Cryptanalysis of indistinguishability obfuscation using GGH13 without ideals

Gu Chunsheng

School of Computer Engineering, Jiangsu University of Technology, China
{chunsheng_gu}@163.com

Abstract. Recently, Albrecht, Davidson and Larraia described a variant of the GGH13 without ideals and presented the distinguishing attacks in simplified branching program security model. Their result partially demonstrates that there seems to be a structural defect in the GGH13 encoding that is not related to the ideal $\langle g \rangle$. However, it is not clear whether a variant of the CGH attack described by Chen, Gentry and Halevi can be used to break a branching program obfuscator instantiated by GGH13 without ideals. Consequently this is left as an open problem by Albrecht, Davidson and Larraia. In this paper, we describe a variant of the CGH attack which breaks the branching program obfuscator using GGH13 without ideals. To achieve this goal, we introduce matrix approximate eigenvalues and build a relationship between the determinant and the rank of a matrix with noise. Our result further strengthens the work of Albrecht, Davidson and Larraia that there is a structural weakness in 'GGH13-type' encodings beyond the presence of $\langle g \rangle$.

Keywords: Cryptanalysis, obfuscation, multilinear maps, approximate eigenvalue, determinant estimate

1 Introduction

Program obfuscation in cryptography makes programs unintelligible and keeps their functionality. In 2013, Garg et al. [20] described the first candidate construction for a general-purpose obfuscation. Since then many different obfuscators are constructed [20,7,9,5,22,30,22], and they are all based on the three candidate graded encoding schemes (GES) (resp. GGH13, CLT13 and GGH15) [19,15,21,16,24]. Unfortunately, the GGH13, CLT13 and GGH15 have been proven to be vulnerable to zero attacks [19,13,10,6,25,17,14], attacks on the overstretched NTRU [1,12,26], and annihilation attacks [28,11].

To immune the above attacks, Garg et al. [22] constructed a provably secure obfuscation in a weak multilinear map model, which aims to prevent the annihilation attack. However, Chen, Gentry and Halevi (CGH) [11] showed that their immunization can not thwart the annihilation attack if the branch program obfuscator is input partionable. It should be noted that the immunised construction in [22] can not be broken by the CGH attack since the dual-inputs used in their construction are not input partionable. As well, Fernando, Rasmussen and

Sahai [18] recently described a defense against input partitioning by applying stamping functions. On the other hand, Albrecht, Davidson and Larraia (ADL) [2] (added Pellet-Mary as author in the updated version in EPRINT [3]) investigated a structural vulnerability of the GGH13 encoding scheme. They proposed a variant of the GGH13 without ideals and presented the distinguishing attacks in simplified branching program and obfuscation security models. However, it is not clear whether a variant of the CGH annihilation attack by Chen, Gentry and Halevi can be used to break a candidate branching program obfuscator instantiate by GGH13 without ideals [2,3].

1.1 Our work

Our main contribution is to describe a variant of the CGH attack which breaks a branching program obfuscator using GGH13 without ideals. The framework of our attack directly follows that of the CGH attack. The core step in the CGH attack is to solve a basis of ideal $\langle g \rangle$, but we cannot perform this step since there are no ideals in the ADL-based obfuscator [2]. Moreover, we cannot find some exact ratios of the bundling scalars and distinguish the ADL-based BP obfuscator by using the rank of a matrix. This is because each entry of the matrix has noise. In order to implement the attack, we solve some approximate ratios of the bundling scalars and build a relationship between the determinant and the rank of a matrix with noise. Therefore, our result further indicates that the structural vulnerability of GGH13 encodings are beyond the presence of ideal.

Our second contribution is to introduce approximate eigenvalues of a matrix to solve the approximate ratios of the bundling scalars used in the ADL-based BP obfuscator. In the BP obfuscator using GGH13 without ideals [2], the multiplicative bundling scalars appear as an approximation factor. That is, when solving the ratios of these bundling scalars in this variant obfuscator, there are noises in the diagonal matrix consisting of the elements returned by the zero-testing procedure. Consequently we can not directly apply the characteristic polynomial of matrix to get the ratios of the bundling scalars, and also can no longer compute their exact ratios. However, we observe that these matrices are diagonal dominated matrix with noise and their inverses are also diagonal dominated matrix with noise. Using this matrix property, we can compute the approximate eigenvalue of the diagonally dominant matrix with noise, and consider them as the approximate ratio of the bundling scalars.

Our final contribution is to estimate the determinant of a matrix with noise. Since in the IO using GGH13 without ideals [2] each term of matrices has noise, as a result these matrices are all full rank with overwhelming probability. So, we can no longer use the rank of matrix to distinguish two equivalent branch program obfuscators. But we observe that the noise magnitude of the matrix in the ADL-based IO is "small" relative to its principal component matrix. Consequently the determinant of matrix is also "small" if the matrix decomposition produces a non-full rank principal component matrix. To this end, we build a relationship between the determinant and the rank of a matrix.

In addition, in the process of attacking a branching program obfuscator using GGH13 without ideals, some matrix properties that we prove might be of independent interest.

Organization. In Section 2 we first recall some preliminaries. In Section 3 we give a branching program obfuscator using GGH13 without ideals. In Section 4 we provide some matrix properties. In Section 5 we describe cryptanalysis of the BP obfuscator using GGH13 without ideals. Finally we conclude the results in this paper.

2 Preliminaries

2.1 Notations

Let $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$ denote the ring of integers, the field of rational numbers, and the field of real numbers. Let a positive integer n be a power of 2. Notation $[n]$ denotes the set $\{1, 2, ..., n\}$. Let $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$, $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$, and $\mathbb{K} = \mathbb{Q}[x]/\langle x^n + 1\rangle$. Vectors are denoted in bold lowercase (e.g. $\mathbf{a}$), and matrices in bold uppercase (e.g. $\mathbf{A}$). We denote by $a[j]$ the j-th entry of $\mathbf{a}$, and $A[i, j]$ the element of the i-th row and j-th column of $\mathbf{A}$. We denote by $\|\mathbf{a}\|_p$ the p-norm of $\mathbf{a}$ and by $\|\mathbf{a}\|$ the ∞-norm. Similarly, for $a \in R$ we let $\|\mathbf{a}\|_p$ (resp. $\|\mathbf{a}\|$) denote the p-norm (resp. ∞-norm) of the coefficient vector corresponding to a. For $\mathbf{A} \in R^{d\times d}$, we define $\|\mathbf{A}\|_\infty = \max\{\|A[i, j]\|, i, j \in [d]\}$.

Let $[a]_q = a \bmod q \in (-q/2, q/2]$. Similarly, for $\mathbf{a} \in \mathbb{Z}^n$ (or $a \in R$), $[\mathbf{a}]_q$ denotes each entry (or each coefficient) $a[j] \in (-q/2, q/2]$ of $\mathbf{a}$ (or a).

Given $\mathbf{c} \in \mathbb{R}^n$, $\sigma > 0$, the Gaussian distribution of a lattice L is defined as $D_{L,\sigma,\mathbf{c}} = \rho_{\sigma,\mathbf{c}}(\mathbf{x})/\rho_{\sigma,\mathbf{c}}(L)$ for $\mathbf{x} \in L$, where $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp(-\pi\|\mathbf{x} - \mathbf{c}\|_2^2/\sigma^2))$, $\rho_{\sigma,\mathbf{c}}(L) = \sum_{\mathbf{x}\in L} \rho_{\sigma,\mathbf{c}}(\mathbf{x})$. In the following, we will write $D_{L,\sigma,\mathbf{0}}$ as $D_{L,\sigma}$. We denote a Gaussian sample as $x \leftarrow D_{L,\sigma}$ (or $d \leftarrow D_{I,\sigma}$) over the lattice L (or ideal lattice I).

An element $a \in R$ is called η-bounded if $\|a\|_\infty \leq \eta$. Moreover, it is easy to verify that for any η-bounded elements $a_1, \cdots a_k \in R$, the element $a = \prod_{i=1}^{k} a_i$ is $(n^{k-1}\eta)$-bounded. By the work in [27], the element $x \leftarrow D_{\mathbb{Z}^n,\sigma,\mathbf{c}}$ is $\sigma\sqrt{n}$-bounded with overwhelming probability. Therefore, we define the truncated Gaussian distribution $\overline{D}_{\mathbb{Z}^n,\sigma,\mathbf{c}}$ by sampling elements from $D_{\mathbb{Z}^n,\sigma,\mathbf{c}}$ and repeating any samples that are not $\sigma\sqrt{n}$-bounded.

2.2 Branching programs

Let λ be the security parameter, $\kappa = \kappa(\lambda)$, $l = l(\lambda)$ and $d = d(\lambda)$. Let inp : $[\kappa] \to [l]^d$ be some fixed 'input' function. All current obfuscators only consider branching programs with $d = 1$ or $d = 2$ [20,7].

Definition 2.1. A matrix branching program BP of length κ, input length l and arity d is defined as follows:

$$\text{BP} := (\kappa, l, d, \text{inp}, \{\mathbf{A}_{k,x_{\text{inp}(k)}}\}_{k\in[\kappa],\text{inp}(k)\in\{0,1\}^d}),$$

where $\mathbf{A}_{k,x_{\mathrm{inp}(k)}} \in \{0,1\}^{w\times w}$ and $|\mathrm{inp}(k)| = d$.

The branching program is associated with the function $f_{\mathrm{BP}} : \{0,1\}^l \to \{0,1\}$, which is defined as

$$f_{\mathrm{BP}}(x) = \begin{cases} 0, & \text{if } \prod_{k=1}^{\kappa} \mathbf{A}_{x_{k,\mathrm{inp}(k)}} = \mathbf{I}; \\ 1, & \text{if } \prod_{k=1}^{\kappa} \mathbf{A}_{x_{k,\mathrm{inp}(k)}} \neq \mathbf{I}. \end{cases}$$

A branching program BP is input partionable if its input bits can be partitioned into two or more independent subsets. We need the following observation in [11].

Lemma 2.2 (Lemma 2.2 [11]). Let BP be an input-partitioned branching program, $[\kappa] = X||Y$. If $x, x' \in \{0,1\}^l$ are two zeros of f_{BP} that differ only in bits that are mapped to steps in X. Then the product of the matrices corresponding to X generates the same result in the evaluation of BP on x and x', namely $\prod_{k\in X} \mathbf{A}_{k,x_{\mathrm{inp}(k)}} = \prod_{k\in X} \mathbf{A}_{k,x'_{\mathrm{inp}(k)}}$.

Similarly, if $x, x' \in \{0,1\}^l$ are two zeros of f_{BP} that differ only in bits that are mapped to steps in Y, then $\prod_{k\in Y} \mathbf{A}_{k,x_{\mathrm{inp}(k)}} = \prod_{k\in Y} \mathbf{A}_{k,x'_{\mathrm{inp}(k)}}$.

2.3 GGH13 without ideals

GGH13 overview. The encoding space of GGH13 is $R_q = R/qR$ where q is some big integer, and its plaintext space $R_g = R/gR$ such that g is a small element in R and is kept secret. An encoding of GGH13 takes the form $y = (e + rg)/z \mod q$, where z is a random secret element in R_q, e is the plaintext element and r is some small random element.

The denominator z enables the levels of the GGH13 scheme. In this paper, we only consider the asymmetric case of GGH13 that uses many different denominators z_i. We say the encoding y is encoded at level S_i if the denominator of y is z_i. It is easy to see that additions and multiplications of encodings can be carried out if they satisfy some level restriction. Namely, adding encodings indexed at the same level S_i generates an encoding at the level S_i, and multiplying two encodings, indexed at the disjoint levels S_i, S_j, generates an encoding at level $S_i \cup S_j$.

The GGH13 scheme also provides a public zero-testing parameter $p_{zt} = h \cdot \prod_{i=1}^{\kappa} z_i/g$, where $h \in R$ such that $\|h\| \ll q$. Given a top-level encoding u indexed at level $[\kappa]$, one can determine whether u encodes zero or not by computing $p_{zt} \cdot u$ and checking if the result is small.

However, a simplified candidate IO over GGH13 exists the annihilation attack introduced by Miles, Sahai and Zhandry [28]. That is, their work constructs two programs that are functionally equivalent, and show how to efficiently distinguish between the obfuscators of these two programs by heuristically computing a basis of $\langle g \rangle$. Then, Chen, Gentry and Halevi [11] extend the annihilation attack in [28] to break the GGHRSW obfuscator instantiated by GGH13 [20] when a branching

program has input partitioning. These works are all first to find a basis of the secret element $\langle g \rangle$.

GGH13 without ideals. We adaptively describe a variant of GGH13 without ideals in [2]. Let $\chi = \overline{D}_{\mathbb{Z}^n,\sigma}$ be the error distribution. Let $e \in R$ be a non-zero element with small coefficients, and $r \leftarrow \chi$ a random element sampled from the distribution χ. We sample z_i uniformly from R_q for $1 \leq i \leq \kappa$, and sample β_i such that $\sqrt[\kappa+1]{q} < \|\beta_i\| < \sqrt[\kappa]{q}$.

An encoding of e indexed at level S_i takes the form $y = (e+r/\beta_i)/z_i \mod q$, where z_i, β_i enables the level structure. Obviously, the encodings also supports addition and multiplication operations. For addition, let y_1, y_2 be two encodings indexed at same level $S \subset [\kappa]$, then their sum results in the encoding $y = y_1 + y_2$ at the level S. For multiplication, given two encodings y_1, y_2 at level $S_1, S_2 \subset [\kappa]$ respectively, their product generates $y = y_1 \cdot y_2$ at the level $S_1 \cup S_2$.

