# Witness Encryption and Null-IO from Evasive LWE

Vinod Vaikuntanathan* MIT
Hoeteck Wee† NTT Research
Daniel Wichs‡ Northeastern U. and NTT Research

August 31, 2022

**Abstract**

Witness encryption (WE) allows us to use an arbitrary NP statement $x$ as a public key to encrypt a message, and the witness $w$ serves as a decryption key. Security ensures that, when the statement $x$ is false, the encrypted message remains computationally hidden. WE appears to be significantly weaker than indistinguishability obfuscation (iO). Indeed, WE is closely related to a highly restricted form of iO that only guarantees security for null circuits (null iO). However, all current approaches towards constructing WE under nice assumptions go through iO. Such constructions are quite complex and are unlikely to lead to practically instantiable schemes. In this work, we revisit a very simple WE and null iO candidate of Chen, Vaikuntanathan and Wee (CRYPTO 2018). We show how to prove its security under a nice and easy-to-state assumption that we refer to as *evasive LWE* following Wee (EUROCRYPT 2022). Roughly speaking, the evasive LWE assumption says the following: assume we have some joint distributions over matrices $\mathbf{P}$, $\mathbf{S}$ and auxiliary information $\mathsf{aux}$ such that

$$(\mathbf{SB} + \mathbf{E}, \mathbf{SP} + \mathbf{E}', \mathsf{aux}) \approx_c (\mathbf{U}, \mathbf{U}', \mathsf{aux}),$$

for a uniformly random (and secret) matrix $\mathbf{B}$, where $\mathbf{U}, \mathbf{U}'$ are uniformly random matrices, and $\mathbf{E}, \mathbf{E}'$ are chosen from the LWE error distribution with appropriate parameters. Then it must also be the case that:

$$(\mathbf{SB} + \mathbf{E}, \mathbf{B}^{-1}(\mathbf{P}), \mathsf{aux}) \approx_c (\mathbf{U}, \mathbf{B}^{-1}(\mathbf{P}), \mathsf{aux}).$$

Essentially the above says that given $\mathbf{SB} + \mathbf{E}$, getting the additional component $\mathbf{B}^{-1}(\mathbf{P})$ is no more useful than just getting the product $(\mathbf{SB} + \mathbf{E}) \cdot \mathbf{B}^{-1}(\mathbf{P}) \approx \mathbf{SP} + \mathbf{E}'$.

*E-mail:vinodv@mit.edu
†E-mail:wee@di.ens.fr
‡E-mail:wichs@ccs.neu.edu

# 1 Introduction

Witness encryption (WE), a notion introduced by Garg, Gentry, Sahai and Waters [GGSW13], allows us to use an arbitrary NP statement $x$ as a public key to encrypt a message. If $x$ is a true statement then any user who knows the corresponding witness $w$ for $x$ will be able to decrypt the message, but if $x$ is a false statement then the encrypted message is computationally hidden. For example, the Clay Mathematics Institute could encrypt $1M worth of bitcoin reward under the NP statement that corresponds to the Riemann hypothesis. If anyone comes up with such a proof, they can use that as the witness to decrypt the ciphertext and recover the reward.

WE is known to be implied by indistinguishability obfuscation (iO) [BGI+01,GGH+13b]. However, iO appears to be a significantly stronger primitive than WE, and provably so with respect to black-box constructions [GMM17]. On an intuitive level, in WE, we only require functionality (ability to correctly decrypt) in a setting where the statement is true and there are no security guarantees. Conversely, we only require security to hold in a setting where the statement is false and there is no functionality requirement. On the other hand, iO requires us to provide security and functionality simultaneously since the obfuscated program needs to function correctly on all inputs while at the same time hiding the code of the original program. Indeed, modulo the LWE assumption, WE is equivalent to a very weak form of iO, referred to as null-iO, where security (indistinguishability of circuits) only needs to hold for null programs that output 0 on all inputs, while functionality needs to hold for all programs [WZ17, GKW17].

Despite WE being seemingly much weaker than iO, the current state-of-the-art in constructions does not reflect this. In particular, a beautiful series of recent works constructs iO under simple-to-state assumptions [AJL+19, JLMS19, Agr19, GJLS21], culminating in a recent break-through that bases iO on well-studied hardness assumptions [JLS21b, JLS21a]. Another recent line of works obtains lattice-inspired iO candidates [Agr19, CHVW19, AP20, BDGM20a, WW21, GP21, BDGM20b, DQV+21] that avoid the use of pairings and are plausibly post-quantum secure, but requires less well-studied assumptions pertaining to variants of LWE with leakage. Both of these routes to iO also incur high computational complexity due to the use of non-black-box recursion (following [BV15, AJ15]), making it almost unimaginable that they could be implemented even for the simplest of programs. Unfortunately, the only known avenue for constructing WE under similarly nice assumptions goes through iO and inherits all of its corresponding complexity.

In this work, we turn our attention to earlier frameworks for constructing iO and witness encryption [GGH+13b, GGSW13, GLW14]: encode the corresponding program or NP instance, represented as a branching programs[1], using multi-linear encodings [GGH13a, CLT15, GGH15]. The ensuing schemes are remarkably simple, direct, reasonably efficient (e.g., implemented in [HHSS17]), and could even achieve plausible post-quantum security. Unfortunately, we have attacks on the iO schemes for read-$c$ branching programs for $c = O(1)$ [CHL+15, MSZ16, CLLT16, ADGM17, CLLT17, CGH17, Pel18, CVW18, CCH+19]. On the other hand, none of these attacks are applicable to the WE schemes.

Arguably the simplest of these WE schemes is due to Chen, Vaikuntanathan and Wee [CVW18] (henceforth CVW) based on GGH15 multi-linear encodings [GGH15, CC17]. It only relies on LWE-style tools/algebra and is very simple to write down, with complexity similar to the iO candidate for read-once branching programs implemented in [HHSS17]. We do not currently know any attacks on the CVW WE scheme, nor do we know how to base it on any nice assumption, other

[1] For iO, we need some additional pre-processing to prevent mixed-input attacks; see Section 7.

than just tautologically assuming its security. This motivates the main question of this work:

> Question: Can we prove security of the CVW scheme for WE (or a variant thereof) under a simple assumption?

For the optimist, the assumption would ideally increase our confidence in the security of the CVW scheme and give us a better understanding of the basis of this security. For the skeptic, the assumption would constitute a simpler and easier target for cryptanalysis. More broadly, the assumption could provide new insights into the security and weakness of GGH15 multi-linear encodings, extending the positive results in [CC17, GKW17, WZ17, GKW18, CVW18].

### 1.1 Our Results

We prove the security of the CVW schemes for WE and null-IO under a variant of Wee's evasive LWE assumption [Wee22], together with LWE with subexponential hardness. We analyze the CVW schemes essentially "as is", with some modifications to the underlying parameters. We proceed to state the assumption and then provide an overview of our security proof.

**Evasive LWE.** Fix some efficiently samplable distributions $(\mathbf{S}, \mathbf{P}, \mathsf{aux})$ over $\mathbb{Z}_q^{n' \times n} \times \mathbb{Z}_q^{n \times t} \times \{0,1\}^*$. We would like to assert statements of the form

$$(\boxed{\mathbf{SB} + \mathbf{E}}, \mathbf{B}^{-1}(\mathbf{P}), \mathsf{aux}) \approx_c (\boxed{\mathbf{C}}, \mathbf{B}^{-1}(\mathbf{P}), \mathsf{aux})$$

where $\mathbf{B} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{C} \leftarrow \mathbb{Z}_q^{n' \times m}$ are uniformly random. Think of parameters $O(n \log q) \le m \le t$, so that $\mathbf{P}$ is wider than $\mathbf{B}$. We have two distinguishing strategies in the literature:

- distinguish $\mathbf{SB} + \mathbf{E}$ from $\mathbf{C}$ given $\mathsf{aux}$;
- compute $(\mathbf{SB} + \mathbf{E}) \cdot \mathbf{B}^{-1}(\mathbf{P}) \approx \mathbf{SP}$ and distinguish the latter from uniform (the aforementioned zeroizing attacks on iO fall into this category).

The evasive LWE assumption essentially asserts that these are the only distinguishing attacks. Namely,

$$\textbf{if} \qquad (\boxed{\mathbf{SB} + \mathbf{E}}, \boxed{\mathbf{SP} + \mathbf{E}'}, \mathsf{aux}) \approx_c (\boxed{\mathbf{C}}, \boxed{\mathbf{C}'}, \mathsf{aux}), \tag{1}$$

$$\textbf{then} \qquad (\boxed{\mathbf{SB} + \mathbf{E}}, \mathbf{B}^{-1}(\mathbf{P}), \mathsf{aux}) \approx_c (\boxed{\mathbf{C}}, \mathbf{B}^{-1}(\mathbf{P}), \mathsf{aux}) \tag{2}$$

where $\mathbf{E}'$ is a fresh noise matrix of sufficiently larger magnitude than $\mathbf{E}$.[2] In [Wee22] (c.f. Section 1.3), the assumption is conceptually similar, but the matrix $\mathbf{B}$ is public and $\mathbf{S}$ is secret and uniformly random, while in our case both $\mathbf{B}, \mathbf{S}$ are secret and $\mathbf{S}$ can be chosen from an arbitrary distribution subject to (1) holding.

First, to give some intuition for the assumption, we begin with two quick sanity checks:

[2] Note that $(\mathbf{SB} + \mathbf{E}) \cdot \mathbf{B}^{-1}(\mathbf{P})$ has rank at most $m$ and therefore cannot be pseudorandom whenever $n', m < t$. Instead, we merely require that the high-order bits of $(\mathbf{SB} + \mathbf{E}) \cdot \mathbf{B}^{-1}(\mathbf{P}) \approx \mathbf{SP}$ are pseudorandom, as formalized by $\mathbf{SP} + \mathbf{E}'$ being pseudorandom.

- If $\mathbf{P}$ is drawn from the uniform distribution over $\mathbb{Z}_q^{m\times t}$, then the evasive LWE assumption holds unconditionally. In particular, (2) follows unconditionally from (1), since $\mathbf{B}^{-1}(\mathbf{P})$ is distributed like a random Gaussian and hence can be simulated without knowing a trapdoor for $\mathbf{B}$. This is the case even if aux can depend on $\mathbf{P}$, as long as it is efficiently sampleable given $\mathbf{P}$.

- If $\mathbf{P} = \mathbf{0}$ or $\mathbf{P}$ is the gadget matrix, then both the pre- and post-conditions are false, so evasive LWE is vacuously unconditionally true.

We will need to rely on a version of evasive LWE where $\mathbf{P}$ is not uniformly random, but we still manage to ensure that (1) holds. We use the evasive LWE assumption to argue that (2) holds in this case as well.

Unfortunately, we show that the evasive LWE assumption is unlikely to hold in its completely full generality with arbitrary aux. In particular, we cook up a highly contrived auxiliary info aux that contains a carefully crafted obfuscated program (containing a trapdoor for $\mathbf{P}$). Under a heuristic obfuscation assumption, we show that for this choice of aux, the pre-condition holds, while the post-condition is clearly violated. This is similar in spirit to the implausibility of differing-inputs obfuscation (diO) with general auxiliary information, as shown in [GGHW14]. See Section 8.2 for details of our counter-example. Nevertheless, analogously to the case of diO, it is still reasonable to assume security with *essentially any* "natural distribution" of aux that is not specifically cooked up to contain a counter-example. This is the route we take in this work. In addition, we also describe in Section 8.2 a class of distributions that are sufficient for our proofs and seem to avoid obfuscated-based counter-examples.

We note that evasive LWE is qualitatively different from the LWE with leakage assumptions used in recent lattice-inspired iO candidates. With the latter, a distinguisher can easily obtain equations of the LWE secrets over the *integers* (which in turn allows zeroizing attacks), whereas the pre-condition in evasive LWE essentially rules this out. Indeed, the variants of LWE with leakage used in [GP21, WW21] have since been broken in [HJL21], whereas the ones in [DQV+21, JLMS19] rely on the pseudorandomness of structured low-degree polynomials over the integers which while plausible, still requires further cryptanalysis (e.g. we do not know how to rule out sum-of-squares attacks, even heuristically).

