Statistical Zeroizing Attack: Cryptanalysis of Candidates of BP Obfuscation over GGH15 Multilinear Map.

Jung Hee Cheon[1,2,3], Wonhee Cho[1], Minki Hhan[1], Jiseung Kim[1], and Changmin Lee[4]

[1] Department of Mathematical Sciences, SNU, Republic of Korea.
{jhcheon, wony0404, hhan_, tory154}@snu.ac.kr
[2] Research Institute of Mathematics (RIM), SNU, Republic of Korea.
[3] Cryptolab, Seoul, Republic of Korea.
[4] ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, INRIA, UCBL), France.
changmin.lee@ens-lyon.fr

Abstract. We present a new cryptanalytic algorithm on obfuscations based on GGH15 multilinear map. Our algorithm, *statistical zeroizing attack*, directly distinguishes two distributions from obfuscation while it follows the zeroizing attack paradigm, that is, it uses evaluations of zeros of obfuscated programs.

Our attack breaks the recent indistinguishability obfuscation candidate suggested by Chen *et al.* (CRYPTO'18) for the optimal parameter settings. More precisely, we show that there are two functionally equivalent branching programs whose CVW obfuscations can be efficiently distinguished by computing the sample variance of evaluations.

This statistical attack gives a new perspective on the security of the indistinguishability obfuscations: we should consider the shape of the distributions of evaluation of obfuscation to ensure security.

In other words, while most of the previous (weak) security proofs have been studied with respect to algebraic attack model or ideal model, our attack shows that this algebraic security is not enough to achieve indistinguishability obfuscation. In particular, we show that the obfuscation scheme suggested by Bartusek *et al.* (TCC'18) does not achieve the desired security in a certain parameter regime, in which their algebraic security proof still holds.

The correctness of statistical zeroizing attacks holds under a mild assumption on the preimage sampling algorithm with a lattice trapdoor. We experimentally verify this assumption for implemented obfuscation by Halevi *et al.* (ACM CCS'17).

Keywords: Cryptanalysis, indistinguishability obfuscation, multilinear map

1 Introduction

Indistinguishability obfuscation (iO) is one of the most powerful tools used to construct many cryptographic applications such as non-interactive multiparty

key exchange and functional encryption [5,18,34]. While constructing a general-purpose iO has been posed as a longstanding open problem, Garg *et al.* [18] first proposed a plausible candidate for the general-purpose iO exploiting a multilinear map in 2013. Starting from this work, many subsequent studies have proposed plausible constructions of iO upon candidate multilinear maps [1–3,6,18,19,25–28,31,32,36].

However, all of the current constructions of multilinear map, essentially classified as GGH13, CLT13 and GGH15 [16,17,20], are merely candidates. These constructions are not known to have the desired security of the multilinear map due to the first class of zeroizing attacks, such as the CHLRS attack and Hu-Jia attack [11, 15, 26]; these attacks commonly exploits several encodings of zero to show the multi-party key exchange protocol instantiated by candidate multilinear maps are not secure.

On the other hand, the first class of zeroizing attacks does not damage the security of current iO constructions from the candidate multilinear maps. It later turns out that most candidates iO fail to achieve the desired security due to subsequent works, *the second class of zeroizing attacks* [9,10,12–15,33], which employs algebraic relations of the top level encodings of zero. In this light, many researches focus on algebraic security of obfuscation using the weak multilinear map models [4, 19, 29] to capture the currently known techniques to analyze obfuscations and multilinear map itself.

Recently, GGH15 multilinear map has been in the spotlight because it is shown that GGH15 and its variants can be exploited to construct provable secure special-purpose obfuscations and other cryptographic applications including constraint pseudorandom functions under the hardness of LWE and its variants [7,8,10,22,35]. Therefore, the GGH15 multilinear map has been believed to be the most plausible candidate for constructing the general-purpose obfuscation.

In this respect, Chen *et al.* [10] proposed a new iO candidate over GGH15, called CVW obfuscation, to be secure against all known attacks. Then, Bartusek *et al.* [4] provided a new candidate over GGH15, called BGMZ obfuscation, which is provably secure against generalized algebraic zeroizing attacks. The security of these two schemes in more general setting remains as an open problem.

1.1 Our Result

We give a new polynomial time cryptanalysis, *statistical zeroizing attack*, on the candidates of iO based on the GGH15 multilinear map. This attack directly distinguishes the distributions from zeros of obfuscated programs instead of finding algebraic relations of evaluations. We particularly exploit the sample variance as a distinguisher of the distributions, while this attack introduces wide class of distinguishing methods. In particular, under an assumption on lattice preimage sampling algorithm with a trapdoor, our attack breaks the security of

- CVW obfuscation for the optimal parameter choice. Further, our attack still works for the relatively small variance σ^2 of Gaussian distribution such as $\sigma = \mathsf{poly}(\lambda)$ for the security parameter λ, and

- BGMZ obfuscation for large variance of Gaussian distribution, e.g. $\sigma = 2^{\lambda}$, which still enables the security proof in the weak GGH15 multilinear map model.[1]

This result refutes the open problem posed in [10] in a certain parameter regime: the CVW obfuscation is not secure even when the adversary gets oracle access to the honest evaluations as matrix products instead of obfuscated program.

Our attack leads a new perspective to the study of iO: we should focus on the statistical properties such as shapes of distributions as well to achieve indistinguishability obfuscation. In particular, the distributions of evaluations should be (almost) the same regardless of the choice of target branching program. Previously, most attacks and constructions only focused on the algebraic structure of evaluations.

Attack Overview. Suppose that the adversary has two functionally equivalent branching programs $\mathbf{M}$ and $\mathbf{N}$, and an obfuscated program $\mathcal{O}(\mathbf{P})$ where $\mathbf{P} = \mathbf{M}$ or $\mathbf{N}$. The purpose of the adversary is to determine whether $\mathbf{P} = \mathbf{M}$ or $\mathbf{N}$. Note that the recent obfuscation constructions compute its output via two processes: the first step is to compute a value, we call *evaluation* here according to the evaluating rules, which is usually to compute a product of given matrices. The second step is to determine the *output* from the size of the evaluation in the first step.

The basic form of statistical zeroizing attack is incredibly simple; just compute the evaluation of obfuscated program (right before computing output) and check if an entry is larger than a threshold value. Since two evaluations of obfuscated programs $\mathcal{O}(\mathbf{M})$ and $\mathcal{O}(\mathbf{N})$ have the different variance, this attack may work.

Technically speaking, we consider a bit complex form of statistical zeroizing attack in this paper to give a rigorous analysis. The above form is simple, but it is hard to check the correctness of attack.[2] Thus we consider the multiple-sample problem instead of one evaluation, and then compute the sample variance. Then we determine $\mathbf{P}$ by checking the inequality of the sample variance and a threshold value. Note that these distributions of evaluations are polynomial-time constructible, i.e. the sampling algorithm is done in polynomial time, since every parameter to do obfuscation process is given to adversary. Therefore the distinguishing algorithm of two distributions implies the distinguishability of two corresponding evaluations by the standard hybrid argument.

Though the attack is conceptually simple, it is difficult to verify that the attack works well for certain obfuscation schemes, and this verification requires several complex computational tasks. Thus we give the sufficient conditions that attack works well using sample variance for a simpler description of the attack.

[1] That is, our attack is lying outside the considered attack class in [4].

[2] The difference of variance is even not enough to distinguish. For example, the distributions that 0 with overwhelming probability cannot be efficiently distinguished though these can have any variance.

And we assign many pagesost papers including appendix and technical computations, which can be found in the full version of this paper [11], to show that those conditions hold under an assumption, dealing with many random variables that might be dependent themselves. We derive many lemmas to deal with such intertwined random variables.

Assumption on Lattice Preimage Sampling. The analysis of attack requires an assumption on lattice preimage sampling algorithm. This assumption states that the variance and kurtosis of products of matrices from preimage sampling have almost the same size as one assumed the independency of those matrices. This assumption is experimentally verified for matrices used in implemented obfuscation scheme [23]. For more detailed description, see Assumption 1 and Appendix C.

Example of Statistical Zeroizing Attack. We give an example to show how our attack intuitively works. We consider a simple construction of GGH15-obfuscation without all safeguards. For brevity we only give the result of evaluation. A detailed description of this simple obfuscation is given in Appendix A. We also do not give a computational analysis of the attack here, but this example still is enough to shows that the two distributions of evaluations from different branching programs may have quite different shape.

We consider two functionally equivalent branching programs

$$\mathbf{M} = \{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}} \quad \textit{and} \quad \mathbf{N} = \{\mathbf{N}_{i,b}\}_{i\in[h],b\in\{0,1\}}$$

where

$$\mathbf{M}_{i,b} = \mathbf{0}^{w\times w} \text{ for all } i,b \text{ and } \mathbf{N}_{i,b} = \begin{cases} \mathbf{I}^{w\times w} & \text{if } i = 1 \\ \mathbf{0}^{w\times w} & \text{otherwise} \end{cases}.$$

For these BPs, the evaluations are of the form

$$\mathcal{O}(\mathbf{M})(\mathbf{x}) = \mathbf{E}_{1,x_{\mathsf{inp}(1)}} \cdot \prod_{k=2}^{h} \mathbf{D}_{k,x_{\mathsf{inp}(k)}} \text{ and}$$

$$\mathcal{O}(\mathbf{N})(\mathbf{x}) = \mathbf{E}_{1,x_{\mathsf{inp}(1)}} \cdot \prod_{k=2}^{h} \mathbf{D}_{k,x_{\mathsf{inp}(k)}} + \mathbf{I} \cdot \mathbf{E}_{2,x_{\mathsf{inp}(2)}} \cdot \prod_{k=3}^{h} \mathbf{D}_{k,x_{\mathsf{inp}(k)}}.$$

Here $\mathbf{D}$'s are preimage-sampled matrices and $\mathbf{E}$'s are error matrices, whose entries are all following discrete Gaussian distribution.

If we choose polynomial-size variances for those matrices, these two distributions have noticeably different shape. Therefore one can hope to distinguish two distribution; indeed, the sample variance will be served as a distinguisher in this paper. Or, more efficiently, one can distinguish them by looking at the size of sample, but this is not easy to show the correctness as noted in above without strong assumption on shape of distributions.

Applicability and Limitation. The class of branching programs constructed from CNF formulas, suggested in [10, Construction 6.4], is in the range of our

attack as well. For example, as we choose two branching programs $\mathbf{N} = \{\mathbf{N}_{i,b}\}$ and $\mathbf{M} = \{\mathbf{M}_{i,b}\}$ as follows: $\mathbf{N}_{1,b}$ as the identity matrix with $w \times w$ size and all other matrices of $\mathbf{M}$ and $\mathbf{N}$ as the zero matrix. These two branching programs $\mathbf{M}$ and $\mathbf{N}$ correspond to some CNF formulas following the construction. This is exactly the same to the target branching programs described in Section 4.2 as an attack example.

On the other hand, there is a class of branching programs that seems robust against our attack: permutation matrix branching programs. For this class of branching programs, the distributions of evaluations except bookend vectors are the same for any choice of permutation branching program $\mathbf{M}$ in many obfuscation constructions (under the assumption on trapdoor matrices). Interestingly, (a variant of) the first candidate iO over the GGH15 multilinear map [18, 20] has targeted such branching programs so it is robust against our attack.

Further, the obfuscation schemes over the CLT13 or GGH13 multilinear maps seems to be secure against statistical zeroizing attack. This is due to the structure of those schemes; encodings CLT13 and GGH13 have large randomness in the zero-testing results compared to the message-dependent parts. In other words, the randomness dominates the zero-testing values and the message only gives negligible perturbation on the zero-testing distributions.

Counter Measures. There are two countermeasures on our attack: 1) modifying construction to obfuscate permutation branching programs and 2) adjusting parameters to rule out our attack. We remark that both countermeasures are plausibly blocking the attack but not in the provable security level.

As noted above, we can simply use the known obfuscations to obfuscate permutation branching programs only. Unfortunately, CVW and BGMZ obfuscations in the suggested form are not appropriate to obfuscate the permutation branching programs.[3] We can modify CVW obfuscation to obfuscate the permutation branching programs; this modified construction is secure against all existing attacks including the attack suggested in this paper. This can be done by choosing the bookends appropriately for permutations. A more precise description is placed in Appendix B. The similar modification works well in BGMZ obfuscation.

Another simple countermeasure for our attack is to take another parameter choice for variance σ, especially to adjust the variance of several discrete Gaussian distributions appropriately. For example, one can consider the following modifications.

[3] Though there is a general transformation from permutation branching program into Type I branching program [10, Claim 6.2], this induces the bookend vector of the form $(\mathbf{v}|-\mathbf{v})$ rather than the implicitly supposed bookend $\mathbf{1}^{1\times w}$ in CVW obfuscation. If we directly obfuscate permutation branching programs, the functionality of them is all-rejection. Indeed, if we obfuscate permutation branching programs using CVW obfuscation as this trivial functionality (without transformation), the iO security for these trivial BPs can be proven by the proof technique of [7].

- For CVW obfuscation, the condition of our attack (using sample variance) does not hold for large σ^2, e.g. $\sigma^2 = \Omega(m^\ell)$ for the sampled dimension m of preimage sampling and the length ℓ of branching program.
- For BGMZ obfuscation, the small choice of σ, e.g. $\sigma^2 = O(\nu)$ for the size bound of the bookend vector's entry ν.

Both countermeasures yield the exponential bound in the first attack condition (See Proposition 3.1). We remark that the preimage sampling procedure with large σ can be done in polynomial time using [21].

It is interesting that the large σ yields countermeasure on CVW obfuscation while it allows the attack on BGMZ obfuscation. This difference comes from the structure of scheme, or the dominating term of evaluation's variance. More precisely, the main parts to induce the difference are

– In BGMZ obfuscation, there are auxiliary random matrices terms, which flood other terms. For large σ, a dominating term moves to the message dependent terms.
– In CVW obfuscation, auxiliary random matrices are only larger than the message dependent terms up to polynomial factor, which gives the enough difference to distinguish. When σ is increased, the ratio is going to exponential and yields noise-flooding.

Open Questions. We also leave some open problems:

1. The presented attack shows some weakness of obfuscation for non-permutation branching program, while this class of branching programs is known to have several advantages compared to permutation branching programs including efficiency [10]. Can we construct a provably secure obfuscation against all zeroizing attack without choosing the permutation branching programs?
2. On the other hand, can we extend the zeroizing attack to more general obfuscation or branching programs such as evasive functions or permutation branching programs? Can we derive a new attack that combines algebraic and statistical structure of evaluations?
3. The candidate witness encryption in [10] shares almost the same structure with the CVW obfuscation but we do not know whether it is secure or not.

