Computer Communications 161 (2020) 202–211

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Security analysis of indistinguishable obfuscation for internet of medical things applications

Zhengjun Jing [a], Chunsheng Gu [a], Yong Li [b], Mengshi Zhang [c], Guangquan Xu [d], Alireza Jolfaei [e], Peizhong Shi [a], Chenkai Tan [a], Xi Zheng [e,*]

[a] Jiangsu University of Technology, Changzhou, China
[b] Changchun University of Technology, Changchun, China
[c] UT Austin, TX, USA
[d] Tianjin University, Tianjin, China
[e] Macquarie University, Sydney, Australia

ARTICLE INFO

Keywords:
Cryptanalysis
Obfuscation
Multilinear maps
Approximate eigenvalue
Determinant estimation
IoMT

ABSTRACT

As a powerful cryptographic primitive, indistinguishable obfuscation has been widely used to protect data privacy on the Internet of Medical Things (IoMT) systems. Basically, the cryptographic technique protects data privacy using a function to obfuscate medical applications to perform outputs computationally indistinguishable. The state-of-the-art obfuscation technique (GGH13) utilizes a variant of the multilinear map to enhance security. However, in such schemes, it can be observed that noise lies in each element of the matrix, which means the matrix is a full rank matrix with a probability of almost 1 and results that it is unable to establish the relationship between the matrix determinant and rank. In this paper, we propose an attack to break such obfuscator. Specifically, we use approximate eigenvalues to remove the influence of noise on the matrix eigenvalues and build a specific relationship between the determinant and matrix rank. Our analysis shows the structural weakness of the state-of-the-art indistinguishable obfuscation mechanism, and we further discuss the future direction to resolve such privacy issues for IoMT applications.

1. Introduction

The applications of the Internet of Medical Things (IoMT) have made our lives convenient significantly [1–4]. However, the concern of IoMT security and privacy arises rapidly [4–9]. Security and privacy have attracted unprecedented attention and cryptography is one of the most promising approaches to resolve such concerns [10–14].

As a powerful cryptographic primitive, indistinguishable obfuscation has been widely used to keep outsourced data private and accessible in Cloud-assisted Internet of Medical Things. Generally, the indistinguishable obfuscation means that obfuscations of two programs with the same function are computationally indistinguishable. In the communication of IoMT devices, in order to prevent the implementation of signcryption on unattended devices from device capture attacks, Shi et al. [15] designed a novel signcryption algorithm with a program obfuscator, which can protect signcryption programs from key-extraction attacks by transforming the programs into unintelligible obfuscated programs. In the Mobile Crowd Sensing (MCS), Shi et al. [16] proposed an anonymous obfuscate authentication scheme, in which the authentication request algorithm is obfuscated into an unintelligible form, and the authentication key is not explicitly used. Thereby, the security of the authentication key can be ensured, even in case the mobile device is lost or encounters the device capture attack. Similarly, on the IoMT paradigm, Kavitha. D et al. [17] proposed a formal model for managing the security threats using program obfuscation technique. Their Indistinguishable Inscrutable Obfuscated Medical Data Transfer can be widely used to eradicate fraud and inside human threats across the standard-compliant devices in a health service center or clinical center. Considering cloud storage security, Zhang M. et al. [18] proposed to construct cloud-based verifiable re-encryption by incorporating the indistinguishable obfuscation. The scheme can achieve the top-tier security even if the cloud proxy is untrusted. Due to the high efficiency of decryption, it can be applied to the light-weight security devices such as nodes in IoMT.

With the fast development of quantum computing, how to design a post-quantum secure indistinguishable obfuscation for future internet of things becomes a new challenge. In 2013, Garg et al. [19] described the first candidate construction for a general-purpose obfuscation. Then, various obfuscators are constructed [19–25], which are all derived from the three candidates of graded encoding schemes (GES) (i.e. GGH13, CLT13, and GGH15) [26–30]. Unfortunately, those candidates have been proven to be vulnerable to zeroizing attacks [26,

* Corresponding author.
E-mail address: james.zheng@mq.edu.au (X. Zheng).

https://doi.org/10.1016/j.comcom.2020.07.033
Received 19 April 2020; Received in revised form 7 July 2020; Accepted 24 July 2020
Available online 30 July 2020

31–37], attacks on the overstretched NTRU [38–40], and annihilation attacks [41,42].

The GGH13 multilinear map [26] is one of the widely used multilinear maps. However, it has been shown to be subject to many zeroizing/annihilation attacks, all of which start by recovering the secret ideal $\langle \mathbf{g} \rangle$ of the multilinear map. Hence, this secret ideal seems to be a weak point of the GGH13 map, and Albrecht, Davidson, Larraia and Pellet-Mary (ADLP) [43,44] investigated the possibility of removing this secret ideal from GGH13. They proposed a variant of GGH13 without ideals, which is immune to the zeroizing attacks. However, they also showed that in two simple cases, the zeroizing attacks could still be adapted against this variant of GGH13 without ideals. They left it as an open question to determine whether attacks of Chen, Gentry and Halevi (CGH) [42], against the Garg et al.'s [19] obfuscator, could also be adapted to this variant of GGH without ideals or not. Here, the ideals used in GGH13 construction is worked in polynomial rings. For example, an instance of GGH13 has a secret short ring element $\mathbf{g} \in R$, generating a principal ideal $\langle \mathbf{g} \rangle$.

1.1. Our work

We expand the work of ADLP [43,44] by showing that GGH13-type graded encoding schemes (GES) may still have algebraic weaknesses even without the presence of algebraic ideals. In this paper, it is demonstrated that the class of attacks can be extended to show that the Garg et al. [19] obfuscator candidate, when instantiated with the ADLP GES, is also susceptible to a variant of an attack that was demonstrated on GGH13. Our contribution is to further establish the equivalence of the scheme of ADLP and the original GGH13 scheme in terms of the security guarantees. In turn, we enhance the original claims that the GGH13 GES may be weaker than first feared, by adapting a new class of attacks to also work without using the vulnerable ideal that is used in GGH13. Adapting the new attack (from [42]) is non-trivial due to the different structures of encodings — resulting in different matrix structures within the branching program. In this paper, we use various matrix techniques to extract bundling scalar ratios and ultimately distinguish obfuscated branching programs using differences in the determinants of the matrices that are exposed.

We first introduce the approximate eigenvalues of a matrix to solve the approximate ratios of the bundling scalars used in the ADLP-based BP obfuscator. In the BP obfuscator using GGH13 without ideals [43,44], the multiplicative bundling scalars play a role as an approximation factor. It means that, when solving the ratios of these bundling scalars in this variant obfuscator, noises lie in the diagonal matrix consisting of the elements returned by the zero-testing procedure. Consequently, we cannot directly apply the characteristic polynomial of the matrix to get the ratios of the bundling scalars and are unable to compute their exact ratios as well. However, we observe that these matrices are a diagonal dominant matrix with noise and their inverses are also diagonal dominant 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.

We then estimate the determinant of a matrix with noise. Since in the IO using GGH13 without ideals [43,44], each element 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 a matrix to distinguish two equivalent branch program obfuscators. However, we observe that the noise of the matrix in the ADLP-based IO is relatively "small" compared to its principal component matrix. Therefore, the determinant of the matrix is also "small" if the principal component matrix generated by the matrix decomposition is a non-full rank matrix. To this end, we build a relationship between the determinant and the rank of a matrix with noise.

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

1.2. Organization

In Section 2, we first introduce some preliminaries. In Section 3, we propose a branching program obfuscator using GGH13 without ideals. In Sections 4 and 5, we provide the matrix properties and describe cryptanalysis, respectively. 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 $\{1, 2, \ldots, n\}$. 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 jth entry of $\mathbf{a}$, and $\mathbf{A}[i, j]$ the element of the ith row and jth column of $\mathbf{A}$. We denote by $\|\mathbf{a}\|_p$ the p-norm of $\mathbf{a}$ and by $\|\mathbf{a}\|_\infty$ the ∞-norm. Similarly, for $a \in R$ we let $\|\mathbf{a}\|_p$ (resp. $\|\mathbf{a}\|_\infty$) denote the p-norm (resp. ∞-norm) of the coefficient vector corresponding to a. For $\mathbf{A} \in R^{d \times d}$, we define $\|\mathbf{A}\|_\infty = \max\{\|\mathbf{A}[i, j]\|, i, j \in [d]\}$.