In this variant, the zero-test parameter is defined as $p_{zt} = \prod_{i=1}^{\kappa} \beta_i z_i$. Similarly, given a top-level encoding u, one can determine whether u encodes zero or not by computing $\delta = p_{zt} \cdot u$ and checking if the result δ is small.

3 BP Obfuscator using GGH13 without Ideals

Let $\mathrm{BP} := (\kappa, l, d, \mathrm{inp}, \{\mathbf{A}_{k,b}\}_{k\in[\kappa],b\in\{0,1\}})$ be the branching program to be obfuscated, where directly using $d = 1$ for notational simplicity. We obfuscate BP by GGHRSW [20] using instantiation of GGH13 without ideals as follows:

Step 1: Dummy branch. We introduce a "dummy branching program":

$$\mathrm{BP}^{'} := (\kappa, l, d, \mathrm{inp}, \{\mathbf{A}^{'}_{k,b}\}_{k\in[\kappa],b\in\{0,1\}}),$$

where every $\mathbf{A}^{'}_{k,b} = \mathbf{I}$ is the identity matrix in $\{0,1\}^{w\times w}$.

Step 2: Random diagonal entries and bookends. Let $s = 2m + w$, where $m = l + 3$ in the original GGHRSW scheme.

For $k \in [\kappa]$, we extend $w \times w$-dimensional matrices into $s \times s$-dimensional matrices

$$\widehat{\mathbf{A}}_{k,b} = \begin{pmatrix} \mathbf{E}_{k,b} & 0 \\ 0 & \mathbf{A}_{k,b} \end{pmatrix}, \quad \widehat{\mathbf{A}}^{\prime}_{k,b} = \begin{pmatrix} \mathbf{E}^{\prime}_{k,b} & 0 \\ 0 & \mathbf{A}^{\prime}_{k,b} \end{pmatrix},$$

where the diagonal matrices $\mathbf{E}_{k,b}, \mathbf{E}^{\prime}_{k,b} \in R^{2m\times 2m}_{\sigma}$ are chosen uniformly at random from the plaintext space.

We also choose four "bookend" vectors as follows:

$$\begin{cases} \widehat{\mathbf{A}}_0 = \left(0^m, \mathbf{e}_0, \mathbf{s}\right), \\ \widehat{\mathbf{A}}^{\prime}_0 = \left(0^m, \mathbf{e}^{\prime}_0, \mathbf{s}^{\prime}\right), \end{cases}$$

$$\begin{cases} \widehat{\mathbf{A}}_{\kappa+1} = \left(\mathbf{e}_{\kappa+1}, \mathbf{0}^m, \mathbf{t}\right)^T, \\ \widehat{\mathbf{A}}^{\prime}_{\kappa+1} = \left(\mathbf{e}^{\prime}_{\kappa+1}, \mathbf{0}^m, \mathbf{t}^{\prime}\right)^T, \end{cases}$$

where $\mathbf{e}_0, \mathbf{e}'_0, \mathbf{e}_{\kappa+1}, \mathbf{e}'_{\kappa+1} \in R^m_\sigma$, and $\mathbf{s}, \mathbf{s}', \mathbf{t}, \mathbf{t}' \in R^w_\sigma$ such that $\mathbf{s} \cdot \mathbf{t}^T = \mathbf{s}' \cdot \mathbf{t}'^T$.

Step 3: Kilian randomization and bundling scalars. We first sample random scalars $\{\epsilon_0, \epsilon'_0, \epsilon_{\kappa+1}, \epsilon'_{\kappa+1}, \epsilon_{k,b}, \epsilon'_{k,b} \leftarrow R_\sigma : k \in [\kappa], b \in \{0,1\}\}$ such that

$$\begin{aligned} \alpha_{j,b} &= \prod\nolimits_{\mathrm{inp}(k)=j} \epsilon_{k,b} = \prod\nolimits_{\mathrm{inp}(k)=j} \epsilon'_{k,b}, \\ \alpha_0 &= \epsilon_0 \epsilon_{\kappa+1} = \epsilon'_0 \epsilon'_{\kappa+1}. \end{aligned}$$

Then, we choose randomly unimodular matrices $\mathbf{P}_0, \mathbf{P}'_0, \mathbf{P}_k, \mathbf{P}'_k \in R^{s\times s}_\sigma, k \in [\kappa]$, and generate randomized matrices as follows:

$$\begin{cases} \widetilde{\mathbf{A}}_0 = \epsilon_0 \widehat{\mathbf{A}}_0 \mathbf{P}_0 \\ \widetilde{\mathbf{A}}_{k,b} = \epsilon_{k,b} \mathbf{P}^{-1}_{k-1} \widehat{\mathbf{A}}_{k,b} \mathbf{P}_k \\ \widetilde{\mathbf{A}}_{\kappa+1} = \epsilon_{\kappa+1} \mathbf{P}^{-1}_\kappa \widehat{\mathbf{A}}_{\kappa+1} \end{cases}, \qquad \begin{cases} \widetilde{\mathbf{A}}'_0 = \epsilon'_0 \widehat{\mathbf{A}}'_0 \mathbf{P}'_0, \\ \widetilde{\mathbf{A}}'_{k,b} = \epsilon'_{k,b} \mathbf{P}'^{-1}_{k-1} \widehat{\mathbf{A}}'_{k,b} \mathbf{P}'_k, \\ \widetilde{\mathbf{A}}'_{\kappa+1} = \epsilon'_{\kappa+1} \mathbf{P}'^{-1}_\kappa \widehat{\mathbf{A}}'_{\kappa+1} \end{cases},$$

where $k \in [\kappa], b \in \{0,1\}$.

Step 4: Encoding using GGH13 without ideals. For $k = 0, \cdots, \kappa+1$, we sample uniformly invertible random elements $z_k \in R_q$, and $\beta_k \in R$ such that $\sqrt[\kappa+3]{q} < \|\beta_k\| < \sqrt[\kappa+2]{q}$. We then choose at random vectors $\mathbf{R}_0, \mathbf{R}'_0, \mathbf{R}_{\kappa+1}, \mathbf{R}'_{\kappa+1} \in R^s_\sigma$, and matrices $\mathbf{R}_{k,b}, \mathbf{R}'_{k,b} \in R^{s\times s}_\sigma$, and set

$$\begin{cases} \overline{\mathbf{A}}_0 = (\widetilde{\mathbf{A}}_0 + \mathbf{R}_0/\beta_0)/z_0 \\ \overline{\mathbf{A}}_{k,b} = (\widetilde{\mathbf{A}}_{k,b} + \mathbf{R}_{k,b}/\beta_k)/z_k \\ \overline{\mathbf{A}}_{\kappa+1} = (\widetilde{\mathbf{A}}_{\kappa+1} + \mathbf{R}_{\kappa+1}/\beta_{\kappa+1})/z_{\kappa+1} \cdot p_{zt} \end{cases},$$

$$\begin{cases} \overline{\mathbf{A}}'_0 = (\widetilde{\mathbf{A}}'_0 + \mathbf{R}'_0/\beta_0)/z_0 \\ \overline{\mathbf{A}}'_{k,b} = (\widetilde{\mathbf{A}}'_{k,b} + \mathbf{R}'_{k,b}/\beta_k)/z_k \\ \overline{\mathbf{A}}'_{\kappa+1} = (\widetilde{\mathbf{A}}'_{\kappa+1} + \mathbf{R}'_{\kappa+1}/\beta_{\kappa+1})/z_{\kappa+1} \cdot p_{zt} \end{cases},$$

where $k \in [\kappa], b \in \{0,1\}$, and $p_{zt} = \prod_{k=0}^{\kappa+1} \beta_k z_k$.

Step 5: Output the obfuscation of BP. The obfuscation $\overline{\mathrm{BP}}$ consists of the following matrices and vectors:

$$\begin{cases} \left\{\overline{\mathbf{A}}_0, \{\overline{\mathbf{A}}_{k,b}\}_{k\in[\kappa],b\in\{0,1\}}, \overline{\mathbf{A}}_{\kappa+1}\right\}, \\ \left\{\overline{\mathbf{A}}'_0, \{\overline{\mathbf{A}}'_{k,b}\}_{k\in[\kappa],b\in\{0,1\}}, \overline{\mathbf{A}}'_{\kappa+1}\right\}. \end{cases}$$

Remark 3.1. (1) To perform Kilian randomization, we use the unimodular matrices $\mathbf{P}_k, \mathbf{P}'_k$. Since if choosing $\mathbf{P}_k$ randomly, then $\|\mathbf{P}_k^{-1} \mod \beta_k\|_\infty \approx \|\beta_k\|$ for $k \in [\kappa]$. Namely, by a change of variable transformation, we cannot rewrite the encodings as

$$\begin{aligned} \overline{\mathbf{A}}_{k,b} &= (\epsilon_{k,b} \mathbf{P}^{-1}_{k-1} \widehat{\mathbf{A}}_{k,b} \mathbf{P}_k + \mathbf{R}_{k,b}/\beta_k)/z_k \\ &= (\epsilon_{k,b} \mathbf{P}^{-1}_{k-1} (\widehat{\mathbf{A}}_{k,b} + \mathbf{R}'_{k,b}/\beta_k) \mathbf{P}_k)/z_k, \end{aligned}$$

such that $\|\mathbf{R}'_{k,b}\|_\infty$ is 'small'.

Because in this case $\|\mathbf{R}'_{k,b}\|_\infty = \|\epsilon_{k,b}^{-1}\mathbf{P}_{k-1}\mathbf{R}_{k,b}\mathbf{P}_k^{-1} \mod \beta_k\|_\infty \approx \|\beta_k\|$. This point is different from the GGH13 encoding since g is small and hence so $\|\mathbf{P}_k^{-1} \mod g\|_\infty$. However for the elements returned by zero-testing, we can write $\mathbf{R}'_{k,b} = \epsilon_{k,b}^{-1}\mathbf{P}_{k-1}\mathbf{R}_{k,b}\mathbf{P}_k^{-1}$ since now all the operations are in the field $\mathbb{K}$.

(2) Alternatively, when choosing randomly $\mathbf{P}_k$ we can also take its adjugate matrix $\mathrm{adj}(\mathbf{P}_k)$ instead of $\mathbf{P}_k^{-1}$.

Evaluation. Given the obfuscation $\overline{\mathrm{BP}}$ and an arbitrary input $\mathbf{x} \in \{0,1\}^l$, we compute an honest evaluation as follows:

$$\begin{aligned}\delta &= \overline{\mathbf{A}}_0 \cdot \prod_{k=1}^{\kappa} \overline{\mathbf{A}}_{k,x_{\mathrm{inp}(k)}} \cdot \overline{\mathbf{A}}_{\kappa+1} \\ &= (\beta_0\widetilde{\mathbf{A}}_0 + \mathbf{R}_0) \cdot \prod_{k=1}^{\kappa}(\beta_k\widetilde{\mathbf{A}}_{k,x_{\mathrm{inp}(k)}} + \mathbf{R}_{k,x_{\mathrm{inp}(k)}}) \cdot (\beta_{\kappa+1}\widetilde{\mathbf{A}}_{\kappa+1} + \mathbf{R}_{\kappa+1}), \\ &= \alpha\beta \cdot \mathbf{s}\prod_{k=1}^{\kappa}\mathbf{A}_{k,x_{inp(k)}}\mathbf{t}^T + o(\beta)\end{aligned}$$

$$\begin{aligned}\delta' &= \overline{\mathbf{A}}'_0 \cdot \prod_{k=1}^{\kappa} \overline{\mathbf{A}}'_{k,x_{\mathrm{inp}(k)}} \cdot \overline{\mathbf{A}}'_{\kappa+1} \\ &= (\beta_0\widetilde{\mathbf{A}}'_0 + \mathbf{R}'_0) \cdot \prod_{k=1}^{\kappa}(\beta_k\widetilde{\mathbf{A}}'_{k,x_{\mathrm{inp}(k)}} + \mathbf{R}'_{k,x_{\mathrm{inp}(k)}}) \cdot (\beta_{\kappa+1}\widetilde{\mathbf{A}}'_{\kappa+1} + \mathbf{R}'_{\kappa+1}), \\ &= \alpha\beta \cdot \mathbf{s}'\mathbf{t}'^T + o(\beta)\end{aligned}$$

where $\alpha = \prod_{j=1}^{l}\alpha_{j,x_j}$ and $\beta = \prod_{j=0}^{\kappa+1}\beta_j$.

If $\prod_{k=1}^{\kappa}\mathbf{A}_{k,x_{inp(k)}} = \mathbf{I}$, then $\|\delta - \delta'\| < q^{\frac{\kappa+1}{\kappa+2}}$ and $\overline{\mathrm{BP}}(\mathbf{x}) = 1$. Otherwise, $\overline{\mathrm{BP}}(\mathbf{x}) = 0$.

4 Matrix Properties

In this section, we give some matrix properties. Let γ, δ be positive numbers such that $\delta/\gamma \leq 2^{-O(\lambda)}$. For simplicity,we denote $R_{\mathbf{A}}[i] = \sum_{j\neq i}|A[i,j]|$ in the following.