**WE and null-IO via GGH15 encodings.** We consider a read-once branching program (BP) specified by values $\mathbf{u} \in \{0,1\}^w, \{\mathbf{M}_{i,b} \in \{0,1\}^{w\times w}\}_{i\in[h],b\in\{0,1\}}$. On input $\mathbf{x} \in \{0,1\}^h$ we define $\mathbf{M}_\mathbf{x} := \prod_{i=1}^h \mathbf{M}_{i,x_i}$, and the output of the branching program is determined by $\mathbf{u}\mathbf{M}_\mathbf{x} \stackrel{?}{=} \mathbf{0}$. (Note that the matrices $\mathbf{M}_{i,b}$ are not necessarily permutations.) The GGH15 encoding of such a branching program $\mathsf{ggh.encode}^{\otimes}(\mathbf{u}, \{\mathbf{M}_{i,b}\})$ is given by

$$\left\{ \underbrace{(\mathbf{u}\mathbf{M}_{1,b} \otimes \mathbf{S}_{1,b})\mathbf{A}_1}, \mathbf{A}_1^{-1}(\underbrace{(\mathbf{M}_{2,b} \otimes \mathbf{S}_{2,b})\mathbf{A}_2}), \ldots, \mathbf{A}_{h-1}^{-1}(\underbrace{(\mathbf{M}_{h,b} \otimes \mathbf{S}_{h,b})\mathbf{A}_h}) \right\}_{b\in\{0,1\}}$$

where $\mathbf{S}_{i,b} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{n\times n}, \mathbf{A}_{i,b} \leftarrow \mathbb{Z}_q^{nw\times O(nw\log q)}$ and we use $\mathbf{A}_{i,b}^{-1}(\cdot)$ to denote random Gaussian preimages, and we use curly underlines in place of noise terms. Given the encoding and any $\mathbf{x} \in \{0,1\}^h$, we can approximate $(\mathbf{u}\mathbf{M}_\mathbf{x} \otimes \mathbf{S}_\mathbf{x})\mathbf{A}_h$ where $\mathbf{M}_\mathbf{x} := \prod_{i=1}^h \mathbf{M}_{i,x_i}, \mathbf{S}_\mathbf{x} := \prod_{i=1}^h \mathbf{S}_{i,x_i}$, and therefore check if $\mathbf{u}\mathbf{M}_\mathbf{x} \stackrel{?}{=} \mathbf{0}$.

For WE, we can embed a CNF formula and the message into the read-once BP, and $\mathbf{x}$ corresponds to a truth assignment. For null-IO, we can take an arbitrary branching program or a NC1 formula and perform some additional pre-processing on it to convert it into a read-once BP, and $\mathbf{x}$ corresponds to a repetition-encoding of the input to the program/formula. In either case, the way we do this ensures that, if the formula is unsatisfiable or the program is a null program, it will be the case that $\mathbf{u}\mathbf{M}_{\mathbf{x}} \neq \mathbf{0}$ for all $\mathbf{x} \in \{0,1\}^h$. (In the case of null iO, this will hold even for values $\mathbf{x}$ that are not valid repetition-encodings of any input.) We show what whenever this condition holds, $\mathsf{ggh.encode}^{\otimes}(\mathbf{u}, \{\mathbf{M}_{i,b}\})$ is pseudorandom and therefore hides $\mathbf{u}, \{\mathbf{M}_{i,b}\}$. The latter immediately implies security of the CVW schemes for WE and null-IO. We sketch the proof of this statement in our technical overview.

**Concurrent Independent Work.** The concurrent and independent work of Tsabary [Tsa22] gives a similar construction of witness encryption and shows security under a variant of evasive LWE, via a similar proof strategy. See Appendix A for a comparison.

## 1.2 Technical Overview

The technical core of this work lies proving the following statement:

**Theorem 1.1** (informal). *Suppose subexponential LWE and evasive LWE holds. If* $\mathbf{u}\mathbf{M}_{\mathbf{x}} \neq \mathbf{0}$ *for all* $\mathbf{x} \in \{0,1\}^h$*, then*

$$\mathsf{ggh.encode}^{\otimes}(\mathbf{u}, \{\mathbf{M}_{i,b}\}) \approx_c \{\mathbf{C}_{1,b}, \mathbf{D}_{2,b}, \ldots, \mathbf{D}_{h,b}\}_{b\in\{0,1\}}$$

*where* $\mathbf{C}_{1,b} \leftarrow \mathbb{Z}_q^{n\times O(nw\log q)}, \mathbf{D}_{i,b} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{O(nw\log q)\times O(nw\log q)}$.

As a warm-up, we prove security under a strengthening of evasive LWE where we omit $\mathbf{SB}+\mathbf{E}$ in the pre-condition, namely we assume:

$$\begin{aligned} &\textbf{if} && (\mathbf{SP}+\mathbf{E}', \mathsf{aux}) \approx_c (\boxed{\mathbf{C}'}, \mathsf{aux}), \\ &\textbf{then} && (\boxed{\mathbf{SB}+\mathbf{E}}, \mathbf{B}^{-1}(\mathbf{P}), \mathsf{aux}) \approx_c (\boxed{\mathbf{C}}, \mathbf{B}^{-1}(\mathbf{P}), \mathsf{aux}) \end{aligned}$$

Intuitively, evasive LWE says that to prove pseudorandomness of $(\mathbf{SB}+\mathbf{E}, \mathbf{B}^{-1}(\mathbf{P}))$, it suffices to "peel off" $\mathbf{B}$ and prove pseudorandomness of the product $\mathbf{SP}+\mathbf{E}'$. Our proof essentially proceeds in two steps:

- We will use evasive LWE to successively "peel off" $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_{h-1}$ in our "encoded program", which leaves us with $2^h$ products $\{(\mathbf{u}\mathbf{M}_{\mathbf{x}} \otimes \mathbf{S}_{\mathbf{x}})\mathbf{A}_h\}_{\mathbf{x}\in\{0,1\}^h}$.
- We then show that these $2^h$ evaluated products are pseudorandom under the LWE assumption, following the BLMR PRF [BLMR13].

We proceed to describe this in more detail.

**Proof idea.** Suppose instead of getting the full "encoded program"

$$\left\{(\underbrace{\mathbf{u}\mathbf{M}_{1,b}\otimes\mathbf{S}_{1,b})\mathbf{A}_1}, \mathbf{A}_1^{-1}(\underbrace{(\mathbf{M}_{2,b}\otimes\mathbf{S}_{2,b})\mathbf{A}_2}), \ldots, \mathbf{A}_{h-1}^{-1}(\underbrace{(\mathbf{M}_{h,b}\otimes\mathbf{S}_{h,b})\mathbf{A}_h})\right\}_{b\in\{0,1\}},$$

we were only given the $2^h$ "evaluated products" (with fresh independent errors):

$$\left\{\underbrace{(\mathbf{u}\mathbf{M}_{\mathbf{x}}\otimes\mathbf{S}_{\mathbf{x}})\mathbf{A}_h}\right\}_{\mathbf{x}\in\{0,1\}^h},$$

which is something we could approximate from evaluating the encoded program on all inputs $\mathbf{x}\in\{0,1\}^h$.[3]

First, by the same security analysis as the BLMR PRF [BLMR13], we can rely on (sub-exponential) LWE to show that such "evaluated products" look pseudorandom. In particular, we have

$$\underbrace{(\mathbf{u}\mathbf{M}_{\mathbf{x}}\otimes\mathbf{S}_{\mathbf{x}})\mathbf{A}_h}\approx\overbrace{(\mathbf{u}\mathbf{M}_{\mathbf{x}}\otimes\mathbf{I})}^{\neq\mathbf{0}}\cdot\overbrace{\underbrace{(\mathbf{I}\otimes\mathbf{S}_{\mathbf{x}})\mathbf{A}_h}}^{\text{pseudorandom}}.$$

Next, we rely on evasive LWE with $\mathbf{B}=\mathbf{A}_{h-1}$ to show that if we were given

$$\left\{\underbrace{(\mathbf{u}\mathbf{M}_{\mathbf{x}}\otimes\mathbf{S}_{\mathbf{x}})\mathbf{A}_{h-1}}\right\}_{\mathbf{x}\in\{0,1\}^{h-1}}, \left\{\mathbf{A}_{h-1}^{-1}(\underbrace{(\mathbf{M}_{h,b}\otimes\mathbf{S}_{h,b})\mathbf{A}_h})\right\}_{b\in\{0,1\}}$$

corresponding to $2^{h-1}$ "evaluated products" for all possible choices of the first $h-1$ bits of the input and the last two components of the "encoded program", the $2^{h-1}$ "evaluated products" would still look pseudorandom.

We repeat the argument with $\mathbf{B}=\mathbf{A}_{h-2},\ldots,\mathbf{A}_1$ until we show that just the first 2 "evaluated products" $\left\{\underbrace{(\mathbf{u}\mathbf{M}_{1,b}\otimes\mathbf{S}_{1,b})\mathbf{A}_1}\right\}_{b\in\{0,1\}}$ look pseudorandom even given all the remaining components of the "encoded program". But the first 2 "evaluated products" are just the first two components of the "encoded program" and hence we can replace them by uniformly random matrices $\{\mathbf{C}_{1,b}\}_{b\in\{0,1\}}$ At this point, we can invoke an argument from [CVW18] to replace the subsequent components of the encoded program by uniformly random Gaussians to complete the proof.[4]

[3] While a polynomial-time adversary cannot evaluate the encoded program on all $2^h$ inputs, it can still efficiently approximate some linear combination of an exponential number of inputs, e.g. the sum of all $2^h$ evaluated products, using $(\underbrace{(\mathbf{u}\mathbf{M}_{1,0}\otimes\mathbf{S}_{1,0})\mathbf{A}_1}+\underbrace{(\mathbf{u}\mathbf{M}_{1,1}\otimes\mathbf{S}_{1,1})\mathbf{A}_1})\cdot\prod_{i=2}^{h}(\mathbf{A}_{i-1}^{-1}(\underbrace{(\mathbf{M}_{i,0}\otimes\mathbf{S}_{i,0})\mathbf{A}_i})+\mathbf{A}_{i-1}^{-1}(\underbrace{(\mathbf{M}_{i,1}\otimes\mathbf{S}_{i,1})\mathbf{A}_i}))$.

[4] The above proof strategy forces us to rely on LWE with sub-exponential security for two distinct reasons. Firstly, in the base case, we rely on LWE with $2^h$ terms. Secondly, we rely on $h$ levels of induction, where each level of the induction incurs a polynomial security loss.

**Example for $h = 3$.** In more detail, let's see an example for $h = 3$. In that case, the proof shows:

$$\begin{aligned}
&\left\{(\underbrace{\mathbf{u}\mathbf{M}_{\mathbf{x}} \otimes \mathbf{S}_{\mathbf{x}})\mathbf{A}_3}\right\}_{\mathbf{x}\in\{0,1\}^3} \approx_c \left\{\mathbf{C}_{\mathbf{x}} \leftarrow \mathbb{Z}_q^{n\times O(nw\log q)}\right\}_{\mathbf{x}\in\{0,1\}^3} \\
\implies \quad &\left\{(\underbrace{\mathbf{u}\mathbf{M}_{\mathbf{x}'} \otimes \mathbf{S}_{\mathbf{x}'})\mathbf{A}_2}\right\}_{\mathbf{x}'\in\{0,1\}^2}, \left\{\mathbf{A}_2^{-1}(\underbrace{(\mathbf{M}_{3,b} \otimes \mathbf{S}_{3,b})\mathbf{A}_3})\right\}_{b\in\{0,1\}} \\
&\approx_c \left\{\mathbf{C}_{\mathbf{x}'} \leftarrow \mathbb{Z}_q^{n\times O(nw\log q)}\right\}_{\mathbf{x}'\in\{0,1\}^2}, \left\{\mathbf{A}_2^{-1}(\underbrace{(\mathbf{M}_{3,b} \otimes \mathbf{S}_{3,b})\mathbf{A}_3})\right\}_{b\in\{0,1\}} \\
\implies \quad &\left\{\underbrace{(\mathbf{u}\mathbf{M}_b \otimes \mathbf{S}_b)\mathbf{A}_1}, \mathbf{A}_1^{-1}(\underbrace{(\mathbf{M}_{2,b} \otimes \mathbf{S}_{2,b})\mathbf{A}_2}), \mathbf{A}_2^{-1}(\underbrace{(\mathbf{M}_{3,b} \otimes \mathbf{S}_{3,b})\mathbf{A}_3})\right\}_{b\in\{0,1\}} \\
&\approx_c \left\{\mathbf{C}_{1,b}, \mathbf{A}_1^{-1}(\underbrace{(\mathbf{M}_{2,b} \otimes \mathbf{S}_{2,b})\mathbf{A}_2}), \mathbf{A}_2^{-1}(\underbrace{(\mathbf{M}_{3,b} \otimes \mathbf{S}_{3,b})\mathbf{A}_3})\right\}_{b\in\{0,1\}} \\
&\approx_c \{\mathbf{C}_{1,b}, \mathbf{D}_{2,b}, \mathbf{D}_{3,b}\}_{b\in\{0,1\}}
\end{aligned}$$

The first statement uses subexponential LWE, and uses security of the BLMR PRF [BLMR13] (as described earlier) asserting pseudorandomness of the set of values $\left\{\underbrace{(\mathbf{I} \otimes \mathbf{S}_{\mathbf{x}})\mathbf{A}_3}\right\}_{\mathbf{x}\in\{0,1\}^3}$, together with the condition $\mathbf{u}\mathbf{M}_{\mathbf{x}} \neq \mathbf{0}$ for all $\mathbf{x}$. The next two $\implies$ corresponds to invocations of evasive LWE. In particular, for the second $\implies$, we invoke the assumption with:

$$\begin{aligned}
\mathbf{S} &= \begin{pmatrix} \mathbf{u}\mathbf{M}_{1,0} \otimes \mathbf{S}_{1,0} \\ \mathbf{u}\mathbf{M}_{1,1} \otimes \mathbf{S}_{1,1} \end{pmatrix} \\
\mathbf{P} &= [\underbrace{(\mathbf{M}_{2,0} \otimes \mathbf{S}_{2,0})\mathbf{A}_2} \| \underbrace{(\mathbf{M}_{2,1} \otimes \mathbf{S}_{2,1})\mathbf{A}_2}] \\
\mathsf{aux} &= \left\{\mathbf{A}_2^{-1}(\underbrace{(\mathbf{M}_{3,b} \otimes \mathbf{S}_{3,b})\mathbf{A}_3})\right\}_{b\in\{0,1\}}
\end{aligned}$$

For this step, we will actually additionally need to use noise flooding to prove the pre-condition. As a result, the noise parameter in $\underbrace{(\mathbf{u}\mathbf{M}_{1,b} \otimes \mathbf{S}_{1,b})\mathbf{A}_1}$ is going to much be larger than that in $\mathbf{A}_{j-1}^{-1}(\underbrace{(\mathbf{M}_{j,b} \otimes \mathbf{S}_{j,b})\mathbf{A}_j}), j = 2, \ldots, h$. The final $\approx_c$ follows from an argument used in [CVW18], repeatedly applying $\mathbf{A}^{-1}(\underbrace{\mathbf{Z}}) \approx_c \mathbf{D}$ (follows from LWE) from "left to right".