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 $a_1, \ldots, a_k \in R$, the element $a = \prod_{i=1}^k a_i$ is $(n^{k-1}\eta)$-bounded. By the work in [45], 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. Definition of branching program (BP)

Let λ be the security parameter, $\kappa = \kappa(\lambda)$, $l = l(\lambda)$ and $d = d(\lambda)$. Let $\mathrm{inp} : [\kappa] \to [l]^d$ be some fixed 'input' function. All current obfuscators only consider branching programs with $d = 1$ or $d = 2$ [19,20].

Definition 1. A matrix branching program BP of length κ, input length l and arity d is defined as follows:

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

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

$$f_{\mathrm{BP}}(x) = \begin{cases} 0, & \text{if } \prod_{k=1}^{\kappa} \mathbf{A}_{x_{k,\mathrm{inp}(k)}} = \mathbf{I}; \\ 1, & \text{if } \prod_{k=1}^{\kappa} \mathbf{A}_{x_{k,\mathrm{inp}(k)}} \neq \mathbf{I}. \end{cases} \tag{2}$$

A branching program BP is input partitionable if its input bits can be partitioned into two or more independent subsets. We need the following observation in [42].

Lemma 1 ([42]). *Let BP be an input-partitioned branching program, $[\kappa] = X \parallel Y$. 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 $\prod_{k \in X} \mathbf{A}_{k,x_{inp(k)}} = \prod_{k \in X} \mathbf{A}_{k,x'_{inp(k)}}$.*

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 $\prod_{k \in Y} \mathbf{A}_{k,x_{inp(k)}} = \prod_{k \in Y} \mathbf{A}_{k,x'_{inp(k)}}$.

2.3. Background of GGH13

2.3.1. 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 $\|h\|_\infty \ll q$. 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 [41]. 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 $\langle \mathbf{g} \rangle$. Then, Chen, Gentry and Halevi [42] extend the annihilation attack in [41] to break the GGHRSW obfuscator instantiated by GGH13 [19] when a branching program has input partitioning. These works are all first to find a basis of the secret element $\langle \mathbf{g} \rangle$.

2.3.2. GGH13 without ideals [43,44]

We adaptively describe a variant of GGH13 without ideals in [43]. 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 $0 \le i \le \kappa + 1$, and sample β_i such that $\sqrt[\kappa+3]{q} < \|\beta_i\|_\infty < \sqrt[\kappa+2]{q}$.

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 $S \subset \{0, \ldots, \kappa + 1\}$, 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 $S_1, S_2 \subset \{0, \ldots, \kappa+1\}$ 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 $p_{zt} = \prod i = 0^{\kappa+1} \beta_i z_i$. 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 $\|\delta\|_\infty$ 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 [19] using instantiation of GGH13 without ideals as follows:

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

$$\mathrm{BP}' = (\kappa, l, d, \mathrm{inp}, \{\mathbf{A}'_{k,b}\}_{k\in[\kappa],b\in\{0,1\}}), \tag{3}$$

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

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

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

We also choose four "bookend" vectors as follows:

$$\begin{cases} \widehat{\mathbf{A}}_0 = \left(0^m, \quad \mathbf{e}_0, \quad \mathbf{s}\right), \\ \widehat{\mathbf{A}}'_0 = \left(0^m, \quad \mathbf{e}'_0, \quad \mathbf{s}'\right), \end{cases} \quad \begin{cases} \widehat{\mathbf{A}}_{\kappa+1} = \left(\mathbf{e}_{\kappa+1}, \quad \mathbf{0}^m, \quad \mathbf{t}\right)^T, \\ \widehat{\mathbf{A}}'_{\kappa+1} = \left(\mathbf{e}'_{\kappa+1}, \quad \mathbf{0}^m, \quad \mathbf{t}'\right)^T, \end{cases}$$

where $\mathbf{e}_0, \mathbf{e}'_0, \mathbf{e}_{\kappa+1}, \mathbf{e}'_{\kappa+1} \in R_\sigma^m$, and $\mathbf{s}, \mathbf{s}', \mathbf{t}, \mathbf{t}' \in R_\sigma^w$ 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 $\alpha_{j,b} = \prod_{\mathrm{inp}(k)=j} \epsilon_{k,b} = \prod_{\mathrm{inp}(k)=j} \epsilon'_{k,b}$, $\alpha_0 = \epsilon_0\epsilon_{\kappa+1} = \epsilon'_0\epsilon'_{\kappa+1}$.

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

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

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

Step 4: Encoding using GGH13 without ideals. For $k = 0, \ldots, \kappa+1$, 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 $\mathbf{R}_0, \mathbf{R}'_0, \mathbf{R}_{\kappa+1}, \mathbf{R}'_{\kappa+1} \in R_\sigma^s$, and matrices $\mathbf{R}_{k,b}, \mathbf{R}'_{k,b} \in R_\sigma^{s\times s}$, and set

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

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 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 $\|\mathbf{P}_k^{-1} \mod \beta_k\|_\infty \approx \|\beta_k\|_\infty$ for $k \in [\kappa]$. Namely, by a change of variable transformation, we cannot rewrite the encodings as $\overline{\mathbf{A}}_{k,b} = (\epsilon_{k,b}\mathbf{P}_{k-1}^{-1}\widehat{\mathbf{A}}_{k,b}\mathbf{P}_k + \mathbf{R}_{k,b}/\beta_k)/z_k = (\epsilon_{k,b}\mathbf{P}_{k-1}^{-1}(\widehat{\mathbf{A}}_{k,b} + \mathbf{R}'_{k,b}/\beta_k)\mathbf{P}_k)/z_k$, such that $\|\mathbf{R}'_{k,b}\|_\infty$ is 'small'.

Because in this case $\|\mathbf{R}'_{k,b}\|_\infty = \|\epsilon_{k,b}^{-1}\mathbf{P}_{k-1}\mathbf{R}_{k,b}\mathbf{P}_k^{-1} \mod \beta_k\|_\infty \approx \|\beta_k\|_\infty$. 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:

$$\begin{aligned} \delta &= \overline{\mathbf{A}}_0 \cdot \prod_{k=1}^{\kappa} \overline{\mathbf{A}}_{k,x_{\mathrm{inp}(k)}} \cdot \overline{\mathbf{A}}_{\kappa+1} \\ &= (\beta_0\widetilde{\mathbf{A}}_0 + \mathbf{R}_0) \cdot \prod_{k=1}^{\kappa} (\beta_k \widetilde{\mathbf{A}}_{k,x_{\mathrm{inp}(k)}} + \mathbf{R}_{k,x_{\mathrm{inp}(k)}}) \cdot (\beta_{\kappa+1}\widetilde{\mathbf{A}}_{\kappa+1} + \mathbf{R}_{\kappa+1}), \\ &= \alpha\beta \cdot \mathbf{s} \prod_{k=1}^{\kappa} \mathbf{A}_{k,x_{inp(k)}} \mathbf{t}^T + o(\beta) \end{aligned} \tag{4}$$

$$\begin{aligned} \delta' &= \overline{\mathbf{A}}'_0 \cdot \prod_{k=1}^{\kappa} \overline{\mathbf{A}}'_{k,x_{\mathrm{inp}(k)}} \cdot \overline{\mathbf{A}}'_{\kappa+1} \\ &= (\beta_0\widetilde{\mathbf{A}}'_0 + \mathbf{R}'_0) \cdot \prod_{k=1}^{\kappa} (\beta_k \widetilde{\mathbf{A}}'_{k,x_{\mathrm{inp}(k)}} + \mathbf{R}'_{k,x_{\mathrm{inp}(k)}}) \cdot (\beta_{\kappa+1}\widetilde{\mathbf{A}}'_{\kappa+1} + \mathbf{R}'_{\kappa+1}), \\ &= \alpha\beta \cdot \mathbf{s}'\mathbf{t}'^T + o(\beta) \end{aligned} \tag{5}$$

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 $\|\delta - \delta'\|_\infty < q^{\frac{\kappa+1}{\kappa+2}}$ 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 $\delta/\gamma \leq 2^{-O(\lambda)}$.