Organization. In Section 2, we introduce preliminary related to the branching program, iO, and lattices. We describe the statistical zeroizing attack in Section 3. In Section 4, we briefly describe CVW obfuscation and its cryptanalysis. In addition, we review BGMZ obfuscation and its cryptanalysis in Section 5.

2 Preliminaries

Notations. $\mathbb{N}, \mathbb{Z}, \mathbb{R}$ denote the sets of natural numbers, integers, and real numbers, respectively. For an integer $q \geq 2$, $\mathbb{Z}_q$ is the set of integers modulo q.

Elements are in $\mathbb{Z}_q$ are usually considered as integers in $[-q/2, q/2)$. We denote the set $\{1, 2, \cdots, h\}$ by $[h]$ for a natural number h.

Lower bold letters means row vectors and capital bold letters denote matrices. In addition, capital italic letters denote random matrices or random variables. For a random variable X, we let $E(X)$ be the expected value of X, $Var(X)$ the variance of X.

The n-dimensional identity matrix is denoted by $\mathbf{I}^{n\times n}$. For a row vector $\mathbf{v}$, a i-th component of $\mathbf{v}$ is denoted by v_i, and for a matrix $\mathbf{A}$, a (i,j)-th entry of a matrix $\mathbf{A}$ is denoted by $a_{i,j}$, respectively. A notation $\mathbf{1}^{a\times b}$ means a $a \times b$ matrix such that all entries are 1. The ℓ_p norm of a vector $\mathbf{v} = (v_i)$ is denoted by $\|\mathbf{v}\|_p = (\sum_i |v_i|^p)^{1/p}$. We denote $\|\mathbf{A}\|_\infty$ by the infinity norm of a matrix $\mathbf{A}$, $\|\mathbf{A}\|_\infty = \max_{i,j} a_{i,j}$ with $\mathbf{A} = (a_{i,j})$.

We use a notation $\mathbf{x} \leftarrow \chi$ to denote the operation of sampling element $\mathbf{x}$ from the distribution χ. Especially, if χ is the uniform distribution on a finite set $\mathbf{X}$, we denote $\mathbf{x} \leftarrow U(\mathbf{X})$.

For two matrices $\mathbf{A} = (a_{i,j}) \in \mathbb{R}^{n\times m}$, $\mathbf{B} \in \mathbb{R}^{k\times \ell}$, the tensor product of matrix $\mathbf{A}$ and $\mathbf{B}$ is defined as

$$\mathbf{A} \otimes \mathbf{B} := \begin{pmatrix} a_{1,1}\cdot\mathbf{B} & \cdots & a_{1,m}\cdot\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{n,1}\cdot\mathbf{B}, & \cdots, & a_{n,m}\cdot\mathbf{B} \end{pmatrix}.$$

For four matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ such that one can form products $\mathbf{A}\cdot\mathbf{C}$ and $\mathbf{B}\cdot\mathbf{D}$, the equation $(\mathbf{A}\otimes\mathbf{B})\cdot(\mathbf{C}\otimes\mathbf{D}) = (\mathbf{A}\cdot\mathbf{C})\otimes(\mathbf{B}\cdot\mathbf{D})$ holds.

2.1 Matrix Branching Program

A matrix branching program (BP) is the set which consists of an index-to-input function and several matrix chains.

Definition 2.1 *A width w, length h, and a s-ary matrix branching program $\mathbf{P}$ over a ℓ-bit input is a set which consists of index-to-input maps $\{\mathsf{inp}_\mu : [h] \to [\ell]\}_{\mu\in[s]}$, sequences of matrices, and two disjoint sets of target matrices*

$$\mathbf{P} = \{(\mathsf{inp}_\mu)_{\mu\in[s]}, \{\mathbf{P}_{i,\boldsymbol{b}} \in \{0,1\}^{w\times w}\}_{i\in[h],\boldsymbol{b}\in\{0,1\}^s}, \mathcal{P}_0, \mathcal{P}_1 \subset \mathbb{Z}^{w\times w}\}.$$

The evaluation of $\mathbf{P}$ on input $\mathbf{x} = (x_i)_{i\in[\ell]} \in \{0,1\}^\ell$ is computed by

$$\mathbf{P}(\mathbf{x}) = \begin{cases} 0 & \text{if } \prod_{i=1}^h \mathbf{P}_{i,(x_{\mathsf{inp}_\mu(i)})_{\mu\in[s]}} \in \mathcal{P}_0 \\ 1 & \text{if } \prod_{i=1}^h \mathbf{P}_{i,(x_{\mathsf{inp}_\mu(i)})_{\mu\in[s]}} \in \mathcal{P}_1 \end{cases}.$$

When $s = 1$ ($s = 2$), the BP is called a single-input (dual-input) BP. In this paper, we usually use $\mathcal{P}_0 = \mathbf{0}^{w\times w}$ and $\mathcal{P}_1$ is the set of all nonzero matrices in $\mathbb{Z}^{w\times w}$. Also, we call $\{\mathbf{P}_{i,\boldsymbol{b}}\}_{\boldsymbol{b}\in\{0,1\}^s}$ the i-th layer of the BP. Remark that CVW obfuscation and BGMZ obfuscation take as input different BP type (e.g. single and dual BP) and the required properties of BP for each obfuscation are different. Therefore, we mention the required properties used to construct an obfuscation again before describing each obfuscation.

2.2 Indistinguishability Obfuscation

Definition 2.2 (Indistinguishability Obfuscation) *A probabilistic polynomial time machine $\mathcal{O}$ is an indistinguishability obfuscation for a circuit class $\mathcal{C} = \{\mathcal{C}_\lambda\}$ if the following conditions are satisfied:*

- *For all security parameters $\lambda \in \mathbb{N}$, for all circuits $C \in \mathcal{C}_\lambda$, for all inputs $\mathbf{x}$, the following probability holds:*

$$\Pr\left[C'(\mathbf{x}) = C(\mathbf{x}) : C' \leftarrow \mathcal{O}(\lambda, C)\right] = 1.$$

- *For any p.p.t distinguisher D, there exists a negligible function α satisfying the following statement: For all security parameters $\lambda \in \mathbb{N}$ and all pairs of circuits C_0, $C_1 \in \mathcal{C}_\lambda$, $C_0(\mathbf{x}) = C_1(\mathbf{x})$ for all inputs $\mathbf{x}$ implies*

$$\left|\Pr\left[D(\mathcal{O}(\lambda, C_0)) = 1\right] - \Pr\left[D(\mathcal{O}(\lambda, C_1)) = 1\right]\right| \leq \alpha(\lambda).$$

2.3 Lattice Trapdoor Background

A lattice $\mathcal{L}$ of dimension n is a discrete additive subgroup of $\mathbb{R}^n$. If $\mathcal{L}$ is generated by the set $\{\mathbf{b}_1, \cdots, \mathbf{b}_n\}$, all elements in $\mathcal{L}$ are of the form $\sum_{i=1}^{n} x_i \cdot \mathbf{b}_i$ for some integers x_i's. In this case, the lattice $\mathcal{L}$ is called the full rank lattice. Throughout this paper, we only consider the full rank lattice. Now we give several definitions and lemmas used in this paper.

For any $\sigma > 0$, the Gaussian function on $\mathbb{R}^n$ centered at $\mathbf{c}$ with parameter σ is defined as

$$\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = e^{-\pi\|\mathbf{x}-\mathbf{c}\|/\sigma^2} \text{ for all } \mathbf{x} \in \mathbb{R}^n.$$

Definition 2.3 (Discrete Gaussian Distribution on Lattices) *For any element $\mathbf{c} \in \mathbb{R}^n$, $\sigma > 0$ and any full rank lattice $\mathcal{L}$ of $\mathbb{R}^n$, the discrete Gaussian distribution over $\mathcal{L}$ is defined as*

$$D_{\mathcal{L},\sigma,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{x})}{\rho_{\sigma,\mathbf{c}}(\mathcal{L})} \text{ for all } \mathbf{x} \in \mathcal{L}$$

where $\rho_{\sigma,\mathbf{c}}(\mathcal{L}) = \sum_{\mathbf{x}\in\mathcal{L}} \rho_{\sigma,\mathbf{c}}(\mathbf{x})$.

Lemma 2.4 ([30]) *For integers $n \geq 1$, $q \geq 2$ and $m \geq 2n\log q$, there is a p.p.t algorithm $\mathsf{TrapSam}(1^n, 1^m, q)$ that outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a trapdoor τ such that $\mathbf{A}$ is statistically indistinguishable from $U(\mathbb{Z}_q^{n\times m})$ with a trapdoor τ.*

Lemma 2.5 ([21]) *There is a p.p.t. algorithm $\mathsf{Sample}(\mathbf{A}, \tau, \mathbf{y}, \sigma)$ that outputs a vector $\mathbf{d}$ from a distribution $D_{\mathbb{Z}^m,\sigma}$. Moreover, if $\sigma \geq 2\sqrt{n\log q}$, then with all but negligible probability, we have*

$$\{\mathbf{A}, \mathbf{d}, \mathbf{y} : \mathbf{y} \leftarrow U(\mathbb{Z}_q^n), \mathbf{d} \leftarrow \mathsf{Sample}(\mathbf{A}, \tau, \mathbf{y}, \sigma)\} \approx_s \{\mathbf{A}, \mathbf{d}, \mathbf{y} : \mathbf{d} \leftarrow D_{\mathbb{Z}^m,\sigma}, \mathbf{Ad} = \mathbf{y}\}.$$

3 Statistical Zeroizing Attack

In this section, we introduce our attack, *statistical zeroizing attack*. We give an abstract model for branching program obfuscation and the attack description in this model. In this attack, we are given two functionally equivalent branching programs $\mathbf{M}$ and $\mathbf{N}$, which will be specified later, and an obfuscated program $\mathcal{O}(\mathbf{P})$ for $\mathbf{P} = \mathbf{M}$ or $\mathbf{N}$. Our purpose is to distinguish whether $\mathbf{P} = \mathbf{M}$ or $\mathbf{P} = \mathbf{N}$. The targeted branching programs of the obfuscation output 0 when the product corresponding to input is zero. The obfuscated program $\mathcal{O}(\mathbf{P})$ consists of

$$\left\{\mathbf{S}, \{\mathbf{D}_{i,\boldsymbol{b}}\}_{1\le i\le h, \boldsymbol{b}\in\{0,1\}^s}, \mathbf{T}, \mathsf{inp} = (\mathsf{inp}_1, \cdots, \mathsf{inp}_s) : [h] \to [\ell]^s, B\right\}$$

where every element is a matrix over $\mathbb{Z}_q$ (possibly identity) except the input function inp. The output of the obfuscated program at $\mathbf{x} = (x_1, \cdots, x_\ell) \in \{0,1\}^\ell$ is computed by considering the value

$$\mathcal{O}(\mathbf{P})(\mathbf{x}) = \mathbf{S} \cdot \prod_{i=1}^{h} \mathbf{D}_{i,\mathbf{x}_{\mathsf{inp}}(i)} \cdot \mathbf{T}$$

where $\boldsymbol{x}_{\mathsf{inp}(i)} = (x_{\mathsf{inp}_1(i)}, \cdots, x_{\mathsf{inp}_s(i)})$. Note that $\mathcal{O}(\mathbf{P})(\mathbf{x})$ can be a matrix, vector or an element (over $\mathbb{Z}_q$). Regard it as matrix/vector/integer over $\mathbb{Z}$ and check the value: if $\|\mathcal{O}(\mathbf{P})(\mathbf{x})\|_\infty < B < q$ then it outputs 0, otherwise outputs 1. We call $\mathcal{O}(\mathbf{P})(\mathbf{x})$ the *evaluation* of the obfuscated program (at $\mathbf{x}$). We also call $\mathcal{O}(\mathbf{P})(\mathbf{x})$ evaluation of zero if $\mathbf{P}(\mathbf{x}) = 0$ in the plain program. We stress that the *output* and *evaluation* of the obfuscated program is different; the output of the obfuscated program is the same to output of original program, and the evaluation is the value $\mathcal{O}(\mathbf{P})(\mathbf{x})$, which is computed right before determining the output.

To distinguish two different obfuscated programs, we see the distribution of valid evaluations of zero of $\mathcal{O}(\mathbf{M})$ and $\mathcal{O}(\mathbf{N})$. For the evaluation of zero, the size of these products is far smaller than q (or B), thus we can obtain the integer value rather than the element in $\mathbb{Z}_q$. Now, if the evaluation is of the matrix or vector form, we consider only the first entry, namely $(1,1)$ entry of the matrix or the first entry of the vector, in the whole procedure of the attack. We call all of these entries by *the first entry* of the evaluation, including the case of the evaluation is just a real value.

Our strategy is to compute the sample variance of the first entries of many independent evaluations which follow the same distribution. The key of the attack is that this variance heavily depends on the plain program of the obfuscated program and the variance is sufficiently different to distinguish for two certain programs. Therefore, from the variance of the several evaluations, we can decide that the obfuscated program is from which program.

Note that one can sample an element following the distribution of obfuscation or its evaluation at fixed point $\mathbf{x} = \mathbf{x}_0$ in polynomial time when the corresponding program is given, since there is no private key in the obfuscation procedure. In this regard, we consider a more general problem which is easier to analyze: Given two polynomial-time constructible distribution $\mathcal{D}_{\mathbf{M}}$ and

$\mathcal{D}_{\mathbf{N}}$ and x sampled from one of them, determine that the sample is from which distribution. In our scenario, $\mathcal{D}_{\mathbf{M}}$ and $\mathcal{D}_{\mathbf{N}}$ are the distribution of $\mathcal{O}(\mathbf{M})(\mathbf{x})$ and $\mathcal{O}(\mathbf{N})(\mathbf{x})$, respectively where the distribution is over all randomness to construct obfuscations.

Since the adversary has one sample in our setting, the actual algorithm proceeds by sampling multiple evaluations itself as follows.

Data: $\mathcal{D}_{\mathbf{M}}, \mathcal{D}_{\mathbf{N}}, x, \kappa$

1. set $B = (\sigma_{\mathbf{M}}^2 + \sigma_{\mathbf{N}}^2)/2$ for $\sigma_{\mathbf{M}}^2 = Var(\mathcal{D}_{\mathbf{M}})$ and $\sigma_{\mathbf{N}}^2 = Var(\mathcal{D}_{\mathbf{N}})$
2. $i \leftarrow [\kappa]$ and let $s_i = x$
3. sample $\{s_j\}_{j\in[i-1]}$ from $\mathcal{D}_{\mathbf{M}}$ and $\{s_j\}_{i+1\le j\le\kappa}$ from $\mathcal{D}_{\mathbf{N}}$
4. compute the sample variance S^2 of $\{s_j\}_{j\in[\kappa]}$
5. if $S^2 < B$, decides $\mathcal{D}_{\mathbf{M}}$, otherwise $\mathcal{D}_{\mathbf{N}}$.