A permutation $p = (p_1, p_2, \cdots, p_n)$ of the numbers $(1, 2, \cdots, n)$ is any rearrangement. The parity of a permutation p is the one of the number of interchanges to restore p to natural order. Consequently, the sign of a permutation p is defined to be the number

$$\pi(p) = \begin{cases} +1 & \text{if the parity of } p \text{ is even,} \\ -1 & \text{if the parity of } p \text{ is odd.} \end{cases}$$

Given a $n \times n$-dimensional matrix $\mathbf{A} = (A[i,j])$, the determinant of A is defined to be the scalar

$$\det(\mathbf{A}) = \sum_p \pi(p)\prod_{i=1}^{n} A[i,p_i], \tag{1}$$

where the sum is taken over the $n!$ permutations p of $(1, 2, \cdots, n)$.

Lemma 4.1 Determinant Inequality. Suppose that $\mathbf{A}$ is an $n \times n$-dimensional matrix over $\mathbb{Q}$ such that $\gamma \leq |A[i,j]| \leq c\gamma$ for $i, j \in [n]$, where $\gamma > 2^{\lambda}$ and $c > 1$. Then with overwhelming probability

$$\gamma^n \leq |\det(\mathbf{A})| \leq n!(c\gamma)^n.$$

Proof. According to the definition of determinant (1),

$$\begin{aligned} |\det(\mathbf{A})| &= \Big| \sum_p \pi(p) \prod_{i=1}^{n} A[i, p_i] \Big| \\ &= \gamma^n \cdot \Big| \sum_p \pi(p) \prod_{i=1}^{n} \frac{A[i, p_i]}{\gamma} \Big| \\ &= \gamma^n \cdot \Big| \sum_p \pi(p) A_p \Big|, \end{aligned}$$

where $A_p = \prod_{i=1}^{n} \frac{A[i,p_i]}{\gamma}$.

By $\gamma \leq |A[i,j]| \leq c\gamma$, we obtain $|\frac{A[i,p_i]}{\gamma}| \geq 1$ and $1 \leq |A_p| \leq c^n$. According to Chernoff-Hoefding inequality, $\big|\sum_p \pi(p) A_p\big| \geq 1$ with overwhelming probability. On the other hand, $\big|\sum_p \pi(p) A_p\big| \leq \sum_p |\pi(p)| c^n = n! c^n$. ∎

Definition 4.2 Matrix Decomposition ($\mathrm{MD}_{\gamma,\delta}$). The decomposition $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_\delta$ is called $\mathrm{MD}_{\gamma,\delta}$ if $\mathbf{A}_1, \mathbf{A}_\delta$ are satisfied

$$\begin{aligned} |A_1[i,j]| &= \Theta(\gamma), \text{for all } i, j \in [n] \\ |A_\delta[i,j]| &= O(\delta), \text{for all } i, j \in [n]. \end{aligned}$$

Lemma 4.3 Determinant Estimate I. Suppose that $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_\delta$ is $\mathrm{MD}_{\gamma,\delta}$ and $\mathrm{rank}(\mathbf{A}_1) < n$. Then $|\det(\mathbf{A})| \leq O(n \cdot n! \cdot \delta\gamma^{n-1})$. In particular, $|\det(\mathbf{A})| = O(\delta\gamma^{n-1})$ when n is constant.

Proof. By the definition of determinant (1),

$$\det(\mathbf{A}) = \sum_p \pi(p) \prod_{i=1}^{n} A[i, p_i] = \sum_p \pi(p) \prod_{i=1}^{n} (A_1[i, p_i] + A_\delta[i, p_i])$$

By $\mathrm{rank}(\mathbf{A}_1) < n$, $\det(\mathbf{A}_1) = 0$. That is, $\sum_p \pi(p) \prod_{i=1}^{n} A_1[i, p_i] = 0$.

We expand $\det(\mathbf{A})$ as follows:

$$\begin{aligned}\det(\mathbf{A}) &= \sum_{p}\pi(p)\prod_{i=1}^{n}(A_1[i,p_i]+A_\delta[i,p_i]) \\ &= \sum_{p}\pi(p)(\underbrace{\sum_{j=1}^{n}A_\delta[j,p_j]\prod_{i\neq j}A_1[i,p_i]}_{B_p}+o(B_p))\end{aligned}$$

So, $|\det(\mathbf{A})| \leq \sum_{p}\pi(p)\sum_{j=1}^{n}O(|A_\delta[j,p_j]\prod_{i\neq j}A_1[i,p_i]|) \leq O(n\cdot n!\cdot\delta\gamma^{n-1})$.

Furthermore, $|\det(\mathbf{A})| \leq O(\delta\gamma^{n-1})$ when n is constant. ∎

Remark 4.4. The result of Lemma 4.3 does not contradict that of Lemma 4.1. Because the former matrix $\mathbf{A}$ is randomly selected, and the latter matrix $\mathbf{A}$ has a special structure such that the rank of the dominant matrix corresponding to its decomposition is less than n.

Moreover, if we assume that the square submatrix obtained by the linearly independent vectors of $\mathbf{A}_1$ in Lemma 4.3 satisfies the condition of Lemma 4.1. Namely, the determinant of the square submatrix can be estimated by applying Lemma 4.1. Accordingly, we can further improve the determinant estimation in Lemma 4.3. Note that the following Lemma 4.5 is not used in this paper.

Lemma 4.5 Determinant Estimate II. Suppose that $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_\delta$ is $\mathrm{MD}_{\gamma,\delta}$ and $\mathrm{rank}(\mathbf{A}_1) = k < n$. Then $|\det(\mathbf{A})| \leq n!(\gamma+\delta^{'})^k(\delta^{'})^{n-k}$, where $\delta^{'} = nk!c^k\delta$ and c is a constant that depends on $\mathbf{A}$. Furthermore, $|\det(\mathbf{A})| \leq O(\delta^{n-k}\gamma^k)$ when n is constant.

Proof. For brevity, $A[i//j]$ represents the matrix of the i-th to k-th rows of $\mathbf{A}$, and $A[i:j]$ the matrix of the i-th to k-th columns of $\mathbf{A}$.

Without loss of generality, assume that the first k rows of $\mathbf{A}_1$ are linearly independent since $\mathrm{rank}(\mathbf{A}_1) = k$. So, the last $n-k$ rows of $\mathbf{A}_1$ are linearly dependent with its first k rows. That is, for $j \in \{k+1,\cdots,n\}$,

$$A_1[j,\cdot] = \sum_{i=1}^{k} y_j[i]A_1[i,\cdot].$$

Now, we write this relationship as matrix-vector form

$$A_1[j,\cdot] = \mathbf{y}_j\mathbf{A}_1^{'} = \mathbf{y}_j(\mathbf{B}_1,\mathbf{B}_2),$$

where $\mathbf{A}_1^{'} = A_1[1//k], \mathbf{B}_1 = A_1^{'}[1:k], \mathbf{B}_2 = A_1^{'}[k+1:n]$.

By Cramer's rule, we compute

$$y_j[i] = \frac{\det(\mathbf{B}_{1,i})}{\det(\mathbf{B}_1)},$$

where for $i \in [k]$, $\mathbf{B}_{1,i} = \left(B_1[1//i-1]//A_1[j,\cdot]//B_1[i+1//k]\right)$.

By $|A_1[i,j]| = \Theta(\gamma)$, we have $c_1\gamma \leq |A_1[i,j]| \leq c_2\gamma$.
Using Lemma 4.1, we yield

$$(c_1\gamma)^k \leq |\det(\mathbf{B}_1)| \leq k!(c_2\gamma)^k,$$
$$(c_1\gamma)^k \leq |\det(\mathbf{B}_{1,i})| \leq k!(c_2\gamma)^k.$$

Let $c = c_2/c_1$. For $j \in \{k+1, \cdots, n\}, i \in [k]$, we get

$$\frac{1}{k!c^k} < y_j[i] < k!c^k.$$

Now, we set

$$\mathbf{Y} = \begin{pmatrix} y_{k+1}[1] & y_{k+1}[2] & \cdots & y_{k+1}[k] \\ y_{k+2}[1] & y_{k+2}[2] & \cdots & y_{k+2}[k] \\ \vdots & \vdots & \cdots & \vdots \\ y_n[1] & y_n[2] & \cdots & y_n[k] \end{pmatrix},$$
$$\mathbf{P} = \begin{pmatrix} \mathbf{I}_k & \mathbf{0} \\ -\mathbf{Y} & \mathbf{I}_{n-k} \end{pmatrix}.$$

Therefore,

$$\mathbf{PA} = \mathbf{PA}_1 + \mathbf{PA}_\delta = \begin{pmatrix} \mathbf{A}_1^{'} \\ \mathbf{0} \end{pmatrix} + \mathbf{A}_{\delta'},$$

By $\mathbf{A}_{\delta'} = \mathbf{PA}_\delta$, we obtain

$$|A_{\delta'}[i,j]| = |\sum_{k=1}^{n} P[i,k]A_\delta[k,j]| \leq nk!c^k\delta$$

So, $\delta^{'} = nk!c^k\delta$.
By the definition of determinant (1),

$$\begin{aligned} \det(\mathbf{A}) &= \det(\mathbf{PA}) \\ &= \det\left(\begin{pmatrix} \mathbf{A}_1^{'} \\ \mathbf{0} \end{pmatrix} + \mathbf{A}_{\delta'} \right) \\ &= \sum_p \pi(p) \prod_{i=1}^{k} (A_1[i,p_i] + A_{\delta'}[i,p_i]) \prod_{i=k+1}^{n} A_{\delta'}[i,p_i] \end{aligned}$$

Since $|A_1[i,p_i]| \leq \gamma$ and $A_{\delta'}[i,p_i] \leq \delta^{'}$, thus

$$|\det(\mathbf{A})| \leq n!(\gamma + \delta^{'})^k(\delta^{'})^{n-k}.$$

Obviously, $|\det(\mathbf{A})| \leq O(\delta^{n-k}\gamma^k)$ when n is constant. ∎

Definition 4.6 Approximate Eigenvalue. Let $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_\delta$ be a (γ, δ)-matrix decomposition. The eigenvalues of $\mathbf{A}$ are called the approximate eigenvalues of $\mathbf{A}_1$.

Definition 4.7 Diagonally Dominant Matrix (DDM). An $n \times n$-dimensional matrix A is diagonally dominant if for all $i \in [k]$,

$$|A[i,i]| \geq \sum\nolimits_{j \neq i} |A[i,j]|.$$

If using a strict inequality ($>$) instead ($\geq$) in the above definition, then A is called strict diagonally dominant matrix (SDDM).

Definition 4.8 (γ, δ)-Diagonally Dominant Matrix ($\mathrm{DDM}_{\gamma,\delta}$). $\mathbf{A}$ is a (γ, δ)-diagonally dominant matrix if $\mathbf{A}$ is satisfied

$$|A[i,j]| = \begin{cases} O(\gamma), & \text{if } i = j \\ O(\delta), & \text{if } i \neq j. \end{cases}$$

Note that we only consider the approximate eigenvalue of diagonally dominant matrix in this paper. In the following we will prove the inverse of a $\mathrm{DDM}_{\gamma,\delta}$ matrix is a $\mathrm{DDM}_{\gamma',\delta'}$ matrix. Moreover, it is not difficult to verify that the product of two $\mathrm{DDM}_{\gamma_i,\delta_i}, i \in [2]$ matrices is a $\mathrm{DDM}_{\gamma_1\gamma_2,\delta_1\gamma_2+\delta_2\gamma_1}$ matrix. By using these properties we can compute the eigenvalues of a $\mathrm{DDM}_{\gamma,\delta}$ matrix as the approximate eigenvalues of its diagonal dominant matrix.

Lemma 4.9. Suppose that $\mathbf{A}$ is a $\mathrm{DDM}_{\gamma,\delta}$ matrix. Then $\mathbf{A}^{-1}$ is a $\mathrm{DDM}_{\gamma^{-1},n\delta/\gamma^{-2}}$ matrix.

Proof. Since $\mathbf{A}$ is a $\mathrm{DDM}_{\gamma,\delta}$ matrix, we can write $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_\delta$ such that $\mathbf{A}_1$ is a diagonal matrix and

$$\begin{aligned} |A_1[i,i]| &= O(\gamma), & i &\in [n] \\ |A_\delta[i,j]| &= O(\delta), & i,j &\in [n]. \end{aligned}$$

So, $\mathbf{A}_1^{-1} = \mathrm{Diag}(A^{-1}[1,1], \cdots, A^{-1}[n,n])$.

Again since $\mathbf{A}$ is a $\mathrm{DDM}_{\gamma,\delta}$ matrix, we have $|A[i,i]| = O(\gamma)$. Without loss of generality, assume $\gamma_{\max}^{-1} = \max\limits_{i \in [n]} \{A^{-1}[i,i]\} = O(\gamma^{-1})$.