We first give the concept of permutation that is used in the definition of determinant. A permutation $p = (p_1, p_2, \ldots, p_n)$ of the numbers $(1, 2, \ldots, n)$ 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

$$\pi(p) = \begin{cases} +1 & \text{if the parity of } p \text{ is even,} \\ -1 & \text{if the parity of } p \text{ is odd.} \end{cases} \tag{6}$$

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

$$\det(\mathbf{A}) = \sum_p \pi(p) \prod_{i=1}^{n} \mathbf{A}[i, p_i], \tag{7}$$

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

Lemma 2 (*Determinant Inequality*). *Suppose that* $\mathbf{A}$ *is an* $n \times n$*-dimensional matrix over* $\mathbb{Q}$ *such that* $\gamma \leq |\mathbf{A}[i, j]| \leq c\gamma$ *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 the determinant in Eq. (7),

$$|\det(\mathbf{A})| = \left| \sum_p \pi(p) \prod_{i=1}^{n} \mathbf{A}[i, p_i] \right| = \gamma^n \cdot \left| \sum_p \pi(p) \prod_{i=1}^{n} \frac{\mathbf{A}[i, p_i]}{\gamma} \right| = \gamma^n \cdot \left| \sum_p \pi(p) \mathbf{A}_p \right|, \tag{8}$$

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

By $\gamma \leq |\mathbf{A}[i, j]| \leq c\gamma$, we obtain $|\frac{\mathbf{A}[i,p_i]}{\gamma}| \geq 1$ and $1 \leq |\mathbf{A}_p| \leq c^n$. According to Chernoff–Hoeffding inequality, $\left| \sum_p \pi(p) \mathbf{A}_p \right| \geq 1$ with overwhelming probability. On the other hand, $\left| \sum_p \pi(p) \mathbf{A}_p \right| \leq \sum_p |\pi(p)| c^n = n! c^n$. □

Definition 2 (*Matrix Decomposition* ($MD_{\gamma,\delta}$)). 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

$$\begin{cases} |\mathbf{A}_1[i, j]| & = \Theta(\gamma), \text{for all } i, j \in [n] \\ |\mathbf{A}_\delta[i, j]| & = O(\delta), \text{for all } i, j \in [n]. \end{cases} \tag{9}$$

Lemma 3 (*Determinant Estimation I*). *Suppose that* $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_\delta$ *is* $MD_{\gamma,\delta}$ *and 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 the determinant in Eq. (7),

$$\det(\mathbf{A}) = \sum_p \pi(p) \prod_{i=1}^{n} \mathbf{A}[i, p_i] = \sum_p \pi(p) \prod_{i=1}^{n} (\mathbf{A}_1[i, p_i] + \mathbf{A}_\delta[i, p_i]) \tag{10}$$

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

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

$$\det(\mathbf{A}) = \sum_p \pi(p) \prod_{i=1}^{n} (\mathbf{A}_1[i, p_i] + \mathbf{A}_\delta[i, p_i]) = \sum_p \pi(p) \left(B_p + o(B_p) \right) \tag{11}$$

where $B_p = \sum_{j=1}^{n} \mathbf{A}_\delta[j, p_j] \prod_{i \neq j} \mathbf{A}_1[i, p_i]$.

So, $|\det(\mathbf{A})| \leq \sum_p \pi(p) \sum_{j=1}^{n} O(|\mathbf{A}_\delta[j, p_j] \prod_{i \neq j} \mathbf{A}_1[i, p_i]|) \leq O(n \cdot n! \cdot \delta\gamma^{n-1})$.

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

Remark 2. The result of Lemma 3 does not contradict that of Lemma 2. 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 3 satisfies the condition of Lemma 2. Namely, the determinant of the square submatrix can be estimated by applying Lemma 2. Accordingly, we can further improve the determinant estimation in Lemma 3.

Definition 3 (*Approximate Eigenvalue*). Let $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_\delta$ be a (γ, δ)-matrix decomposition. The eigenvalues of $\mathbf{A}$ are defined as the approximate eigenvalues of $\mathbf{A}_1$.

Definition 4 (*Diagonally Dominant Matrix (DDM)*). An $n \times n$-dimensional matrix $\mathbf{A}$ is diagonally dominant if for all $i \in [k]$, $|\mathbf{A}[i, i]| \geq \sum_{j \neq i} |\mathbf{A}[i, j]|$.

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

Definition 5 ((γ, δ)-*Diagonally Dominant Matrix* ($DDM_{\gamma,\delta}$)). $\mathbf{A}$ is a (γ, δ)-diagonally dominant matrix if $\mathbf{A}$ is satisfied

$$|\mathbf{A}[i, j]| = \begin{cases} O(\gamma), & \text{if } i = j \\ O(\delta), & \text{if } i \neq j. \end{cases} \tag{12}$$

Note that in this paper we only consider the approximate eigenvalue of diagonally dominant matrix. In the following we will prove the inverse of a $\mathrm{DDM}_{\gamma,\delta}$ matrix is a $\mathrm{DDM}_{\gamma',\delta'}$ matrix where γ', δ' depend on γ, δ. 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. *Suppose that* $\mathbf{A}$ *is a* $DDM_{\gamma,\delta}$ *matrix. Then* $\mathbf{A}^{-1}$ *is a* $DDM_{\gamma^{-1}, n\delta/\gamma^{-2}}$ *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 $|\mathbf{A}_1[i, i]| = O(\gamma), \quad i \in [n]$, $|\mathbf{A}_\delta[i, j]| = O(\delta), \quad i, j \in [n]$. So, $\mathbf{A}_1^{-1} = \mathrm{Diag}(\mathbf{A}^{-1}[1, 1], \ldots, \mathbf{A}^{-1}[n, n])$.

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

By $\|\mathbf{A}_1^{-1}\mathbf{A}_\delta\|_\infty \leq n\|\mathbf{A}_1^{-1}\|_\infty \|\mathbf{A}_\delta\|_\infty \leq O(n^2\gamma_{\max}^{-1}\delta) = O(n^2\gamma^{-1}\delta) < 1/n$, we have $(\mathbf{A}_1 + \mathbf{A}_\delta)^{-1} = (\mathbf{I} + \mathbf{A}_1^{-1}\mathbf{A}_\delta)^{-1}\mathbf{A}_1^{-1} = (\mathbf{I} - \mathbf{A}_1^{-1}\mathbf{A}_\delta + (\mathbf{A}_1^{-1}\mathbf{A}_\delta)^2 - \cdots)\mathbf{A}_1^{-1} = \mathbf{A}_1^{-1} + \mathbf{A}_{\delta'}$. 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,

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

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

$$|\mathbf{A}^{-1}[i, j]| = \begin{cases} O(\gamma^{-1}), & \text{if } i = j \\ O(n^2\delta\gamma^{-2}), & \text{if } i \neq j. \end{cases} \tag{13}$$

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

Remark 3. 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 [43,44] 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 observe that

some approximate ratios of the bundling scalars can be recovered by applying the matrix properties described in the above section. Using 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 [42]. 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

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

Similarly, for the dummy branch we have

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

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 $\alpha_x, \alpha_y, \alpha_z$ et al. our attack requires to use the following results in [42].

Lemma 5 ([42]). *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. For brevity, we incorporate the index of the left (resp. right) bookend vector into the interval X (resp. Z) and write as $X' = \{0\} \cup X$ (resp. $Z' = Z \cup \{\kappa+1\}$). We also define the values of the index function: $\mathrm{inp}(0) = 0, \mathrm{inp}(\kappa+1) = \kappa+1$. Given the input $f^{(i,b,j)}$, we can evaluate the obfuscator and write the result in the form of a matrix product:

$$\begin{aligned}
w &= \overline{\mathbf{A}}_0 \cdot \prod_{k=1}^{\kappa} \overline{\mathbf{A}}_{k,f_{\mathrm{inp}(k)}} \cdot \overline{\mathbf{A}}_{\kappa+1} - \overline{\mathbf{A}}'_0 \cdot \prod_{k=1}^{\kappa} \overline{\mathbf{A}}'_{k,f_{\mathrm{inp}(k)}} \cdot \overline{\mathbf{A}}'_{\kappa+1}\\
&= \prod_{k\in X'} \overline{\mathbf{A}}_{k,f_{\mathrm{inp}(k)}} \cdot \prod_{k\in Y} \overline{\mathbf{A}}_{k,f_{\mathrm{inp}(k)}} \cdot \prod_{k\in Z'} \overline{\mathbf{A}}_{k,f_{\mathrm{inp}(k)}}\\
&\quad - \prod_{k\in X'} \overline{\mathbf{A}}'_{k,f_{\mathrm{inp}(k)}} \cdot \prod_{k\in Y} \overline{\mathbf{A}}'_{k,f_{\mathrm{inp}(k)}} \cdot \prod_{k\in Z'} \overline{\mathbf{A}}'_{k,f_{\mathrm{inp}(k)}}\\
&= \prod_{k\in X'} (\beta_k\widetilde{\mathbf{A}}_{k,f_{\mathrm{inp}(k)}} + \mathbf{R}_{k,f_{\mathrm{inp}(k)}}) \cdot \prod_{k\in Y} (\beta_k\widetilde{\mathbf{A}}_{k,f_{\mathrm{inp}(k)}} + \mathbf{R}_{k,f_{\mathrm{inp}(k)}})\\
&\quad \cdot \prod_{k\in Z'} (\beta_k\widetilde{\mathbf{A}}_{k,f_{\mathrm{inp}(k)}} + \mathbf{R}_{k,f_{\mathrm{inp}(k)}})\\
&\quad - \prod_{k\in X'} (\beta_k\widetilde{\mathbf{A}}'_{k,f_{\mathrm{inp}(k)}} + \mathbf{R}'_{k,f_{\mathrm{inp}(k)}}) \cdot \prod_{k\in Y} (\beta_k\widetilde{\mathbf{A}}'_{k,f_{\mathrm{inp}(k)}} + \mathbf{R}'_{k,f_{\mathrm{inp}(k)}})\\
&\quad \cdot \prod_{k\in Z'} (\beta_k\widetilde{\mathbf{A}}'_{k,f_{\mathrm{inp}(k)}} + \mathbf{R}'_{k,f_{\mathrm{inp}(k)}})\\
&= (\beta_{X'}\mathbf{S}_{x^{(i)}} + \mathbf{R}_{x^{(i)}})(\beta_Y\mathbf{S}_{y^{(b)}} + \mathbf{R}_{y^{(b)}})(\beta_{Z'}\mathbf{S}_{z^{(j)}} + \mathbf{R}_{z^{(j)}})\\
&\quad - (\beta_{X'}\mathbf{S}'_{x^{(i)}} + \mathbf{R}'_{x^{(i)}})(\beta_Y\mathbf{S}'_{y^{(b)}} + \mathbf{R}'_{y^{(b)}})(\beta_{Z'}\mathbf{S}'_{z^{(j)}} + \mathbf{R}'_{z^{(j)}})\\
&= \begin{pmatrix} \beta_{X'}\mathbf{S}_{x^{(i)}} + \mathbf{R}_{x^{(i)}} & -(\beta_{X'}\mathbf{S}'_{x^{(i)}} + \mathbf{R}'_{x^{(i)}}) \end{pmatrix}\\
&\quad \times \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} \beta_{Z'}\mathbf{S}_{z^{(j)}} + \mathbf{R}_{z^{(j)}} \\ \beta_{Z'}\mathbf{S}'_{z^{(j)}} + \mathbf{R}'_{z^{(j)}} \end{pmatrix},
\end{aligned} \tag{14}$$

where

$$\beta_{X'} = \prod_{k\in X'} \beta_k, \quad \beta_Y = \prod_{k\in Y} \beta_k, \quad \beta_{Z'} = \prod_{k\in Z'} \beta_k,$$
$$\mathbf{R}_{x^{(i)}} = \prod_{k\in X'} (\beta_k\widetilde{\mathbf{A}}_{k,f_{\mathrm{inp}(k)}} + \mathbf{R}_{k,f_{\mathrm{inp}(k)}}) - \beta_{X'}\mathbf{S}_{x^{(i)}},$$

and all other matrices $\mathbf{R}$'s in Eq. (14) are similarly defined.

It is easy to verify that w in Eq. (14) holds not only modulo q but also over the ring R. This is because $f^{(i,b,j)} = x^{(i)}y^{(b)}z^{(j)}$ is a zero of the function.

Let $b \in \{0,1\}$ and i, j range over $2s$ inputs. We can obtain the matrices:

$$\begin{aligned}
\mathbf{W}_b &= \mathbf{X}\mathbf{Y}_b\mathbf{Z}\\
&= \begin{pmatrix} \cdots \\ \beta_{X'}\mathbf{S}_{x^{(i)}} + \mathbf{R}_{x^{(i)}} & -(\beta_{X'}\mathbf{S}'_{x^{(i)}} + \mathbf{R}'_{x^{(i)}}) \\ \cdots \end{pmatrix}\\
&\quad \times \begin{pmatrix} \beta_Y\mathbf{S}_{y^{(b)}} + \mathbf{R}_{y^{(b)}} & 0 \\ 0 & \beta_Y\mathbf{S}'_{y^{(b)}} + \mathbf{R}'_{y^{(b)}} \end{pmatrix} \times \begin{pmatrix} \cdots, & \beta_{Z'}\mathbf{S}_{z^{(j)}} + \mathbf{R}_{z^{(j)}}, & \cdots \\ & \beta_{Z'}\mathbf{S}'_{z^{(j)}} + \mathbf{R}'_{z^{(j)}} & \end{pmatrix},
\end{aligned} \tag{15}$$

where $\mathbf{X}, \mathbf{Y}_b, \mathbf{Z} \in R^{2s\times 2s}$ are full rank with high probability.

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:

$$\mathbf{Y}_1\mathbf{Y}_0^{-1} = \begin{pmatrix} \beta_Y\mathbf{S}_{y^{(1)}} + \mathbf{R}_{y^{(1)}}, & 0 \\ 0 & \beta_Y\mathbf{S}'_{y^{(1)}} + \mathbf{R}'_{y^{(1)}} \end{pmatrix} \cdot \begin{pmatrix} \beta_Y\mathbf{S}_{y^{(0)}} + \mathbf{R}_{y^{(0)}}, & 0 \\ 0 & \beta_Y\mathbf{S}'_{y^{(0)}} + \mathbf{R}'_{y^{(0)}} \end{pmatrix}^{-1} \tag{16}$$

According to the BP obfuscator construction, we have

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

where $\mathbf{A}_1 = \beta_Y\alpha_{y^{(0)}}\widehat{\mathbf{A}}_{y^{(0)}}$, $\mathbf{A}_\delta = \mathbf{P}_{y_1}\mathbf{R}_{y^{(0)}}\mathbf{P}^{-1}_{z_1}$, and $\mathbf{A}_1$ is a diagonal matrix.

By the parameter settings, it is easy to verify that $\delta = \|\mathbf{A}_\delta\|_\infty = O(s^2n^{1.5}\sigma^3)$ and $\gamma = \max_{i\in[n]}\|A_1[i,i]\|_\infty \approx O(\beta_Y\alpha_{y^{(0)}})$ such that $\delta/\gamma \le 2^{-O(\lambda)}$. So, by Lemma 4 we get $(\beta_Y\mathbf{S}_{y^{(0)}} + \mathbf{R}_{y^{(0)}})^{-1} = \mathbf{P}^{-1}_{z_1}(\mathbf{A}_1^{-1} + \mathbf{A}_{\delta'})\mathbf{P}_{y_1}$, where $\delta' = n\delta/\gamma^2$.

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

$$\begin{aligned}
&(\beta_Y\mathbf{S}_{y^{(1)}} + \mathbf{R}_{y^{(1)}})(\beta_Y\mathbf{S}_{y^{(0)}} + \mathbf{R}_{y^{(0)}})^{-1}\\
=&(\beta_Y\alpha_{y^{(1)}}\mathbf{P}^{-1}_{y_1}\widehat{\mathbf{A}}_{y^{(1)}}\mathbf{P}_{z_1} + \mathbf{R}_{y^{(1)}})\mathbf{P}^{-1}_{z_1}(\mathbf{A}_1^{-1} + \mathbf{A}_{\delta'})\mathbf{P}_{y_1}\\
=&\frac{\alpha_{y^{(1)}}}{\alpha_{y^{(0)}}}\mathbf{P}^{-1}_{y_1}(\widehat{\mathbf{A}}_{y^{(1)}}\widehat{\mathbf{A}}^{-1}_{y^{(0)}})\mathbf{P}_{y_1} + \mathbf{R}\\
=&\frac{\alpha_{y^{(1)}}}{\alpha_{y^{(0)}}}\mathbf{P}^{-1}_{y_1}\begin{pmatrix}\mathbf{E}_{y^{(1)}} & 0\\ 0 & \mathbf{A}_{y^{(1)}}\end{pmatrix}\begin{pmatrix}\mathbf{E}_{y^{(0)}} & 0\\ 0 & \mathbf{A}_{y^{(0)}}\end{pmatrix}^{-1}\mathbf{P}_{y_1} + \mathbf{R}\\
=&\frac{\alpha_{y^{(1)}}}{\alpha_{y^{(0)}}}\mathbf{P}^{-1}_{y_1}\begin{pmatrix}\mathbf{E}_{y^{(1)}}\mathbf{E}^{-1}_{y^{(0)}} & 0\\ 0 & \mathbf{A}_{y^{(1)}}\mathbf{A}^{-1}_{y^{(0)}}\end{pmatrix}\mathbf{P}_{y_1} + \mathbf{R}\\
\approx&\frac{\alpha_{y^{(1)}}}{\alpha_{y^{(0)}}}\mathbf{P}^{-1}_{y_1}\begin{pmatrix}\mathbf{E}_{y^{(1)}}\mathbf{E}^{-1}_{y^{(0)}} & 0\\ 0 & \mathbf{A}_{y^{(1)}}\mathbf{A}^{-1}_{y^{(0)}}\end{pmatrix}\mathbf{P}_{y_1},
\end{aligned} \tag{17}$$

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

By Lemma 1, we have $\mathbf{A}_{y^{(1)}}\mathbf{A}^{-1}_{y^{(0)}} = \mathbf{I}^{w\times w}$. 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, $\frac{\alpha_{y(1)}}{\alpha_{y(0)}} = \frac{\alpha'_{y(1)}}{\alpha'_{y(0)}}$, 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 4. 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 $\|\delta_x\|_\infty, \|\delta_{x'}\|_\infty < q$ according to [43], 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 $\|\beta\|_\infty \geq q$. Note that its updated version [44] has set the parameters such that $\|\delta_x\|_\infty, \|\delta_{x'}\|_\infty > q$.