The choice of κ is specified later in Proposition 3.1. We also remark that the overall time complexity of algorithm is $O(\kappa \cdot T_{\mathsf{sample}})$ plus small computation for sample variance, where T_{sample} is the time complexity for sampling algorithms. The advantage of this algorithm is, by the standard hybrid arguemnt, $\mathsf{adv}_{\mathsf{mult}}/\kappa$ where $\mathsf{adv}_{\mathsf{mult}} = 0.98$ is the advantage of distinguishing algorithm by sample variance when κ samples are given as inputs instead of one sample as in Proposition 3.1.

In the next subsection, we analyze the distinguishing algorithm using sample variance for general distributions instead of iO when the multiple samples are given. Then we go back to the actual attack for iO for the concrete obfuscations in Section 4 and 5 by showing the attack conditions hold well.

3.1 Distinguishing Distributions using Sample Variance

Now we give the detailed analysis of distinguishing by sample variance. In this algorithm, we compute the variance of the samples, and check whether the distance between the sample variance and the expected variance of $\mathcal{D}_{\mathbf{M}}$ and $\mathcal{D}_{\mathbf{N}}$. If the distance from the sample variance to the variance of $\mathcal{D}_{\mathbf{M}}$ is less than the distance to the variance of $\mathcal{D}_{\mathbf{N}}$, we decide the given samples are from $\mathcal{D}_{\mathbf{M}}$. Otherwise we decide the samples are from $\mathcal{D}_{\mathbf{N}}$. The result of this method is stated in the following proposition.

Proposition 3.1 *Suppose that two random variables $X_{\mathbf{M}}$ and $X_{\mathbf{N}}$ that follow polynomial time constructible distributions $\mathcal{D}_{\mathbf{N}}$ and $\mathcal{D}_{\mathbf{M}}$ and have the means $\mu_{\boldsymbol{M}}$ and $\mu_{\boldsymbol{N}}$ and the variances $\sigma_{\mathbf{N}}^2$ and $\sigma_{\mathbf{M}}^2$, respectively. For the security parameter λ and polynomials $p, q, r = \mathsf{poly}(\lambda)$, there is a polynomial time algorithm that distinguishes $\mathcal{D}_{\mathbf{M}}$ and $\mathcal{D}_{\mathbf{N}}$ with non-negligible advantage when $O(p\cdot(\sqrt{q}+\sqrt{r})) = poly(\lambda)$ independent samples from $\mathcal{D}_{\mathbf{P}}$ are given and the following conditions hold:*

$$\left|\frac{\max(\sigma_{\boldsymbol{N}}^2, \sigma_{\boldsymbol{M}}^2)}{\sigma_{\boldsymbol{N}}^2 - \sigma_{\boldsymbol{M}}^2}\right| \le p \quad \left|\frac{E[(X_{\mathbf{N}} - \mu_{\mathbf{N}})^4]}{\sigma_{\boldsymbol{N}}^4}\right| \le q, \text{ and } \left|\frac{E[(X_{\mathbf{M}} - \mu_{\mathbf{M}})^4]}{\sigma_{\boldsymbol{M}}^4}\right| \le r.$$

In other words, if two known distributions satisfy the conditions, we can solve the distinguishing problem of two distribution with multiple samples. Thus to cryptanalyze the concrete obfuscation schemes, it suffice to show the conditions in Proposition 3.1. We conclude this section by giving the proof of this proposition.

Proof (Proposition 3.1). We call a definition and useful lemmas first.

Lemma 3.2 (Chebyshev's inequality) *Let X be a random variable with a finite expected value μ and a finite variance $\sigma^2 > 0$. Then, it holds that*

$$\Pr[|X - \mu| \geq k\sigma] \leq 1/k^2$$

for any real number $k > 0$.

Definition 3.3 (Sample variance) *Given random n samples $x_1, x_2, \cdots, x_n$ of $\mathcal{D}$, the sample variance of $\mathcal{D}$ is defined by*

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the sample mean.

Definition 3.4 (Kurtosis) *Let X be a random variable with a finite expected value μ and a finite variance $\sigma^2 > 0$. The kurtosis of X is defined by*

$$Kurt[X] = \frac{E[(X-\mu)^4]}{E[(X-\mu)^2]^2} = \frac{E[(X-\mu)^4]}{\sigma^4}.$$

Lemma 3.5 *Let S^2 be the sample variance of size κ samples of a distribution $\mathcal{D}$. Let X be a random variable following $\mathcal{D}$ and $\mu_n = E[(X - E[X])^n]$ be the n-th central moment. Then the variance of S^2 satisfies*

$$Var(S^2) = \frac{1}{\kappa}\left(\mu_4 - \frac{\kappa-3}{\kappa-1}\mu_2^2\right).$$

Now we return to the proof. Suppose that all of the conditions hold for polynomials $p, q, r \in \mathsf{poly}(\lambda)$ and $\sigma^2_{\mathbf{M}} < \sigma^2_{\mathbf{N}}$. By Lemma 3.2 and 3.5, we compute the 99% confidence interval of variance of S^2 as follows

$$\Pr\left[|S^2 - \sigma^2_{\boldsymbol{P}}| \geq 10 \cdot \sqrt{\frac{1}{\kappa}\cdot\left(E[(X_{\boldsymbol{P}} - \mu_{\boldsymbol{P}})^4] - \frac{\kappa-1}{\kappa-3}\cdot\sigma^4_{\boldsymbol{P}}\right)}\right] \leq \frac{1}{100}$$

with κ number of samples. If κ is sufficiently large, the two intervals of sample variance for $\mathbf{M}$ and $\mathbf{N}$ are disjoint. So we can distinguish two distributions by checking the size of sample variance.

More precisely, if $\kappa \geq 100 \cdot (p \cdot \sqrt{q} + p \cdot \sqrt{r})^2$ that is $\mathsf{poly}(\lambda)$, we have $\sigma^2_{\boldsymbol{M}} + 10\sigma^2_{\boldsymbol{M}} \cdot \sqrt{\frac{1}{\kappa}\cdot\left(\frac{E[(X_{\mathbf{M}}-\mu_{\mathbf{M}})^4]}{\sigma^4_{\mathbf{M}}} - \frac{\kappa-1}{\kappa-3}\right)} < \sigma^2_{\boldsymbol{N}} - 10\sigma^2_{\boldsymbol{N}} \cdot \sqrt{\frac{1}{\kappa}\cdot\left(\frac{E[(X_{\mathbf{N}}-\mu_{\boldsymbol{N}})^4]}{\sigma^4_{\mathbf{N}}} - \frac{\kappa-1}{\kappa-3}\right)}$.

Thus the algorithm decides the answer by checking if the sample variance is included in which interval; we do not care the case that it is not included both. This algorithm succeeds with probability at least 0.99 for each input, i.e. the advantage of algorithm is at least 0.98. Note that this algorithm only does the polynomial number of sampling and computing the variance, thus the running time is polynomial. □

4 Cryptanalysis of CVW Obfuscation

In this section, we briefly describe the construction of CVW obfuscation scheme and show that the statistical zeroizing attack works well for CVW obfuscation.

4.1 Construction of CVW Obfuscation

Chen, Vaikuntanathan and Wee proposed a new candidate of iO which is robust against all existing attacks. We here give a brief description of the candidate scheme. For more details, we refer to original paper [10].

First, we start with the description of BPs they used. The authors use single-input binary BPs, *i.e.*, $\mathsf{inp} = \mathsf{inp}_1$. They employ a new function, called an input-to-index map $\bar{\omega}$: $\{0,1\}^\ell \to \{0,1\}^h$ such that $\bar{\omega}(\mathbf{x})_i = \mathbf{x}_{\mathsf{inp}(i)}$ for all $i \in [h]$, $\mathbf{x} \in \{0,1\}^\ell$. As used in the paper [10], we denote the $\prod_{i=1}^h \mathbf{M}_{i,\bar{\omega}(\mathbf{x})_i}$ by $\mathbf{M}_{\bar{\omega}(\mathbf{x})}$ or simply $\mathbf{M}_{\mathbf{x}}$. We sometimes abuse the notion $\mathbf{M}_{i,x_i}$ to denote $\mathbf{M}_{i,\bar{\omega}(\mathbf{x})_i}$.

A target BP $\mathbf{P} = \{\mathsf{inp}, \{\mathbf{P}_{i,b}\}_{i\in[h],b\in\{0,1\}}, \mathcal{P}_0, \mathcal{P}_1\}$, which is called *Type I* BP in the original paper, satisfies the following conditions.

1. All the matrices $\mathbf{P}_{i,b}$ are $w \times w$ matrices.
2. For a vector $\mathbf{v} = \mathbf{1}^{1\times w}$, the target sets $\mathcal{P}_0, \mathcal{P}_1$ satisfies $\mathbf{v}\cdot\mathcal{P}_0 = \{\mathbf{0}^{1\times w}\}$, $\mathbf{v}\cdot\mathcal{P}_1 \neq \{\mathbf{0}^{1\times w}\}$.[4]
3. An index length h is set to $(\lambda+1)\cdot\ell$ with the security parameter λ.
4. An index-to-input function satisfies $\mathsf{inp}(i) = (i \mod \ell)$. Thus, index-to-input function iterates $\lambda+1$ times.

Construction. CVW obfuscation is a probabilistic polynomial time algorithm which takes as input a BP $\mathbf{P}$ with an input length ℓ, and outputs an obfuscated program preserving the functionality. The algorithm process consists of the following steps. Here we use new parameters $n, m, q, t := (w+2n\ell)\cdot n, \sigma$ for the construction. We will specify the parameter settings later.

- Sample bundling matrices $\{\mathbf{R}_{i,b} \in \mathbb{Z}^{2n\ell\times 2n\ell}\}_{i\in[h],b\in\{0,1\}}$ such that $(\mathbf{1}^{1\times 2\ell}\otimes\mathbf{I}^{n\times n})\cdot\mathbf{R}_{\mathbf{x}'}\cdot(\mathbf{1}^{2\ell\times 1}\otimes\mathbf{I}^{n\times n}) = \mathbf{0} \iff \mathbf{x}' \in \bar{\omega}(\{0,1\}^\ell)$ for all $\mathbf{x}' \in \{0,1\}^h$.

[4] As noted in the remark of introduction, it is assumed implicitly that $\mathbf{v} = \mathbf{1}^{1\times w}$ for the targeted BP, while the definition of Type I BP uses $\mathbf{v} \in \{0,1\}^{1\times w}$.

More precisely, $\mathbf{R}_{i,b}$ is a block diagonal matrix $\mathsf{diag}(\mathbf{R}_{i,b}^{(1)}, \mathbf{R}_{i,b}^{(2)}, \cdots, \mathbf{R}_{i,b}^{(\ell)})$. Each $\mathbf{R}_{i,b}^{(k)} \in \mathbb{Z}^{2n \times 2n}$ is one of the following three cases.

$$\mathbf{R}_{i,b}^{(k)} = \begin{cases} \mathbf{I}^{2n \times 2n} & \text{if } \mathsf{inp}(i) \neq k \\ \begin{pmatrix} \tilde{\mathbf{R}}_{i,b}^{(k)} & \\ & \mathbf{I}^{n \times n} \end{pmatrix}, \tilde{\mathbf{R}}_{i,b}^{(k)} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{n \times n} & \text{if } \mathsf{inp}(i) = k \text{ and } i \leq \lambda\ell \\ \begin{pmatrix} -\mathbf{I}^{n \times n} & \\ & \prod_{j=0}^{\lambda-1} \tilde{\mathbf{R}}_{k+j\ell,b}^{(k)} \end{pmatrix} & \text{if } \mathsf{inp}(i) = k \text{ and } i > \lambda\ell \end{cases}$$

- Sample matrices $\{\mathbf{S}_{i,b} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{n \times n}\}_{i \in [h], b \in \{0,1\}}$ and compute

$$\mathbf{J} := (\mathbf{1}^{1 \times (w+2n\ell)} \otimes \mathbf{I}^{n \times n}) \in \mathbb{Z}^{n \times t}$$
$$\hat{\mathbf{S}}_{i,b} := \begin{pmatrix} \mathbf{P}_{i,b} \otimes \mathbf{S}_{i,b} & \\ & \mathbf{R}_{i,b} \otimes \mathbf{S}_{i,b} \end{pmatrix} \in \mathbb{Z}^{t \times t}$$
$$\mathbf{L} := (\mathbf{1}^{(w+2n\ell) \times 1} \otimes \mathbf{I}^{n \times n}) \in \mathbb{Z}^{t \times n}$$

- Sample $(\mathbf{A}_i, \tau_i) \leftarrow \mathsf{TrapSam}(1^t, 1^m, q)$ for $0 \leq i \leq h-1$, $\mathbf{A}_h \leftarrow U(\mathbb{Z}_q^{n \times n})$, $\{\mathbf{E}_{i,b} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{t \times m}\}_{i \in [h-1], b \in \{0,1\}}$ and $\{\mathbf{E}_{h,b} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{t \times n}\}_{b \in \{0,1\}}$.
- Run Sample algorithms to obtain

$$\mathbf{D}_{i,b} \in \mathbb{Z}^{m \times m} \leftarrow \mathsf{Sample}(\mathbf{A}_{i-1}, \tau_{i-1}, \hat{\mathbf{S}}_{i,b} \cdot \mathbf{A}_i + \mathbf{E}_{i,b}, \sigma) \text{ for } 1 \leq i \leq h-1,$$
$$\mathbf{D}_{h,b} \in \mathbb{Z}^{m \times n} \leftarrow \mathsf{Sample}(\mathbf{A}_{h-1}, \tau_{h-1}, \hat{\mathbf{S}}_{h,b} \cdot \mathbf{L} \cdot \mathbf{A}_h + \mathbf{E}_{h,b}, \sigma).$$

- Define $\mathbf{A}_{\mathbf{J}}$ as a matrix $\mathbf{J} \cdot \mathbf{A}_0 \in \mathbb{Z}^{n \times m}$ and outputs matrices

$$\left\{\mathsf{inp}, \mathbf{A}_{\mathbf{J}}, \{\mathbf{D}_{i,b}\}_{i \in [h], b \in \{0,1\}}\right\}.$$

Evaluation. Evaluation process consists of two steps. The first step is to compute a matrix $\mathbf{A}_{\mathbf{J}} \cdot \mathbf{D}_{\bar{\omega}(\mathbf{x})} \bmod q$. The last step is size comparison: If $\|\mathbf{A}_{\mathbf{J}} \cdot \mathbf{D}_{\bar{\omega}(\mathbf{x})} \bmod q\|_\infty \leq B$, output 0 for some fixed B. Otherwise, output 1.