**Tying up the loose ends.** More generally, we invoke evasive LWE $h - 1$ times, where each statement contains up to $2^h$ terms, so the size of the evasive LWE instances are as large as $2^h \cdot \mathrm{poly}(\lambda)$. With each invocation of evasive LWE, we also incur a multiplicative polynomial security loss (in the size of the instance), and therefore our total security loss is $(2^h \cdot \mathrm{poly}(\lambda))^{O(h)}$.

To extend the argument to the setting where $\mathbf{SB} + \mathbf{E}$ is also provided in the pre-condition, we observe that

$$\begin{aligned}
&(\mathbf{SB} + \mathbf{E}, \mathbf{S}, \mathsf{aux}) \approx_c (\mathbf{C}, \mathbf{S}, \mathsf{aux}) \ \wedge \ (\mathbf{SP} + \mathbf{E}', \mathbf{S}, \mathsf{aux}) \approx_c (\mathbf{C}', \mathbf{S}, \mathsf{aux}) \\
\implies &(\mathbf{SB} + \mathbf{E}, \mathbf{SP} + \mathbf{E}', \mathbf{S}, \mathsf{aux}) \approx_c (\mathbf{SB} + \mathbf{E}, \mathbf{C}', \mathbf{S}, \mathsf{aux}) \approx_c (\mathbf{C}, \mathbf{C}', \mathbf{S}, \mathsf{aux})
\end{aligned}$$

This allows us to treat pseudorandomness of $\mathbf{SB} + \mathbf{E}$ and that of $\mathbf{SP} + \mathbf{E}'$ separately, where the former will rely on security of the BLMR PRF (which holds even if the distinguisher gets

$\{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}}$) and the latter uses the argument as before. This step is important as it captures the fact that the adversary can in fact compute $2^{2h}-1$ evaluated products $\{(\mathbf{u}\mathbf{M}_{\mathbf{x}'}\otimes\mathbf{S}_{\mathbf{x}'})\mathbf{A}_j\}_{\mathbf{x}'\in\{0,1\}^j,j\in[h]}$ corresponding to all possible prefixes $\mathbf{x}'$ of length at most $h$.

### 1.3 Discussion

**Comparison with [Wee22].** Wee's evasive LWE assumption in [Wee22] considers distributions over pairs of matrices $(\mathbf{A}', \mathbf{P})$ together with auxiliary input aux and stipulates that

$$\textbf{if} \quad (\mathbf{A}', \mathbf{B}, \mathbf{P}, \boxed{\mathbf{s}\mathbf{A}+\mathbf{e}'}, \boxed{\mathbf{s}\mathbf{B}+\mathbf{e}}, \boxed{\mathbf{s}\mathbf{P}+\mathbf{e}''}, \mathsf{aux}) \approx_c (\mathbf{A}', \mathbf{B}, \mathbf{P}, \mathbf{c}', \mathbf{c}, \mathbf{c}'', \mathsf{aux}),$$

$$\textbf{then} \quad (\mathbf{A}', \mathbf{B}, \boxed{\mathbf{s}\mathbf{A}+\mathbf{e}'}, \boxed{\mathbf{s}\mathbf{B}+\mathbf{e}}, \mathbf{B}^{-1}(\mathbf{P}), \mathsf{aux}) \approx_c (\mathbf{A}', \mathbf{B}, \mathbf{c}', \mathbf{c}, \mathbf{B}^{-1}(\mathbf{P}), \mathsf{aux})$$

For the applications in [Wee22], the auxiliary input includes the coin tosses used to sample $\mathbf{A}', \mathbf{P}$, which rules out obfuscation-based counter-examples.

In [Wee22], evasive LWE was used to build ciphertext-policy ABE for circuits and optimal broadcast encryption schemes. The schemes are very different from the ones analyzed and in particular, do not rely on GGH15 encodings. The techniques are also quite different: in [Wee22], evasive LWE is only invoked once, whereas in this work, we invoke evasive LWE $h$ times. For ease of comparison, we reproduce the informal description of the CP-ABE scheme described in [Wee22, Section 2.1] below:

$$\begin{aligned}
\mathsf{mpk} &:= \mathbf{A}_0, \mathbf{B}_0 \leftarrow \mathbb{Z}_q^{n\times m}, \mathbf{B}_1 \leftarrow \mathbb{Z}_q^{mn\times m^2}, \mathbf{A} \leftarrow \mathbb{Z}_q^{n\times \ell m} \\
\mathsf{ct}_f &:= \mathbf{s}_0\mathbf{B}_0,\ \mathbf{s}(\mathbf{A}_f\otimes\mathbf{I}_m)+\mathbf{s}_0\mathbf{A}_0+\mu\cdot\mathbf{g},\ \mathbf{s}\mathbf{B}_1,\ \text{where } \mathbf{s}\leftarrow\mathbb{Z}_q^{mn}, \mathbf{s}_0\leftarrow\mathbb{Z}_q^n \\
\mathsf{sk}_{\mathbf{x}} &:= \mathbf{B}_0^{-1}(\mathbf{A}_0\mathbf{r}), \mathbf{B}_1^{-1}((\mathbf{A}-\mathbf{x}\otimes\mathbf{G})\otimes\mathbf{r}), \mathbf{r},\ \text{where } \mathbf{r}\leftarrow\mathcal{D}_{\mathbb{Z},\chi}^m
\end{aligned}$$

**WE proof strategies.** It is instructive to compare our proof strategy with that for WE in [GLW14] (henceforth GLW) based on static assumptions over multi-linear encodings; unfortunately, existing candidate instantiations for these assumptions are broken due to zeroizing attacks. Our proof uses $O(h)$ hybrids and evasive LWE instances of size $2^h\cdot\mathrm{poly}(\lambda)$, whereas the GLW proof, based on the notion of positional WE, uses $2^h$ hybrids and problem instances of size $\mathrm{poly}(\lambda)$.

**Zeroizing attacks and iO vs WE.** What iO and WE have in common is that they require handling an exponential number of possible evaluations for both correctness and security. A key difficulty in constructing post-quantum iO arises from the fact that all known approaches yield schemes in the *zeroizing regime* [CHL+15] wherein an attacker can easily obtain sufficiently many equations in low-norm secret values —low-norm LWE secrets, error vectors, or both— over the integers that information-theoretically determine these secret values.[5] These equations arise naturally from the interaction of the correctness constraints and the security requirements, and could in turn be exploited to yield a *zeroizing attack* on the scheme [MSZ16, CLLT16, ADGM17, CLLT17, CGH17, Pel18, CVW18, CCH+19, HJL21]. In order to rule out zeroizing attacks, current approaches to

[5] As a point of comparison, we have examples such as $k$-LWE [LPSS14] and inner product functional encryption [ALS16] based on LWE where it is easy to obtain a few such equations, but the equations do *not* information-theoretically determine the secret values.

post-quantum iO rely on some form of pseudorandomness of low-norm values over the integers [AJL+19,Agr19,CHVW19] to argue that the leakages in the *zeroizing regime* do not lend themselves to an attack. As mentioned earlier in the introduction, the evasive LWE assumption is qualitatively different from these assumptions as it does not refer to pseudorandomness of low-norm values.

**Weak multi-linear map models.** Prior works analyzed security of iO and WE candidates in the so-called weak multi-linear map models, e.g. [GMM+16, BGMZ18, CHVW19]. Most of these models (notably [CVW18, Section 11.3] and [CHVW19]) immediately yield a statement similar to Lemma 5.2 (used in the proof of Theorem 1.1), whereas our proof of Lemma 5.2 from evasive LWE requires a careful inductive argument combined with noise flooding.

**On security losses.** The CVW18-type schemes are the most promising (and currently only) approach towards practical witness encryption, which begs the question: are the schemes secure and the underlying design principle sound? Towards answering these questions, we follow the cryptographic tradition of relating the security of the schemes to a simpler assumption. As is often the case, the parameters we achieve in our security reduction are far from practical. Nonetheless, they constitute some evidence that the underlying design is indeed sound, and the first step in a broader research agenda. Indeed, many NIST post-quantum candidates and the sub-field of "tight security" (e.g. for TLS 1.3) start with provably secure schemes with poor parameters, and the parameters for practical instantiations are based on the best-known attacks on the scheme and often more aggressive than the parameters given by the security reduction.

**Looking ahead.** Looking ahead, we see 4 possible scenarios, starting with the most optimistic and ambitious:

1. In a few years, we have witness encryption based on LWE, as has been the case for several lattice-based schemes where the initial candidates were based on non-standard assumptions (outside the zeroizing regime), such as fully homomorphic encryption and its multi-key variant, attribute-based encryption and predicate encryption, and the Fiat-Shamir heuristic. If so, we hope that the insights and techniques developed in this work play a small role, but even if not, the ensuing witness encryption scheme will almost certainly be substantially more complex than the CVW scheme. This would place us in a world analogous to the state of the art for discrete-log and pairing-based cryptography: while we do have fairly efficient schemes based on standard assumptions like DDH, the most practical schemes as well as the ones being deployed are often the ones for which we only know how to prove security in the generic group model, possibly augmented with random oracles.

2. The evasive LWE assumption survives cryptanalysis: this gives us confidence in the CVW18 WE, and the techniques in this work would likely further enable other cryptographic constructions based on evasive LWE as well as GGH15 multi-linear encodings.

3. The evasive LWE assumption is broken but the CVW18 WE scheme is not. This would require new and valuable cryptanalytic advances beyond the state-of-the-art zeroizing attacks. The current statement of evasive LWE is fairly general, and an attack could guide us towards identifying restricted variants of the assumption that would suffice for our analysis of the CVW18 scheme and more generally yield new insights into GGH15 multi-linear encodings.

4. The CVW18 scheme (and thus evasive LWE) is broken. This would be a fairly exciting result in cryptanalysis, and we hope that our statement of evasive LWE plays an important role as an intermediate and easier target for cryptanalysis.

We believe any of these scenarios would advance our current scientific understanding of lattice-based cryptography and assumptions (hardness and/or attacks).

# 2 Preliminaries

**Notations.** We use boldface lower case for row vectors (e.g. $\mathbf{v}$) and boldface upper case for matrices (e.g. $\mathbf{V}$). For integral vectors and matrices (i.e., those over $\mathbb{Z}$), we use the notation $|\mathbf{v}|, |\mathbf{V}|$ to denote the maximum absolute value over all the entries. We use $v \leftarrow \mathcal{D}$ to denote a random sample from a distribution $\mathcal{D}$, as well as $v \leftarrow S$ to denote a uniformly random sample from a set $S$. We also use $\mathcal{U}(S)$ to denote the uniform distribution over a set $S$. We use $\approx_s$ and $\approx_c$ as the abbreviation for statistically close and computationally indistinguishable.