5.3. Annihilation attack

Chen, Gentry and Halevi [42] have extended the annihilation attack introduced by Miles, Sahai and Zhandry [41] 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 to extend the CGH attack to attack the BP obfuscators over GGH13 without ideals [43]. 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 [42].

Example 1 (*[42]*). 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:	**I**	⋯	**I**	**I**	**I**	**I**	**I**	⋯	**I**
	1:	**I**	⋯	**I**	**I**	**I**	**I**	**I**	⋯	**I**
B'=	0:	**I**	⋯	**I**	**I**	**I**	**I**	**I**	⋯	**I**
	1:	**I**	⋯	**I**	**P**	**I**	$\mathbf{P}^{-1}$	**I**	⋯	**I**
Steps	0:		X		u	v	w		Z	
Input bits	1:	*	⋯	*	j_1	j_2	j_1	*	⋯	*

Unlike [42], 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 [42]. However, we here are working on $\mathbb{K}$, not mod $\langle g \rangle$ and hence we can take $v_0 = 1$, $v_1 \approx \alpha'_1/\alpha'_0$ and $\zeta_{00} = 1$, $\zeta_{11} \approx \alpha_1\alpha'_1/\alpha_0\alpha'_0$

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 $f_{\mu\nu}^{(i,j)}$:

$$\begin{aligned}\mathrm{Eval}(f_{\mu\nu}^{(i,j)}) &= \overline{\mathbf{A}}_0 \cdot \prod_{k=1}^{\kappa} \overline{\mathbf{A}}_{k,x_{\mathrm{inp}(k)}} \cdot \overline{\mathbf{A}}_{\kappa+1} - \overline{\mathbf{A}}'_0 \cdot \prod_{k=1}^{\kappa} \overline{\mathbf{A}}'_{k,x_{\mathrm{inp}(k)}} \cdot \overline{\mathbf{A}}'_{\kappa+1} \\ &= (\beta_0\widetilde{\mathbf{A}}_0 + \mathbf{R}_0) \cdot \prod_{k=1}^{\kappa} (\beta_k\widetilde{\mathbf{A}}_{k,x_{\mathrm{inp}(k)}} + \mathbf{R}_{k,x_{\mathrm{inp}(k)}}) \cdot (\beta_{\kappa+1}\widetilde{\mathbf{A}}_{\kappa+1} + \mathbf{R}_{\kappa+1}) \\ &\quad - (\beta_0\widetilde{\mathbf{A}}'_0 + \mathbf{R}'_0) \cdot \prod_{k=1}^{\kappa} (\beta_k\widetilde{\mathbf{A}}'_{k,x_{\mathrm{inp}(k)}} + \mathbf{R}'_{k,x_{\mathrm{inp}(k)}})(\beta_{\kappa+1}\widetilde{\mathbf{A}}'_{\kappa+1} + \mathbf{R}'_{\kappa+1}).\end{aligned} \tag{18}$$

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

$$\begin{aligned}A[i,j] = &\ \mathrm{Eval}(f_{11}^{(i,j)}) \cdot \zeta_{00} \cdot v_1v_0 - \mathrm{Eval}(f_{10}^{(i,j)}) \cdot \zeta_{00} \cdot v_1v_1 \\ &- \mathrm{Eval}(f_{01}^{(i,j)}) \cdot \zeta_{11} \cdot v_0v_0 - \mathrm{Eval}(f_{00}^{(i,j)}) \cdot \zeta_{11} \cdot v_0v_1,\end{aligned}$$

where all the computations are operated in $\mathbb{K}$. As a consequence, choosing sufficient 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 of 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. Parameter analysis

5.4.1. The matrix $\mathbf{D}_Y$

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 $\overline{\beta} = \max_{0\leq k\leq \kappa+1}\{\beta_k\}$ such that $\|\overline{\beta}\|_\infty = \max_{0\leq k\leq \kappa+1}\{\|\beta_k\|_\infty\}$. 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

$$\begin{aligned}\mathbf{A}_Y^{\mu\nu} &= \prod_{k\in Y}(\beta_k\widetilde{\mathbf{A}}_{k,x_{\mathrm{inp}(k)}} + \mathbf{R}_{k,x_{\mathrm{inp}(k)}}) \\ &= (\beta_u\widetilde{\mathbf{A}}_{u,\mu} + \mathbf{R}_{u,\mu}) \cdot (\beta_v\widetilde{\mathbf{A}}_{v,\nu} + \mathbf{R}_{v,\nu}) \cdot (\beta_w\widetilde{\mathbf{A}}_{w,\mu} + \mathbf{R}_{w,\mu}) \\ &= \alpha_\mu\alpha'_\nu \cdot \mathbf{P}_{u-1}^{-1} \cdot \left(\beta_u\widehat{\mathbf{A}}_{u,\mu} + \widehat{\mathbf{R}}_{u,\mu}\right) \cdot \left(\beta_v\widehat{\mathbf{A}}_{v,\nu} + \widehat{\mathbf{R}}_{v,\nu}\right) \cdot \left(\beta_w\widehat{\mathbf{A}}_{w,\mu} + \widehat{\mathbf{R}}_{w,\mu}\right) \cdot \mathbf{P}_w \\ &= \alpha_\mu\alpha'_\nu \cdot \mathbf{P}_{u-1}^{-1} \cdot \left(\mathbf{C}_Y^{\mu\nu} + (\mathbf{D}_Y^{\mu\nu}) + O(\overline{\beta}^{|Y|-2})\mathbf{E}_Y^{\mu\nu}\right) \cdot \mathbf{P}_w \\ &= \alpha_\mu\alpha'_\nu \cdot \mathbf{P}_{u-1}^{-1} \cdot \left(\mathbf{C}_Y^{\mu\nu} + \mathbf{D}_Y^{\mu\nu} + O(\overline{\beta})\mathbf{E}_Y^{\mu\nu}\right) \cdot \mathbf{P}_w,\end{aligned} \tag{19}$$

where
$\widehat{\mathbf{R}}_{u,\mu} = \frac{1}{\epsilon_{u,\mu}}\mathbf{P}_{u-1}\mathbf{R}_{u,\mu}\mathbf{P}_u^{-1}$, $\widehat{\mathbf{R}}_{v,\nu} = \frac{1}{\epsilon_{v,\nu}}\mathbf{P}_u\mathbf{R}_{v,\nu}\mathbf{P}_v^{-1}$, $\widehat{\mathbf{R}}_{w,\mu} = \frac{1}{\epsilon_{w,\mu}}\mathbf{P}_v\mathbf{R}_{w,\mu}\mathbf{P}_w^{-1}$,
$\mathbf{C}_Y^{\mu\nu} = \beta_Y\widehat{\mathbf{A}}_{u,\mu}\widehat{\mathbf{A}}_{v,\nu}\widehat{\mathbf{A}}_{w,\mu}$, $\mathbf{D}_Y^{\mu\nu} = \beta_{uw}\widehat{\mathbf{A}}_{u,\mu}\widehat{\mathbf{R}}_{v,\nu}\widehat{\mathbf{A}}_{w,\mu} + \beta_{uv}\widehat{\mathbf{A}}_{u,\mu}\widehat{\mathbf{A}}_{v,\nu}\widehat{\mathbf{R}}_{w,\mu} + \beta_{vw}\widehat{\mathbf{R}}_{u,\mu}\widehat{\mathbf{A}}_{v,\nu}\widehat{\mathbf{A}}_{w,\mu}$.