By $\|\mathbf{A}_1^{-1}\mathbf{A}_\delta\| \leq \|\mathbf{A}_1^{-1}\|\|\mathbf{A}_\delta\| \leq O(n\gamma_{\max}^{-1}\delta) = O(n\gamma^{-1}\delta) \ll 1$, we have

$$\begin{aligned} \mathbf{A}^{-1} &= (\mathbf{A}_1 + \mathbf{A}_\delta)^{-1} \\ &= (\mathbf{I} + \mathbf{A}_1^{-1}\mathbf{A}_\delta)^{-1}\mathbf{A}_1^{-1} \\ &= (\mathbf{I} - \mathbf{A}_1^{-1}\mathbf{A}_\delta + (\mathbf{A}_1^{-1}\mathbf{A}_\delta)^2 - \cdots)\mathbf{A}_1^{-1} \\ &= \mathbf{A}_1^{-1} + \mathbf{A}_{\delta'} \end{aligned}$$

where $\mathbf{A}_{\delta'} = (-\mathbf{A}_1^{-1}\mathbf{A}_\delta + (\mathbf{A}_1^{-1}\mathbf{A}_\delta)^2 - \cdots)\mathbf{A}_1^{-1}$.

Again,

$$\begin{aligned}\|\mathbf{A}_{\delta'}\| &\leq (\|\mathbf{A}_1^{-1}\mathbf{A}_\delta\| + \|(\mathbf{A}_1^{-1}\mathbf{A}_\delta)^2\| + \cdots)\|\mathbf{A}_1^{-1}\| \\ &= \sum_{i=1}^{\infty}(O(n\gamma^{-1}\delta))^i O(\gamma^{-1}) \\ &= \frac{O(n\gamma^{-1}\delta)}{1 - O(n\gamma^{-1}\delta)}O(\gamma^{-1}) \\ &= O(n\delta\gamma^{-2}).\end{aligned}$$

Consequently, $|\mathbf{A}_{\delta'}[i,j]| = O(n\delta\gamma^{-2})$ for all $i, j \in [n]$, and hence

$$|\mathbf{A}^{-1}[i,j]| = \begin{cases} O(\gamma^{-1}), & \text{if } i = j \\ O(n\delta\gamma^{-2}), & \text{if } i \neq j \end{cases}.$$

Therefore, $\mathbf{A}^{-1}$ is a $\mathrm{DDM}_{\gamma^{-1}, n\delta/\gamma^{-2}}$ matrix. ∎

Remark 4.10. Although the results of all the lemmas above are given over the field $\mathbb{Q}$, they can be directly extended to the field $\mathbb{K} = \mathbb{Q}[x]/\langle f(x)\rangle$. Note that in this case we require to use the norm of the elements in $\mathbb{K}$, instead of using the absolute value in $\mathbb{Q}$.

5 Cryptanalysis

Since the BP obfuscator using GGH13 without ideals [2] no longer uses ideals, we cannot obtain a basis of the ideal β_k as that of the CGH attack. Also, we cannot find some exact representatives of the bundling scalars due to the noise. However, we can recover some approximate ratios of the bundling scalars by applying the matrix properties described in the above section. Applying these approximate ratios, we present a variant of the CGH attack to break the BP obfuscator using GGH13 without ideals.

5.1 Branching program with input partitioning

We first adaptively recall the branching program with input partitioning in [11]. Let $X||Y||Z = [\kappa]$ be a 3-partition of the branching program steps. For a 3-partition input $f = xyz$, we use $\mathbf{S}_x$ (resp. $\mathbf{S}_y, \mathbf{S}_z$) to denote the plaintext product matrix of function branch in the X (resp. Y, Z) interval, and $\mathbf{S}'_x$ (resp. $\mathbf{S}'_y, \mathbf{S}'_z$) the plaintext product matrix of dummy branch in the X (resp. Y, Z) interval. In addition, we denote by $|S|$ the number of elements in a set S.

For the function branch, it is easy to obtain

$$\begin{aligned}
\mathbf{S}_x &= \widetilde{\mathbf{A}}_0 \prod\nolimits_{k\in X} \widetilde{\mathbf{A}}_{k,u_{\mathrm{inp}(k)}} = \epsilon_0\alpha_x\widehat{\mathbf{A}}_0 \times \prod\nolimits_{k\in X} \widehat{\mathbf{A}}_{k,u_{\mathrm{inp}(k)}} \times \mathbf{P}_{y_1} \\
&= \epsilon_0\alpha_x\widehat{\mathbf{A}}_0 \times \widehat{\mathbf{A}}_x \times \mathbf{P}_{y_1}, \\
\mathbf{S}_y &= \prod\nolimits_{k\in Y} \widetilde{\mathbf{A}}_{k,u_{\mathrm{inp}(k)}} = \alpha_y\mathbf{P}_{y_1}^{-1} \times \prod\nolimits_{k\in Y} \widehat{\mathbf{A}}_{k,u_{\mathrm{inp}(k)}} \times \mathbf{P}_{z_1} \\
&= \alpha_y\mathbf{P}_{y_1}^{-1} \times \widehat{\mathbf{A}}_y \times \mathbf{P}_{z_1}, \\
\mathbf{S}_z &= \prod\nolimits_{k\in Z} \widetilde{\mathbf{A}}_{k,u_{\mathrm{inp}(k)}} \times \widetilde{\mathbf{A}}_{\kappa+1} = \epsilon_{\kappa+1}\alpha_z\mathbf{P}_{z_1}^{-1} \times \prod\nolimits_{k\in Z} \widehat{\mathbf{A}}_{k,u_{\mathrm{inp}(k)}} \times \widehat{\mathbf{A}}_{\kappa+1} \\
&= \epsilon_{\kappa+1}\alpha_z\mathbf{P}_{z_1}^{-1} \times \widehat{\mathbf{A}}_z \times \widehat{\mathbf{A}}_{\kappa+1}.
\end{aligned}$$

Similarly, for the dummy branch we have

$$\begin{aligned}
\mathbf{S}'_x &= \widetilde{\mathbf{A}}'_0 \prod\nolimits_{k\in X} \widetilde{\mathbf{A}}'_{k,u_{\mathrm{inp}(k)}} = \epsilon'_0\alpha'_x\widehat{\mathbf{A}}'_0 \times \prod\nolimits_{k\in X} \widehat{\mathbf{A}}'_{k,u_{\mathrm{inp}(k)}} \times \mathbf{P}'_{y_1} \\
&= \epsilon'_0\alpha'_x\widehat{\mathbf{A}}'_0 \times \widehat{\mathbf{A}}'_x \times \mathbf{P}'_{y_1}, \\
\mathbf{S}'_y &= \prod\nolimits_{k\in Y} \widetilde{\mathbf{A}}'_{k,u_{\mathrm{inp}(k)}} = \alpha'_y\mathbf{P}'^{-1}_{y_1} \times \prod\nolimits_{k\in Y} \widehat{\mathbf{A}}'_{k,u_{\mathrm{inp}(k)}} \times \mathbf{P}'_{z_1} \\
&= \alpha'_y\mathbf{P}'^{-1}_{y_1} \times \widehat{\mathbf{A}}'_y \times \mathbf{P}'_{z_1}, \\
\mathbf{S}'_z &= \prod\nolimits_{k\in Z} \widetilde{\mathbf{A}}'_{k,u_{\mathrm{inp}(k)}} \times \widetilde{\mathbf{A}}'_{\kappa+1} = \epsilon'_{\kappa+1}\alpha'_z\mathbf{P}'^{-1}_{z_1} \times \prod\nolimits_{k\in Z} \widehat{\mathbf{A}}'_{k,u_{\mathrm{inp}(k)}} \times \widehat{\mathbf{A}}'_{\kappa+1} \\
&= \epsilon'_{\kappa+1}\alpha'_z\mathbf{P}'^{-1}_{z_1} \times \widehat{\mathbf{A}}'_z \times \widehat{\mathbf{A}}'_{\kappa+1},
\end{aligned}$$

where the scalars $\alpha_x, \alpha_y, \alpha_z$, etc. are the product of all the $\epsilon_{k,b}$ in the corresponding branch, and $y_1 = |X|$, $z_1 = |(X||Y)|$.

For these bundling scalars $\alpha_x, \alpha_y, \alpha_z$, etc., our attack requires to use the following results in [11].

Lemma 5.1 (Lemma 2.3 [11]). Suppose that $f^{(i,j,t)} = x^{(i)}y^{(j)}z^{(t)}$ are some 3-partition inputs that are all zeros of the function. Then $\alpha_{x^{(1)}}/\alpha_{x'^{(1)}} = \alpha_{x^{(2)}}/\alpha_{x'^{(2)}} = \cdots$, and similarly $\alpha_{y^{(1)}}/\alpha_{y'^{(1)}} = \alpha_{y^{(2)}}/\alpha_{y'^{(2)}} = \cdots$ and $\alpha_{z^{(1)}}/\alpha_{z'^{(1)}} = \alpha_{z^{(2)}}/\alpha_{z'^{(2)}} = \cdots$.

5.2 Generating approximate ratios of the bundling scalars

Without loss of generality, we assume that the branching program is 3-partitioned. Let $f^{(i,b,j)} = x^{(i)}y^{(b)}z^{(j)}$ be a 3-partition input of the form $X||Y||Z$ that is an input of a zero of the function. Let i, j range over $2s$ inputs and for $b \in \{0,1\}$, then we first obtain the matrices:

$$\begin{aligned}
\mathbf{W}_b &= \mathbf{X}\mathbf{Y}_b\mathbf{Z} \\
&= \begin{pmatrix} \cdots \\ \beta_X\mathbf{S}_{x^{(i)}} + \mathbf{R}_{x^{(i)}}, -\beta_X\mathbf{S}'_{x^{(i)}} + \mathbf{R}'_{x^{(i)}} \\ \cdots \end{pmatrix} \times \\
&\quad \begin{pmatrix} \beta_Y\mathbf{S}_{y^{(b)}} + \mathbf{R}_{y^{(b)}}, & 0 \\ 0 & \beta_Y\mathbf{S}'_{y^{(b)}} + \mathbf{R}'_{y^{(b)}} \end{pmatrix} \times \begin{pmatrix} \cdots, \beta_Z\mathbf{S}_{z^{(j)}} + \mathbf{R}_{z^{(j)}}, \cdots \\ \beta_Z\mathbf{S}'_{z^{(j)}} + \mathbf{R}'_{z^{(j)}}, \end{pmatrix},
\end{aligned}$$

where $\mathbf{X}, \mathbf{Y}_b, \mathbf{Z} \in R^{2s\times 2s}$ are full rank with high probability, and β_X (resp. β_Y and β_Z) is equal to the product $\prod_{k\in X}\beta_k$ (resp. $\prod_{k\in Y}\beta_k$ and $\prod_{k\in Z}\beta_k$).

Then, we compute the characteristic polynomial of $\mathbf{W}_1\mathbf{W}_0^{-1}$ over $\mathbb{K}$ that is equal to the characteristic polynomial of $\mathbf{Y}_1\mathbf{Y}_0^{-1}$.

Now we analyze $\mathbf{Y}_1\mathbf{Y}_0^{-1}$ over $\mathbb{K}$ as follows:

$$\mathbf{Y}_1\mathbf{Y}_0^{-1} = \begin{pmatrix} \beta_Y\mathbf{S}_{y^{(1)}} + \mathbf{R}_{y^{(1)}}, & 0 \\ 0 & \beta_Y\mathbf{S}'_{y^{(1)}} + \mathbf{R}'_{y^{(1)}} \end{pmatrix} \begin{pmatrix} \beta_Y\mathbf{S}_{y^{(0)}} + \mathbf{R}_{y^{(0)}}, & 0 \\ 0 & \beta_Y\mathbf{S}'_{y^{(0)}} + \mathbf{R}'_{y^{(0)}} \end{pmatrix}^{-1}$$

According to the BP obfuscator construction, we have

$$\begin{aligned} \beta_Y\mathbf{S}_{y^{(0)}} + \mathbf{R}_{y^{(0)}} &= \beta_Y\alpha_{y^{(0)}}\mathbf{P}_{y_1}^{-1}\widehat{\mathbf{A}}_{y^{(0)}}\mathbf{P}_{z_1} + \mathbf{R}_{y^{(0)}} \\ &= \mathbf{P}_{y_1}^{-1}(\underbrace{\beta_Y\alpha_{y^{(0)}}\widehat{\mathbf{A}}_{y^{(0)}}}_{\mathbf{A}_1} + \underbrace{\mathbf{P}_{y_1}\mathbf{R}_{y^{(0)}}\mathbf{P}_{z_1}^{-1}}_{\mathbf{A}_\delta})\mathbf{P}_{z_1}, \end{aligned}$$

where $\mathbf{A}_1$ is a diagonal matrix.

By the parameter settings, it is easy to verify that $\delta = \|\mathbf{A}_\delta\| = O(s^2n^{1.5}\sigma^3)$ and $\gamma = \max_{i\in[n]}\|A_1[i,i]\| \approx O(\beta_Y\alpha_{y^{(0)}})$ such that $\delta/\gamma \le 2^{-O(\lambda)}$. So, by Lemma 4.9 we get

$$(\beta_Y\mathbf{S}_{y^{(0)}} + \mathbf{R}_{y^{(0)}})^{-1} = \mathbf{P}_{z_1}^{-1}(\mathbf{A}_1^{-1} + \mathbf{A}_{\delta'})\mathbf{P}_{y_1},$$

where $\delta' = n\delta/\gamma^2$.