**Tensor product.** The tensor product (Kronecker product) for matrices $\mathbf{A} = (a_{i,j}) \in \mathbb{Z}^{\ell \times m}$, $\mathbf{B} \in \mathbb{Z}^{n \times p}$ is defined as

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{1,1}\mathbf{B}, & \dots, & a_{1,m}\mathbf{B} \\ \dots, & \dots, & \dots \\ a_{\ell,1}\mathbf{B}, & \dots, & a_{\ell,m}\mathbf{B} \end{bmatrix} \in \mathbb{Z}^{\ell n \times mp}.$$

The mixed-product property for tensor product says that

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD})$$

We adopt the convention that matrix multiplication takes precedence over tensor product, so that we can write $\mathbf{A} \otimes \mathbf{BC}$ to mean $\mathbf{A} \otimes (\mathbf{BC})$.

## 2.1 Lattices background

We use $\mathcal{D}_{\mathbb{Z},\chi}$ to denote the discrete Gaussian distribution over $\mathbb{Z}$ with standard deviation $\chi$.

**Learning with errors (LWE).** Given $n, m, q, \chi \in \mathbb{N}$, the $\mathsf{LWE}_{n,m,q,\chi}$ assumption states that

$$(\mathbf{A}, \mathbf{sA} + \mathbf{e}) \approx_c (\mathbf{A}, \mathbf{c})$$

where

$$\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{s} \leftarrow \mathbb{Z}_q^n, \mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^m, \mathbf{c} \leftarrow \mathbb{Z}_q^m$$

We rely on the LWE assumption with sub-exponential hardness (for time, advantage and modulus-to-noise ratio), namely for some $\delta > 0$, indistinguishability holds against adversaries running in time $2^{n^\delta}$ with advantage at most $2^{-n^\delta}$, as long as $q/\chi \leq 2^{n^\delta}$.

**Trapdoor and preimage sampling.** Given any $\mathbf{Z} \in \mathbb{Z}_q^{n \times n'}$, $\sigma > 0$, we use $\mathbf{B}^{-1}(\mathbf{Z}, \sigma)$ to denote the distribution of a matrix $\mathbf{Y}$ sampled from $\mathcal{D}_{\mathbb{Z}^{m \times n'}, \sigma}$ conditioned on $\mathbf{BY} = \mathbf{Z} \pmod q$. We sometimes suppress $\sigma$ when the context is clear.

There is a p.p.t. algorithm $\mathsf{TrapGen}(1^n, q)$ that, given the modulus $q \geq 2$ and dimension $n$, outputs $\mathbf{B} \approx_s U(\mathbb{Z}_q^{n \times 2n \log q})$ with a trapdoor $\tau$. Moreover, there is a p.p.t. algorithm that given $(\mathbf{B}, \tau) \leftarrow \mathsf{TrapGen}(1^n, q)$, $\mathbf{Z} \in \mathbb{Z}_q^{n \times n'}$, and $\sigma \geq 2\sqrt{n \log q}$, outputs a sample from $\mathbf{B}^{-1}(\mathbf{Z}, \sigma)$.

### 2.2 Matrix branching programs

A (matrix) branching program $\Gamma$ with width $w$ and length $h$ is a set

$$\Gamma = \left\{ \mathbf{u} \in \{0,1\}^{1 \times w}, \left\{ \mathbf{M}_{i,b} \in \{0,1\}^{w \times w} \right\}_{i \in [h], b \in \{0,1\}}, \varpi : \{0,1\}^\ell \to \{0,1\}^h \right\}$$

where $w$ is called width of branching program and $\varpi$ an input-to-index function. We say that a branching program $\Gamma$ computes a function $f : \{0,1\}^\ell \to \{0,1\}$ if

$$\forall \mathbf{x} \in \{0,1\}^\ell : \ \mathbf{u} \prod_{i=1}^{h} \mathbf{M}_{i,\varpi(x)} = \mathbf{0} \iff f(\mathbf{x}) = 1$$

For simplicity, we only consider "oblivious" branching programs, where $\varpi : \{0,1\}^\ell \to \{0,1\}^h$ that outputs $h/\ell$ copies of $\mathbf{x}$, i.e. $\varpi(\mathbf{x}) = \mathbf{x}|\mathbf{x}|\cdots|\mathbf{x}$ . We denote $c := h/\ell$ and call this a read-$c$ branching program. For most of the paper, we will focus on read-once branching programs, with $c = 1, \ell = h$ and $\varpi$ being the identity function, and where we write $\mathbf{M}_{\mathbf{x}} := \prod_{i=1}^{h} \mathbf{M}_{i,x_i}$.

## 3 Evasive LWE

We proceed to provide a formal statement of our evasive LWE assumption, stated informally in Section 1.1.

**Evasive LWE.** Let $\mathsf{Samp}$ be a PPT algorithm that on input $1^\lambda$, outputs

$$\mathbf{S} \in \mathbb{Z}_q^{n' \times n}, \mathbf{P} \in \mathbb{Z}_q^{n \times t}, \mathsf{aux} \in \{0,1\}^*$$

We define the following advantage functions:

$$\begin{aligned} \mathsf{Adv}^{\mathrm{PRE}}_{\mathcal{A}_0}(\lambda) &:= \Pr[\mathcal{A}_0(\boxed{\mathbf{SB} + \mathbf{E}}, \boxed{\mathbf{SP} + \mathbf{E}'}, \mathsf{aux}) = 1] \\ &\quad - \Pr[\mathcal{A}_0(\mathbf{C}, \mathbf{C}', \mathsf{aux}) = 1], \qquad (3) \\ \mathsf{Adv}^{\mathrm{POST}}_{\mathcal{A}_1}(\lambda) &:= \Pr[\mathcal{A}_1(\boxed{\mathbf{SB} + \mathbf{E}}, \mathbf{D}, \mathsf{aux}) = 1] \\ &\quad - \Pr[\mathcal{A}_1(\mathbf{C}, \mathbf{D}, \mathsf{aux}) = 1] \qquad (4) \end{aligned}$$

where

$$\begin{aligned} &(\mathbf{S}, \mathbf{P}, \mathsf{aux}) \leftarrow \mathsf{Samp}(1^\lambda), \\ &\mathbf{B} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{E} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{n' \times m}, \mathbf{E}' \leftarrow \mathcal{D}_{\mathbb{Z},\chi'}^{n' \times t}, \\ &\mathbf{C} \leftarrow \mathbb{Z}_q^{n' \times m}, \mathbf{C}' \leftarrow \mathbb{Z}_q^{n' \times t}, \\ &\mathbf{D} \leftarrow \mathbf{B}^{-1}(\mathbf{P}, \chi) \end{aligned}$$

We say that the *evasive LWE* assumption holds if for every PPT Samp there exists some polynomial $Q(\cdot)$ such that for every PPT $\mathcal{A}_1$, there exists another PPT $\mathcal{A}_0$ such that

$$\mathsf{Adv}^{\mathrm{PRE}}_{\mathcal{A}_0}(\lambda) \geq \mathsf{Adv}^{\mathrm{POST}}_{\mathcal{A}_1}(\lambda)/Q(\lambda) - \mathrm{negl}(\lambda)$$

and $\mathsf{Time}(\mathcal{A}_0) \leq \mathsf{Time}(\mathcal{A}_1) \cdot Q(\lambda)$. We consider parameter settings for which $\chi' \ll \chi$ so that the pre-condition is stronger, which in turn makes evasive LWE weaker. See Section 8 for further discussion.

# 4 GGH15 Encodings

We describe (generalized) GGH15 encodings, following [GGH15, CC17, CVW18]. We find it helpful to break down the description into two separate algorithms ggh.encode and ggh.encode$^{\otimes}$. The former is more general, and refers to matrices $\widehat{\mathbf{S}}_{i,b}$, whereas the latter instantiates $\widehat{\mathbf{S}}_{i,b}$ with $\mathbf{M}_{i,b} \otimes \mathbf{S}_{i,b}$.

**Construction 4.1** (GGH15 Encodings)**.** *The randomized algorithm* ggh.encode *takes the following inputs*

- *parameters* $1^\lambda$, $h, m, q, \hat{n}_0, \hat{n} \in \mathbb{N}$ *and Gaussian parameters* $\chi, \chi', \chi'', \chi'''$*;*
- *matrices* $\widehat{\mathbf{S}}_{1,b} \in \mathbb{Z}_q^{\hat{n}_0 \times \hat{n}}, \widehat{\mathbf{S}}_{2,b}, \ldots, \widehat{\mathbf{S}}_{h,b} \in \mathbb{Z}_q^{\hat{n} \times \hat{n}}, b \in \{0,1\}$*;*

*and*

- *samples* $\mathbf{A}_i, \tau_{\mathbf{A}_i} \leftarrow \mathsf{TrapGen}(1^{\hat{n}}, q)$ *for* $i = 1, \ldots, h$*,*
- *samples* $\mathbf{E}_{1,b} \leftarrow \mathcal{D}^{\hat{n}_0 \times m}_{\mathbb{Z},\chi}, \mathbf{E}_{2,b}, \ldots, \mathbf{E}_{h,b} \leftarrow \mathcal{D}^{\hat{n} \times m}_{\mathbb{Z},\chi'''}$ *for* $b \in \{0,1\}$*,*[6]
- *outputs*

$$\left\{\widehat{\mathbf{S}}_{1,b}\mathbf{A}_1 + \mathbf{E}_{1,b}\right\}_{b\in\{0,1\}}, \left\{\mathbf{A}_{i-1}^{-1}(\widehat{\mathbf{S}}_{i,b}\mathbf{A}_i + \mathbf{E}_{i,b})\right\}_{i=2,\ldots,h,b\in\{0,1\}}$$

*where* $\mathbf{A}_{i-1}^{-1}(\cdot)$ *is computed with Gaussian parameter* $\chi''$ *using* $\tau_{\mathbf{A}_{i-1}}$*.*

**Construction 4.2** ($\otimes$-GGH15 Encodings)**.** *The randomized algorithm* ggh.encode$^{\otimes}$ *takes as input*

$$\mathbf{u} \in \{0,1\}^w, \left\{\mathbf{M}_{i,b} \in \{0,1\}^{w\times w}\right\}_{i\in[h],b\in\{0,1\}}$$

*and*

- *samples* $\mathbf{S}_{i,b} \leftarrow \mathcal{D}^{n\times n}_{\mathbb{Z},O(1)}$*,*
- *sets* $\widehat{\mathbf{S}}_{i,b} := \begin{cases} \mathbf{u}\mathbf{M}_{1,b} \otimes \mathbf{S}_{1,b} & \text{if } i = 1 \\ \mathbf{M}_{i,b} \otimes \mathbf{S}_{i,b} & \text{if } i > 1 \end{cases}$
- *outputs* $\mathsf{ggh.encode}(\left\{\widehat{\mathbf{S}}_{i,b}\right\}_{i\in[h],b\in\{0,1\}})$ *with* $\hat{n}_0 = n, \hat{n} = wn$*, i.e.,*

$$\left\{(\mathbf{u}\mathbf{M}_{1,b} \otimes \mathbf{S}_{1,b})\mathbf{A}_1 + \mathbf{E}_{1,b}\right\}_{b\in\{0,1\}}, \left\{\mathbf{A}_{i-1}^{-1}((\mathbf{M}_{i,b} \otimes \mathbf{S}_{i,b})\mathbf{A}_i + \mathbf{E}_{i,b})\right\}_{i=2,\ldots,h,b\in\{0,1\}}$$

[6] Prior works all use $\chi = \chi'''$. Looking ahead, we require $\chi \gg \chi'''$.

**Correctness.** The next lemma from [CVW18, Lemma 5.3] (also [GGH15, CC17]) captures the functionality provided by $\mathsf{ggh.encode}^{\otimes}$, namely for all $\mathbf{x} = (x_1, \ldots, x_h) \in \{0,1\}^h$:

$$\mathbf{C}_{1,x_1} \cdot \mathbf{D}_{2,x_2} \cdots \mathbf{D}_{h,x_h} \approx (\mathbf{u}\mathbf{M}_{\mathbf{x}} \otimes \mathbf{S}_{\mathbf{x}}) \cdot \mathbf{A}_h$$

where $\mathbf{M}_{\mathbf{x}} := \prod_{i=1}^{h} \mathbf{M}_{i,x_i}, \mathbf{S}_{\mathbf{x}} := \prod_{i=1}^{h} \mathbf{S}_{i,x_i}$.

**Lemma 4.3** (Correctness). *We have for all $\mathbf{x} \in \{0,1\}^h$: w.h.p. over*

$$(\mathbf{C}_{1,0}, \mathbf{C}_{1,1}, \mathbf{D}_{2,0}, \mathbf{D}_{2,1}, \ldots, \mathbf{D}_{h,0}, \mathbf{D}_{h,1}) \leftarrow \mathsf{ggh.encode}^{\otimes}(\mathbf{u}, \{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}})$$

*we have*

$$|\mathbf{C}_{1,x_1} \cdot \mathbf{D}_{2,x_2} \cdots \mathbf{D}_{h,x_h} - (\mathbf{u}\mathbf{M}_{\mathbf{x}} \otimes \mathbf{S}_{\mathbf{x}}) \cdot \mathbf{A}_h| \le h \cdot \chi \cdot (\lambda n w(\chi'' + \chi''') \log q)^h$$

# 5 Pseudorandomness of GGH15 Encodings from Evasive LWE

In this section, we prove Theorem 1.1 in the introduction, i.e., pseudorandomness of GGH15 encodings under subexponential LWE and evasive LWE.