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

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

$$\begin{aligned}\mathbf{D}_Y &= \mathbf{D}_Y^{11} - \mathbf{D}_Y^{10} - \mathbf{D}_Y^{01} + \mathbf{D}_Y^{00} \\ &= \left(\beta_{uw}\widehat{\mathbf{A}}_{u,1}\widehat{\mathbf{R}}_{v,1}\widehat{\mathbf{A}}_{w,1} + \beta_{uv}\widehat{\mathbf{A}}_{u,1}\widehat{\mathbf{A}}_{v,1}\widehat{\mathbf{R}}_{w,1} + \beta_{vw}\widehat{\mathbf{R}}_{u,1}\widehat{\mathbf{A}}_{v,1}\widehat{\mathbf{A}}_{w,1}\right) \\ &\quad - \left(\beta_{uw}\widehat{\mathbf{A}}_{u,1}\widehat{\mathbf{R}}_{v,0}\widehat{\mathbf{A}}_{w,1} + \beta_{uv}\widehat{\mathbf{A}}_{u,1}\widehat{\mathbf{A}}_{v,0}\widehat{\mathbf{R}}_{w,1} + \beta_{vw}\widehat{\mathbf{R}}_{u,1}\widehat{\mathbf{A}}_{v,0}\widehat{\mathbf{A}}_{w,1}\right) \\ &\quad - \left(\beta_{uw}\widehat{\mathbf{A}}_{u,0}\widehat{\mathbf{R}}_{v,1}\widehat{\mathbf{A}}_{w,0} + \beta_{uv}\widehat{\mathbf{A}}_{u,0}\widehat{\mathbf{A}}_{v,1}\widehat{\mathbf{R}}_{w,0} + \beta_{vw}\widehat{\mathbf{R}}_{u,0}\widehat{\mathbf{A}}_{v,1}\widehat{\mathbf{A}}_{w,0}\right) \\ &\quad + \left(\beta_{uw}\widehat{\mathbf{A}}_{u,0}\widehat{\mathbf{R}}_{v,0}\widehat{\mathbf{A}}_{w,0} + \beta_{uv}\widehat{\mathbf{A}}_{u,0}\widehat{\mathbf{A}}_{v,0}\widehat{\mathbf{R}}_{w,0} + \beta_{vw}\widehat{\mathbf{R}}_{u,0}\widehat{\mathbf{A}}_{v,0}\widehat{\mathbf{A}}_{w,0}\right).\end{aligned} \tag{20}$$

Now it is completely analogous to the method in [42] 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 $\boldsymbol{A}$

To analyze $\mathbf{A}$, we let $X = \{x_1, x_2, \ldots, x_x\}$, $Y = \{u, v, w\}$, $Z = \{z_1, z_2, \ldots, z_z\}$. 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, 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:

$$\begin{cases} \mathbf{A}^i_X &= \alpha_{x^{(i)}} \cdot \mathbf{P}_0^{-1}\big(\mathbf{C}^i_X + \mathbf{D}^i_X + O(\overline{\beta}^{|X|-2})\mathbf{E}^i_X\big)\mathbf{P}_{u-1} \\ \mathbf{A}^{\mu\nu}_Y &= \alpha_\mu \alpha'_\nu \cdot \mathbf{P}_{u-1}^{-1}\Big(\mathbf{C}^{\mu\nu}_Y + \mathbf{D}^{\mu\nu}_Y + O(\overline{\beta}^{|Y|-2})\mathbf{E}^{\mu\nu}_Y\Big)\mathbf{P}_w \\ \mathbf{A}^j_Z &= \alpha_{z^{(j)}} \cdot \mathbf{P}_w^{-1}\big(\mathbf{C}^j_Z + \mathbf{D}^j_Z + O(\overline{\beta}^{|Z|-2})\mathbf{E}^j_Z\big)\mathbf{P}_\kappa \end{cases} \tag{21}$$

$$\begin{cases} \mathbf{A}'^i_X &= \alpha'_{x^{(i)}} \cdot \mathbf{P}'^{-1}_0\big(\mathbf{C}'^i_X + \mathbf{D}'^i_X + O(\overline{\beta}^{|X|-2})\mathbf{E}'^i_X\big)\mathbf{P}'_{u-1} \\ \mathbf{A}'^{\mu\nu}_Y &= \alpha_\mu \alpha'_\nu \cdot \mathbf{P}'^{-1}_{u-1}\Big(\mathbf{C}'^{\mu\nu}_Y + \mathbf{D}'^{\mu\nu}_Y + O(\overline{\beta}^{|Y|-2})\mathbf{E}'^{\mu\nu}_Y\Big)\mathbf{P}'_w \\ \mathbf{A}'^j_Z &= \alpha'_{z^{(j)}} \cdot \mathbf{P}'^{-1}_w\big(\mathbf{C}'^j_Z + \mathbf{D}'^j_Z + O(\overline{\beta}^{|Z|-2})\mathbf{E}'^j_Z\big)\mathbf{P}'_\kappa \end{cases} \tag{22}$$

In Eqs. (21) and (22), except for the unspecified small noise matrices $\mathbf{E}^i_X, \mathbf{E}'^i_X, \mathbf{E}^j_Z, \mathbf{E}'^j_Z$, we also use the following notations

$$\mathbf{C}^i_X = \beta_X \cdot \prod_{k\in X} \widehat{\mathbf{A}}_{k,u_{\mathrm{inp}(k)}}, \quad \mathbf{C}'^i_X = \beta_X \cdot \prod_{k\in X} \widehat{\mathbf{A}}'_{k,u_{\mathrm{inp}(k)}},$$
$$\mathbf{C}^j_Z = \beta_Z \cdot \prod_{k\in Z} \widehat{\mathbf{A}}_{k,u_{\mathrm{inp}(k)}}, \quad \mathbf{C}'^j_Z = \beta_Z \cdot \prod_{k\in Z} \widehat{\mathbf{A}}'_{k,u_{\mathrm{inp}(k)}},$$
$$\mathbf{D}^i_X = \sum_{k\in X} \beta_{X_k} \widehat{\mathbf{A}}_{x_1,u_{\mathrm{inp}(x_1)}} \cdots \widehat{\mathbf{A}}_{k-1,u_{\mathrm{inp}(k-1)}} \widehat{\mathbf{R}}_{k,u_{\mathrm{inp}(k)}} \cdot \widehat{\mathbf{A}}_{k+1,u_{\mathrm{inp}(k+1)}} \cdots \widehat{\mathbf{A}}_{x_x,u_{\mathrm{inp}(x_x)}},$$
$$\mathbf{D}'^i_X = \sum_{k\in X} \beta_{X_k} \widehat{\mathbf{A}}'_{x_1,u_{\mathrm{inp}(x_1)}} \cdots \widehat{\mathbf{A}}'_{k-1,u_{\mathrm{inp}(k-1)}} \widehat{\mathbf{R}}'_{k,u_{\mathrm{inp}(k)}} \cdot \widehat{\mathbf{A}}'_{k+1,u_{\mathrm{inp}(k+1)}} \cdots \widehat{\mathbf{A}}'_{x_x,u_{\mathrm{inp}(x_x)}},$$
$$\mathbf{D}^j_Z = \sum_{k\in Z} \beta_{Z_k} \widehat{\mathbf{A}}_{z_1,u_{\mathrm{inp}(z_1)}} \cdots \widehat{\mathbf{A}}_{k-1,u_{\mathrm{inp}(k-1)}} \widehat{\mathbf{R}}_{k,u_{\mathrm{inp}(k)}} \cdot \widehat{\mathbf{A}}_{k+1,u_{\mathrm{inp}(k+1)}} \cdots \widehat{\mathbf{A}}_{z_z,u_{\mathrm{inp}(z_z)}},$$
$$\mathbf{D}'^j_Z = \sum_{k\in Z} \beta_{Z_k} \widehat{\mathbf{A}}'_{z_1,u_{\mathrm{inp}(z_1)}} \cdots \widehat{\mathbf{A}}'_{k-1,u_{\mathrm{inp}(k-1)}} \widehat{\mathbf{R}}'_{k,u_{\mathrm{inp}(k)}} \cdot \widehat{\mathbf{A}}'_{k+1,u_{\mathrm{inp}(k+1)}} \cdots \widehat{\mathbf{A}}'_{z_z,u_{\mathrm{inp}(z_z)}},$$

where

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

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

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

$$\mathbf{C}_0 = \beta_0\widehat{\mathbf{A}}_0, \quad \mathbf{C}_{\kappa+1} = \beta_{\kappa+1}\widehat{\mathbf{A}}_{\kappa+1}, \quad \mathbf{C}'_0 = \beta_0\widehat{\mathbf{A}}'_0, \quad \mathbf{C}'_{\kappa+1} = \beta_{\kappa+1}\widehat{\mathbf{A}}'_{\kappa+1}$$
$$\widehat{\mathbf{R}}_0 = \frac{1}{\epsilon_0}\mathbf{R}_0\mathbf{P}_0^{-1}, \quad \widehat{\mathbf{R}'}_0 = \frac{1}{\epsilon'_0}\mathbf{R}'_0\mathbf{P}'^{-1}_0, \quad \widehat{\mathbf{R}}_{\kappa+1} = \frac{1}{\epsilon_{\kappa+1}}\mathbf{P}_\kappa\mathbf{R}_{\kappa+1}, \quad \widehat{\mathbf{R}'}_{\kappa+1} = \frac{1}{\epsilon'_{\kappa+1}}\mathbf{P}'_\kappa\mathbf{R}'_{\kappa+1}.$$