Thus, we can compute the function branching part of $\mathbf{Y}_1\mathbf{Y}_0^{-1}$ as follows:

$$\begin{aligned} &(\beta_Y\mathbf{S}_{y^{(1)}} + \mathbf{R}_{y^{(1)}})(\beta_Y\mathbf{S}_{y^{(0)}} + \mathbf{R}_{y^{(0)}})^{-1} \\ &= (\beta_Y\alpha_{y^{(1)}}\mathbf{P}_{y_1}^{-1}\widehat{\mathbf{A}}_{y^{(1)}}\mathbf{P}_{z_1} + \mathbf{R}_{y^{(1)}})\mathbf{P}_{z_1}^{-1}(\mathbf{A}_1^{-1} + \mathbf{A}_{\delta'})\mathbf{P}_{y_1} \\ &= \frac{\alpha_{y^{(1)}}}{\alpha_{y^{(0)}}}\mathbf{P}_{y_1}^{-1}(\widehat{\mathbf{A}}_{y^{(1)}}\widehat{\mathbf{A}}_{y^{(0)}}^{-1})\mathbf{P}_{y_1} + \mathbf{R} \\ &= \frac{\alpha_{y^{(1)}}}{\alpha_{y^{(0)}}}\mathbf{P}_{y_1}^{-1}\begin{pmatrix} \mathbf{E}_{y^{(1)}} & 0 \\ 0 & \mathbf{A}_{y^{(1)}} \end{pmatrix}\begin{pmatrix} \mathbf{E}_{y^{(0)}} & 0 \\ 0 & \mathbf{A}_{y^{(0)}} \end{pmatrix}^{-1}\mathbf{P}_{y_1} + \mathbf{R} \\ &= \frac{\alpha_{y^{(1)}}}{\alpha_{y^{(0)}}}\mathbf{P}_{y_1}^{-1}\begin{pmatrix} \mathbf{E}_{y^{(1)}}\mathbf{E}_{y^{(0)}}^{-1} & 0 \\ 0 & \mathbf{A}_{y^{(1)}}\mathbf{A}_{y^{(0)}}^{-1} \end{pmatrix}\mathbf{P}_{y_1} + \mathbf{R} \\ &\approx \frac{\alpha_{y^{(1)}}}{\alpha_{y^{(0)}}}\mathbf{P}_{y_1}^{-1}\begin{pmatrix} \mathbf{E}_{y^{(1)}}\mathbf{E}_{y^{(0)}}^{-1} & 0 \\ 0 & \mathbf{A}_{y^{(1)}}\mathbf{A}_{y^{(0)}}^{-1} \end{pmatrix}\mathbf{P}_{y_1}, \end{aligned}$$

where $\mathbf{R} = \beta_Y\alpha_{y^{(1)}}\mathbf{P}_{y_1}^{-1}\widehat{\mathbf{A}}_{y^{(1)}}\mathbf{A}_{\delta'}\mathbf{P}_{y_1} + \mathbf{R}_{y^{(1)}}\mathbf{P}_{z_1}^{-1}(\mathbf{A}_1^{-1} + \mathbf{A}_{\delta'})\mathbf{P}_{y_1}$ such that $\|\mathbf{R}\| \approx O(\beta_Y^{-1})$.

By Lemma 2.2, we have $\mathbf{A}_{y^{(1)}}\mathbf{A}_{y^{(0)}}^{-1} = \mathbf{I}^{w\times w}$. As a consequence, $\frac{\alpha_{y^{(1)}}}{\alpha_{y^{(0)}}} \in \mathbb{K}$ is an approximate eigenvalue of the function branch part of multiplicity at least w. Likewise, $\frac{\alpha'_{y^{(1)}}}{\alpha'_{y^{(0)}}} \in \mathbb{K}$ is an approximate eigenvalue of the dummy branch of

multiplicity at least w. Again by Lemma 5.1, $\frac{\alpha_{y(1)}}{\alpha_{y(0)}} = \frac{\alpha'_{y(1)}}{\alpha'_{y(0)}}$, and therefore $\frac{\alpha_{y(1)}}{\alpha_{y(0)}}$ is the approximate eigenvalue of $\mathbf{Y}_1\mathbf{Y}_0^{-1}$ of multiplicity at least $2w$.

Thus, we can find all roots of the characteristic polynomial of $\mathbf{W}_1\mathbf{W}_0^{-1}$ over $\mathbb{K}$ and consider at least $2w$ approximately equal roots as the approximate value of $\frac{\alpha_{y(1)}}{\alpha_{y(0)}}$.

Remark 5.2. We observe that for two inputs $\mathbf{x}, \mathbf{x}' \in \{0,1\}^l$ that differ only in $x_j = 1$ and $x'_j = 0$, if the branching program evaluates to zero for them, namely $\delta_x = \alpha_x\beta \cdot \mathbf{st}^T + o(\beta)$ and $\delta_{x'} = \alpha_{x'}\beta \cdot \mathbf{st}^T + o(\beta)$. As a consequence, if we take the setting of parameters with $\|\delta_x\|, \|\delta_{x'}\| < q$ according to [2], then $\frac{\alpha_{j,1}}{\alpha_{j,0}} \approx \frac{\delta_x}{\delta_{x'}}$. The advantage of this simple attack method is that it has not related to the input-partition of the branching program. However, it is not difficult to avoid this attack by setting $\|\beta\| \geq q$. Note that its updated version [3] has set the parameters such that $\|\delta_x\|, \|\delta_{x'}\| > q$.

5.3 Annihilation attack

Chen, Gentry and Halevi [11] have extended the annihilation attack introduced by Miles, Sahai and Zhandry [28] to break the GGH13-based branching program obfuscators with the padded random diagonal entries by using the ratios of the bundling scalars. However, it is not clear whether the CGH attack can be extended to attack the BP obfuscators over GGH13 without ideals [2]. Here we further generalize the CGH attack to break this candidate IO over GGH13 without ideals by applying the approximate ratios of the bundling scalars.

To simplify our attack description, we use the same running example used by Chen, Gentry and Halevi [11].

Example 5.3 (Example 3.1 [11]). The two programs B, B' have the identity matrix for both 0 and 1 in all the steps except for the two steps u, w that are a permutation matrix P and its inverse P^{-1} for B'. Here we require the steps u, v, w belong to the interval Y such that $u < v < w$ and the input bit j_2 does not control any steps before u or after w. The programs B, B' that compute the constant-zero function concretely define as follows:

B=	0:	$\mathbf{I}$	$\cdots$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\cdots$	$\mathbf{I}$
	1:	$\mathbf{I}$	$\cdots$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\cdots$	$\mathbf{I}$
B'=	0:	$\mathbf{I}$	$\cdots$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\cdots$	$\mathbf{I}$
	1:	$\mathbf{I}$	$\cdots$	$\mathbf{I}$	$\mathbf{P}$	$\mathbf{I}$	$\mathbf{P}^{-1}$	$\mathbf{I}$	$\cdots$	$\mathbf{I}$
Steps	0:		X		u	v	w		Z	
Input bits	1:	*	$\cdots$	*	j_1	j_2	j_1	*	$\cdots$	*

Unlike [11], in the above subsection we can only compute the approximate ratios of α_1/α_0 and α'_1/α'_0, not their exact ratios. Since these ratios are approximate, consequently we cannot compute four scalars $v_0, v_1, \zeta_{00}, \zeta_{11} \in R$ as that in [11]. However, we here are working on $\mathbb{K}$, not mod $\langle g \rangle$ and hence we can take

$$v_0 = 1, \quad v_1 \approx \alpha'_1/\alpha'_0 \quad \text{and} \quad \zeta_{00} = 1, \quad \zeta_{11} \approx \alpha_1\alpha'_1/\alpha_0\alpha'_0$$

We let $f_{\mu\nu}^{(i,j)} = x^{(i)}\mu\nu z^{(j)}$ be an input for a zero of the function, where $x^{(i)}$ is the bits controlled in the step interval X, $\mu\nu$ the two distinguished bits controlled in the step interval Y, and $z^{(j)}$ the bits controlled in the step interval Z. We denote by $\mathrm{Eval}(f_{\mu\nu}^{(i,j)})$ the value returned by honest evaluating the obfuscated BP on the input $f_{\mu\nu}^{(i,j)}$:

$$\begin{aligned}\mathrm{Eval}(f_{\mu\nu}^{(i,j)}) &= \overline{\mathbf{A}}_0 \cdot \prod_{k=1}^{\kappa} \overline{\mathbf{A}}_{k,x_{\mathrm{inp}(k)}} \cdot \overline{\mathbf{A}}_{\kappa+1} - \overline{\mathbf{A}}'_0 \cdot \prod_{k=1}^{\kappa} \overline{\mathbf{A}}'_{k,x_{\mathrm{inp}(k)}} \cdot \overline{\mathbf{A}}'_{\kappa+1} \\ &= (\beta_0 \widetilde{\mathbf{A}}_0 + \mathbf{R}_0) \cdot \prod_{k=1}^{\kappa} (\beta_k \widetilde{\mathbf{A}}_{k,x_{\mathrm{inp}(k)}} + \mathbf{R}_{k,x_{\mathrm{inp}(k)}}) \cdot (\beta_{\kappa+1} \widetilde{\mathbf{A}}_{\kappa+1} + \mathbf{R}_{\kappa+1}) \\ &\quad - (\beta_0 \widetilde{\mathbf{A}}'_0 + \mathbf{R}'_0) \cdot \prod_{k=1}^{\kappa} (\beta_k \widetilde{\mathbf{A}}'_{k,x_{\mathrm{inp}(k)}} + \mathbf{R}'_{k,x_{\mathrm{inp}(k)}}) \cdot (\beta_{\kappa+1} \widetilde{\mathbf{A}}'_{\kappa+1} + \mathbf{R}'_{\kappa+1})\end{aligned}$$

To perform our attack, we select many different inputs $f_{\mu\nu}^{(i,j)}$ that are all zeros of the function, and for each i, j we set

$$\begin{aligned}A[i,j] = &\ \mathrm{Eval}(f_{11}^{(i,j)}) \cdot \zeta_{00} \cdot v_1 v_0 - \mathrm{Eval}(f_{10}^{(i,j)}) \cdot \zeta_{00} \cdot v_1 v_1 \\ &- \mathrm{Eval}(f_{01}^{(i,j)}) \cdot \zeta_{11} \cdot v_0 v_0 - \mathrm{Eval}(f_{00}^{(i,j)}) \cdot \zeta_{11} \cdot v_0 v_1,\end{aligned}$$

where all the computations are operated in $\mathbb{K}$. Choosing enough inputs $f_{\mu\nu}^{(i,j)}$, we can obtain a matrix $\mathbf{A}$.

In the following, we first analyze the rank of the submatrix corresponding to the interval Y in the matrix $\mathbf{A}$. Then we show that $\mathbf{A}$ has a non-full rank matrix decomposition for the program B, whereas for the program B', there is no such decomposition with high probability. Finally, we describe a distinguishing attack between the programs B and B'.

5.4 Analysis

5.4.1 The Matrix $\mathbf{D}_{\boldsymbol{Y}}$

Assume that the step interval Y only consists of the steps u, v, w, namely $|Y| = 3$, and $\mu\nu \in \{0,1\}^2$ are any two input bits corresponding to Y. For simplicity, let $\overline{\beta} = \max\limits_{0 \le k \le \kappa+1} \{\beta_k\}$ such that $\|\overline{\beta}\| = \max\limits_{0 \le k \le \kappa+1} \{\|\beta_k\|\}$. We write $\beta_{uv} = \beta_u \beta_v$, and similarly for β_{uw}, β_{vw}.

Then the matrix in the function branch of Y has the form

$$\begin{aligned}
\mathbf{A}_Y^{\mu\nu} &= \prod\nolimits_{k\in Y}(\beta_k \widetilde{\mathbf{A}}_{k,x_{\mathrm{inp}(k)}} + \mathbf{R}_{k,x_{\mathrm{inp}(k)}}) \\
&= (\beta_u \widetilde{\mathbf{A}}_{u,\mu} + \mathbf{R}_{u,\mu})(\beta_v \widetilde{\mathbf{A}}_{v,\nu} + \mathbf{R}_{v,\nu})(\beta_w \widetilde{\mathbf{A}}_{w,\mu} + \mathbf{R}_{w,\mu}) \\
&= \alpha_\mu \alpha'_\nu \cdot \mathbf{P}_{u-1}^{-1} \cdot \big(\beta_u \widehat{\mathbf{A}}_{u,\mu} + \underbrace{\frac{1}{\epsilon_{u,\mu}} \mathbf{P}_{u-1} \mathbf{R}_{u,\mu} \mathbf{P}_u^{-1}}_{:=\widehat{\mathbf{R}}_{u,\mu}}\big) \\
&\qquad \big(\beta_v \widehat{\mathbf{A}}_{v,\nu} + \underbrace{\frac{1}{\epsilon_{v,\nu}} \mathbf{P}_u \mathbf{R}_{v,\nu} \mathbf{P}_v^{-1}}_{:=\widehat{\mathbf{R}}_{v,\nu}}\big)\big(\beta_w \widehat{\mathbf{A}}_{w,\mu} + \underbrace{\frac{1}{\epsilon_{w,\mu}} \mathbf{P}_v \mathbf{R}_{w,\mu} \mathbf{P}_w^{-1}}_{:=\widehat{\mathbf{R}}_{w,\mu}}\big) \cdot \mathbf{P}_w \\
&= \alpha_\mu \alpha'_\nu \cdot \mathbf{P}_{u-1}^{-1} \cdot \Bigg(\underbrace{\beta_Y \widehat{\mathbf{A}}_{u,\mu} \widehat{\mathbf{A}}_{v,\nu} \widehat{\mathbf{A}}_{w,\mu}}_{:=\mathbf{C}_Y^{\mu\nu}} \\
&\qquad + \big(\underbrace{\beta_{uw} \widehat{\mathbf{A}}_{u,\mu} \widehat{\mathbf{R}}_{v,\nu} \widehat{\mathbf{A}}_{w,\mu} + \beta_{uv} \widehat{\mathbf{A}}_{u,\mu} \widehat{\mathbf{A}}_{v,\nu} \widehat{\mathbf{R}}_{w,\mu} + \beta_{vw} \widehat{\mathbf{R}}_{u,\mu} \widehat{\mathbf{A}}_{v,\nu} \widehat{\mathbf{A}}_{w,\mu}}_{:=\mathbf{D}_Y^{\mu\nu}} \big) \\
&\qquad + O(\overline{\beta}^{|Y|-2}) \mathbf{E}_Y^{\mu\nu} \Bigg) \cdot \mathbf{P}_w \\
&= \alpha_\mu \alpha'_\nu \cdot \mathbf{P}_{u-1}^{-1} \cdot \Big(\mathbf{C}_Y^{\mu\nu} + \mathbf{D}_Y^{\mu\nu} + O(\overline{\beta}) \mathbf{E}_Y^{\mu\nu} \Big) \cdot \mathbf{P}_w,
\end{aligned}$$

where all the computations above are operated in $\mathbb{K}$. Notice that in the above $\|\mathbf{E}_Y^{\mu\nu}\| = \lambda^{O(1)}$, and $\alpha_\mu = \epsilon_{u,\mu}\epsilon_{w,\mu}, \alpha'_\nu = \epsilon_{v,\nu}$.