Parameters. Let λ and λ_{LWE} for the security parameters of obfuscation itself and underlying LWE problem satisfying $\lambda_{LWE} = \mathsf{poly}(\lambda)$ and the following constraints. Set $n = \Omega(\lambda_{LWE} \log q)$ and $\chi = D_{\mathbb{Z}, 2\sqrt{\lambda_{LWE}}}$. Moreover, for the trapdoor functionality, $m = \Omega(t \log q)$ and $\sigma = \Omega(\sqrt{t \log q})$ for $t = (w + 2n\ell) \cdot n$. $B \geq (w+2n\ell) \cdot h \cdot (m \cdot \sigma^2 \sqrt{n(w+2n\ell)}\sigma)^h$ and $q = B \cdot \omega(\mathsf{poly}(\lambda))$ for correctness, and $q \leq (\sigma/\lambda_{LWE}) \cdot 2^{\lambda_{LWE}^{1-\epsilon}}$ for a fixed $\epsilon \in (0,1)$ for security. For more details, we refer readers to the original paper [10].

Remark 1. The original paper [10] only uses one security parameter λ, but the correctness does not hold in that setting. Instead, the trick that uses two security parameters λ and λ_{LWE} resolves this problem as in [4].

Zerotest Functionality. From the construction of the obfuscation, the following equality always holds, which is essentially what we need.

$$[\mathbf{A}_{\mathbf{J}} \cdot \mathbf{D}_{\bar{\omega}(\mathbf{x})}]_q = \left[\mathbf{J} \cdot \left(\prod_{i=1}^{h} \hat{\mathbf{S}}_{i,x_i} \right) \cdot \mathbf{A}_h + \mathbf{J} \cdot \sum_{j=1}^{h} \left(\left(\prod_{i=1}^{j-1} \hat{\mathbf{S}}_{i,x_i} \right) \cdot \mathbf{E}_{j,x_j} \cdot \prod_{k=j+1}^{h} \mathbf{D}_{k,x_k} \right) \right]_q$$

The honest evaluation with $\mathbf{P}_{\mathbf{x}} = \mathbf{0}^{w \times w}$ gives $\hat{\mathbf{S}}_{\mathbf{x}} = \mathbf{0}^{t \times t}$ due to the construction of $\mathbf{R}_{i,b}$ is zero for the valid evaluation. Then, the following inequality holds:

$$\begin{aligned}
\|[\mathbf{A}_{\mathbf{J}} \cdot \mathbf{D}_{\bar{\omega}(\mathbf{x})}]_q\|_\infty &= \left\| \left[\mathbf{J} \cdot \sum_{j=1}^{h} \left(\left(\prod_{i=1}^{j-1} \hat{\mathbf{S}}_{i,x_i} \right) \cdot \mathbf{E}_{j,x_j} \cdot \prod_{k=j+1}^{h} \mathbf{D}_{k,x_k} \right) \right]_q \right\|_\infty && (1) \\
&\leq \left\| \mathbf{J} \cdot \sum_{j=1}^{h} \left(\left(\prod_{i=1}^{j-1} \hat{\mathbf{S}}_{i,x_i} \right) \cdot \mathbf{E}_{j,x_j} \cdot \prod_{k=j+1}^{h} \mathbf{D}_{k,x_k} \right) \right\|_\infty && (2) \\
&\leq h \cdot \left(\max_{i,b} \|\hat{\mathbf{S}}_{i,b}\| \cdot \sigma \cdot m \right)^h \leq B && (3)
\end{aligned}$$

for all but negligible probability due to the choice of B. If $\mathbf{P}_{\mathbf{x}}$ is not the zero matrix, then $\hat{\mathbf{S}}_{\mathbf{x}}$ is also not the zero matrix with overwhelming probability. It implies that $\|[\mathbf{A}_{\mathbf{J}} \cdot \mathbf{D}_{\bar{\omega}(\mathbf{x})}]_q\|_\infty$ is larger than B with overwhelming probability because of $\mathbf{A}_h \leftarrow U(\mathbb{Z}_q^{n \times n})$.

4.2 Cryptanalysis of CVW Obfuscation

We apply the statistical zeroizing attack to the CVW obfuscation. As stated in Section 3, it is enough to show that the conditions of Proposition 3.1 hold. We only consider small variance σ^2 so that $\sigma = \mathsf{poly}(\lambda)$, and sufficiently large ℓ.[5] This includes the optimal parameter choice as well.

Our targeted two functionally equivalent BPs $\mathbf{M} = \{\mathbf{M}_{i,b}\}_{i \in [h], b \in \{0,1\}}$ and $\mathbf{N} = \{\mathbf{N}_{i,b}\}_{i \in [h], b \in \{0,1\}}$ are of the form

$$\mathbf{M}_{i,b} = \mathbf{0}^{w \times w} \text{ for all } i, b \text{ and } \mathbf{N}_{i,b} = \begin{cases} \mathbf{1}^{w \times w} & \text{if } i = 1 \\ \mathbf{0}^{w \times w} & \text{otherwise} \end{cases}.$$

Suppose that we have an obfuscated program $\mathcal{O}(\mathbf{P})$ for $\mathbf{P} = \mathbf{M}$ or $\mathbf{P} = \mathbf{N}$. Our goal is to determine whether the program $\mathcal{O}(\mathbf{P})$ is an obfuscation of $\mathbf{M}$ or $\mathbf{N}$.

By the standard hybrid argument, it suffices to distinguish the distributions $\mathcal{D}_{\mathbf{M}}$ or $\mathcal{D}_{\mathbf{N}}$ where $\mathcal{D}_{\mathbf{M}}$ and $\mathcal{D}_{\mathbf{N}}$ is the distributions of the (1,1) entry of evaluation at a fixed vector $\mathbf{x}$ of the obfuscated program of $\mathbf{M}$ or $\mathbf{N}$, respectively. To exploit Proposition 3.1, we transform the CVW construction into the language

[5] Indeed, the attack requires the condition $\sigma^4 < m^\ell / n^{\ell+1}$.

of random variables. We denote the random matrix by the capital italic words whose entry follows a distribution that corresponds to the distribution of entry of the bold matrix. For example, the entry of random matrix $E_{i,b}$ follows the distribution $\mathcal{D}_{\mathbb{Z},\sigma}$ since the matrix $\mathbf{E}_{i,b}$ is chosen from $\mathcal{D}_{\mathbb{Z},\sigma}^{t\times m}$ in the CVW construction. More precisely, we define random matrices $\tilde{R}_{i,b}^{(k)}$ following $\mathcal{D}_{\mathbb{Z},\sigma}^{n\times n}$, $S_{i,b}$ following $\mathcal{D}_{\mathbb{Z},\sigma}^{n\times n}$ and A_i as in the trapdoor sampling algorithm. Then we obtain random matrices $\hat{S}_{i,b}^{(\mathbf{P})}$, $R_{i,b}^{(\mathbf{P})}$, $E_{i,b}^{(\mathbf{P})}$ and $D_{i,b}^{(\mathbf{P})}$ as in the construction of CVW obfuscation for the branching programs $\mathbf{P} = \mathbf{M}$ or $\mathbf{N}$. We note that only $\hat{S}_{i,b}^{(\mathbf{P})}$ and $D_{i,b}^{(\mathbf{P})}$ depend on the choice of branching program, but we put $\mathbf{P}$ in some other random variables for convenience of distinction.

Under this setting, it suffices to show the following proposition.

Proposition 4.1 *For a security parameter λ, fix the Gaussian variance parameter $\sigma = \mathsf{poly}(\lambda)$. Then, there are two functionally equivalent branching programs $\mathbf{M}$ and $\mathbf{N}$ with sufficiently large input length ℓ satisfying the following statement: let $Z_{\mathbf{M}}$ and $Z_{\mathbf{N}}$ be random variables satisfying*

$$Z_{\mathbf{M}} = \left[\left(\mathbf{J}\cdot A_0\cdot D_{\bar{\omega}(\mathbf{x})}^{(\mathbf{M})}\right)_{(1,1)}\right]_q, \quad Z_{\mathbf{N}} = \left[\left(\mathbf{J}\cdot A_0\cdot D_{\bar{\omega}(\mathbf{x})}^{(\mathbf{N})}\right)_{(1,1)}\right]_q$$

where every random matrix is defined as the above. Let $\mu_{\mathbf{M}}$ and $\mu_{\mathbf{N}}$, $\sigma_{\mathbf{M}}^2$ and $\sigma_{\mathbf{N}}^2$ be mean and variance of the random variables of $Z_{\mathbf{M}}$ and $Z_{\mathbf{N}}$, respectively. Then, it holds that

$$\left|\frac{\max(\sigma_{\boldsymbol{N}}^2,\sigma_{\boldsymbol{M}}^2)}{\sigma_{\boldsymbol{N}}^2-\sigma_{\boldsymbol{M}}^2}\right| \le p, \quad \left|\frac{E[(Z_{\mathbf{N}}-\mu_{\mathbf{N}})^4]}{\sigma_{\boldsymbol{N}}^4}\right| \le q, \text{ and } \quad \left|\frac{E[(Z_{\mathbf{M}}-\mu_{\mathbf{M}})^4]}{\sigma_{\mathbf{M}}^4}\right| \le q.$$

for some $p, q = \mathsf{poly}(\lambda)$ under Assumption 1.

We remark that since the random matrices D's are dependent each other, we need to assume the statistical property for verifying conditions of Proposition 4.1 as follows.

Assumption 1 *For an integer $0 \le k \le h-2$ and $\mathbf{P} = \mathbf{M}$ or $\mathbf{N}$, let $\hat{D}_k^{(\mathbf{P})}$ be a random matrix such that $\hat{D}_k^{(\mathbf{P})} = \prod_{i=k+2}^{h} D_i^{(\mathbf{P})}$, where $D_i^{(\mathbf{P})}$ is the random matrix which follows a distribution corresponding preimage-sampled matrix $\mathbf{D}_i^{(\mathbf{P})}$. Then, the following equations hold*

1. *the variance is approximated by the same one assumed that D's are independent Gaussian, that is, it holds that*

$$Var[\hat{D}_k^{(\mathbf{P})}] = \Theta\left(m^{h-k-2}(\sigma^2)^{h-k-1}\right).$$

2. *the kurtosis is bounded by constant, that is, it holds that*

$$\frac{E[(\hat{D}_k{}^{(\mathbf{P})} - E[\hat{D}_k{}^{(\mathbf{P})}])^4]}{Var[\hat{D}_k{}^{(\mathbf{P})}]^2} = O(\mathsf{poly}(\lambda)).$$

We experimentally verify this assumption using the implementation of GGH15 BP obfuscation by Halevi *et al.* [23]. More detailed experimental results are presented in Appendix C. We remark that if we assume that D's are independent matrices that have discrete Gaussian entry with the variance σ^2, the following computations hold:

- the variance of $\hat{D}_k^{(\mathbf{P})}$ is exactly $m^{h-k-2} \cdot (\sigma^2)^{h-k-1}$, and
- the kurtosis of $\hat{D}_k^{(\mathbf{P})}$ is $3 \cdot (1 + 2/m)^{h-k} = \Theta(1)$.

The honest evaluation of the CVW obfuscation $[\mathbf{A}_{\mathbf{J}} \cdot \mathbf{D}^{(\mathbf{P})}_{\bar{\omega}(\mathbf{x})}]_q$ is the matrix of the form

$$\mathbf{J} \cdot \sum_{j=0}^{h-1} \left(\left(\prod_{i=1}^{j} \hat{\mathbf{S}}_{i,x_i} \right) \cdot \mathbf{E}_{j+1,x_{j+1}} \cdot \prod_{k=j+2}^{h} \mathbf{D}^{(\mathbf{P})}_{k,x_k} \right),$$

which does not contain the term including the trapdoor matrices $\mathbf{A}_i$ for $i = 0, \cdots, h-1$. Thus, to establish the statistical properties including variance in Proposition 4.1, it suffices to analyze the statistical properties of the random matrices $\hat{S}^{(\mathbf{P})}_{i,b}$, $E^{(\mathbf{P})}_{i,b}$, $D^{(\mathbf{P})}_{i,b}$ and their products.

By the definition of $Z_{\mathbf{P}}$ with $\mathbf{P} = \mathbf{M}$ or $\mathbf{P} = \mathbf{N}$, it is rewritten as

$$Z_{\mathbf{P}} = \mathbf{J} \cdot \sum_{j=0}^{h-1} \left(\left(\prod_{i=1}^{j} \hat{S}_{i,x_i} \right) \cdot E_{j+1,x_{j+1}} \cdot \prod_{k=j+2}^{h} D^{(\mathbf{P})}_{k,x_k} \right).$$

Now we give the lemmas to prove Proposition 4.1. The proofs of lemmas can be found in the full version of this paper [11]. The proof of Proposition 4.1 using the lemmas is placed in the concluding part of this section.

For the convenience of the statement, let $(Z^{(\mathbf{M})}_{1,1})_j$ be random variables of $(1,1)$-th entry of the random matrices

$$\mathbf{J} \cdot \prod_{i=1}^{j} \hat{S}^{(\mathbf{M})}_i \cdot E^{(\mathbf{M})}_{j+1} \cdot \prod_{k=j+2}^{h} D^{(\mathbf{M})}_k$$

for $j = 0, 1, \cdots, h-1$. In this notation, $Z_{\mathbf{M}}$ is the summation of $(Z^{(\mathbf{M})}_{1,1})_j$ for $j \in \{0, 1, \cdots, h-1\}$. Similarly, we define $(Z^{(\mathbf{N})}_{1,1})_j$ for all $j = 0, \cdots, h-1$. We employ additional notations constants c, d and (possibly polynomial) c_0 such that for all $0 \leq k \leq h-2$,

$$c \leq \frac{Var[\hat{D}^{(\mathbf{P})}_k]}{m^{h-k-2}(\sigma^2)^{h-k-1}} \leq d \quad \text{and} \quad \frac{E[(\hat{D}_k{}^{(\mathbf{P})} - E[\hat{D}_k{}^{(\mathbf{P})}])^4]}{Var[\hat{D}_k{}^{(\mathbf{P})}]^2} \leq c_0.$$

We remark that variances of many terms for $\mathbf{M}$ and $\mathbf{N}$ are *exactly the same* since the only D_1, $\hat{S}_1$ are different and the different terms in products of $\hat{S}$ are canceled for $j \geq 2$. Note that most of lemmas hold under Assumption 1, but we omit this repeated statement *under Assumption 1* for brevity.