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

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

By the definition of the bundling scalars and their approximate ratios that solve in the above subsection, it is easy to verify that $\alpha_1\alpha'_1 \cdot \zeta_{00} \cdot \upsilon_1\upsilon_0 \approx \alpha_1\alpha'_0 \cdot \zeta_{00} \cdot \upsilon_1\upsilon_1 \approx \alpha_0\alpha'_1 \cdot \zeta_{11} \cdot \upsilon_0\upsilon_0 \approx \alpha_0\alpha'_0 \cdot \zeta_{11} \cdot \upsilon_0\upsilon_1$, 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:

$$A[i,j] = F[i,j] - F'[i,j] + O(\overline{\beta}^{\kappa}), \tag{23}$$

where

$$F[i,j] = \mathbf{x}_i\mathbf{C}_Y\mathbf{z}_j + \mathbf{x}_i\mathbf{D}_Y\mathbf{z}_j + \mathbf{e}_i\mathbf{C}_Y\mathbf{z}_j + \mathbf{r}_i\mathbf{C}_Y\mathbf{z}_j + \mathbf{x}_i\mathbf{C}_Y\mathbf{w}_j;$$

$$F'[i,j] = \mathbf{x}'_i\mathbf{C}'_Y\mathbf{z}'_j + \mathbf{x}'_i\mathbf{D}'_Y\mathbf{z}'_j + \mathbf{e}'_i\mathbf{C}'_Y\mathbf{z}'_j + \mathbf{x}'_i\mathbf{C}'_Y\mathbf{z}'_j + \mathbf{x}'_i\mathbf{C}'_Y\mathbf{w}'_j.$$

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 $\mathbf{x}_i, \mathbf{x}'_i, \mathbf{e}_i, \mathbf{e}'_i = \begin{pmatrix} 0^m & \$^m & \$^w \end{pmatrix}$, $\mathbf{z}_j, \mathbf{z}'_j, \mathbf{f}_j$, $\mathbf{f}'_j = \begin{pmatrix} \$^m & 0^m & \$^w \end{pmatrix}^T$, 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:

$$\begin{aligned} \mathbf{F} &= \mathbf{X}\mathbf{C}_Y\mathbf{Z} + \mathbf{X}\mathbf{D}_Y\mathbf{Z} + \mathbf{E}\mathbf{C}_Y\mathbf{Z} + \mathbf{R}\mathbf{C}_Y\mathbf{Z} + \mathbf{X}\mathbf{C}_Y\mathbf{W} \\ &= \begin{pmatrix} 0 & \mathbf{X}_2 & \mathbf{X}_3 \end{pmatrix} \begin{pmatrix} \mathbf{C}_{1,1} & 0 & 0 \\ 0 & \mathbf{C}_{2,2} & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Z}_1 \\ 0 \\ \mathbf{Z}_3 \end{pmatrix} \\ &\quad + \begin{pmatrix} 0 & \mathbf{X}_2 & \mathbf{X}_3 \end{pmatrix} \begin{pmatrix} \mathbf{D}_{1,1} & \mathbf{D}_{1,2} & \mathbf{D}_{1,3} \\ \mathbf{D}_{2,1} & \mathbf{D}_{2,2} & \mathbf{D}_{2,3} \\ \mathbf{D}_{3,1} & \mathbf{D}_{3,2} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Z}_1 \\ 0 \\ \mathbf{Z}_3 \end{pmatrix} \\ &\quad + \begin{pmatrix} 0 & \mathbf{E}_2 & \mathbf{E}_3 \end{pmatrix} \begin{pmatrix} \mathbf{C}_{1,1} & 0 & 0 \\ 0 & \mathbf{C}_{2,2} & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Z}_1 \\ 0 \\ \mathbf{Z}_3 \end{pmatrix} \\ &\quad + \begin{pmatrix} \mathbf{R}_1 & \mathbf{R}_2 & \mathbf{R}_3 \end{pmatrix} \begin{pmatrix} \mathbf{C}_{1,1} & 0 & 0 \\ 0 & \mathbf{C}_{2,2} & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Z}_1 \\ 0 \\ \mathbf{Z}_3 \end{pmatrix} \\ &\quad + \begin{pmatrix} 0 & \mathbf{X}_2 & \mathbf{X}_3 \end{pmatrix} \begin{pmatrix} \mathbf{C}_{1,1} & 0 & 0 \\ 0 & \mathbf{C}_{2,2} & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \mathbf{W}_3 \end{pmatrix} \\ &= \big(\mathbf{X}_2\mathbf{D}_{2,1} + \mathbf{X}_3\mathbf{D}_{3,1} + \mathbf{R}_1\mathbf{C}_{1,1}\big)\mathbf{Z}_1 + \mathbf{X}_2\big(\mathbf{D}_{2,3}\mathbf{Z}_3 + \mathbf{C}_{2,2}\mathbf{W}_2\big) \end{aligned} \tag{24}$$

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 $\mathbf{F}$ cannot 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 6. *Let $\xi = 4m+1$, $\gamma = \|\overline{\beta}^{\kappa+1}\|_\infty$ and $\delta = \|\overline{\beta}^{\kappa}\|_\infty$. Suppose there exist sufficiently many inputs $u^{i,j}_{\mu\nu}$ that are all the zero of the function. Then when m is constant, with high probability*

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

When $m = poly(\lambda)$, under the following heuristic assumption

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

Proof. According to the analysis of $\mathbf{A}$, for B we have $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_\delta$, where $\mathbf{A}_1 = \mathbf{F} - \mathbf{F}'$, $\mathbf{A}_\delta = O(\overline{\beta}^\kappa)\mathbf{E}$ and $\mathbf{E}$ is a matrix whose entries are polynomials with small norm over $\mathbb{K}$.

Thus, for 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 $\|\det(\mathbf{A})\|_\infty = O(\gamma^{\xi-1}\delta)$ by Lemma 3.

However, for 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 $\|\det(\mathbf{A})\|_\infty = O(\gamma^\xi)$ for B' by Lemma 2.

When $m = \text{poly}(\lambda)$ we heuristically assume that $\|\det(\mathbf{A})\|_\infty$ 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 $\|\det(\mathbf{A})\|_\infty = O(\xi! \cdot \xi\gamma^{\xi-1}\delta)$ by Lemma 3, and for B' the result directly follows the heuristic assumption. □

5.5. Analysis of recent immunization

In order to prevent the annihilation attack [41], Garg et al. [24] (a merged version of [46,47]) constructed a variant of BP obfuscator whose security is proved in the weakened idealized model. However, Chen, Gentry and Halevi [42] observed that this variant could not thwart the annihilation attack if the branching program is input partitioning. This attack result is not contradictory to the security proof in [24], 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 2 m × 2 m matrices $\mathbf{E}_{k,b}, \mathbf{E}'_{k,b}$ instead of the diagonal ones, and takes the bookend vectors as $\widehat{\mathbf{A}}_0, \widehat{\mathbf{A}}'_0 = \left(0^{2m}, \quad \$^w\right)$ and $\widehat{\mathbf{A}}_{\kappa+1}, \widehat{\mathbf{A}}'_{\kappa+1} = \left(\$^{2m}, \quad \$^w\right)$.