**Theorem 5.1** (Theorem 1.1, restated). *Fix $\{\mathbf{M}_{i,b} \in \{0,1\}^{w\times w}\}_{i\in[h],b\in\{0,1\}}$, $\mathbf{u} \in \{0,1\}^w$, such that for all $\mathbf{x} \in \{0,1\}^h$, we have $\mathbf{u}\mathbf{M}_{\mathbf{x}} \neq \mathbf{0}$. Then, by LWE and the evasive LWE assumption, we have*

$$\begin{aligned}
&\mathsf{ggh.encode}^{\otimes}(\mathbf{u}, \{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}}) \\
=&\left\{(\mathbf{u}\mathbf{M}_{1,b} \otimes \mathbf{S}_{1,b})\mathbf{A}_1 + \mathbf{E}_{1,b}\right\}_{b\in\{0,1\}}, \left\{\mathbf{A}_{i-1}^{-1}((\mathbf{M}_{i,b} \otimes \mathbf{S}_{i,b})\mathbf{A}_i + \mathbf{E}_{i,b})\right\}_{i=2,\ldots,h,b\in\{0,1\}} \\
\approx_c&\left\{\mathcal{U}(\mathbb{Z}_q^{\hat{n}_0\times m})\right\}_{b\in\{0,1\}}, \left\{\mathcal{D}_{\mathbb{Z},\chi''}^{m\times m}\right\}_{i=2,\ldots,h,b\in\{0,1\}}
\end{aligned}$$

An overview of the proof is given in Section 1.2. We proceed to describe the parameter settings, followed by an overview of the proof structure and then the proof.

**Remark 5.1** (Parameter settings.). Here, $1^\lambda$ denotes the security parameter and in particular, the running time of the adversary is $\mathrm{poly}(\lambda)$. We rely on $2^{n^\delta}$-hardness for LWE (i.e., indistinguishability against adversaries running in time $2^{n^\delta}$ and a modulus-to-noise ratio of $2^{n^\delta}$), and set the parameters so that

$$\begin{aligned}
2^{n^\delta} &\ge \max\{2^{h^2\lambda}, q/\chi'''\} && \text{LWE hardness} \\
\chi' &= \lambda^h \cdot \chi''' \cdot \lambda^{\omega(1)} && \text{noise flooding} \\
\chi &= \chi' \cdot \lambda^{\omega(1)} && \text{evasive LWE} \\
q &\ge 4h \cdot \chi \cdot (\lambda n w(\chi'' + \chi''') \log q)^h && \text{correctness} \\
\chi'' &= 2\sqrt{nw \log q} && \text{trapdoor sampling}
\end{aligned}$$

The first line comes from the fact that we need to instantiate Lemma 5.3 with hardness $2^{h^2\lambda} \gg (2^h \cdot \mathrm{poly}(\lambda))^{\omega(1)}$ to accommodate the fact that the instances have size up to $2^h \cdot \mathrm{poly}(\lambda)$ (we think of the corresponding instantiation of evasive LWE as using security parameter $\lambda' = 2^h\lambda$ so that

$n' \leq 2^h \cdot \mathrm{poly}(\lambda)$ is bounded by $\mathrm{poly}(\lambda')$), and we iterate the security loss from evasive LWE a total of $h$ times. We can realize the above constraints with

$$n = (h^2\lambda)^{1/\delta}, \quad q = 2^{n^\delta} = 2^{h^2\lambda}, \quad \chi''' = O(n), \quad m = O(\sqrt{nw \log q})$$

The main differences from prior instantiations is that we use $\chi \gg \chi'''$ (whereas prior works use $\chi = \chi'''$) and that $n$ is much larger as a function of $h$.

**Proof structure.** We break down the proof of Theorem 5.1 into two separate lemmas: Lemmas 5.2 and 5.3.

- In the first lemma, we show that if the $2^h$ "evaluated products" (with fresh independent errors)

$$\left\{\underbrace{(\mathbf{u}\mathbf{M}_{\mathbf{x}} \otimes \mathbf{S}_{\mathbf{x}})\mathbf{A}_h}\right\}_{\mathbf{x}\in\{0,1\}^h} \tag{5}$$

  are pseudorandom, then the "encoded program" $\mathsf{ggh.encode}^{\otimes}(\mathbf{u}, \{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}})$ given by

$$\{(\mathbf{u}\mathbf{M}_{1,b} \otimes \mathbf{S}_{1,b})\mathbf{A}_1 + \mathbf{E}_{1,b}\}_{b\in\{0,1\}}, \{\mathbf{A}_{i-1}^{-1}((\mathbf{M}_{i,b} \otimes \mathbf{S}_{i,b})\mathbf{A}_i + \mathbf{E}_{i,b})\}_{i=2,\ldots,h,b\in\{0,1\}}$$

  is pseudorandom. This step relies on $h-1$ invocations of evasive LWE. In fact, we prove a more general statement that does not depend on properties of the matrices $\{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}}$ or the tensor product structure in $\mathbf{M}_{i,b} \otimes \mathbf{S}_{i,b}$. Specifically, the formalization refers to matrices $\hat{\mathbf{S}}_{i,b}$ in place of $\mathbf{M}_{i,b} \otimes \mathbf{S}_{i,b}$.

- In the second lemma, we show that the $2^h$ evaluated products in (5) are pseudorandom under the (standard) LWE assumption, provided $\mathbf{u}\mathbf{M}_{\mathbf{x}} \neq \mathbf{0}$ for all $\mathbf{x} \in \{0,1\}^h$.

**Lemma 5.2.** *Fix some distributions for* $\left\{\widehat{\mathbf{S}}_{i,b}\right\}_{i\in[h],b\in\{0,1\}}$. *Suppose for all* $j \in [h]$, *we have:*

$$\left\{\boxed{\widehat{\mathbf{S}}_{\mathbf{x}'}\mathbf{A}_j + \mathbf{E}_{\mathbf{x}'}}\right\}_{\mathbf{x}'\in\{0,1\}^j}, \left\{\widehat{\mathbf{S}}_{i,b}\right\}_{i\in[h],b\in\{0,1\}} \approx_c \left\{\mathcal{U}(\mathbb{Z}_q^{\hat{n}_0\times m})\right\}_{\mathbf{x}'\in\{0,1\}^j}, \left\{\widehat{\mathbf{S}}_{i,b}\right\}_{i\in[h],b\in\{0,1\}} \tag{6}$$

*where*

$$\mathbf{A}_j \leftarrow \mathbb{Z}_q^{\hat{n}\times m}, \mathbf{E}_{\mathbf{x}'} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{\hat{n}_0\times m}$$

*Then, by the evasive LWE assumption, we have*

$$\begin{aligned}
&\mathsf{ggh.encode}(\left\{\widehat{\mathbf{S}}_{i,b}\right\}_{i\in[h],b\in\{0,1\}}) \\
&= \left\{\widehat{\mathbf{S}}_{1,b}\mathbf{A}_1 + \mathbf{E}_{1,b}\right\}_{b\in\{0,1\}}, \left\{\mathbf{A}_{i-1}^{-1}(\widehat{\mathbf{S}}_{i,b}\mathbf{A}_i + \mathbf{E}_{i,b})\right\}_{i=2,\ldots,h,b\in\{0,1\}} \\
&\approx_c \left\{\mathcal{U}(\mathbb{Z}_q^{\hat{n}_0\times m})\right\}_{b\in\{0,1\}}, \left\{\mathcal{D}_{\mathbb{Z},\chi''}^{m\times m}\right\}_{i=2,\ldots,h,b\in\{0,1\}}
\end{aligned}$$

*Proof.* The proof proceeds in two steps.

Step 1. First, we show that:

$$\begin{aligned} & \left\{\boxed{\widehat{\mathbf{S}}_{1,b}\mathbf{A}_1+\mathbf{E}_{1,b}}\right\}_{b\in\{0,1\}}, \left\{\mathbf{A}_{i-1}^{-1}(\widehat{\mathbf{S}}_{i,b}\mathbf{A}_i+\mathbf{E}_{i,b})\right\}_{i=2,\ldots,h,b\in\{0,1\}} \\ \approx_c & \left\{\mathcal{U}(\mathbb{Z}_q^{\hat{n}_0\times m})\right\}_{b\in\{0,1\}}, \left\{\mathbf{A}_{i-1}^{-1}(\widehat{\mathbf{S}}_{i,b}\mathbf{A}_i+\mathbf{E}_{i,b})\right\}_{i=2,\ldots,h,b\in\{0,1\}} \end{aligned}$$

This proceeds via a proof by induction on $j = h, \ldots, 1$ that:

$$\begin{aligned} & \left\{\boxed{\widehat{\mathbf{S}}_{\mathbf{x}'}\mathbf{A}_j+\mathbf{E}_{\mathbf{x}'}}\right\}_{\mathbf{x}'\in\{0,1\}^j}, \left\{\mathbf{A}_{i-1}^{-1}(\widehat{\mathbf{S}}_{i,b}\mathbf{A}_i+\mathbf{E}_{i,b})\right\}_{i\geq j+1,b\in\{0,1\}}, \left\{\widehat{\mathbf{S}}_{i,b}\right\}_{i\in[h],b\in\{0,1\}} \\ \approx_c & \left\{\mathcal{U}(\mathbb{Z}_q^{\hat{n}_0\times m})\right\}_{\mathbf{x}'\in\{0,1\}^j}, \left\{\mathbf{A}_{i-1}^{-1}(\widehat{\mathbf{S}}_{i,b}\mathbf{A}_i+\mathbf{E}_{i,b})\right\}_{i\geq j+1,b\in\{0,1\}}, \left\{\widehat{\mathbf{S}}_{i,b}\right\}_{i\in[h],b\in\{0,1\}} \end{aligned} \tag{7}$$

where we have additionally augmented the distinguisher's view with $\left\{\widehat{\mathbf{S}}_{i,b}\right\}_{i\in[h],b\in\{0,1\}}$. The base case $j = h$ corresponds to the pre-condition (6) in the lemma. For the inductive step, suppose (7) holds for some $j$ and we would like to deduce the same statement for $j-1$.

We want to invoke our evasive LWE assumption with

$$\begin{aligned} & n' = 2^{j-1}\hat{n}_0, \quad t = 2m = O(\hat{n}_0 \log q) \\ & \mathbf{S} = \widehat{\mathbf{S}}_{j-1} := \left\{\widehat{\mathbf{S}}_{\mathbf{x}'}\right\}_{\mathbf{x}'\in\{0,1\}^{j-1}} \in \mathbb{Z}_q^{2^{j-1}\hat{n}_0\times\hat{n}}, \\ & \mathbf{P} = \widehat{\mathbf{S}}_{j,0}\mathbf{A}_j + \mathbf{E}_{j,0} \| \widehat{\mathbf{S}}_{j,1}\mathbf{A}_j + \mathbf{E}_{j,1} \in \mathbb{Z}_q^{\hat{n}\times 2m} \\ & \mathsf{aux} = \left\{\mathbf{A}_{i-1}^{-1}(\widehat{\mathbf{S}}_{i,b}\mathbf{A}_i+\mathbf{E}_{i,b})\right\}_{i\geq j+1,b\in\{0,1\}}, \left\{\widehat{\mathbf{S}}_{i,b}\right\}_{i\in[h],b\in\{0,1\}}, \\ & \mathbf{B} = \mathbf{A}_{j-1}, \\ & \mathbf{E} = \mathbf{E}_{j-1} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{2^{j-1}\hat{n}_0\times m}, \\ & \mathbf{E}' = \mathbf{E}'_{j-1} \leftarrow \mathcal{D}_{\mathbb{Z},\chi'}^{2^{j-1}\hat{n}_0\times 2m} \end{aligned}$$

where $\{\cdot\}_{\mathbf{x}'\in\{0,1\}^{j-1}}$ denotes stacking the matrices vertically.