Lemma 4.2 $E[(Z_{1,1}^{(\mathbf{M})})_j] = E[(Z_{1,1}^{(\mathbf{N})})_j] = 0$ *for all* $j = 0, \cdots, h-1$.

Lemma 4.3 $E[(Z_{1,1}^{(\mathbf{M})})_{\mu_1} \cdot (Z_{1,1}^{(\mathbf{M})})_{\mu_2}] = E[(Z_{1,1}^{(\mathbf{N})})_{\mu_1} \cdot (Z_{1,1}^{(\mathbf{N})})_{\mu_2}] = 0$ *for* $\mu_1 \neq \mu_2$.

Lemma 4.4 ($j = 0$) *It holds that*

$$Var[(Z_{1,1}^{(\mathbf{M})})_0] = Var[(Z_{1,1}^{(\mathbf{N})})_0] = \Theta\left((w + 2n\ell) \cdot m^{h-1} \cdot \sigma^{2h}\right) \text{ and}$$
$$\left|\frac{E[(Z_{1,1}^{(\mathbf{M})})_0^4]}{Var[(Z_{1,1}^{(\mathbf{M})})_0]^2}\right|, \left|\frac{E[(Z_{1,1}^{(\mathbf{N})})_0^4]}{Var[(Z_{1,1}^{(\mathbf{N})})_0]^2}\right| \leq 3c_0 \cdot (w + 2n\ell)^2 \cdot m^2 \cdot \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda).$$

Lemma 4.5 ($j = 1$) *It holds that*

$$Var[(Z_{1,1}^{(\mathbf{M})})_1] = \Theta\left(\left(n^3\sigma^2 + (2\ell - 1) \cdot n^2\right) \cdot m^{h-2}(\sigma^2)^h\right),$$
$$Var[(Z_{1,1}^{(\mathbf{N})})_1] = \Theta\left(w^3 \cdot n \cdot m^{h-2}(\sigma^2)^h\right) + Var[(Z_{1,1}^{(\mathbf{M})})_1]$$
$$\left|\frac{E[(Z_{1,1}^{(\mathbf{M})})_1^4]}{Var[(Z_{1,1}^{(\mathbf{M})})_1]^2}\right|, \left|\frac{E[(Z_{1,1}^{(\mathbf{N})})_1^4]}{Var[(Z_{1,1}^{(\mathbf{N})})_1]^2}\right| \leq 27c_0 \cdot (w + 2n\ell)^4 n^2 m^2 \cdot \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda).$$

Lemma 4.6 ($1 < j \leq \lambda \cdot \ell$) *Let* j *be a fixed integer with* $j = \ell \cdot j_1 + j_2 > 1$ *for* $0 \leq j_2 < \ell$ *and* $2 \leq j \leq \lambda \cdot \ell$. *Then, it holds that*

$$Var[(Z_{1,1}^{(\mathbf{M})})_j] = Var[(Z_{1,1}^{(\mathbf{N})})_j]$$
$$= \Theta\left(\left(j_2 n^{j+j_1+2}(\sigma^2)^{j_1+1} + (\ell - j_2)n^{j+j_1+1}(\sigma^2)^{j_1} + \ell n^{j+1}\right)m^{h-j-1}(\sigma^2)^h\right).$$

Moreover, it holds that

$$\left|\frac{E[(Z_{1,1}^{(\mathbf{M})})_j^4]}{Var[(Z_{1,1}^{(\mathbf{M})})_j]^2}\right|, \left|\frac{E[(Z_{1,1}^{(\mathbf{N})})_j^4]}{Var[(Z_{1,1}^{(\mathbf{N})})_j]^2}\right| \leq 27c_0(w + 2n\ell)^4 n^2 m^2 \left(1 + \frac{2}{n}\right)^{j_1+j-1} \left(\frac{d}{c}\right)^2$$
$$= \mathsf{poly}(\lambda).$$

Lemma 4.7 ($j > \lambda \cdot \ell$)) *Let* j *be a fixed integer with* $j = \ell \cdot j_1 + j_2 > 1$ *for* $0 \leq j_2 < \ell$ *and* $j > \lambda \cdot \ell$. *Then, it holds that*

$$Var[(Z_{1,1}^{(\mathbf{M})})_j] = Var[(Z_{1,1}^{(\mathbf{N})})_j]$$
$$= \Theta\left(\left((\ell + j_2) \cdot n^{\lambda+j+1} \cdot (\sigma^2)^\lambda + (\ell - j_2) \cdot n^{j+1}\right) \cdot m^{h-j-1} \cdot (\sigma^2)^h\right).$$

In addition, it holds that

$$\left|\frac{E[(Z_{1,1}^{(\mathbf{M})})_j^4]}{Var[(Z_{1,1}^{(\mathbf{M})})_j]^2}\right|, \left|\frac{E[(Z_{1,1}^{(\mathbf{N})})_j^4]}{Var[(Z_{1,1}^{(\mathbf{N})})_j]^2}\right| \leq 27c_0(w + 2n\ell)^4 n^2 m^2 \left(1 + \frac{2}{n}\right)^{\lambda+j-2} \left(\frac{d}{c}\right)^2$$
$$= \mathsf{poly}(\lambda).$$

Now we give a proof of the proposition 4.1 using above lemmas.

Proof (of Proposition 4.1). Fix ℓ be a sufficiently large so that $\sigma^4 < m^\ell/n^{\ell+1}$ and choose BP $\mathbf{M}$ and $\mathbf{N}$ as the given in the first page of this section. These two branching programs have the same functionality and length.

Using the results of lemmas, we can prove the proposition by analyzing the summation of random matrices. We first verify the results for $Z_{\mathbf{M}}$. The similar result holds for $Z_{\mathbf{N}}$ since the bounds of lemmas are almost same.

From Lemma 4.2, 4.3 and the definition of $Z_{\mathbf{M}}$, we have

$$Var[Z_{\mathbf{M}}] = E\left[(\sum_{j=0}^{h-1}(Z_{1,1}^{(\mathbf{M})})_j)^2\right] = E\left[\sum_{j=0}^{h-1}(Z_{1,1}^{(\mathbf{M})})_j^2\right] = \sum_{j=0}^{h-1} Var[(Z_{1,1}^{(\mathbf{M})})_j].$$

On the other hands, applying to the Cauchy-Schwarz inequality, it also holds

$$E[Z_{\mathbf{M}}^4] = E\left[(\sum_{j=0}^{h-1}(Z_{1,1}^{(\mathbf{M})})_j)^4\right] \leq E\left[h^3 \cdot (\sum_{j=0}^{h-1}(Z_{1,1}^{(\mathbf{M})})_j^4)\right].$$

When dividing both sides by $Var[Z_{\mathbf{M}}]^2$, we obtain the inequality

$$\begin{aligned}\left|\frac{E[Z_{\mathbf{M}}^4]}{Var[Z_{\mathbf{M}}]^2}\right| &\leq \left|\frac{E[h^3 \cdot (\sum_{j=0}^{h-1}(Z_{1,1}^{(\mathbf{M})})_j^4)]}{Var[Z_{\mathbf{M}}]^2}\right| = h^3 \cdot \left|\frac{E[\sum_{j=0}^{h-1}(Z_{1,1}^{(\mathbf{M})})_j^4]}{Var[Z_{\mathbf{M}}]^2}\right| \\ &= h^3 \cdot \sum_{j=0}^{h-1}\left|\frac{E[(Z_{1,1}^{(\mathbf{M})})_j^4]}{Var[Z_{\mathbf{M}}]^2}\right| \leq h^3 \cdot \sum_{j=0}^{h-1}\left|\frac{E[(Z_{1,1}^{(\mathbf{M})})_j^4]}{Var[(Z_{1,1}^{(\mathbf{M})})_j]^2}\right|.\end{aligned}$$

By Lemma 4.4,4.5,4.6 and 4.7, $\left|\frac{E[(Z_{1,1}^{(\mathbf{M})})_j^4]}{Var[(Z_{1,1}^{(\mathbf{M})})_j]^2}\right|$ is bounded by $\mathsf{poly}(\lambda)$ for all $j = 0, 1, \cdots, h-1$. Therefore, the following inequality holds.

$$\left|\frac{E[Z_{\mathbf{M}}^4]}{Var[Z_{\mathbf{M}}]^2}\right| \leq \mathsf{poly}(\lambda) =: q(\lambda)$$

The same holds for $\mathbf{N}$ as well.

Moreover, $Var[Z_{\mathbf{N}}] - Var[Z_{\mathbf{M}}] = \Theta\left(w^3 \cdot n \cdot m^{h-2}(\sigma^2)^h\right)$ holds by Lemma 4.5. Then the values $\left|Var[(Z_{1,1}^{(\mathbf{M})})_j]/(Var[Z_{\mathbf{N}}] - Var[Z_{\mathbf{M}}])\right|$ is bounded by $\mathsf{poly}(\lambda)$ for every j since $\sigma^4 < m^\ell/n^{\ell+1}$. This implies the first condition also holds. □

Remark 2. In the original paper [10], the authors give two different choice of the distributions of $\mathbf{E}_{i,b}$; $\mathcal{D}_{\mathbb{Z},\sigma}$ with corresponding dimension in Section 11, and $\chi = \mathcal{D}_{\mathbb{Z},2\sqrt{\lambda_{LWE}}}$ with appropriate dimension in Section 5. This paper focus on $\mathcal{D}_{\mathbb{Z},\sigma}$ but the result still holds for $\chi = \mathcal{D}_{\mathbb{Z},2\sqrt{\lambda_{LWE}}}$ with slight modification.

5 Cryptanalysis of BGMZ Obfuscation

In this section, we briefly review the BGMZ obfuscation and apply the statistical zeroizing attack on BGMZ obfuscation for exponentially large variance σ. Note that the security proof of BGMZ obfuscation under GGH15 zeroizing model (and underlying BPUA assumption) is independent of the parameter σ, so our attack implies that the algebraic security proof is not enough to achieve the ideal security of iO.

5.1 Construction of BGMZ Obfuscation

Bartusek *et al.* proposed a new candidate of iO which is provably secure in the GGH15 zeroizing model. We briefly review the construction of this scheme. For more detail, we refer to the original paper [4].

We start with the conditions of BP they used. The authors use a dual-input binary BP's. *i.e.*, $\mathsf{inp}(i) = (\mathsf{inp}_1(i), \mathsf{inp}_2(i))$. For simplicity, they use the notation $\boldsymbol{x}(i) = (x_{\mathsf{inp}_1(i)}, x_{\mathsf{inp}_2(i)})$. Moreover, they employ the new parameter $\eta = \mathsf{poly}(\ell, \lambda)$ with $\eta \geq \ell^4$ which decides the minimum number of the BP layer for the security parameter λ and input length ℓ.

The targeted BP $\mathbf{P}$ also satisfies the following conditions.

1. All the matrices $\{\mathbf{P}_{i,\boldsymbol{b}}\}_{i\in[h],\boldsymbol{b}\in\{0,1\}^2}$ are $w \times w$ matrices.
2. $\prod_{i=1}^{h} \mathbf{P}_{i,\boldsymbol{x}(i)} = \mathbf{0}^{w\times w}$.
3. Each pair of input bits (j, k) is read in at least $4\ell^2$ different layers of branching program.
4. There exist layers $i_1 < i_2 < \cdots < i_\eta$ such that $\mathsf{inp}_1(i_1), \cdots, \mathsf{inp}_1(i_\eta)$ cycles η/ℓ times through $[\ell]$.

To obfuscate a branching program that does not satisfy the condition 3 or 4, one pads the identity matrices to satisfy the conditions while preserving the functionality.

Remark 3. The original construction consider the straddling set and asymmetric level structures to prohibit *invalid* evaluations. The description below omitted them because our attack only exploits the valid evaluations whose results are the same regardless of them.

Construction. BGMZ obfuscation is a probabilistic polynomial time algorithm which takes as input a BP $\mathbf{P}$ with a length h, and outputs an obfuscated program with the same functionality. We use several parameter such as $n, m, q, t := (w + 1) \cdot n, \sigma, \nu, g$ in the construction. We will describe the setting for new parameters such as g, ν later.

The obfuscation procedure consists of the following steps.

- Sample $(\mathbf{A}_i, \tau_i) \leftarrow \mathsf{TrapSam}(1^t, 1^m, q)$ for $0 \leq i \leq h-1$, $\mathbf{A}_h \leftarrow U(\mathbb{Z}_q^{t\times m})$, $\{\mathbf{E}_{i,\boldsymbol{b}} \leftarrow \chi^{t\times m}\}_{i\in[h-1],\boldsymbol{b}\in\{0,1\}^2}$ and $\mathbf{E}_h \leftarrow \chi^{t\times m}$ where $t := (w + 1) \cdot n$.

