

[51] Ilan Komargodski, Moni Naor, and Eylon Yogev. 2014. Secret-Sharing for NP. 254–273. https://doi.org/10.1007/978-3-662-45608-8_14

[52] Venkata Koppula, Allison Bishop Lewko, and Brent Waters. 2015. Indistinguishability Obfuscation for Turing Machines with Unbounded Memory. 419–428. https://doi.org/10.1145/2746539.2746614

[53] Venkata Koppula, Kim Ramchen, and Brent Waters. 2015. Separations in Circular Security for Arbitrary Length Key Cycles. 378–400. https://doi.org/10.1007/978-3-662-46497-7_15

[54] Venkata Koppula and Brent Waters. 2016. Circular Security Separations for Arbitrary Length Cycles from LWE. 681–700. https://doi.org/10.1007/978-3-662-53008-5_24

[55] Huijia Lin. 2016. Indistinguishability Obfuscation from Constant-Degree Graded Encoding Schemes. 28–57. https://doi.org/10.1007/978-3-662-49890-3_2

[56] Huijia Lin. 2017. Indistinguishability Obfuscation from SXDH on 5-Linear Maps and Locality-5 PRGs. 599–629. https://doi.org/10.1007/978-3-319-63688-7_20
[57] Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. 2016. Indistinguishability Obfuscation with Non-trivial Efficiency. 447–462. https://doi.org/10.1007/978-3-662-49387-8_17
[58] Huijia Lin and Stefano Tessaro. 2017. Indistinguishability Obfuscation from Trilinear Maps and Block-Wise Local PRGs. 630–660. https://doi.org/10.1007/978-3-319-63688-7_21
[59] Huijia Lin and Vinod Vaikuntanathan. 2016. Indistinguishability Obfuscation from DDH-Like Assumptions on Constant-Degree Graded Encodings. 11–20. https://doi.org/10.1109/FOCS.2016.11
[60] Mohammad Mahmoody, Ameer Mohammed, and Soheil Nematihaji. 2015. More on Impossibility of Virtual Black-Box Obfuscation in Idealized Models. *IACR Cryptology ePrint Archive* 2015 (2015), 632. http://eprint.iacr.org/2015/632
[61] Antonio Marcedone and Claudio Orlandi. 2014. Obfuscation ⇒ (IND-CPA Security ⇏ Circular Security). 77–90. https://doi.org/10.1007/978-3-319-10879-7_5
[62] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. 2004. Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. 21–39. https://doi.org/10.1007/978-3-540-24638-1_2
[63] Daniele Micciancio. 2019. *From linear functions to fully homomorphic encryption. https://bacrypto.github.io/presentations/2018.11.30-Micciancio-FHE.pdf.* Technical Report.
[64] Daniele Micciancio and Chris Peikert. 2012. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. 700–718. https://doi.org/10.1007/978-3-642-29011-4_41
[65] Brice Minaud and Pierre-Alain Fouque. 2015. Cryptanalysis of the New Multilinear Map over the Integers. Cryptology ePrint Archive, Report 2015/941. http://eprint.iacr.org/.
[66] Rafael Pass, Karn Seth, and Sidharth Telang. 2014. Indistinguishability Obfuscation from Semantically-Secure Multilinear Encodings. 500–517. https://doi.org/10.1007/978-3-662-44371-2_28
[67] Rafael Pass and abhi shelat. 2016. Impossibility of VBB Obfuscation with Ideal Constant-Degree Graded Encodings. 3–17. https://doi.org/10.1007/978-3-662-49096-9_1
[68] Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. 2017. Pseudorandomness of ring-LWE for any ring and modulus. 461–473. https://doi.org/10.1145/3055399.3055489
[69] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. 2008. A Framework for Efficient and Composable Oblivious Transfer. 554–571. https://doi.org/10.1007/978-3-540-85174-5_31
[70] Oded Regev. 2005. On lattices, learning with errors, random linear codes, and cryptography. 84–93. https://doi.org/10.1145/1060590.1060603
[71] Ron Rothblum. 2013. On the Circular Security of Bit-Encryption. 579–598. https://doi.org/10.1007/978-3-642-36594-2_32
[72] Amit Sahai and Brent Waters. 2014. How to use indistinguishability obfuscation: deniable encryption, and more. 475–484. https://doi.org/10.1145/2591796.2591825
[73] Huijia Lin Samuel B. Hopkins, Aayush Jain. 2021. Counterexamples to New Circular Security Assumptions Underlying iO. (2021). manuscript.
[74] Hoeteck Wee and Daniel Wichs. 2020. Candidate Obfuscation via Oblivious LWE Sampling. Cryptology ePrint Archive, Report 2020/1042. https://eprint.iacr.org/2020/1042.
[75] Daniel Wichs and Giorgos Zirdelis. 2017. Obfuscating Compute-and-Compare Programs under LWE. 600–611. https://doi.org/10.1109/FOCS.2017.61