First, we verify that the pre-condition of evasive LWE is satisfied. Observe that

$$\begin{aligned} & \widehat{\mathbf{S}}_{j-1}\mathbf{A}_{j-1}+\mathbf{E}_{j-1}, \boxed{\widehat{\mathbf{S}}_{j-1}\mathbf{P}+\mathbf{E}'_{j-1}}, \mathsf{aux} \\ \approx_s & \widehat{\mathbf{S}}_{j-1}\mathbf{A}_{j-1}+\mathbf{E}_{j-1}, \boxed{[\widehat{\mathbf{S}}_{j-1}\widehat{\mathbf{S}}_{j,0}\mathbf{A}_j \| \widehat{\mathbf{S}}_{j-1}\widehat{\mathbf{S}}_{j,1}\mathbf{A}_j] + \mathbf{E}'_{j-1}}, \mathsf{aux} \\ \approx_c & \boxed{\widehat{\mathbf{S}}_{j-1}\mathbf{A}_{j-1}+\mathbf{E}_{j-1}}, \mathcal{U}(\mathbb{Z}_q^{2^{j-1}\hat{n}_0\times 2m}), \mathsf{aux} \\ \approx_c & \mathcal{U}(\mathbb{Z}_q^{2^{j-1}\hat{n}_0\times m}), \mathcal{U}(\mathbb{Z}_q^{2^{j-1}\hat{n}_0\times 2m}), \mathsf{aux} \end{aligned}$$

where

- the $\approx_s$ uses noise flooding to deduce that $\mathbf{E}'_{j-1} \approx_s \mathbf{E}'_{j-1} + \widehat{\mathbf{S}}_{j-1}\cdot[\mathbf{E}_{j,0}\|\mathbf{E}_{j,1}]$;

- the first $\approx_c$ follows from the induction hypothesis (7) for $j$, since we can expand $[\widehat{\mathbf{S}}_{j-1}\widehat{\mathbf{S}}_{j,0}\mathbf{A}_j \| \widehat{\mathbf{S}}_{j-1}\widehat{\mathbf{S}}_{j,1}\mathbf{A}_j] + \mathbf{E}'_{j-1}$ as $\left\{\widehat{\mathbf{S}}_{\mathbf{x}'}\mathbf{A}_j + \mathbf{E}_{\mathbf{x}'}\right\}_{\mathbf{x}'\in\{0,1\}^j}$, together with the observation that given $\left\{\widehat{\mathbf{S}}_{i,b}\right\}_{i\in[h],b\in\{0,1\}}$ in $\mathsf{aux}$, we can sample a random $\mathbf{A}_{j-1}$ and simulate $\widehat{\mathbf{S}}_{j-1}\mathbf{A}_{j-1} + \mathbf{E}_{j-1}$ (note that $\mathsf{aux}$ depends on $\mathbf{A}_j, \ldots, \mathbf{A}_h$ but not $\mathbf{A}_{j-1}$);
- the second $\approx_c$ follows from the pre-condition in (6), along with the fact that given $\left\{\widehat{\mathbf{S}}_{i,b}\right\}_{i\in[h],b\in\{0,1\}}$, we can simulate $\mathsf{aux}$ by sampling $\mathbf{A}_j, \ldots, \mathbf{A}_h$ along with the respective trapdoors.

Then, it follows from evasive LWE that

$$\begin{aligned}&\boxed{\widehat{\mathbf{S}}_{j-1}\mathbf{A}_{j-1} + \mathbf{E}_{j-1}}, \mathbf{A}_{j-1}^{-1}(\mathbf{P}_j), \mathsf{aux}\\ \approx_c\ &\mathcal{U}(\mathbb{Z}_q^{2^{j-1}\hat{n}_0\times m}), \mathbf{A}_{j-1}^{-1}(\mathbf{P}_j), \mathsf{aux}\end{aligned}$$

which corresponds to the statement in (7) for $j-1$. This completes the proof of the inductive step.

To complete the proof of this step, we need to write down the parameters for evasive LWE. Let $\mathcal{A}_j$ denote an adversary that breaks the statement in (7). Then, evasive LWE with security parameter $\lambda' = 2^j\lambda$ (so that $n' = 2^{j-1}\hat{n}_0$ is bounded by $\mathrm{poly}(\lambda')$) tells us:

$$\mathsf{Adv}(\mathcal{A}_j) \geq \mathsf{Adv}(\mathcal{A}_{j-1})/\mathrm{poly}(2^j\lambda), \quad \mathsf{Time}(\mathcal{A}_j) \leq \mathsf{Time}(\mathcal{A}_{j-1}) \cdot \mathrm{poly}(2^j\lambda)$$

which implies:

$$\mathsf{Adv}(\mathcal{A}_h) \geq \mathsf{Adv}(\mathcal{A}_1)/\mathrm{poly}(2^{h^2}\lambda^h), \quad \mathsf{Time}(\mathcal{A}_h) \leq \mathsf{Time}(\mathcal{A}_1) \cdot \mathrm{poly}(2^{h^2}\lambda^h)$$

That is, we will need $\mathrm{poly}(2^{h^2}\lambda^h)$ hardness for the pre-condition in (6). We account for this when setting the final parameters in Remark 5.1.

Step 2. Next, we show that

$$\left\{\mathbf{A}_{i-1}^{-1}(\widehat{\mathbf{S}}_{i,b}\mathbf{A}_i + \mathbf{E}_{i,b})\right\}_{i=2,\ldots,h,b\in\{0,1\}} \approx_c \left\{\mathcal{D}_{\mathbb{Z},\chi''}^{m\times m}\right\}_{i=2,\ldots,h,b\in\{0,1\}}$$

This proceeds exactly as in the proof of [CVW18, Lemma 5.11]: for $j = 2, \ldots, h$, we replace $\left\{\mathbf{A}_{j-1}^{-1}(\widehat{\mathbf{S}}_{j,b}\mathbf{A}_i + \mathbf{E}_{j,b})\right\}_{b\in\{0,1\}}$ with $\left\{\mathcal{D}_{\mathbb{Z},\chi''}^{m\times m}\right\}_{b\in\{0,1\}}$, using

$$\begin{aligned}&\left\{\mathbf{A}_{j-1}^{-1}(\widehat{\mathbf{S}}_{j,b}\mathbf{A}_j + \mathbf{E}_{j,b})\right\}_{b\in\{0,1\}}, \widehat{\mathbf{S}}_{j,0}, \widehat{\mathbf{S}}_{j,1}, \mathbf{A}_j, \tau_{\mathbf{A}_j}\\ \approx_c\ &\left\{\mathcal{D}_{\mathbb{Z},\chi''}^{m\times m}\right\}_{b\in\{0,1\}}, \widehat{\mathbf{S}}_{j,0}, \widehat{\mathbf{S}}_{j,1}, \mathbf{A}_j, \tau_{\mathbf{A}_j}\end{aligned}$$

which in turn follows from LWE [CVW18, Lemma 4.4]. □

**Lemma 5.3.** *Fix $\mathbf{u} \in \{0,1\}^w, \{\mathbf{M}_{i,b} \in \{0,1\}^{w\times w}\}_{i\in[h],b\in\{0,1\}}$ such that for all $\mathbf{x} \in \{0,1\}^h$, we have $\mathbf{u}\mathbf{M}_{\mathbf{x}} \neq \mathbf{0}$. Then, by the LWE assumption, for all $j \in [h]$, we have:*

$$\left\{\boxed{(\mathbf{u}\mathbf{M}_{\mathbf{x}'} \otimes \mathbf{S}_{\mathbf{x}'})\mathbf{A}_j + \mathbf{E}_{\mathbf{x}'}}\right\}_{\mathbf{x}'\in\{0,1\}^j}, \{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}} \approx_c \left\{\mathcal{U}(\mathbb{Z}_q^{n\times m})\right\}_{\mathbf{x}'\in\{0,1\}^j}, \{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}}$$

*where $\mathbf{A}_j \leftarrow \mathbb{Z}_q^{\hat{n}\times m}, \mathbf{E}_{\mathbf{x}'} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{\hat{n}_0\times m}$.*

Similar statements were shown and used in [CVW18,CHVW19] for the special case $j = h$, and where $\{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}}$ were not provided to the adversary. The proof is essentially the same as before, since $\mathbf{u}\mathbf{M}_{\mathbf{x}} \neq \mathbf{0}$ for all $\mathbf{x} \in \{0,1\}^h$ implies $\mathbf{u}\mathbf{M}_{\mathbf{x}'} \neq \mathbf{0}$ for all $\mathbf{x}' \in \{0,1\}^j$.

*Proof.* The proof proceeds in three steps:

- First, by the mixed-product property of tensor products and noise flooding, we have

$$(\mathbf{u}\mathbf{M}_{\mathbf{x}'} \otimes \mathbf{S}_{\mathbf{x}'})\mathbf{A}_j + \mathbf{E}_{\mathbf{x}'} \approx_s (\mathbf{u}\mathbf{M}_{\mathbf{x}'} \otimes \mathbf{I}_n) \cdot (\mathbf{I}_w \otimes \mathbf{S}_{\mathbf{x}'})\mathbf{A}_j + \mathcal{D}^{nw\times m}_{\mathbb{Z},\chi'}) + \mathbf{E}_{\mathbf{x}'}$$

- Next, by the security of the BLMR PRF [BLMR13, BPR12] (also [CVW18, Lemma 7.4]), we have:

$$\left\{\boxed{(\mathbf{I}_w \otimes \mathbf{S}_{\mathbf{x}'})\mathbf{A}_j + \mathcal{D}^{nw\times m}_{\mathbb{Z},\chi'}}\right\}_{\mathbf{x}'\in\{0,1\}^j}, \{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}}$$
$$\approx_c \left\{\boxed{\mathcal{U}(\mathbb{Z}_q^{nw\times m})}\right\}_{\mathbf{x}'\in\{0,1\}^j}, \{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}}$$

  where we use $((\mathbf{I}_w \otimes \mathbf{S})\mathbf{A} + \mathbf{E}, \mathbf{S}) \approx_c (\mathcal{U}(\mathbb{Z}_q^{nw\times m}), \mathbf{S})$, which in turn follows from LWE [BLMR13,CC17].

- Finally, for all $\mathbf{x}' \in \{0,1\}^j$, we have $\mathbf{u}\mathbf{M}_{\mathbf{x}'} \cdot \mathcal{U}(\mathbb{Z}_q^{nw\times m}) \approx_s \mathcal{U}(\mathbb{Z}_q^{n\times m})$, since $\mathbf{u}\mathbf{M}_{\mathbf{x}'} \neq \mathbf{0}$.

This completes the proof. □

We proceed to complete the proof of Theorem 5.1.

*Proof of Theorem 5.1.* We instantiate Lemma 5.2 with:

$$\widehat{\mathbf{S}}_{i,b} = \begin{cases} \mathbf{u}\mathbf{M}_{1,b} \otimes \mathbf{S}_{1,b} & \text{if } i = 1 \\ \mathbf{M}_{i,b} \otimes \mathbf{S}_{i,b} & \text{if } i > 1 \end{cases}$$

The pre-condition in (6) is satisfied, via Lemma 5.3. □

# 6 Witness Encryption

## 6.1 Definition

We recall the definition of witness encryption from [GGSW13].

**Definition 6.1** (Witness encryption [GGSW13])**.** *A witness encryption scheme for an NP language $L$ (with corresponding witness relation $R$) consists of the following two p.p.t. algorithms:*

**Encryption.** $\mathsf{Enc}(1^\lambda, \Psi, \mu)$ *takes as input a security parameter $1^\lambda$, an instance $\Psi \in \{0,1\}^{\mathrm{poly}(\lambda)}$, and a message $\mu \in \{0,1\}$, outputs a ciphertext* $\mathsf{ct}$.