- Sample matrices $\mathbf{B}_{i,\boldsymbol{b}} \in \mathbb{Z}_\nu^{g\times g}$ and invertible matrices $\mathbf{R}_i \in \mathbb{Z}_q^{(m+g)\times(m+g)}$ randomly.
- Sample matrices $\{\mathbf{S}_{i,\boldsymbol{b}} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{n\times n}\}_{i\in[h-1],\boldsymbol{b}\in\{0,1\}^2}$ and a final encoding $\mathbf{D}_h$ as

$$\mathbf{D}_h \in \mathbb{Z}^{m\times m} \leftarrow \mathsf{Sample}(\mathbf{A}_{h-1}, \tau_{h-1}, \begin{pmatrix} \mathbf{I}^{wn\times wn} & \\ & \mathbf{0}^{n\times n} \end{pmatrix} \cdot \mathbf{A}_h + \mathbf{E}_h, \sigma),$$

  and compute bookend vectors $\mathbf{v}$ and $\mathbf{w}$ as

$$\mathbf{v} = [\mathbf{v}' \cdot \mathbf{J} \cdot \mathbf{A}_0 \mid \mathbf{b}_v] \cdot \mathbf{R}_1,$$
$$\hat{\mathbf{S}}_{i,\boldsymbol{b}} := \begin{pmatrix} \mathbf{P}_{i,\boldsymbol{b}} \otimes \mathbf{S}_{i,\boldsymbol{b}} & \\ & \mathbf{S}_{i,\boldsymbol{b}} \end{pmatrix} \in \mathbb{Z}^{t\times t}$$
$$\mathbf{w}^T = \mathbf{R}_h^{-1} \cdot \begin{pmatrix} \mathbf{D}_h \cdot \mathbf{w}'^T \\ \mathbf{b}_w^T \end{pmatrix}$$

  where $\mathbf{v}' \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^n$, $\mathbf{w}' \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$, $\mathbf{b}_v, \mathbf{b}_w \leftarrow U(\mathbb{Z}_\nu^k)$ and $\mathbf{J} := [\mathbf{J}'|\mathbf{I}^{n\times n}]$ with a randomly chosen matrix $\mathbf{J}' \leftarrow \{0,1\}^{n\times wn}$.
- Compute matrices

$$\mathbf{D}_{i,} \in \mathbb{Z}^{m\times m} \leftarrow \mathsf{Sample}(\mathbf{A}_{i-1}, \tau_{i-1}, \hat{\mathbf{S}}_{i,\boldsymbol{b}} \cdot \mathbf{A}_i + \mathbf{E}_{i,\boldsymbol{b}}, \sigma) \text{ with } 1 \le i \le h-1,$$
$$\text{and } \mathbf{C}_{i,\boldsymbol{b}} = \mathbf{R}_i^{-1} \cdot \begin{pmatrix} \mathbf{D}_{i,\boldsymbol{b}} & \\ & \mathbf{B}_{i,\boldsymbol{b}} \end{pmatrix} \cdot \mathbf{R}_{i+1} \text{ with } i = 1, \cdots, h-1.$$

Evaluation. Outputs 0 if $|\mathbf{v} \cdot \prod_{i=1}^{h-1} \mathbf{C}_{i,\boldsymbol{x}(i)} \cdot \mathbf{w}^T| \le B$. Otherwise, outputs 1.

Parameters. We first consider several security parameters. Let λ and $\lambda_{LWE} = \mathsf{poly}(\lambda)$ be security parameters depending on the obfuscation itself and the hardness of LWE satisfying following constraints, respectively. Set $n = \Omega(\lambda_{LWE} \log q)$, $\chi = \mathcal{D}_{\mathbb{Z},s}$ with $s = \Omega(\sqrt{n})$. Moreover, for the trapdoor functionality, we set $m = \Omega(t \log q)$ and $\sigma = \Omega(\sqrt{t \log q})$. In addition, they use parameters $g = 5$ and $\nu = 2^\lambda$. For correctness we set zerotest bound $B = (m \cdot \beta \cdot \sigma \cdot \sqrt{t})^{h+1} + (k \cdot \nu)^{h+1}$ and $B \cdot \omega(\mathsf{poly}(\lambda)) \le q \le (\sigma/\lambda_{LWE}) \cdot 2^{\lambda_{LWE}^{1-\epsilon}}$ for some fixed $\epsilon \in (0,1)$. For more detail we refer readers to the original paper [4].

Zerotest Functionality. From the construction of obfuscation, the following equality always holds if $\mathbf{C} := \prod_{i=1}^{h-1} \mathbf{C}_{i,\boldsymbol{x}(i)}$ is an encoding of zero computed by honest evaluation.

$$\begin{aligned}
&\|[\mathbf{v} \cdot \mathbf{C} \cdot \mathbf{w}^T]_q\|_\infty \\
&= \left\| \left[\mathbf{v}' \cdot \mathbf{J} \cdot \sum_{j=1}^{h} ((\prod_{i=1}^{j-1} \hat{\mathbf{S}}_{i,\boldsymbol{x}(i)}) \cdot \mathbf{E}_{j,\boldsymbol{x}(j)} \cdot \prod_{k=j+1}^{h} \mathbf{D}_{k,\boldsymbol{x}(k)} \cdot \mathbf{w}'^T + \mathbf{b}_v \cdot \prod_{i=1}^{h-1} \mathbf{B}_{i,\boldsymbol{x}(i)} \cdot \mathbf{b}_w^T \right]_q \right\|_\infty \\
&\le \left\| \mathbf{v}' \cdot \mathbf{J} \cdot \sum_{j=1}^{h} ((\prod_{i=1}^{j-1} \hat{\mathbf{S}}_{i,\boldsymbol{x}(i)}) \cdot \mathbf{E}_{j,\boldsymbol{x}(j)} \cdot \prod_{k=j+1}^{h} \mathbf{D}_{k,\boldsymbol{x}(k)} \cdot \mathbf{w}'^T + \mathbf{b}_v \cdot \prod_{i=1}^{h-1} \mathbf{B}_{i,\boldsymbol{x}(i)} \cdot \mathbf{b}_w^T \right\|_\infty
\end{aligned}$$

$$\leq \sigma^2 \cdot m^2 \cdot (m \cdot \beta \cdot \sigma \cdot \sqrt{t})^{h-1} + (k \cdot \nu)^{h+1}$$

Since $\|[\mathbf{v} \cdot \mathbf{C} \cdot \mathbf{w}^T]_q\|_\infty$ is bounded by $\sigma^2 \cdot m^2 \cdot (m \cdot \beta \cdot \sigma \cdot \sqrt{t})^{h-1} + (k \cdot \nu)^{h+1} \leq B$ for all but negligible probability. Moreover, if $\prod_{i=1}^{h} \mathbf{P}_{i,\boldsymbol{x}(i)}$ is a nonzero matrix, then $\prod_{i=1}^{h} \hat{\mathbf{S}}_{i,\boldsymbol{x}(i)}$ is also nonzero matrix. Thus, $\|[\mathbf{v} \cdot \mathbf{C} \cdot \mathbf{w}^T]_q\|_\infty$ is larger than B with overwhelming probability because of $\mathbf{A}_h \leftarrow U(\mathbb{Z}_q^{t \times m})$.

5.2 Cryptanalysis of BGMZ Obfuscation

In this section, we analyze the conditions for the statistical zeroizing attack on the BGMZ obfuscation when we assume $\sigma \geq \nu = 2^\lambda$. (More precisely, the same result holds when $\sigma^2 \geq \nu^2 g / 12m$.). As in Section 4.2, the notation written in the capital italic words are regarded as the random matrix whose entry follows a distribution that corresponds to the distribution of entry of the bold-written matrix.

The targeted BPs are $\mathbf{M} = \{\mathbf{M}_{i,\boldsymbol{b}}\}_{i \in [h], \boldsymbol{b} \in \{0,1\}^2}$ and $\mathbf{N} = \{\mathbf{N}_{i,\boldsymbol{b}}\}_{i \in [h], \boldsymbol{b} \in \{0,1\}^2}$ such that

$$\mathbf{M}_{i,\boldsymbol{b}} = \mathbf{0}^{w \times w} \text{ for all } i, b \text{ and } \mathbf{N}_{i,\boldsymbol{b}} = \begin{cases} \mathbf{I}^{w \times w} & \text{if } i = 1 \\ \mathbf{0}^{w \times w} & \text{otherwise} \end{cases}.$$

Note that two branching programs always output zero. Now we suppose that we have polynomially many samples from the one of two distributions $\mathcal{D}_{\mathbf{M}}$ and $\mathcal{D}_{\mathbf{N}}$, where $\mathcal{D}_{\mathbf{M}}$ and $\mathcal{D}_{\mathbf{N}}$ are the distributions of the evaluations of obfuscations of $\mathbf{M}$ and $\mathbf{N}$.

Then our purpose is to distinguish whether the samples come from $\mathcal{D}_{\mathbf{M}}$ or $\mathcal{D}_{\mathbf{N}}$ by Proposition 3.1. We obtain random matrices $S_{i,\mathbf{b}}^{(\mathbf{P})}$, $E_{i,\boldsymbol{b}}^{(\mathbf{P})}$, $D_{i,\boldsymbol{b}}^{(\mathbf{P})}$ and $C_{i,\boldsymbol{b}}^{(\mathbf{P})}$ as in the construction of BGMZ obfuscation for branching programs $\mathbf{P} = \mathbf{M}$ or $\mathbf{N}$. Thus, it suffices to prove the following proposition.

Proposition 5.1 *Let λ be a security parameter and σ the Gaussian variance parameter satisfying $\sigma^2 \geq \nu^2 g / 12m$ for parameters m, ν and g of BGMZ obfuscation. Then, there are two functionally equivalent branching programs $\mathbf{M}$ and $\mathbf{N}$ satisfying the following statement: let $Z_{\mathbf{M}}$ and $Z_{\mathbf{N}}$ be random variables satisfying*

$$Z_{\mathbf{M}} = \left[v \cdot \prod_{i=1}^{h-1} C_{i,\boldsymbol{x}(i)}^{(\mathbf{M})} \cdot w^T \right]_q \text{ and } Z_{\mathbf{N}} = \left[v \cdot \prod_{i=1}^{h-1} C_{i,\boldsymbol{x}(i)}^{(\mathbf{N})} \cdot w^T \right]_q.$$

where every random matrix is defined as the above. Let $\mu_{\mathbf{M}}$ and $\mu_{\mathbf{N}}$, $\sigma_{\mathbf{M}}^2$ and $\sigma_{\mathbf{N}}^2$ be mean and variance of the random variables of $Z_{\mathbf{M}}$ and $Z_{\mathbf{N}}$, respectively. Then, it holds that

$$\left| \frac{\max(\sigma_{\boldsymbol{N}}^2, \sigma_{\boldsymbol{M}}^2)}{\sigma_{\boldsymbol{N}}^2 - \sigma_{\boldsymbol{M}}^2} \right| \leq p, \quad \left| \frac{E[(Z_{\mathbf{N}} - \mu_{\mathbf{N}})^4]}{\sigma_{\mathbf{N}}^4} \right| \leq q, \text{ and } \quad \left| \frac{E[(Z_{\mathbf{M}} - \mu_{\mathbf{M}})^4]}{\sigma_{\mathbf{M}}^4} \right| \leq q.$$

for some $p, q = \mathsf{poly}(\lambda)$ under Assumption 1.

Note that Assumption 1 (for BGMZ obfuscation) is also needed to verify the proposition. With the honest evaluation $\left[\mathbf{v}\cdot\prod_{i=1}^{h-1}\mathbf{C}_{i,\boldsymbol{x}(i)}\cdot\mathbf{w}^T\right]_q$ of the BGMZ obfuscation, we obtain the integer of the form

$$\mathbf{v}'\cdot\mathbf{J}\sum_{j=1}^{h}((\prod_{i=1}^{j-1}\hat{\mathbf{S}}_{i,\boldsymbol{x}(i)})\mathbf{E}_{j,\boldsymbol{x}(j)}\prod_{k=j+1}^{h}\mathbf{D}_{k,\boldsymbol{x}(k)}\cdot\mathbf{w}'^T+\mathbf{b}_v\cdot\prod_{i=1}^{h-1}\mathbf{B}_{i,\boldsymbol{x}(i)}\cdot\mathbf{b}_w^T$$

which does not contain the term including trapdoor matrices $\mathbf{A}_i$'s. Thus, similarly to the CVW obfuscation case, we need to analyze the statistical properties of the random vectors $v'^{(\mathbf{P})}$, $w'^{(\mathbf{P})}$, $b_v^{(\mathbf{P})}$, $b_w^{(\mathbf{P})}$ and random matrices $\hat{S}_{i,\boldsymbol{b}}^{(\mathbf{P})}$, $E_{i,\boldsymbol{b}}^{(\mathbf{P})}$, $D_{i,\boldsymbol{b}}^{(\mathbf{P})}$ and their products to prove the statistical properties including the variance in Proposition 5.1.

The proof of Proposition 5.1 is based on the following lemmas and placed in the concluding part of this section. All proofs of these lemmas can be found in the full version [11]. Note that most lemmas in this section also hold under Assumption 1 as the section 4.2, so we omit repeated *under Assumption 1* in statements. Notations c_0, c, and d are similarly defined as Section 4.

For $j=0,1,\cdots,h-1$, let $(Z^{(\mathbf{M})})_j$ be a random variable of the form

$$v'^{(\mathbf{M})}\cdot J^{(\mathbf{M})}\cdot\prod_{i=1}^{j}\hat{S}_{i,\boldsymbol{x}(i)}^{(\mathbf{M})}\cdot E_{j+1,\boldsymbol{x}(j+1)}^{(\mathbf{M})}\cdot\prod_{k=j+2}^{h}D_{k,\boldsymbol{x}(k)}^{(\mathbf{M})}\cdot w'^{(\mathbf{M})^T},$$

and for $j=h$, $(Z^{(\mathbf{M})})_h$ a random variable of the form

$$b_v^{(\mathbf{M})}\cdot\prod_{i=1}^{h-1}B_{i,\boldsymbol{x}(i)}^{(\mathbf{M})}\cdot b_w^{(\mathbf{M})^T}.$$

We similarly define $(Z^{(\mathbf{N})})_j$ for $j=0,1,\cdots,h$, and $Z_{\mathbf{P}}=\sum_{i=0}^{h}(Z^{(\mathbf{P})})_j$ for $\mathbf{P}=\mathbf{M}$ and $\mathbf{N}$.