By $\mathbf{D}_Y^{\mu\nu}$ we define

$$\begin{aligned}
\mathbf{D}_Y &= \mathbf{D}_Y^{11} - \mathbf{D}_Y^{10} - \mathbf{D}_Y^{01} + \mathbf{D}_Y^{00} \\
&= \quad \big(\beta_{uw} \widehat{\mathbf{A}}_{u,1} \widehat{\mathbf{R}}_{v,1} \widehat{\mathbf{A}}_{w,1} + \beta_{uv} \widehat{\mathbf{A}}_{u,1} \widehat{\mathbf{A}}_{v,1} \widehat{\mathbf{R}}_{w,1} + \beta_{vw} \widehat{\mathbf{R}}_{u,1} \widehat{\mathbf{A}}_{v,1} \widehat{\mathbf{A}}_{w,1}\big) \\
&\quad - \big(\beta_{uw} \widehat{\mathbf{A}}_{u,1} \widehat{\mathbf{R}}_{v,0} \widehat{\mathbf{A}}_{w,1} + \beta_{uv} \widehat{\mathbf{A}}_{u,1} \widehat{\mathbf{A}}_{v,0} \widehat{\mathbf{R}}_{w,1} + \beta_{vw} \widehat{\mathbf{R}}_{u,1} \widehat{\mathbf{A}}_{v,0} \widehat{\mathbf{A}}_{w,1}\big) \\
&\quad - \big(\beta_{uw} \widehat{\mathbf{A}}_{u,0} \widehat{\mathbf{R}}_{v,1} \widehat{\mathbf{A}}_{w,0} + \beta_{uv} \widehat{\mathbf{A}}_{u,0} \widehat{\mathbf{A}}_{v,1} \widehat{\mathbf{R}}_{w,0} + \beta_{vw} \widehat{\mathbf{R}}_{u,0} \widehat{\mathbf{A}}_{v,1} \widehat{\mathbf{A}}_{w,0}\big) \\
&\quad + \big(\beta_{uw} \widehat{\mathbf{A}}_{u,0} \widehat{\mathbf{R}}_{v,0} \widehat{\mathbf{A}}_{w,0} + \beta_{uv} \widehat{\mathbf{A}}_{u,0} \widehat{\mathbf{A}}_{v,0} \widehat{\mathbf{R}}_{w,0} + \beta_{vw} \widehat{\mathbf{R}}_{u,0} \widehat{\mathbf{A}}_{v,0} \widehat{\mathbf{A}}_{w,0}\big).
\end{aligned} \tag{2}$$

Now it is completely analogous to the method in [11] to show $\mathbf{D}_Y \in \begin{pmatrix} * & * \\ * & 0^{w\times w} \end{pmatrix}$ when evaluating B, but not with high probability when evaluating B'.

Similarly, we can define the matrix $\mathbf{D}'_Y$ in the dummy branch for the step interval Y, and use the same method to prove $\mathbf{D}'_Y \in \begin{pmatrix} * & * \\ * & 0^{w\times w} \end{pmatrix}$ regardless of whether the branching program is B or B'.

5.4.2 The Matrix A

To analyze $\mathbf{A}$, we let $X=\{x_1,x_2,\cdots,x_x\}, Y=\{u,v,w\}, Z=\{z_1,z_2,\cdots,z_z\}$. We denote by $\alpha_{x^{(i)}}$ (resp. $\alpha_{z^{(j)}}$) the product of the bundling scalars of the function branch corresponding to X (resp. Z), and similarly for $\alpha'_{x^{(i)}}$, $\alpha'_{z^{(j)}}$ corresponding to the dummy branch. Moreover by Lemma 5.1, we have $\alpha_{x^{(i)}}\alpha_{z^{(j)}} = \alpha'_{x^{(i)}}\alpha'_{z^{(j)}}$ and denote this product by $\alpha_{(i,j)}$. We also write $\beta_{X_k}=\beta_X/\beta_k$ and $\beta_{Z_k}=\beta_Z/\beta_k$.

Similar to the simplification of $\mathbf{A}_Y^{\mu\nu}$ in the function branch corresponding to Y, it is easy to simplify all the matrices associated to the intervals X,Y,Z as follows:

$$\begin{cases} \mathbf{A}_X^i &= \alpha_{x^{(i)}}\cdot\mathbf{P}_0^{-1}\big(\mathbf{C}_X^i+\mathbf{D}_X^i+O(\overline{\beta}^{|X|-2})\mathbf{E}_X^i\big)\mathbf{P}_{u-1},\\ \mathbf{A}_Y^{\mu\nu} &= \alpha_\mu\alpha'_\nu\cdot\mathbf{P}_{u-1}^{-1}\Big(\mathbf{C}_Y^{\mu\nu}+\mathbf{D}_Y^{\mu\nu}+O(\overline{\beta}^{|Y|-2})\mathbf{E}_Y^{\mu\nu}\Big)\mathbf{P}_w\\ \mathbf{A}_Z^j &= \alpha_{z^{(j)}}\cdot\mathbf{P}_w^{-1}\big(\mathbf{C}_Z^j+\mathbf{D}_Z^j+O(\overline{\beta}^{|Z|-2})\mathbf{E}_Z^j\big)\mathbf{P}_\kappa,\\ \mathbf{A'}_X^i &= \alpha'_{x^{(i)}}\cdot\mathbf{P'}_0^{-1}\big(\mathbf{C'}_X^i+\mathbf{D'}_X^i+O(\overline{\beta}^{|X|-2})\mathbf{E'}_X^i\big)\mathbf{P'}_{u-1},\\ \mathbf{A'}_Y^{\mu\nu} &= \alpha_\mu\alpha'_\nu\cdot\mathbf{P'}_{u-1}^{-1}\Big(\mathbf{C'}_Y^{\mu\nu}+\mathbf{D'}_Y^{\mu\nu}+O(\overline{\beta}^{|Y|-2})\mathbf{E'}_Y^{\mu\nu}\Big)\mathbf{P'}_w\\ \mathbf{A'}_Z^j &= \alpha'_{z^{(j)}}\cdot\mathbf{P'}_w^{-1}\big(\mathbf{C'}_Z^j+\mathbf{D'}_Z^j+O(\overline{\beta}^{|Z|-2})\mathbf{E'}_Z^j\big)\mathbf{P'}_\kappa. \end{cases} \tag{3}$$

In Equ. (3), except for the unspecified small noise matrices $\mathbf{E}_X^i,\mathbf{E'}_X^i,\mathbf{E}_Z^j,\mathbf{E'}_Z^j$, we also use the following notations

$$\mathbf{C}_X^i=\beta_X\cdot\prod_{k\in X}\widehat{\mathbf{A}}_{k,u_{\mathrm{inp}(k)}},\quad \mathbf{C'}_X^i=\beta_X\cdot\prod_{k\in X}\widehat{\mathbf{A}}'_{k,u_{\mathrm{inp}(k)}},$$

$$\mathbf{C}_Z^j=\beta_Z\cdot\prod_{k\in Z}\widehat{\mathbf{A}}_{k,u_{\mathrm{inp}(k)}},\quad \mathbf{C'}_Z^j=\beta_Z\cdot\prod_{k\in Z}\widehat{\mathbf{A}}'_{k,u_{\mathrm{inp}(k)}},$$

$$\mathbf{D}_X^i=\sum_{k\in X}\beta_{X_k}\widehat{\mathbf{A}}_{x_1,u_{\mathrm{inp}(x_1)}}\cdots\widehat{\mathbf{A}}_{k-1,u_{\mathrm{inp}(k-1)}}\widehat{\mathbf{R}}_{k,u_{\mathrm{inp}(k)}}\widehat{\mathbf{A}}_{k+1,u_{\mathrm{inp}(k+1)}}\cdots\widehat{\mathbf{A}}_{x_x,u_{\mathrm{inp}(x_x)}},$$

$$\mathbf{D'}_X^i=\sum_{k\in X}\beta_{X_k}\widehat{\mathbf{A}}'_{x_1,u_{\mathrm{inp}(x_1)}}\cdots\widehat{\mathbf{A}}'_{k-1,u_{\mathrm{inp}(k-1)}}\widehat{\mathbf{R}}'_{k,u_{\mathrm{inp}(k)}}\widehat{\mathbf{A}}'_{k+1,u_{\mathrm{inp}(k+1)}}\cdots\widehat{\mathbf{A}}'_{x_x,u_{\mathrm{inp}(x_x)}},$$

$$\mathbf{D}_Z^j=\sum_{k\in Z}\beta_{Z_k}\widehat{\mathbf{A}}_{z_1,u_{\mathrm{inp}(z_1)}}\cdots\widehat{\mathbf{A}}_{k-1,u_{\mathrm{inp}(k-1)}}\widehat{\mathbf{R}}_{k,u_{\mathrm{inp}(k)}}\widehat{\mathbf{A}}_{k+1,u_{\mathrm{inp}(k+1)}}\cdots\widehat{\mathbf{A}}_{z_z,u_{\mathrm{inp}(z_z)}},$$

$$\mathbf{D'}_Z^j=\sum_{k\in Z}\beta_{Z_k}\widehat{\mathbf{A}}'_{z_1,u_{\mathrm{inp}(z_1)}}\cdots\widehat{\mathbf{A}}'_{k-1,u_{\mathrm{inp}(k-1)}}\widehat{\mathbf{R}}'_{k,u_{\mathrm{inp}(k)}}\widehat{\mathbf{A}}'_{k+1,u_{\mathrm{inp}(k+1)}}\cdots\widehat{\mathbf{A}}'_{z_z,u_{\mathrm{inp}(z_z)}},$$

where

$$\widehat{\mathbf{R}}_{k,u_{\mathrm{inp}(k)}}=\frac{1}{\epsilon_{k,u_{\mathrm{inp}(k)}}}\mathbf{P}_{k-1}\mathbf{R}_{k,u_{\mathrm{inp}(k)}}\mathbf{P}_k^{-1},$$

$$\widehat{\mathbf{R}}'_{k,u_{\mathrm{inp}(k)}}=\frac{1}{\epsilon_{k,u_{\mathrm{inp}(k)}}}\mathbf{P'}_{k-1}\mathbf{R'}_{k,u_{\mathrm{inp}(k)}}\mathbf{P'}_k^{-1}.$$