**Decryption.** $\mathsf{Dec}(\mathsf{ct}, x)$ *takes as input a ciphertext* $\mathsf{ct}$ *and string $x \in \{0,1\}^{\mathrm{poly}(\lambda)}$, outputs a message $\mu$ or the symbol $\perp$.*

*These algorithms satisfy*

**Correctness.** *For any security parameter $\lambda$, for any $\mu \in \{0,1\}$, and for any $\Psi \in L$ such that $R(\Psi, x)$ holds, we have that*

$$\Pr[\mathsf{Dec}(\mathsf{Enc}(1^\lambda, \Psi, \mu), x) = \mu] \geq 1 - \mathrm{negl}(\lambda).$$

**Soundness.** *For any p.p.t. adversary $A$, there exists a negligible function $\mathrm{negl}(\cdot)$ such that for any $\Psi \notin L$, we have*

$$\left|\Pr[A(\mathsf{Enc}(1^\lambda, \Psi, 0)) = 1] - \Pr[A(\mathsf{Enc}(1^\lambda, \Psi, 1)) = 1]\right| \leq \mathrm{negl}(\lambda).$$

### 6.2 CVW WE Scheme

To build a witness encryption scheme for all of NP, it suffices to build one for the class of CNF formulas. We describe the CVW18 scheme [CVW18, Section 10]:

**Construction 6.2** (CVW witness encryption)**.** *We construct a witness encryption scheme for the class of CNF formula as follows:*

**Encryption.** $\mathsf{Enc}(1^\lambda, \Psi, \mu)$ *proceeds as follows:*

- *Apply [CVW18, Constructions 6.4,10.2] to the CNF $\Psi$ (of $c$ clauses and $h$ literals) to obtain a read-once branching program $\mathbf{u} = (1 \ \cdots \ 1) \in \{0,1\}^{c+1}$ and $\{\mathbf{M}_{i,b} \in \{0,1\}^{(c+1)\times(c+1)}\}_{i\in[h],b\in\{0,1\}}$ such that for all $\mathbf{x} \in \{0,1\}^h$:*

$$\mathbf{u}\mathbf{M}_\mathbf{x} = \begin{cases} (\mathbf{0}\|\mu) & \text{if } \Psi(\mathbf{x}) = 1 \\ (\neq \mathbf{0}\|\mu) & \text{if } \Psi(\mathbf{x}) = 0 \end{cases}.$$

  *That is, the program computes $\Psi(\mathbf{x}) = 1 \wedge \mu = 0$. Concretely,*

  1. *Initialization: for all $i \in [\ell], b \in \{0,1\}$, Let $\mathbf{M}_{i,b} := \begin{pmatrix} \mathbf{I}_c & \\ & \mu \end{pmatrix}$.*
  2. *If $x_i$ appears in $\psi_j$: set the $j^{th}$ entry on the diagonal of $\mathbf{M}_{i,1}$ to be 0.*
  3. *If $\bar{x}_i$ appears in $\psi_j$: set the $j^{th}$ entry on the diagonal of $\mathbf{M}_{i,0}$ to be 0.*

- *Output*

$$\mathsf{ct} = \mathsf{ggh.encode}^{\otimes}(\mathbf{u}, \{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}})$$

**Decryption.** $\mathsf{Dec}(\mathsf{ct}, \mathbf{x})$ *takes as input $\mathsf{ct} = \{\mathbf{C}_{1,b}, \mathbf{D}_{2,b}, \ldots, \mathbf{D}_{h,b}\}_{b\in\{0,1\}}$ and $\mathbf{x} \in \{0,1\}^h$, outputs 0 if $|\mathbf{C}_{1,x_1} \cdot \mathbf{D}_{2,x_2} \cdots \mathbf{D}_{h,x_h}| \leq B = h \cdot \chi \cdot (\lambda n w(\chi'' + \chi''') \log q)^h$, and 1 otherwise.*

We will set the parameters as in Remark 5.1 with $w = c + 1$. Correctness follows readily from that of ggh.encode. Security follows readily from Theorem 5.1, together with the fact that if $\Psi$ is not satisfiable, then $\mathbf{u}\mathbf{M}_\mathbf{x} \neq \mathbf{0}$ for all $\mathbf{x} \in \{0,1\}^h$.

# 7 Null iO

We analyze the CVW iO scheme for branching programs in [CVW18, Section 11], incorporating simplifications from [CHVW19, Section 6]. The same paper presents a $\mathrm{poly}(\lambda)^{O(c)}$ attack on the iO scheme for read-$c$ branching programs, which could be avoided by artificially padding the branching program when $c$ is small. Here, we show that the scheme is secure as a null-iO scheme for any $c$ even without padding, assuming subexponential LWE and evasive LWE.

## 7.1 Definition

**Definition 7.1.** *An obfuscation scheme* $\mathsf{Obf}$ *is a null-iO scheme if it satisfies the following properties:*

**Correctness:** *There is a negligible function $\nu$ such that for all circuits $C : \{0,1\}^\ell \to \{0,1\}$:*

$$\Pr[\forall x \in \{0,1\}^n : C(x) = \widetilde{C}(x) \mid \widetilde{C} \leftarrow \mathsf{Obf}(1^\lambda, C)] \geq 1 - \nu(\lambda),$$

*where the probability is over the coin tosses of* $\mathsf{Obf}$.

**Security:** *Let $C = \{C_\lambda\}, C' = \{C'_\lambda\}$ be two circuit ensembles, such that $C, C'$ have equal input length and circuit size and furthermore are* everywhere null*, meaning that $\forall x : C(x) = C'(x) = 0$. Then we require that: $\mathsf{Obf}(1^\lambda, C_\lambda) \approx_c \mathsf{Obf}(1^\lambda, C'_\lambda)$.*

## 7.2 CVW null-iO Scheme

**Construction 7.2** (CVW null-IO)**.**

**Obfuscation.** *On input a branching program $\mathbf{u}, \{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}}$ computing a function $C : \{0,1\}^\ell \to \{0,1\}$,*

- *Following [CHVW19, Section 6], we may assume WLOG (at the cost of increasing the width $w$) that*[7]

$$\forall \mathbf{x}' \in \{0,1\}^h : \mathbf{u}\mathbf{M}_{\mathbf{x}'} = \mathbf{0} \iff \mathbf{x}' \in \varpi(\{0,1\}^\ell) \wedge C(\varpi^{-1}(\mathbf{x}')) = 1$$

- *Output*

$$\mathsf{ggh.encode}^{\otimes}(\mathbf{u}, \{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}})$$

**Evaluation.** *On input $\{\mathbf{C}_{1,b}, \mathbf{D}_{2,b}, \ldots, \mathbf{D}_{h,b}\}_{b\in\{0,1\}}$ and $\mathbf{x} \in \{0,1\}^\ell$, outputs 0 if $|\mathbf{C}_{1,\varpi(x)_1} \cdot \mathbf{D}_{2,\varpi(x)_2} \cdots \mathbf{D}_{h,\varpi(x)_h}| \leq B = h \cdot \chi \cdot (\lambda n w(\chi'' + \chi''') \log q)^h$, and 1 otherwise.*

We will set the parameters as in Remark 5.1. Correctness follows readily from that of ggh.encode. Security follows readily from Theorem 5.1, together with the fact that if $C$ is the null program, then $\mathbf{u}\mathbf{M}_{\mathbf{x}'} \neq \mathbf{0}$ for all $\mathbf{x}' \in \{0,1\}^h$.

[7] This basically follows from the fact that we can compute $\mathbf{x}' \stackrel{?}{\in} \varpi(\{0,1\}^\ell)$ using a read-once matrix branching program.

# 8 Cryptanalysis of Evasive LWE

## 8.1 Algorithmic attacks

The known algorithmic attacks essentially fall into one of two categories:

- Attacks on LWE: namely break pseudorandomness of $\mathbf{SB} + \mathbf{E}$ given $\mathsf{aux}$, which is ruled out via the pre-condition;
- Zero-izing attacks: here, given $\mathsf{aux}$, an attacker is able to compute a short vector $\mathbf{z}$ such that $\mathbf{SPz} \bmod q \approx \mathbf{0}$ has low-norm; these attacks are also ruled out via the pre-condition. But first, observe that such a $\mathbf{z}$ breaks the post-condition, since

  $$(\mathbf{SB} + \mathbf{E}) \cdot \mathbf{B}^{-1}(\mathbf{P}) \cdot \mathbf{z} = \mathbf{SPz} + \mathbf{E} \cdot \mathbf{B}^{-1}(\mathbf{P}) \cdot \mathbf{z} \approx \mathbf{0}$$

  and therefore an attacker can distinguish $\mathbf{SB} + \mathbf{E}$ from a random $\mathbf{C}$. On the other hand, we also have

  $$(\mathbf{SP} + \mathbf{E}') \cdot \mathbf{z} \approx \mathbf{0}$$

  and therefore an attacker can also distinguish $\mathbf{SP} + \mathbf{E}'$ from a random $\mathbf{C}'$, which violates the pre-condition.

**A direct attack strategy that fails.** It is instructive to consider the following direct attack strategy: Let $\mathsf{aux} = \mathbf{P}$. Find any (big) $\mathbf{x}$ via Gaussian elimination such that: $\mathbf{Px} = \mathbf{0}$ and $\mathbf{Kx}$ is small (but non-zero), where $\mathbf{K} = \mathbf{B}^{-1}(\mathbf{P})$. This would yield a distinguisher for the post-condition since $(\mathbf{SB} + \mathbf{E}) \cdot \mathbf{K} \cdot \mathbf{x}$ is small whereas $\mathbf{C} \cdot \mathbf{K} \cdot \mathbf{x}$ is not small. We provide three explanations why this attack does not work:

- The matrix $\binom{\mathbf{P}}{\mathbf{K}}$ is a $(n+m) \times t$ matrix but only has rank at most $m$. This is because $[\mathbf{I} \mid -\mathbf{B}]\binom{\mathbf{P}}{\mathbf{K}} = \mathbf{0}$ (that is, the top rows are a linear combination of the bottom ones). Therefore, not every system of linear equations $\binom{\mathbf{P}}{\mathbf{K}}\mathbf{x} = \mathbf{z}$ has a solution $\mathbf{x}$.
- Any solution $\mathbf{x}$ for which $\mathbf{Kx}$ is small yields a solution $\mathbf{Kx}$ to SIS with respect to the random matrix $\mathbf{B}$. Therefore, attacks of this type are already ruled out by SIS.
- More generally, the assumption provably holds when $\mathbf{P}$ is uniformly random and $\mathsf{aux}$ is an efficient function of $\mathbf{P}$ that is independent of $\mathbf{S}$. Therefore, an attack on the assumption must crucially exploit some properties of $\mathbf{P}, \mathsf{aux}$ and fundamentally different from the one here.

## 8.2 Auxiliary inputs

Next, we describe a (heuristic) auxiliary-input attack on our assumption based on general obfuscation, and describe a restricted class of $(\mathbf{P}, \mathsf{aux})$ that avoid these attack, while still sufficient for our security proofs.

**A (heuristic) auxiliary-input attack.** Suppose $\mathbf{S} \leftarrow \mathbb{Z}_q^{2m\times n}, \mathbf{P} \leftarrow \mathbb{Z}_q^{n\times 2m}$ (that is, $n' = t = 2m$). Let $\mathsf{aux}$ be an obfuscation of the follow program $\Pi_{\mathbf{P},\tau}$ which has $\mathbf{P}$ and a corresponding trapdoor $\tau$ hard-wired, and on input $\mathbf{C} \in \mathbb{Z}_q^{2m\times m}, \mathbf{D} \in \mathbb{Z}_q^{m\times 2m}$,

- use $\tau$ to solve for $\mathbf{S}_0$ such that $|\mathbf{C}\cdot\mathbf{D} - \mathbf{S}_0\cdot\mathbf{P}|$ is small
- if $|\mathbf{D}|$ is small and such a $\mathbf{S}$ exists, output $1$, else output $0$.

Observe that $\Pi_{\mathbf{P},\tau}$ would output $1$ on input $(\mathbf{SB}+\mathbf{E}, \mathbf{B}^{-1}(\mathbf{P}))$, and $0$ on input $(\mathcal{U}(\mathbb{Z}_q^{n'\times m}), \mathbf{B}^{-1}(\mathbf{P}))$, which yields a distinguisher for the post-condition. On the other hand, by LWE, $(\mathbf{SB}+\mathbf{E}, \mathbf{SP}+\mathbf{E}')$ is pseudorandom. Moreover, given oracle access to $\Pi_{\mathbf{P},\tau}$, it is statistically hard to find an accepting input. This means that the pre-condition would hold given an ideal obfuscation of $\Pi_{\mathbf{P},\tau}$.

**Restricted class $(\mathbf{P}, \mathsf{aux})$.** We consider $(\mathbf{P}, \mathsf{aux})$ of the form:

$$\begin{aligned}\mathbf{P} :=& [\widehat{\mathbf{S}}_{1,0}\mathbf{A}_1 \| \widehat{\mathbf{S}}_{1,1}\mathbf{A}_1] + \mathbf{E}_1\\ \mathsf{aux} :=& (\mathbf{A}_1^{-1}([\widehat{\mathbf{S}}_{2,0}\mathbf{A}_2 \| \widehat{\mathbf{S}}_{2,1}\mathbf{A}_2] + \mathbf{E}_2), \ldots, \mathbf{A}_{\ell-1}^{-1}([\widehat{\mathbf{S}}_{\ell,0}\mathbf{A}_\ell \| \widehat{\mathbf{S}}_{\ell,1}\mathbf{A}_\ell] + \mathbf{E}_\ell),\\ & \widehat{\mathbf{S}}_{1,0}, \widehat{\mathbf{S}}_{1,1}, \ldots, \widehat{\mathbf{S}}_{\ell,0}, \widehat{\mathbf{S}}_{\ell,1}, \mathsf{aux}_0)\end{aligned}$$

where $\widehat{\mathbf{S}}_{1,0}, \widehat{\mathbf{S}}_{1,1}, \ldots, \widehat{\mathbf{S}}_{\ell,0}, \widehat{\mathbf{S}}_{\ell,1}, \mathsf{aux}_0$ are "public-coin" (by requiring that $\mathsf{aux}$ also contains the coin tosses used to sample $\widehat{\mathbf{S}}_{i,b}, \mathsf{aux}_0$) and independent of the random matrices $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$. Note that

- the private randomness for $\mathsf{aux}$ are only used in sampling (i) $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$ along with the respective trapdoors, (ii) $\mathbf{E}_1, \ldots, \mathbf{E}_\ell$, as well as (iii) $\mathbf{A}_1^{-1}(\cdot), \ldots, \mathbf{A}_\ell^{-1}(\cdot)$;
- we only require that the $\widehat{\mathbf{S}}_{i,b}$'s are "public-coin" and do not require that they compute a tensor of the form $\mathbf{M}_{i,b} \otimes \mathbf{S}_{i,b}$;
- this restricted class of $(\mathbf{P}, \mathsf{aux})$ is sufficient for our security reductions in Lemma 5.2.