Lemma 5.2 *$E[(Z^{(\mathbf{M})})_j]=E[(Z^{(\mathbf{N})})_j]=0$ for all $j=0,1,\cdots,h$.*

Lemma 5.3 *$E[(Z^{(\mathbf{M})})_{\mu_1}\cdot(Z^{(\mathbf{M})})_{\mu_2}]=E[(Z^{(\mathbf{N})})_{\mu_1}\cdot(Z^{(\mathbf{N})})_{\mu_2}]=0$ for $\mu_1\neq\mu_2$.*

Lemma 5.4 **($j=0$)** *It holds that*

$$Var[(Z^{(\mathbf{M})})_0]=Var[(Z^{(\mathbf{N})})_0]=\Theta\left(wn\cdot m^h\cdot(\sigma^2)^{h+1}\cdot s^2\right),$$
$$\left|\frac{E[(Z^{(\mathbf{M})})_0^4]}{Var[(Z^{(\mathbf{M})})_0]^2}\right|,\ \left|\frac{E[(Z^{(\mathbf{N})})_0^4]}{Var[(Z^{(\mathbf{N})})_0]^2}\right|\leq 108c_0(w+1)^2\cdot n^2m^4\cdot\left(\frac{d}{c}\right)^2=\mathsf{poly}(\lambda).$$

Lemma 5.5 **($j=1$)** *It holds that*

$$Var[(Z^{(\mathbf{M})})_1]=\Theta\left(n^2m^{h-1}\cdot(\sigma^2)^{h+1}\cdot s^2\right),$$
$$Var[(Z^{(\mathbf{N})})_1]=\Theta\left(wn^3m^{h-1}\cdot(\sigma^2)^{h+1}\cdot s^2\right)+Var[(Z^{(\mathbf{M})})_1]$$

Moreover, it holds that

$$\left|\frac{E[(Z^{(\mathbf{M})})_1^4]}{Var[(Z^{(\mathbf{M})})_1]^2}\right| \leq 81c_0 \cdot n^4 m^4 \cdot \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda),$$
$$\left|\frac{E[(Z^{(\mathbf{N})})_1^4]}{Var[(Z^{(\mathbf{N})})_1]^2}\right| \leq 324c_0(w+1)^2 \cdot n^6 m^4 \cdot \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda).$$

Lemma 5.6 ($2 \leq j \leq h-1$) *It holds that*

$$Var[(Z^{(\mathbf{M})})_j] = Var[(Z^{(\mathbf{N})})_j] = \Theta\left(n^{j+1} m^{h-j} \cdot (\sigma^2)^{h+1} \cdot s^2\right).$$

Moreover, it holds that

$$\left|\frac{E[(Z^{(\mathbf{M})})_j^4]}{Var[(Z^{(\mathbf{M})})_j]^2}\right|, \left|\frac{E[(Z^{(\mathbf{N})})_j^4]}{Var[(Z^{(\mathbf{N})})_j]^2}\right| \leq 81c_0 \cdot n^4 m^4 \left(1+\frac{2}{n}\right)^{j-1} \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda).$$

Lemma 5.7 ($j = h$) *It holds that*

$$Var[(Z^{(\mathbf{M})})_h] = Var[(Z^{(\mathbf{N})})_h] = g^h \cdot \left\{\frac{1}{12} \cdot \nu(\nu+2)\right\}^{h+1}.$$

Moreover, it holds that

$$E[(Z^{(\mathbf{M})})_h^4], E[(Z^{(\mathbf{N})})_h^4] \leq 27 \cdot (g^2)^4 \cdot \{g(g+2)\}^{h-2} \cdot \left\{\frac{1}{12} \cdot \nu(\nu+2)\right\}^{2(h+1)}.$$

Now we give a proof of the proposition 5.1 using the above lemmas.

Proof (of Proposition 5.1). Choose BPs $\mathbf{M}$ and $\mathbf{N}$ as given in the first page of this section. They have the same functionality and length.

Note that elements $(Z^{(\mathbf{M})})_j$ in the above Lemmas are of the form

$$(Z^{(\mathbf{M})})_j = v'^{(\mathbf{M})} \cdot J^{(\mathbf{M})} \cdot \prod_{i=1}^{j} \hat{S}^{(\mathbf{M})}_{i,\boldsymbol{x}(i)} \cdot E^{(\mathbf{M})}_{j+1,\boldsymbol{x}(j+1)} \cdot \prod_{k=j+2}^{h} D^{(\mathbf{M})}_{k,\boldsymbol{x}(k)} \cdot w'^{(\mathbf{M})^T} \quad \textit{for } j < h$$
$$(Z^{(\mathbf{M})})_h = b_v^{(\mathbf{M})} \cdot \prod_{i=1}^{h-1} B^{(\mathbf{M})}_{i,\boldsymbol{x}(i)} \cdot b_w^{(\mathbf{M})^T}$$

Let $Z_{\mathbf{M}}$ be the summation of $(Z^{(\mathbf{M})})_j$ for $j \in \{0, 1, \cdots, h\}$. From Lemma 5.3, we have

$$Var[Z_{\mathbf{M}}] = E\left[(\sum_{i=0}^{h}(Z^{(\mathbf{M})})_i)^2\right] = E\left[\sum_{i=0}^{h}(Z^{(\mathbf{M})})_i^2\right] = \sum_{i=0}^{h} Var[(Z^{(\mathbf{M})})_i],$$

$$E[Z_{\mathbf{M}}^4] = E\left[(\sum_{i=0}^{h}(Z^{(\mathbf{M})})_i)^4\right] \leq E\left[(h+1)^3 \cdot (\sum_{i=0}^{h}(Z^{(\mathbf{M})})_i^4)\right].$$

After dividing both sides by $Var[Z_{\mathbf{M}}]^2$, we obtain the following inequality

$$\begin{aligned}\left|\frac{E[Z_{\mathbf{M}}^4]}{Var[Z_{\mathbf{M}}]^2}\right| &\leq \left|\frac{E[(h+1)^3\cdot(\sum_{i=0}^{h}(Z^{(\mathbf{M})})_i^4)]}{Var[Z_{\mathbf{M}}]^2}\right| = (h+1)^3\cdot\left|\frac{E[\sum_{i=0}^{h}(Z^{(\mathbf{M})})_i^4]}{Var[Z_{\mathbf{M}}]^2}\right| \\ &= (h+1)^3\cdot\sum_{i=0}^{h}\left|\frac{E[(Z^{(\mathbf{M})})_i^4]}{Var[Z_{\mathbf{M}}]^2}\right| \\ &\leq (h+1)^3\cdot\left(\sum_{i=0}^{h-1}\left|\frac{E[(Z^{(\mathbf{M})})_i^4]}{Var[(Z^{(\mathbf{M})})_i]^2}\right| + \left|\frac{E[(Z^{(\mathbf{M})})_h^4]}{Var[Z_{\mathbf{M}}]^2}\right|\right)\end{aligned}$$

By Lemma 5.4,5.5, 5.6 and 5.7, $\left|\frac{E[(Z^{(\mathbf{M})})_i^4]}{Var[(Z^{(\mathbf{M})})_i]^2}\right|$ is bounded by $\mathsf{poly}(\lambda)$ for all $i = 0, 1, \cdots, h-1$ regardless of $\mathbf{P} = \mathbf{M}$ or $\mathbf{P} = \mathbf{N}$. Since $\sigma^2 \geq \nu^2 g/12m$, we obtain the following upper bound.

$$\begin{aligned}\left|\frac{E[(Z^{(\mathbf{M})})_h^4]}{Var[Z_{\mathbf{M}}]^2}\right| &\leq \left|\frac{E[(Z^{(\mathbf{M})})_h^4]}{Var[(Z^{(\mathbf{M})})_0]^2}\right| \\ &= O\left((g^2)^4\cdot\left(\frac{g(g+2)}{m^2}\right)^{h-2}\cdot\left(\frac{\nu(\nu+2)}{12\sigma^2}\right)^{h+1}\right) \\ &= \mathsf{poly}(\lambda)\end{aligned}$$

Thus the kurtosis is bounded by polynomial of security parameter λ.

Moreover, by the definition of $Z_{\mathbf{N}}$ and $Z_{\mathbf{M}}$ and lemmas, we obtain the equality $|\sigma_{\mathbf{N}}^2 - \sigma_{\mathbf{M}}^2| = \Theta\left(wn^3m^{h-1}\cdot(\sigma^2)^{h+1}\cdot s^2\right)$. Using lemmas, $\left|\frac{\max(\sigma_{\mathbf{N}}^2, \sigma_{\mathbf{M}}^2)}{\sigma_{\mathbf{N}}^2 - \sigma_{\mathbf{M}}^2}\right|$ is bounded by $\mathsf{poly}(\lambda)$. □

Acknowledgments. We sincerely thank to James Bartusek, Fermi Ma and anonymous reviewers of Eurocrypt 2019 for noting the errors in the earlier version of this paper. We also thank to the anonymous reviewers of Crypto 2019 for their careful comments.

The authors of Seoul National University were supported by Institute for Information & communication Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2016-6-00598, The mathematical structure of functional encryption and its analysis), and the ARO and DARPA under Contract No.W911NF-15-C-0227. The author of ENS de Lyon was supported by the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).

References

1. Prabhanjan Vijendra Ananth, Divya Gupta, Yuval Ishai, and Amit Sahai. Optimizing obfuscation: Avoiding barrington's theorem. In *ACM CCS 2014*, pages 646–658, 2014.

2. Saikrishna Badrinarayanan, Eric Miles, Amit Sahai, and Mark Zhandry. Post-zeroizing obfuscation: New mathematical tools, and the case of evasive circuits. In *EUROCRYPT 2016, Part II*, pages 764–791, 2016.
3. Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. In *EUROCRYPT 2014*, pages 221–238, 2014.
4. James Bartusek, Jiaxin Guan, Fermi Ma, and Mark Zhandry. Return of GGH15: provable security against zeroizing attacks. In *TCC 2018, Part II*, pages 544–574, 2018.
5. Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. *Algorithmica*, 79(4):1233–1285, 2017.
6. Zvika Brakerski and Guy N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In *TCC 2014*, pages 1–25, 2014.
7. Zvika Brakerski, Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Obfuscating conjunctions under entropic ring LWE. In *ITCS 2016*, pages 147–156, 2016.
8. Ran Canetti and Yilei Chen. Constraint-hiding constrained prfs for NC^1 from LWE. In *EUROCRYPT 2017, Part I*, pages 446–476, 2017.
9. Yilei Chen, Craig Gentry, and Shai Halevi. Cryptanalyses of candidate branching program obfuscators. In *EUROCRYPT 2017, Part III*, pages 278–307, 2017.
10. Yilei Chen, Vinod Vaikuntanathan, and Hoeteck Wee. GGH15 beyond permutation branching programs: Proofs, attacks, and candidates. In *CRYPTO 2018, Part II*, pages 577–607, 2018.
11. 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, 2018. full version of this paper, https://eprint.iacr.org/2018/1081.
12. Jung Hee Cheon, Minki Hhan, Jiseung Kim, and Changmin Lee. Cryptanalyses of branching program obfuscations over GGH13 multilinear map from the NTRU problem. In *CRYPTO 2018, Part III*, pages 184–210, 2018.
13. Jung Hee Cheon, Minki Hhan, Jiseung Kim, and Changmin Lee. Cryptanalysis on the HHSS obfuscation arising from absence of safeguards. *IEEE Access*, 6:40096–40104, 2018.
14. Jean-Sébastien Coron, Craig Gentry, Shai Halevi, Tancrède Lepoint, Hemanta K. Maji, Eric Miles, Mariana Raykova, Amit Sahai, and Mehdi Tibouchi. Zeroizing without low-level zeroes: New MMAP attacks and their limitations. In *CRYPTO 2015, Part I*, pages 247–266, 2015.
15. Jean-Sébastien Coron, Moon Sung Lee, Tancrède Lepoint, and Mehdi Tibouchi. Zeroizing attacks on indistinguishability obfuscation over CLT13. In *PKC 2017, Part I*, pages 41–58, 2017.
16. Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In *CRYPTO 2013, Part I*, pages 476–493, 2013.
17. Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *EUROCRYPT 2013*, pages 1–17, 2013.
18. 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, 2013.
19. Sanjam Garg, Eric Miles, Pratyay Mukherjee, Amit Sahai, Akshayaram Srinivasan, and Mark Zhandry. Secure obfuscation in a weak multilinear map model. In *TCC 2016-B, Part II*, pages 241–268, 2016.

20. Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In *TCC 2015, Part II*, pages 498–527, 2015.
21. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *40th STOC*, pages 197–206, 2008.
22. Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In *58th FOCS*, pages 612–621, 2017.
23. Shai Halevi, Tzipora Halevi, Victor Shoup, and Noah Stephens-Davidowitz. Implementing bp-obfuscation using graph-induced encoding. In *ACM CCS 2017*, pages 783–798. ACM, 2017.
24. Shai Halevi, Tzipora Halevi, Victor Shoup, and Noah Stephens-Davidowitz. Implementing bp-obfuscation using graph-induced encoding. https://github.com/shaih/BPobfus, 2017.
25. Huijia Lin. Indistinguishability obfuscation from constant-degree graded encoding schemes. In *EUROCRYPT 2016, Part I*, pages 28–57, 2016.
26. Huijia Lin. Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 prgs. In *CRYPTO 2017, Part I*, pages 599–629, 2017.
27. Huijia Lin and Stefano Tessaro. Indistinguishability obfuscation from trilinear maps and block-wise local prgs. In *CRYPTO 2017, Part I*, pages 630–660, 2017.
28. Huijia Lin and Vinod Vaikuntanathan. Indistinguishability obfuscation from ddh-like assumptions on constant-degree graded encodings. In *57th FOCS*, pages 11–20, 2016.
29. Fermi Ma and Mark Zhandry. The mmap strikes back: Obfuscation and new multilinear maps immune to CLT13 zeroizing attacks. In *TCC 2018, Part II*, pages 513–543, 2018.
30. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT 2012*, pages 700–718, 2012.
31. Eric Miles, Amit Sahai, and Mor Weiss. Protecting obfuscation against arithmetic attacks. *IACR Cryptology ePrint Archive*, 2014:878, 2014.
32. Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. In *CRYPTO 2014, Part I*, pages 500–517, 2014.
33. Alice Pellet-Mary. Quantum attacks against indistinguishablility obfuscators proved secure in the weak multilinear map model. In *CRYPTO 2018, Part III*, pages 153–183, 2018.
34. Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *STOC 2014*, pages 475–484, 2014.
35. Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In *58th FOCS*, pages 600–611, 2017.
36. Joe Zimmerman. How to obfuscate programs directly. In *EUROCRYPT 2015, Part II*, pages 439–467, 2015.

A Simple GGH15 obfuscation

We briefly describe the construction of single input BP obfuscation based GGH15 without safeguard.

For an index to input function $\mathsf{inp} : [h] \to [\ell]$, let

$$\mathbf{P} = \left\{\mathsf{inp}, \{\mathbf{P}_{i,b} \in \{0,1\}^{w\times w}\}_{i\in[h],b\in\{0,1\}}, \mathcal{P}_0 = \mathbf{0}^{w\times w}, \mathcal{P}_1 = \mathbb{Z}^{w\times w} \setminus \mathcal{P}_0\right\}$$

be a single input BP.

For parameters $w, m, q, B \in \mathbb{N}$ and $\sigma \in \mathbb{R}^+$, the BP obfuscation based GGH15 consists of the matrices and input function, namely

$$\mathcal{O}(\mathbf{P}) = \left\{\mathsf{inp}, \mathbf{A}_0, \{\mathbf{D}_{i,b} \in \mathbb{Z}^{m\times m}\}_{i\in[h],b\in\{0,1\}}\right\}.$$

In this case, the matrix $\mathbf{T}$ in the abstract model is the identity matrix and $\mathbf{S} = \mathbf{A}_0$. The output of the obfuscation at $\mathbf{x}$ is computed as follows: compute the matrix $\mathbf{A}_0 \cdot \prod_{i=1}^{h} \mathbf{D}_{i,x_{\mathsf{inp}(i)}}$ mod q and compare its $\|\cdot\|_\infty$ to a zerotest bound B. If it is less than B, outputs zero. Otherwise, outputs 1.