Thus, we now can simplify $\mathrm{Eval}(f_{\mu\nu}^{(i,j)})$ as follows:

$$
\begin{aligned}
&\mathrm{Eval}(f^{(i,j)}_{\mu\nu}) \\
&= \alpha_0\alpha_{(i,j)}\alpha_\mu\alpha'_\nu \cdot \Bigg(\big(\underbrace{\beta_0\widehat{\mathbf{A}}_0}_{:=\mathbf{C}_0} + \widehat{\mathbf{R}}_0 \big)\big(\mathbf{C}^i_X + \mathbf{D}^i_X + O(\overline{\beta}^{|X|-2})\mathbf{E}^i_X\big) \\
&\quad \big(\mathbf{C}^{\mu\nu}_Y + \mathbf{D}^{\mu\nu}_Y + O(\overline{\beta}^{|Y|-2})\mathbf{E}^{\mu\nu}_Y\big)\big(\mathbf{C}^j_Z + \mathbf{D}^j_Z + O(\overline{\beta}^{|Z|-2})\mathbf{E}^j_Z\big)\big(\underbrace{\beta_{\kappa+1}\widehat{\mathbf{A}}_{\kappa+1}}_{:=\mathbf{C}_{\kappa+1}} + \widehat{\mathbf{R}}_{\kappa+1}\big) \\
&- \big(\underbrace{\beta_0\widehat{\mathbf{A}}'_0}_{:=\mathbf{C}'_0} + \widehat{\mathbf{R}}'_0\big)\big(\mathbf{C}'^i_X + \mathbf{D}'^i_X + O(\overline{\beta}^{|X|-2})\mathbf{E}'^i_X\big)\big(\mathbf{C}'^{\mu\nu}_Y + \mathbf{D}'^{\mu\nu}_Y + O(\overline{\beta}^{|Y|-2})\mathbf{E}'^{\mu\nu}_Y\big) \\
&\quad \big(\mathbf{C}'^j_Z + \mathbf{D}'^j_Z + O(\overline{\beta}^{|Z|-2})\mathbf{E}'^j_Z\big)\big(\underbrace{\beta_{\kappa+1}\widehat{\mathbf{A}}'_{\kappa+1}}_{:=\mathbf{C}'_{\kappa+1}} + \widehat{\mathbf{R}}'_{\kappa+1}\big)\Bigg) \\
&= \alpha_0\alpha_{(i,j)}\alpha_\mu\alpha'_\nu \cdot \Big(\mathbf{C}_0\big(\mathbf{C}^i_X + \mathbf{D}^i_X\big)\big(\mathbf{C}^{\mu\nu}_Y + \mathbf{D}^{\mu\nu}_Y\big)\big(\mathbf{C}^j_Z + \mathbf{D}^j_Z\big)\mathbf{C}_{\kappa+1} \\
&\qquad + \widehat{\mathbf{R}}_0\mathbf{C}^i_X\mathbf{C}^{\mu\nu}_Y\mathbf{C}^j_Z\mathbf{C}_{\kappa+1} + \mathbf{C}_0\mathbf{C}^i_X\mathbf{C}^{\mu\nu}_Y\mathbf{C}^j_Z\widehat{\mathbf{R}}_{\kappa+1} \\
&\quad - \mathbf{C}'_0\big(\mathbf{C}'^i_X + \mathbf{D}'^i_X\big)\big(\mathbf{C}'^{\mu\nu}_Y + \mathbf{D}'^{\mu\nu}_Y\big)\big(\mathbf{C}'^j_Z + \mathbf{D}'^j_Z\big)\mathbf{C}'_{\kappa+1} \\
&\qquad - \widehat{\mathbf{R}}'_0\mathbf{C}'^i_X\mathbf{C}'^{\mu\nu}_Y\mathbf{C}'^j_Z\mathbf{C}'_{\kappa+1} - \mathbf{C}'_0\mathbf{C}'^i_X\mathbf{C}'^{\mu\nu}_Y\mathbf{C}'^j_Z\widehat{\mathbf{R}}'_{\kappa+1} + O(\overline{\beta}^{\kappa})\Big) \\
&= \alpha_0\alpha_{(i,j)}\alpha_\mu\alpha'_\nu \cdot \Big(\mathbf{C}_0\big(\mathbf{C}^i_X\mathbf{C}^{\mu\nu}_Y\mathbf{D}^j_Z + \mathbf{C}^i_X\mathbf{D}^{\mu\nu}_Y\mathbf{C}^j_Z + \mathbf{D}^i_X\mathbf{C}^{\mu\nu}_Y\mathbf{C}^j_Z\big)\mathbf{C}_{\kappa+1} \\
&\qquad + \widehat{\mathbf{R}}_0\mathbf{C}^i_X\mathbf{C}^{\mu\nu}_Y\mathbf{C}^j_Z\mathbf{C}_{\kappa+1} + \mathbf{C}_0\mathbf{C}^i_X\mathbf{C}^{\mu\nu}_Y\mathbf{C}^j_Z\widehat{\mathbf{R}}_{\kappa+1} \\
&\quad - \mathbf{C}'_0\big(\mathbf{C}'^i_X\mathbf{C}'^{\mu\nu}_Y\mathbf{D}'^j_Z + \mathbf{C}'^i_X\mathbf{D}'^{\mu\nu}_Y\mathbf{C}'^j_Z + \mathbf{D}'^i_X\mathbf{C}'^{\mu\nu}_Y\mathbf{C}'^j_Z\big)\mathbf{C}'_{\kappa+1} \\
&\qquad - \widehat{\mathbf{R}}'_0\mathbf{C}'^i_X\mathbf{C}'^{\mu\nu}_Y\mathbf{C}'^j_Z\mathbf{C}'_{\kappa+1} - \mathbf{C}'_0\mathbf{C}'^i_X\mathbf{C}'^{\mu\nu}_Y\mathbf{C}'^j_Z\widehat{\mathbf{R}}'_{\kappa+1} + O(\overline{\beta}^{\kappa})\Big),
\end{aligned}
$$

where

$$
\begin{aligned}
\widehat{\mathbf{R}}_0 &= \frac{1}{\epsilon_0}\mathbf{R}_0\mathbf{P}_0^{-1}, \qquad & \widehat{\mathbf{R}'}_0 &= \frac{1}{\epsilon'_0}\mathbf{R}'_0\mathbf{P}'^{-1}_0, \\
\widehat{\mathbf{R}}_{\kappa+1} &= \frac{1}{\epsilon_{\kappa+1}}\mathbf{P}_{\kappa+1}\mathbf{R}_{\kappa+1}, & \widehat{\mathbf{R}'}_{\kappa+1} &= \frac{1}{\epsilon'_{\kappa+1}}\mathbf{P}'_{\kappa+1}\mathbf{R}'_{\kappa+1}.
\end{aligned}
$$

To further simplify $A[i,j]$, we define

$$
\begin{aligned}
\mathbf{C}_Y &= \mathbf{C}^{11}_Y - \mathbf{C}^{10}_Y - \mathbf{C}^{01}_Y + \mathbf{C}^{00}_Y, \quad & \mathbf{C}'_Y &= \mathbf{C}'^{11}_Y - \mathbf{C}'^{10}_Y - \mathbf{C}'^{01}_Y + \mathbf{C}'^{00}_Y \\
\mathbf{x}_i &= \mathbf{C}_0\mathbf{C}^i_X, \quad & \mathbf{x}'_i &= \mathbf{C}'_0\mathbf{C}'^i_X, \quad & \mathbf{z}_j &= \mathbf{C}^j_Z\mathbf{C}_{\kappa+1}, \quad & \mathbf{z}'_j &= \mathbf{C}'^j_Z\mathbf{C}'_{\kappa+1} \\
\mathbf{e}_i &= \mathbf{C}_0\mathbf{D}^i_X, \quad & \mathbf{e}'_i &= \mathbf{C}'_0\mathbf{D}'^i_X, \quad & \mathbf{f}_j &= \mathbf{D}^j_Z\mathbf{C}_{\kappa+1}, \quad & \mathbf{f}'_j &= \mathbf{D}'^j_Z\mathbf{C}'_{\kappa+1} \\
\mathbf{r}_i &= \widehat{\mathbf{R}}_0\mathbf{C}^i_X, \quad & \mathbf{r}'_i &= \widehat{\mathbf{R}}'_0\mathbf{C}'^i_X, \quad & \mathbf{w}_j &= \mathbf{C}^j_Z\widehat{\mathbf{R}}_{\kappa+1}, \quad & \mathbf{w}'_j &= \mathbf{C}'^j_Z\widehat{\mathbf{R}}'_{\kappa+1}
\end{aligned}
$$

By the definition of the bundling scalars and their approximate ratios that solve in the above subsection, it is easy to verify that

$$\alpha_1\alpha_1' \cdot \zeta_{00} \cdot v_1v_0 \approx \alpha_1\alpha_0' \cdot \zeta_{00} \cdot v_1v_1 \approx \alpha_0\alpha_1' \cdot \zeta_{11} \cdot v_0v_0 \approx \alpha_0\alpha_0' \cdot \zeta_{11} \cdot v_0v_1,$$

where the approximate accuracy is $O(\overline{\beta}^{-1})$.

As a consequence, we can incorporate these approximate scalars into the matrices corresponding to $x^{(i)}$ and $z^{(j)}$ respectively and can rewrite $A[i,j]$ as follows:

$$\begin{aligned} A[i,j] = \quad & \big(\underbrace{\mathbf{x}_i\mathbf{C}_Y\mathbf{z}_j + \mathbf{x}_i\mathbf{D}_Y\mathbf{z}_j + \mathbf{e}_i\mathbf{C}_Y\mathbf{z}_j + \mathbf{r}_i\mathbf{C}_Y\mathbf{z}_j + \mathbf{x}_i\mathbf{C}_Y\mathbf{w}_j}_{:=F[i,j]}\big) \\ & - \big(\underbrace{\mathbf{x}_i'\mathbf{C}_Y'\mathbf{z}_j' + \mathbf{x}_i'\mathbf{D}_Y'\mathbf{z}_j' + \mathbf{e}_i'\mathbf{C}_Y'\mathbf{z}_j' + \mathbf{x}_i'\mathbf{C}_Y'\mathbf{z}_j' + \mathbf{x}_i'\mathbf{C}_Y'\mathbf{w}_j'}_{:=F'[i,j]}\big) + O(\overline{\beta}^{\kappa}), \end{aligned}$$

In the following we first analyze the matrix $\mathbf{F}$ generated by the term $F[i,j]$ from the function branch with $i,j \in [\xi]$, where $\xi \geq 2m+1$.

According to the construction structure of the obfuscated BP, for program B we have the vectors $\mathbf{x}_i, \mathbf{x}_i', \mathbf{e}_i, \mathbf{e}_i' = \big(0^m\ \$^m\ \$w\big)$, $\mathbf{z}_j, \mathbf{z}_j', \mathbf{f}_j, \mathbf{f}_j' = \big(\$^m\ 0^m\ \$w\big)^T$, and the matrices

$$\mathbf{C}_Y, \mathbf{C}_Y' \in \begin{pmatrix} \$^{m\times m} & 0^{m\times m} & 0^{m\times w} \\ 0^{m\times m} & \$^{m\times m} & 0^{m\times w} \\ 0^{m\times m} & 0^{m\times m} & 0^{w\times w} \end{pmatrix}, \quad \mathbf{D}_Y, \mathbf{D}_Y' \in \begin{pmatrix} \$^{m\times m} & \$^{m\times m} & \$^{m\times w} \\ \$^{m\times m} & \$^{m\times m} & \$^{m\times w} \\ \$^{m\times m} & \$^{m\times m} & 0^{w\times w} \end{pmatrix}.$$

Moreover, for the program B' everything else is the same except that $\mathbf{D}_Y$ is arbitrary by the analysis of $\mathbf{D}_Y$ in the previous subsection.

Thus for B we can write $\mathbf{F}$ by the block form and simplify it to determine its rank as follows:

$$\begin{aligned} \mathbf{F} &= \mathbf{X}\mathbf{C}_Y\mathbf{Z} + \mathbf{X}\mathbf{D}_Y\mathbf{Z} + \mathbf{E}\mathbf{C}_Y\mathbf{Z} + \mathbf{R}\mathbf{C}_Y\mathbf{Z} + \mathbf{X}\mathbf{C}_Y\mathbf{W} \\ &= \big(0\ \mathbf{X}_2\ \mathbf{X}_3\big) \begin{pmatrix} \mathbf{C}_{1,1} & 0 & 0 \\ 0 & \mathbf{C}_{2,2} & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Z}_1 \\ 0 \\ \mathbf{Z}_3 \end{pmatrix} + \big(0\ \mathbf{X}_2\ \mathbf{X}_3\big) \begin{pmatrix} \mathbf{D}_{1,1} & \mathbf{D}_{1,2} & \mathbf{D}_{1,3} \\ \mathbf{D}_{2,1} & \mathbf{D}_{2,2} & \mathbf{D}_{2,3} \\ \mathbf{D}_{3,1} & \mathbf{D}_{3,2} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Z}_1 \\ 0 \\ \mathbf{Z}_3 \end{pmatrix} \\ &\quad + \big(0\ \mathbf{E}_2\ \mathbf{E}_3\big) \begin{pmatrix} \mathbf{C}_{1,1} & 0 & 0 \\ 0 & \mathbf{C}_{2,2} & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Z}_1 \\ 0 \\ \mathbf{Z}_3 \end{pmatrix} + \big(\mathbf{R}_1\ \mathbf{R}_2\ \mathbf{R}_3\big) \begin{pmatrix} \mathbf{C}_{1,1} & 0 & 0 \\ 0 & \mathbf{C}_{2,2} & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Z}_1 \\ 0 \\ \mathbf{Z}_3 \end{pmatrix} \\ &\quad + \big(0\ \mathbf{X}_2\ \mathbf{X}_3\big) \begin{pmatrix} \mathbf{C}_{1,1} & 0 & 0 \\ 0 & \mathbf{C}_{2,2} & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \mathbf{W}_3 \end{pmatrix} \\ &= \big(\mathbf{X}_2\mathbf{D}_{2,1} + \mathbf{X}_3\mathbf{D}_{3,1} + \mathbf{R}_1\mathbf{C}_{1,1}\big)\mathbf{Z}_1 + \mathbf{X}_2\big(\mathbf{D}_{2,3}\mathbf{Z}_3 + \mathbf{C}_{2,2}\mathbf{W}_2\big) \end{aligned} \tag{4}$$

Since the rank of $\mathbf{Z}_1$ and $\mathbf{X}_2$ is at most m, consequently the rank of $\mathbf{F}$ is at most $2m$.

However, the rank of $\mathbf{F}$ for B' is at least $2m+1$ with high probability. Since $\mathbf{D}_{3,3}$ is a non-zero block matrix, as a result with high probability $\mathbf{F}$ can not be decomposed into the sum of two matrices with rank m.