Next, we argue that this restricted class do not capture the obfuscation-based $\mathsf{aux}$. The reasons are two-fold:

- The matrices $\widehat{\mathbf{S}}_{i,b}$'s are public-coin and given to the distinguisher as part of $\mathsf{aux}$, so any secret information (e.g. matrix trapdoors $\tau$) embedded into these matrices will also be provided to the distinguisher in the pre-condition "in the clear".
- The matrices $\widehat{\mathbf{S}}_{i,b}$'s are independent of the random matrices $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$ and in particular cannot depend on trapdoors for any of these matrices.

### 8.3 A special case

We consider a special case for evasive LWE that is closely related to the WE and null-IO scheme. Suppose $\mathbf{u}_i\mathbf{M} \neq 0$ for all $i \in [N]$. Then, evasive LWE (plus LWE) tells us that the following distribution is pseudorandom:

$$\{\mathbf{S}_{1,i}, (\mathbf{u}_i \otimes \mathbf{S}_{1,i})\mathbf{A}_1 + \mathbf{E}_{1,i}\}_{i\in[N]}, \mathbf{S}_2, \mathbf{A}_1^{-1}((\mathbf{M}\otimes\mathbf{S}_2)\mathbf{A}_2 + \mathbf{E}_2)$$

We observe that for the case $N = 1$, such a statement follows from LWE. In fact, it suffices to prove pseudorandomness of

$$(\mathbf{u} \otimes \mathbf{I})\mathbf{A}_1, \mathbf{S}_2, \mathbf{A}^{-1}((\mathbf{M} \otimes \mathbf{S}_2)\mathbf{A}_2 + \mathbf{E}_2)$$

- First, we apply LWE with secret $\mathbf{A}_2$ to replace $(\mathbf{I} \otimes \mathbf{S}_2)\mathbf{A}_2$ with a random $\mathbf{P}$.
- Next, by LWE and adapting an argument from [CVW18], we have

  $$(\mathbf{u} \otimes \mathbf{I})\mathbf{A}_2, \mathbf{A}_2^{-1}((\mathbf{M} \otimes \mathbf{I})\mathbf{P} + \mathbf{E}) \approx_c (\mathbf{u} \otimes \mathbf{I})\mathbf{A}_2, ((\mathbf{u} \otimes \mathbf{I})\mathbf{A}_2)^{-1}((\mathbf{u}\mathbf{M} \otimes \mathbf{I})\mathbf{P} + \mathbf{E}')$$

  The idea is to treat the part of $\mathbf{A}_2$ that is perfectly hidden given $(\mathbf{u} \otimes \mathbf{I})\mathbf{A}_2$ as the LWE secret.
- The rest of the proof follows from a statistical argument.


### Acknowledgements

We thank the reviewers for helpful and meticulous feedback. VV was supported by DARPA under Agreement No. HR00112020023, a grant from the MIT-IBM Watson AI, a grant from Analog Devices, a Microsoft Trustworthy AI grant, and a DARPA Young Faculty Award. DW was supported by NSF grant CNS-1750795, CNS-2055510, and the Alfred P. Sloan Research Fellowship.


## References


[ADGM17] Daniel Apon, Nico Döttling, Sanjam Garg, and Pratyay Mukherjee. Cryptanalysis of indistinguishability obfuscations of circuits over GGH13. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *ICALP 2017*, volume 80 of *LIPIcs*, pages 38:1–38:16. Schloss Dagstuhl, July 2017. 2, 8

[Agr19] Shweta Agrawal. Indistinguishability obfuscation without multilinear maps: New methods for bootstrapping and instantiation. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 191–225. Springer, Heidelberg, May 2019. 2, 9

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 308–326. Springer, Heidelberg, August 2015. 2

[AJL+19] Prabhanjan Ananth, Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. Indistinguishability obfuscation without multilinear maps: New paradigms via low degree weak pseudorandomness and security amplification. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 284–332. Springer, Heidelberg, August 2019. 2, 9

[ALS16] Shweta Agrawal, Benoît Libert, and Damien Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 333–362. Springer, Heidelberg, August 2016. 8

[AP20] Shweta Agrawal and Alice Pellet-Mary. Indistinguishability obfuscation without maps: Attacks and fixes for noisy linear FE. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 110–140. Springer, Heidelberg, May 2020. 2

[BDGM20a] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Candidate iO from homomorphic encryption schemes. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 79–109. Springer, Heidelberg, May 2020. 2

[BDGM20b] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Factoring and pairings are not necessary for io: Circular-secure lwe suffices. Cryptology ePrint Archive, Report 2020/1024, 2020. 2

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Heidelberg, August 2001. 2

[BGMZ18] James Bartusek, Jiaxin Guan, Fermi Ma, and Mark Zhandry. Return of GGH15: Provable security against zeroizing attacks. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 544–574. Springer, Heidelberg, November 2018. 9

[BLMR13] Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key homomorphic PRFs and their applications. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 410–428. Springer, Heidelberg, August 2013. 5, 6, 7, 17

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 719–737. Springer, Heidelberg, April 2012. 17

[BV15] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In Venkatesan Guruswami, editor, *56th FOCS*, pages 171–190. IEEE Computer Society Press, October 2015. 2

[CC17] Ran Canetti and Yilei Chen. Constraint-hiding constrained PRFs for $NC^1$ from LWE. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 446–476. Springer, Heidelberg, April / May 2017. 2, 3, 12, 13, 17

[CCH+19] Jung Hee Cheon, Wonhee Cho, Minki Hhan, Jiseung Kim, and Changmin Lee. Statistical zeroizing attack: Cryptanalysis of candidates of BP obfuscation over GGH15 multilinear map. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 253–283. Springer, Heidelberg, August 2019. 2, 8

[CGH17] Yilei Chen, Craig Gentry, and Shai Halevi. Cryptanalyses of candidate branching program obfuscators. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 278–307. Springer, Heidelberg, April / May 2017. 2, 8

[CHL+15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 3–12. Springer, Heidelberg, April 2015. 2, 8

[CHVW19] Yilei Chen, Minki Hhan, Vinod Vaikuntanathan, and Hoeteck Wee. Matrix PRFs: Constructions, attacks, and applications to obfuscation. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part I*, volume 11891 of *LNCS*, pages 55–80. Springer, Heidelberg, December 2019. 2, 9, 17, 19

[CLLT16] Jean-Sébastien Coron, Moon Sung Lee, Tancrède Lepoint, and Mehdi Tibouchi. Cryptanalysis of GGH15 multilinear maps. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 607–628. Springer, Heidelberg, August 2016. 2, 8

[CLLT17] Jean-Sébastien Coron, Moon Sung Lee, Tancrède Lepoint, and Mehdi Tibouchi. Zeroizing attacks on indistinguishability obfuscation over CLT13. In Serge Fehr, editor, *PKC 2017, Part I*, volume 10174 of *LNCS*, pages 41–58. Springer, Heidelberg, March 2017. 2, 8

[CLT15] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. New multilinear maps over the integers. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 267–286. Springer, Heidelberg, August 2015. 2

[CVW18] Yilei Chen, Vinod Vaikuntanathan, and Hoeteck Wee. GGH15 beyond permutation branching programs: Proofs, attacks, and candidates. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 577–607. Springer, Heidelberg, August 2018. 2, 3, 6, 7, 8, 9, 12, 13, 16, 17, 18, 19, 22

[DQV+21] Lalita Devadas, Willy Quach, Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Succinct LWE sampling, random polynomials, and obfuscation. In *TCC*, 2021. 2, 4

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 1–17. Springer, Heidelberg, May 2013. 2

[GGH+13b] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013. 2

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 498–527. Springer, Heidelberg, March 2015. 2, 12, 13

[GGHW14] Sanjam Garg, Craig Gentry, Shai Halevi, and Daniel Wichs. On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 518–535. Springer, Heidelberg, August 2014. 4

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 467–476. ACM Press, June 2013. 2, 17

[GJLS21] Romain Gay, Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from simple-to-state hard problems: New assumptions, new techniques, and simplification. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part III*, volume 12698 of *LNCS*, pages 97–126. Springer, Heidelberg, October 2021. 2

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In Chris Umans, editor, *58th FOCS*, pages 612–621. IEEE Computer Society Press, October 2017. 2, 3

[GKW18] Rishab Goyal, Venkata Koppula, and Brent Waters. Collusion resistant traitor tracing from learning with errors. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 660–670. ACM Press, June 2018. 3

[GLW14] Craig Gentry, Allison B. Lewko, and Brent Waters. Witness encryption from instance independent assumptions. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 426–443. Springer, Heidelberg, August 2014. 2, 8

[GMM+16] Sanjam Garg, Eric Miles, Pratyay Mukherjee, Amit Sahai, Akshayaram Srinivasan, and Mark Zhandry. Secure obfuscation in a weak multilinear map model. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 241–268. Springer, Heidelberg, October / November 2016. 9

[GMM17] Sanjam Garg, Mohammad Mahmoody, and Ameer Mohammed. Lower bounds on obfuscation from all-or-nothing encryption primitives. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 661–695. Springer, Heidelberg, August 2017. 2

[GP21] Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. In *STOC*, 2021. 2, 4

[HHSS17] Shai Halevi, Tzipora Halevi, Victor Shoup, and Noah Stephens-Davidowitz. Implementing BP-obfuscation using graph-induced encoding. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 783–798. ACM Press, October / November 2017. 2

[HJL21] Samuel B. Hopkins, Aayush Jain, and Huijia Lin. Counterexamples to new circular security assumptions underlying iO. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 673–700, Virtual Event, August 2021. Springer, Heidelberg. 4, 8

[JLMS19] Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. How to leverage hardness of constant-degree expanding polynomials overa $\mathbb{R}$ to build $i\mathcal{O}$. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 251–281. Springer, Heidelberg, May 2019. 2, 4

[JLS21a] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from LPN over f_p, dlin, and prgs in nc^0. *IACR Cryptol. ePrint Arch.*, page 1334, 2021. 2

[JLS21b] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In *STOC*, 2021. 2

[LPSS14] San Ling, Duong Hieu Phan, Damien Stehlé, and Ron Steinfeld. Hardness of k-LWE and applications in traitor tracing. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 315–334. Springer, Heidelberg, August 2014. 8

[MSZ16] Eric Miles, Amit Sahai, and Mark Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over GGH13. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 629–658. Springer, Heidelberg, August 2016. 2, 8

[Pel18] Alice Pellet-Mary. Quantum attacks against indistinguishablility obfuscators proved secure in the weak multilinear map model. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 153–183. Springer, Heidelberg, August 2018. 2, 8

[Tsa22] Rotem Tsabary. Candidate witness encryption from lattice techniques. In *CRYPTO*, 2022. 5, 26, 27

[Wee22] Hoeteck Wee. Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In *Eurocrypt*, 2022. 3, 8, 26, 27

[WW21] Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious LWE sampling. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part III*, volume 12698 of *LNCS*, pages 127–156. Springer, Heidelberg, October 2021. 2, 4

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In Chris Umans, editor, *58th FOCS*, pages 600–611. IEEE Computer Society Press, October 2017. 2, 3

# A Comparison with [Tsa22]

An independent work of Tsabary [Tsa22] (also independent of [Wee22]) presents a new witness encryption scheme under a variant of evasive LWE. We describe some high-level differences between the two works:

- Tsabary [Tsa22] presents a new witness encryption scheme that uses read-many branching programs and does not consider null-IO. We prove security of *existing* candidate WE and null-IO schemes in CVW, where the former uses read-once (matrix) branching programs.

- The formulation of evasive LWE in [Tsa22] allows $(\mathbf{P}, \mathsf{aux})$ to depend on $\mathbf{B}$, whereas ours and that in [Wee22] does not. In particular, our formulation of evasive LWE is more conservative.
- The analysis in [Tsa22] relies on a formulation of evasive LWE with polynomial hardness and oracle access to a possibly exponential number of matrices, whereas we crucially rely on evasive LWE with instances of exponential size $2^h$ (which in turn requires a careful setting of parameters). In our security reduction, the adversary receives all possible partial evaluated products, whereas the adversary in [Tsa22] only has oracle access to these quantities. Note that in both analysis, the complexity of the adversary could double with each invocation of evasive LWE, so that we would necessarily need to consider adversaries running in time at least $2^h$, for which there is no real distinction between receiving and oracle access to all possible partial products.