The algorithm to construct an obfuscated program $\mathcal{O}(\mathbf{P})$ proceeds as follows:

- Sample matrices $(\mathbf{A}_i, \tau_i) \leftarrow \mathsf{TrapSam}(1^w, 1^m, q)$ for $i = 0, 1, \cdots, h-1$, $\mathbf{A}_h \leftarrow U(\mathbb{Z}_q^{w\times m})$ and $\mathbf{E}_{i,b} \leftarrow \chi^{w\times m}$ where χ is a distribution related to the hardness of LWE problem.
- By using the trapdoor τ_i, sample matrices

$$\mathbf{D}_{i,b} \in \mathbb{Z}^{m\times m} \leftarrow \mathsf{Sample}(\mathbf{A}_{i-1}, \tau_{i-1}, \mathbf{P}_{i,b} \cdot \mathbf{A}_i + \mathbf{E}_{i,b}, \sigma) \text{ with } 1 \le i \le h.$$

- Output matrices $\{\mathbf{A}_0, \{\mathbf{D}_{i,b} \in \mathbb{Z}^{m\times m}\}_{i\in[h],b\in\{0,1\}}\}$.

Then, we observe the product $\mathcal{O}(\mathbf{P})(\mathbf{x}) = [\mathbf{A}_0 \cdot \prod_{i=1}^{h} \mathbf{D}_{i,x_{\mathsf{inp}(i)}}]_q$ is equal to

$$\prod_{i=1}^{h} \mathbf{P}_{i,x_{\mathsf{inp}(i)}} \cdot \mathbf{A}_h + \sum_{j=1}^{h} \left(\left(\prod_{i=1}^{j-1} \mathbf{P}_{i,x_{\mathsf{inp}(i)}} \right) \cdot \mathbf{E}_{j,x_{\mathsf{inp}(j)}} \cdot \prod_{k=j+1}^{h} \mathbf{D}_{i,x_{\mathsf{inp}(k)}} \right)$$

over $\mathbb{Z}_q$. If $\prod_{i=1}^{h} \mathbf{P}_{i,x_{\mathsf{inp}(i)}} = \mathbf{0}^{w\times w}$, then $\mathcal{O}(\mathbf{P})(\mathbf{x})$ can be regarded as a summation of matrices over integers instead of $\mathbb{Z}_q$ under the certain choice of parameters as follows

$$\mathcal{O}(\mathbf{P})(\mathbf{x}) = \left[\mathbf{A}_0 \cdot \prod_{i=1}^{h} \mathbf{D}_{i,x_{\mathsf{inp}(i)}}\right]_q = \sum_{j=1}^{h} \left(\left(\prod_{i=1}^{j-1} \mathbf{P}_{i,x_{\mathsf{inp}(i)}} \right) \cdot \mathbf{E}_{j,x_{\mathsf{inp}(j)}} \cdot \prod_{k=j+1}^{h} \mathbf{D}_{i,x_{\mathsf{inp}(k)}} \right)$$

since the infinity norm of the above matrix is less than $B \ll q$. Note that the evaluation values only rely on the matrices $\mathbf{P}_{i,b}$, $\mathbf{E}_{i,b}$ and $\mathbf{D}_{i,b}$. Thus, the evaluation result depends on the message matrices $\mathbf{P}_{i,b}$.

Suppose that we have two functionally equivalent BPs $\mathbf{M} = \{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}}$ and $\mathbf{N} = \{\mathbf{N}_{i,b}\}_{i\in[h],b\in\{0,1\}}$ satisfies

$$\mathbf{M}_{i,b} = \mathbf{0}^{w\times w} \text{ for all } i, b \text{ and } \mathbf{N}_{i,b} = \begin{cases} \mathbf{I}^{w\times w} & \text{if } i = 1 \\ \mathbf{0}^{w\times w} & \text{otherwise} \end{cases},$$

and an obfuscated program $\mathcal{O}(\mathbf{P})$. The goal of adversary is to determine whether $\mathbf{P}$ is $\mathbf{M}$ or not. For all $\mathbf{x} \in \{0,1\}^\ell$, the evaluation of the obfuscation is of the form

$$\mathcal{O}(\mathbf{M})(\mathbf{x}) = \mathbf{E}_{1,x_{\mathsf{inp}(1)}} \cdot \prod_{k=2}^{h} \mathbf{D}_{k,x_{\mathsf{inp}(k)}} \text{ and}$$

$$\mathcal{O}(\mathbf{N})(\mathbf{x}) = \mathbf{E}_{1,x_{\mathsf{inp}(1)}} \cdot \prod_{k=2}^{h} \mathbf{D}_{k,x_{\mathsf{inp}(k)}} + \mathbf{I} \cdot \mathbf{E}_{2,x_{\mathsf{inp}(2)}} \cdot \prod_{k=3}^{h} \mathbf{D}_{k,x_{\mathsf{inp}(k)}}.$$

Note that they correspond to the distributions $\mathcal{D}_{\mathbf{M}}$ and $\mathcal{D}_{\mathbf{N}}$ for a fixed vector $\mathbf{x}$. These equations show the difference of two distributions in this case.

B Modified CVW Obfuscation

We give a modification of CVW obfuscation, which can obfuscate the permutation matrix branching programs. This modification is, as far as we know, robust against all existing attacks. We first describe the transformation of branching programs. Then, we describe the modification of CVW obfuscation.

B.1 Transformation of Branching Programs

We first introduce the transformation from single-input permutation matrix branching programs to *Type I* BP. This transformation is applicable to BPs which outputs 0 when the product of BP matrices is the identity matrix. The output of transformation is a new branching program that outputs 0 when the product of BP matrices is the zero matrix. Through this transformation, the width of branching program is doubled. Note that this is adapted version of [10, Claim 6.2].

We are given a branching program with input size ℓ

$$\mathbf{P} = \left\{ \{\mathbf{P}_{i,b} \in \{0,1\}^{w \times w}\}_{i \in [h], b \in \{0,1\}}, \mathsf{inp} : [h] \to [\ell] \right\}$$

where the evaluation of $\mathbf{P}$ at $\mathbf{x} \in \{0,1\}^{\ell}$ is computed by

$$\mathbf{P}(\mathbf{x}) = \begin{cases} 0 & \text{if } \prod_{i=1}^{h} \mathbf{P}_{i,(x_{\mathsf{inp}(i)})} = \mathbf{I}_w \\ 1 & \text{otherwise} \end{cases}$$

Then the transformation is done by changing branching program matrices as

$$\mathbf{P}' = \left\{ \left\{ \mathbf{P}'_{i,b} = \begin{pmatrix} \mathbf{P}_{i,b} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_w \end{pmatrix} \in \{0,1\}^{2w \times 2w} \right\}_{i \in [h], b \in \{0,1\}}, \mathsf{inp} : [h] \to [\ell] \right\}$$

and the evaluation is similar but uses new vectors $\mathbf{v}' = (\mathbf{v} | -\mathbf{v})$ and $\mathbf{w}' = (\mathbf{w} | \mathbf{w})$ for $\mathbf{v}, \mathbf{w} \in \mathbb{Z}^w$:

$$\mathbf{P}'(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{v}' \cdot \prod_{i=1}^{h} \mathbf{P}'_{i,(x_{\mathsf{inp}(i)})} \cdot \mathbf{w}'^{T} = 0 \\ 1 & \text{otherwise} \end{cases}$$

We will choose $\mathbf{v}$ and $\mathbf{w}$ as random Gaussian vectors. Note that the resulting branching program is also a permutation BP.

B.2 Modification of CVW Obfuscation

We give here how to modify the CVW obfuscation to be applicable to the resulting permutation BPs of the above transform. We also assume that the index length $h = (\lambda+1)\cdot\ell$ and the index-to-input function satisfies $\mathsf{inp}(i) = (i \mod \ell)$ as in the CVW obfuscation. We also assume that the BP is $(\lambda+1)$-input repetition BP as in the original construction. The changed parts are written in red. Note that the targeted BPs have width $2w$. Thus we set $t := (2w + 2n\ell)\cdot n$.

- Sample bundling matrices $\{\mathbf{R}_{i,b} \in \mathbb{Z}^{2n\ell\times 2n\ell}\}_{i\in[h],b\in\{0,1\}}$ such that $(\mathbf{1}^{1\times 2\ell}\otimes \mathbf{I}^{n\times n})\cdot\mathbf{R}_{\mathbf{x}'}\cdot(\mathbf{1}^{2\ell\times 1}\otimes\mathbf{I}^{n\times n}) = \mathbf{0} \iff \mathbf{x}' \in \bar{\omega}(\{0,1\}^\ell)$ for all $\mathbf{x}' \in \{0,1\}^h$. More precisely, $\mathbf{R}_{i,b}$ is a block diagonal matrix $\mathsf{diag}(\mathbf{R}^{(1)}_{i,b}, \mathbf{R}^{(2)}_{i,b}, \cdots, \mathbf{R}^{(\ell)}_{i,b})$. Each $\mathbf{R}^{(k)}_{i,b} \in \mathbb{Z}^{2n\times 2n}$ is one of the following three cases.

$$\mathbf{R}^{(k)}_{i,b} = \begin{cases} \mathbf{I}^{2n\times 2n} & \text{if } \mathsf{inp}(i) \neq k \\ \begin{pmatrix} \tilde{\mathbf{R}}^{(k)}_{i,b} & \\ & \mathbf{I}^{n\times n} \end{pmatrix}, \tilde{\mathbf{R}}^{(k)}_{i,b} \leftarrow \mathcal{D}^{n\times n}_{\mathbb{Z},\sigma} & \text{if } \mathsf{inp}(i) = k \text{ and } i \leq \lambda\ell \\ \begin{pmatrix} -\mathbf{I}^{n\times n} & \\ & \prod_{j=0}^{\lambda-1} \tilde{\mathbf{R}}^{(k)}_{k+j\ell,b} \end{pmatrix} & \text{if } \mathsf{inp}(i) = k \text{ and } i > \lambda\ell \end{cases}$$

- Sample matrices $\{\mathbf{S}_{i,b} \leftarrow \mathcal{D}^{n\times n}_{\mathbb{Z},\sigma}\}_{i\in[h],b\in\{0,1\}}$, bookend vectors $\mathbf{v} \leftarrow \mathcal{D}^{w}_{\mathbb{Z},\sigma}$ and $\mathbf{w} \leftarrow \mathcal{D}^{w}_{\mathbb{Z},\sigma}$ and compute

$$\mathbf{J} := ((\mathbf{v}| - \mathbf{v}|\mathbf{1}^{1\times 2n\ell})\otimes\mathbf{I}^{n\times n}) \in \mathbb{Z}^{n\times t}$$
$$\hat{\mathbf{S}}_{i,b} := \begin{pmatrix} \mathbf{P}_{i,b}\otimes\mathbf{S}_{i,b} & \\ & \mathbf{R}_{i,b}\otimes\mathbf{S}_{i,b} \end{pmatrix} \in \mathbb{Z}^{t\times t}$$
$$\mathbf{L} := ((\mathbf{w}|\mathbf{w}|\mathbf{1}^{1\times 2n\ell})^T\otimes\mathbf{I}^{n\times n}) \in \mathbb{Z}^{t\times n}$$

- Sample $(\mathbf{A}_i, \tau_i) \leftarrow \mathsf{TrapSam}(1^t, 1^m, q)$ for $0 \leq i \leq h-1$, $\mathbf{A}_h \leftarrow U(\mathbb{Z}_q^{n\times n})$, $\{\mathbf{E}_{i,b} \leftarrow \mathcal{D}^{t\times m}_{\mathbb{Z},\sigma}\}_{i\in[h-1],b\in\{0,1\}}$ and $\{\mathbf{E}_{h,b} \leftarrow \mathcal{D}^{t\times n}_{\mathbb{Z},\sigma}\}_{b\in\{0,1\}}$.
- Run Sample algorithms to obtain

$$\mathbf{D}_{i,b} \in \mathbb{Z}^{m\times m} \leftarrow \mathsf{Sample}(\mathbf{A}_{i-1}, \tau_{i-1}, \hat{\mathbf{S}}_{i,b}\cdot\mathbf{A}_i + \mathbf{E}_{i,b}, \sigma) \text{ for } 1 \leq i \leq h-1,$$
$$\mathbf{D}_{h,b} \in \mathbb{Z}^{m\times n} \leftarrow \mathsf{Sample}(\mathbf{A}_{h-1}, \tau_{h-1}, \hat{\mathbf{S}}_{h,b}\cdot\mathbf{L}\cdot\mathbf{A}_h + \mathbf{E}_{h,b}, \sigma).$$

- Define $\mathbf{A}_{\mathbf{J}}$ as a matrix $\mathbf{J}\cdot\mathbf{A}_0 \in \mathbb{Z}^{n\times m}$ and outputs matrices

$$\left\{\mathsf{inp}, \mathbf{A}_{\mathbf{J}}, \{\mathbf{D}_{i,b}\}_{i\in[h],b\in\{0,1\}}\right\}.$$

We omit the procedure and correctness of evaluation that are almost the same as the original one.

C Assumptions of lattice preimage sampling

In this section we provide the experimental results of Assumption 1. Our experiments are built upon the preimage sampling algorithm in the [24], an implementation of BP obfuscation [23].[6] The results imply that the variance and kurtosis move almost the same as one assumed independency, the correctness of attack only requires much relaxed assumption.

Parameters			Experiments		Expected
#products	m	$\log_2 \sigma_x^2$	$\log_2 S^2$	$E[X^4]/\sigma^4$	$\log_2 \sigma^2$
2	2191	34.9	80.8	2.937	80.8
2	2771	35.2	81.4	2.702	81.7
2	3352	35.4	82.4	2.677	82.5
3	2771	35.2	128.7	3.025	128.4
4	3352	35.4	177.0	2.900	176.8
5	3932	35.6	225.9	3.068	225.9
7	5621	36.1	328.1	3.210	327.5

Table 1. Experiment results on statistical value of preimage sampling. #products stands for the number of producted preimage matrices, σ_x^2 the variance of preimage sampling, S^2 the sample variance, $E[X^4]/\sigma^4$ the sample kurtosis and σ^2 the expected variance. Every experiment is done using 100 samples. The expected variance is computed under the assumption on independency of D's. Every expected kurtosis assuming independency of D's is about 3.

[6] We also verify the correctness of the attack itself for [23], but with *large entry* BPs. It requires very large number of samples (say 2^{20} but polynomially many) to verify the attack with binary entry BPs, which is not easy to experiment because the obfuscation/evaluation of [23] takes long time (say few minutes to obtain one evaluation).