Furthermore, the rank of $\mathbf{F}'$ for B and B' is at most $2m$. The analysis of $\mathbf{F}'$ is exactly similar to the analysis of $\mathbf{F}$ for B.

Theorem 5.4. Let $\xi = 4m+1$, $\gamma = \|\overline{\beta}^{\kappa+1}\|$ and $\delta = \|\overline{\beta}^{\kappa}\|$. Suppose there exist sufficiently many inputs $u_{\mu\nu}^{i,j}$ that are all the zero of the function. Then when m is constant, with high probability

$$\text{the program is } \begin{cases} B', & \text{if } \|\det(\mathbf{A})\| = O(\gamma^{\xi}); \\ B, & \text{if } \|\det(\mathbf{A})\| = O(\gamma^{\xi-1}\delta). \end{cases}$$

When $m = \text{poly}(\lambda)$, using heuristical assumption

$$\text{the program is } \begin{cases} B', & \text{if } \|\det(\mathbf{A})\| = O(\xi! \cdot \gamma^{\xi}); \\ B, & \text{if } \|\det(\mathbf{A})\| = O(\xi! \cdot \xi\gamma^{\xi-1}\delta). \end{cases}$$

Proof. According to the analysis of $\mathbf{A}$, for the program B we have

$$\mathbf{A} = \underbrace{\mathbf{F} - \mathbf{F}'}_{:=\mathbf{A}_1} + \underbrace{O(\overline{\beta}^{\kappa})\mathbf{E}}_{:=\mathbf{A}_\delta},$$

where $\mathbf{E}$ is a matrix whose entries are polynomials with small norm over $\mathbb{K}$.

Thus, for the program B there exists a (γ, δ)-matrix decomposition $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_\delta$. Since the rank of $\mathbf{A}_1$ is at most $4m < \xi$, consequently when m is constant we have $\|\det(\mathbf{A})\| = O(\gamma^{\xi-1}\delta)$ by Lemma 4.3.

However, for the program B' with high probability there is no such (γ, δ)-matrix decomposition with a non-full rank $\mathbf{A}_1$. Therefore when m is constant we get $\|\det(\mathbf{A})\| = O(\gamma^{\xi})$ for B' by Lemma 4.1.

When $m = \text{poly}(\lambda)$ we heuristically assume that $\|\det(\mathbf{A})\|$ is approximately equal to $O(\xi! \cdot \gamma^{\xi})$ if $\mathbf{A}$ has no (γ, δ)-matrix decomposition such that $\mathbf{A}_1$ is a non-full rank matrix. Note that this heuristic assumption is supported by our computation experiment.

For B, therefore, we have $\|\det(\mathbf{A})\| = O(\xi! \cdot \xi\gamma^{\xi-1}\delta)$ by Lemma 4.3, and for B' the result directly follows the heuristic assumption. ∎

5.5 Analysis of Recent Immunization

To prevent the annihilation attack [28], Garg et al. [22] (a merged version of [23,29]) constructed a variant of BP obfuscator whose security is proved in the weakened idealized model. However, Chen, Gentry and Halevi [11] observed that this variant can not thwart the annihilation attack if the branching program is input partitioning. This attack result is not contradictory to the security proof in [22], as their immunized variant only considers dual input branching programs that are no input partitioning.

Similarly, we can also extend our attack to this immunized variant using instantiation of GGH13 without ideals. In this case, the variant uses fully random $2m \times 2m$ matrices $\mathbf{E}_{k,b}, \mathbf{E}'_{k,b}$ instead of the diagonal ones, and takes the bookend vectors as $\widehat{\mathbf{A}}_0, \widehat{\mathbf{A}}'_0 = \left(0^{2m}, \$^w\right)$ and $\widehat{\mathbf{A}}_{\kappa+1}, \widehat{\mathbf{A}}'_{\kappa+1} = \left(\$^{2m}, \$^w\right)$.

Observe that the algorithm that solves approximate ratios of the bundling scalars still works. Moreover, the analysis of the matrix $\mathbf{D}_Y$ in Equ. (2) remains the same. For the rank of $\mathbf{F}$ in Equ. (4), we analyze $\mathbf{F}$ for the program B in Example 5.3 as follows:

$$\begin{aligned}
\mathbf{F} &= \mathbf{X}\mathbf{C}_Y\mathbf{Z} + \mathbf{X}\mathbf{D}_Y\mathbf{Z} + \mathbf{E}\mathbf{C}_Y\mathbf{Z} + \mathbf{R}\mathbf{C}_Y\mathbf{Z} + \mathbf{X}\mathbf{C}_Y\mathbf{W} \\
&= \begin{pmatrix} 0 & \mathbf{X}_2 \end{pmatrix} \begin{pmatrix} \mathbf{C}_{1,1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix} + \begin{pmatrix} 0 & \mathbf{X}_2 \end{pmatrix} \begin{pmatrix} \mathbf{D}_{1,1} & \mathbf{D}_{1,2} \\ \mathbf{D}_{2,1} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix} \\
&\quad + \begin{pmatrix} 0 & \mathbf{E}_2 \end{pmatrix} \begin{pmatrix} \mathbf{C}_{1,1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix} + \begin{pmatrix} \mathbf{R}_1 & \mathbf{R}_2 \end{pmatrix} \begin{pmatrix} \mathbf{C}_{1,1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix} \\
&\quad + \begin{pmatrix} 0 & \mathbf{X}_2 \end{pmatrix} \begin{pmatrix} \mathbf{C}_{1,1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{pmatrix} \\
&= \left(\mathbf{X}_2\mathbf{D}_{2,1} + \mathbf{R}_1\mathbf{C}_{1,1}\right)\mathbf{Z}_1,
\end{aligned} \tag{5}$$

where $\{\mathbf{C}_{i,j}, \mathbf{D}_{i,j}\}_{i,j\in[2]}$ are blocks of the matrices $\mathbf{C}_Y, \mathbf{D}_Y$ with dimensions $(2m|w) \times (2m|w)$, $\{\mathbf{X}_i, \mathbf{E}_i, \mathbf{R}_i\}_{i\in[2]}$ are blocks of the matrices $\mathbf{X}, \mathbf{E}, \mathbf{R}$ with dimensions $\xi \times (2m|w)$, and $\{\mathbf{Z}_j, \mathbf{W}_j\}_{j\in[2]}$ are blocks of the matrices $\mathbf{Z}, \mathbf{W}$ with dimensions $(2m|w) \times \xi$. It is easy to verify that the rank of $\mathbf{F}$ is at most $2m$. On the other hand, for the program B' we will add another matrix $\mathbf{X}_2\mathbf{D}_{2,2}\mathbf{Z}_2$ to $\mathbf{F}$ in Equ. (5) since with high probability $\mathbf{D}_{2,2}$ is not a "0" matrix. Therefore, we can use the same algorithm in Section 5.3 to distinguish between B and B'.

As well, we can also adapt the original variant proposed by Garg et al. [23] to a new variant using GGH13 without ideals using β_i^2 instead of g^2. It is not difficult to verify that our attack can still generalize to this new immunized variant instantiated by GGH13 without ideals if the branching program is input-partitioning.

6 Conclusions

In this paper, we show how to break a branching program obfuscator using G-GH13 without ideals by extending the CGH attack when the branching program is input partionable. Consequently, we solve an open problem in [2] presented by Albrecht, Davidson and Larraia. Our work demonstrates that the security of the obfuscator using GGH13 without ideals [2] is essentially equivalent to that of the GGH13-based obfuscator [20]. Furthermore, our work further strengthens the work in [2,3] that there is a structural weakness in 'GGH13-type' encodings beyond the presence of $\langle g \rangle$.

While the immunized construction in [22] proposed by Garg et al. can prevent input-partitioning attack, their weakened graded encoding model does not explicitly include this requirement of input non-partitioning. Therefore, it is still

an open problem how to construct a branching program obfuscator with input-partitioning or improve the weakened graded encoding model to enable this input requirement.

References

1. M. Albrecht, S. Bai, and L. Ducas. A subfield lattice attack on overstretched NTRU assumptions Cryptanalysis of some FHE and Graded Encoding Schemes. CRYPTO 2016, LNCS 9814, pp.153-178.
2. M. Albrecht, A. Davidson, E. Larraia. Notes On GGH13 Without The Presence Of Ideals. IMACC 2017, LNCS 10655, pp. 135-158.
3. M. Albrecht, A. Davidson, E. Larraia, and A. Pellet–Mary. Notes On GGH13 Without The Presence Of Ideals. http://eprint.iacr.org/2017/906.
4. D. Apon, N. Döttling, S. Garg, and P. Mukherjee. Cryptanalysis of indistinguishability obfuscations of circuits over GGH13. http://eprint.iacr.org/2016/1003.
5. Prabhanjan Vijendra Ananth, Divya Gupta, Yuval Ishai, and Amit Sahai. Optimizing obfuscation: Avoiding Barrington's theorem. ACM CCS 2014, pp. 646-658.
6. Z. Brakerskiy, C. Gentry, S. Halevi, T. Lepoint, A. Sahai, M. Tibouchi. Cryptanalysis of the Quadratic Zero-Testing of GGH. http://eprint.iacr.org/2015/845.
7. B. Barak, S. Garg, Y. T. Kalai, O. Paneth, and A. Sahai. Protecting obfuscation against algebraic attacks. EUROCRYPT 2014, LNCS 8441, pp. 221-238.
8. D. Boneh and A. Silverberg. Applications of multilinear forms to cryptography. Contemporary Mathematics, 324:71-90, 2003.
9. D. Boneh and M. Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. CRYPTO 2014, LNCS 8616, pp. 480-499.
10. J. S. Coron, C. Gentry, S. Halevi, T. Lepoint, H. K. Maji, E. Miles, M. Raykova, A. Sahai, M. Tibouchi. Zeroizing Without Low-Level Zeroes New MMAP Attacks and Their Limitations. http://eprint.iacr.org/2015/596.
11. Yilei Chen, C. Gentry, and S. Halevi. Cryptanalyses of candidate branching program obfuscators. EUROCRYPT 2017, pp. 278-307, 2017.
12. J. H. Cheon, J. Jeong, and C. Lee. An algorithm for ntru problems and cryptanalysis of the GGH multilinear map without a low-level encoding of zero. LMS Journal of Computation and Mathematics 2016, 19(A):255-266.
13. J. H. Cheon, K. Han, C. Lee, H. Ryu, and D. Stehle. Cryptanalysis of the Multilinear Map over the Integers. EUROCRYPT 2015, Part I, LNCS 9056, pp. 3-12.
14. J. H. Cheon, P. A. Fouque, C. Lee, B. Minaud, H. Ryu. Cryptanalysis of the New CLT Multilinear Map over the Integers. EUROCRYPT 2016, LNCS 9665, pp.509-536.
15. J. S. Coron, T. Lepoint, and M. Tibouchi. Practical multilinear maps over the integers. CRYPTO 2013, LNCS 8042, pp. 476-493.
16. J. S. Coron, T. Lepoint, and M. Tibouchi. New Multilinear Maps over the Integers. CRYPTO 2015, LNCS 9215, pp. 267-286.
17. J. S. Coron, M. S. Lee, T. Lepoint, and M. Tibouchi. Cryptanalysis of GGH15 Multilinear Maps. CRYPTO 2016, LNCS 9815, pp. 607-628.
18. R. Fernando, P. M. R. Rasmussen, and A. Sahai. Preventing CLT Attacks on Obfuscation with Linear Overhead. ASIACRYPT 2017, Part III, LNCS 10626, pp. 242-271.

19. S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. EUROCRYPT 2013, LNCS 7881, pp. 1-17.
20. S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. FOCS 2013, pp.40-49.
21. C. Gentry, S. Gorbunov, and S. Halevi. Graph-induced multilinear maps from lattices. TCC 2015, Part II, LNCS 9015, pp. 498-527.
22. S. Garg, E. Miles, P. Mukherjee, A. Sahai, A. Srinivasan, and M. Zhandry. Secure obfuscation in a weak multilinear map model. TCC 2016, LNCS 9986, pp. 241-268.
23. S. Garg, P. Mukherjee, and A. Srinivasan. Obfuscation without the vulnerabilities of multilinear maps. http://eprint.iacr.org/2016/390.
24. S. Halevi. Graded Encoding, Variations on a Scheme. http://eprint.iacr.org/2015/866.
25. Yupu Hu and Huiwen Jia. Cryptanalysis of GGH Map. EUROCRYPT 2016, LNCS 9665, pp. 537-565.
26. P. Kirchner and P. Fouque. Revisiting lattice attacks on overstretched NTRU parameters. EUROCRYPT 2017, LNCS 10210, pp. 3-26.
27. D. Micciancio and O. Regev. Worst-Case to Average-Case Reductions Based on Gaussian Measures, SIAM Journal on Computing, 37(1):267-302, 2007.
28. E. Miles, A. Sahai, and M. Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over GGH13. CRYPTO 2016, Part II, LNCS 9815, pp. 629-658.
29. E. Miles, A. Sahai, and M. Zhandry. Secure obfuscation in a weak multilinear map model: A simple construction secure against all known attacks. http://eprint.iacr.org/2016/588.
30. J. Zimmerman. How to obfuscate programs directly. EUROCRYPT 2015, Part II, LNCS 9057, pp. 439-467.