Observe that the algorithm that solves approximate ratios of the bundling scalars still works. Moreover, the analysis of the matrix $\mathbf{D}_Y$ in Eq. (20) remains the same. For the rank of $\mathbf{F}$ in Eq. (24), we analyze $\mathbf{F}$ for the program B in Example 1 as follows:

$$\begin{aligned}
\mathbf{F} &= \mathbf{X}\mathbf{C}_Y\mathbf{Z} + \mathbf{X}\mathbf{D}_Y\mathbf{Z} + \mathbf{E}\mathbf{C}_Y\mathbf{Z} + \mathbf{R}\mathbf{C}_Y\mathbf{Z} + \mathbf{X}\mathbf{C}_Y\mathbf{W} \\
&= \begin{pmatrix} 0 & \mathbf{X}_2 \end{pmatrix} \begin{pmatrix} \mathbf{C}_{1,1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix} + \begin{pmatrix} 0 & \mathbf{X}_2 \end{pmatrix} \begin{pmatrix} \mathbf{D}_{1,1} & \mathbf{D}_{1,2} \\ \mathbf{D}_{2,1} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix} \\
&\quad + \begin{pmatrix} 0 & \mathbf{E}_2 \end{pmatrix} \begin{pmatrix} \mathbf{C}_{1,1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix} + \begin{pmatrix} \mathbf{R}_1 & \mathbf{R}_2 \end{pmatrix} \begin{pmatrix} \mathbf{C}_{1,1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix} \\
&\quad + \begin{pmatrix} 0 & \mathbf{X}_2 \end{pmatrix} \begin{pmatrix} \mathbf{C}_{1,1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{pmatrix} \\
&= \left(\mathbf{X}_2\mathbf{D}_{2,1} + \mathbf{R}_1\mathbf{C}_{1,1}\right)\mathbf{Z}_1,
\end{aligned} \tag{25}$$

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 Eq. (25) since with high probability $\mathbf{D}_{2,2}$ is not a "0" matrix. Therefore, we can use the same algorithm in Section 5 to distinguish between B and B'.

As well, we can also adapt the original variant proposed by Garg et al. [46] 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

Although the increasing popularity of IoMT applications provides many benefits, it also raises serious security and privacy concerns. In this paper, we present a variant of the CGH attack, which breaks an industry-scale IoMT application obfuscator using GGH13 without ideals when the application is input partitionable. Consequently, we resolve an open question of Albrecht, Davidson, Larraia and Pellet-Mary in [43,44]. Our work demonstrates that the security of the obfuscator using GGH13 without ideals [43] is essentially equivalent to that of the GGH13-based obfuscator [19]. Furthermore, our results further strengthen the ones in [43,44] that there is a structural weakness in 'GGH13-type' encodings beyond the presence of $\langle \mathbf{g} \rangle$.

While the immunized construction in [24] can prevent the input-partitioning attack, their weakened graded encoding model does not explicitly include this requirement of non-partitioning input. Therefore, it is still an open problem about how to construct an obfuscator for industry-scale IoMT application with input-partitioning or improve the weakened graded encoding model in [24] to satisfy this input requirement.

CRediT authorship contribution statement

Zhengjun Jing: Conceptualization, Methodology, Writing - original draft. **Chunsheng Gu:** Methodology, Formal analysis. **Yong Li:** Validation. **Mengshi Zhang:** Writing - review & editing. **Guangquan Xu:** Supervision. **Alireza Jolfaei:** Visualization, Investigation. **Peizhong Shi:** Data curation. **Chenkai Tan:** Software. **Xi Zheng:** Investigation, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61672270; No. 61602216 and No. 51705220); the Industry-University-Research Cooperation Project of Jiangsu Province (No. BY2018309), the Research and Innovation Project for College Graduates of Jiangsu Province (No. SJCX19_0713).

References

[1] A. Gatouillat, Y. Badr, B. Massot, E. Sejdić, Internet of medical things: A review of recent contributions dealing with cyber-physical systems in medicine, IEEE Internet Things J. 5 (5) (2018) 3810–3822.

[2] G. Hatzivasilis, O. Soultatos, S. Ioannidis, C. Verikoukis, G. Demetriou, C. Tsatsoulis, Review of security and privacy for the Internet of Medical Things (IoMT), in: 2019 15th International Conference on Distributed Computing in Sensor Systems, DCOSS, IEEE, 2019, pp. 457–464.

[3] H. Fouad, N.M. Mahmoud, M.S. El Issawi, H. Al-Feel, Distributed and scalable computing framework for improving request processing of wearable IoT assisted medical sensors on pervasive computing system, Comput. Commun. 151 (2020) 257–265.

[4] Z. Yue, S. Ding, L. Zhao, Y. Zhang, Z. Cao, M. Tanveer, A. Jolfaei, X. Zheng, Privacy-preserving time series medical images analysis using a hybrid deep learning framework, ACM Trans. Internet Technol. 37 (4) (2019) 1–22.

[5] B. Han, Z. Yin-Liang, Z. Chang-Peng, An object proxy-based dynamic layer replacement to protect IoMT applications, Secur. Commun. Netw. 2019 (2019) http://dx.doi.org/10.1155/2019/2798571.

[6] F. Alsubaei, A. Abuhussein, V. Shandilya, S. Shiva, IoMT-SAF: Internet of medical things security assessment framework, Internet Things 8 (2019) 100–123.

[7] A. Farouk, A. Alahmadi, S. Ghose, A. Mashatan, Blockchain platform for industrial healthcare: Vision and future opportunities, Comput. Commun. 154 (2020) 223–235.

[8] G. Xu, Y. Zhang, A.K. Sangaiah, X. Li, A. Castiglione, X. Zheng, CSP-E2: An abuse-free contract signing protocol with low-storage TTP for energy-efficient electronic transaction ecosystems, Inform. Sci. 476 (2019) 505–515.

[9] W. Wang, X. Wang, D. Feng, J. Liu, Z. Han, X. Zhang, Exploring permission-induced risk in android applications for malicious application detection, IEEE Transactions on Information Forensics and Security 9 (11) (2014) 1869–1882.

[10] R.M. Aileni, G. Suciu, IoMT: A blockchain perspective, in: Decentralised Internet of Things, Springer, 2020, pp. 199–215.
[11] J. Mo, Z. Hu, Y. Lin, Cryptanalysis and security improvement of two authentication schemes for healthcare systems using wireless medical sensor networks, Secur. Commun. Netw. 2020 (2020) http://dx.doi.org/10.1155/2020/5047379.
[12] G. Xu, W. Zhou, A.K. Sangaiah, Y. Zhang, X. Zheng, Q. Tang, N. Xiong, K. Liang, X. Zhou, A security-enhanced certificateless aggregate signature authentication protocol for invanets, IEEE Netw. 34 (2) (2020) 22–29.
[13] W. Wang, M. Zhao, J. Wang, Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network, Journal of Ambient Intelligence and Humanized Computing 10 (8) (2019) 3035–3043.
[14] W. Wang, Y. Shang, Y. He, Y. Li, J. Liu, Botmark: automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors, Information Sciences 511 (2020) 284–296.
[15] Y. Shi, J. Han, X. Wang, J. Gao, H. Fan, An obfuscatable aggregatable signcryption scheme for unattended devices in IoT systems, IEEE Internet Things J. 4 (4) (2017) 1067–1081.
[16] Y. Shi, Q. Zhang, J. Liang, Z. He, H. Fan, Obfuscatable anonymous authentication scheme for mobile crowd sensing, IEEE Syst. J. 13 (3) (2018) 2918–2929.
[17] D. Kavitha, C. Subramaniam, Security threat management by software obfuscation for privacy in internet of medical thing (IoMT) application, J. Comput. Theor. Nanosci. 14 (7) (2017) 3100–3114.
[18] M. Zhang, Y. Jiang, H. Shen, B. Li, W. Susilo, Cloud-based data-sharing scheme using verifiable and cca-secure re-encryption from indistinguishability obfuscation, in: International Conference on Information Security and Cryptology, Springer, 2018, pp. 240–259.
[19] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, B. Waters, Candidate indistinguishability obfuscation and functional encryption for all circuits, SIAM J. Comput. 45 (3) (2016) 882–929.
[20] B. Barak, S. Garg, Y.T. Kalai, O. Paneth, A. Sahai, Protecting obfuscation against algebraic attacks, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2014, pp. 221–238.
[21] D. Boneh, A. Silverberg, Applications of multilinear forms to cryptography, Contemp. Math. 324 (1) (2003) 71–90.
[22] D. Boneh, M. Zhandry, Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation, Algorithmica 79 (4) (2017) 1233–1285.
[23] P. Ananth, D. Gupta, Y. Ishai, A. Sahai, Optimizing obfuscation: Avoiding barrington's theorem, in: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2014, pp. 646–658.
[24] S. Garg, E. Miles, P. Mukherjee, A. Sahai, A. Srinivasan, M. Zhandry, Secure obfuscation in a weak multilinear map model, in: Theory of Cryptography Conference, Springer, 2016, pp. 241–268.
[25] J. Zimmerman, How to obfuscate programs directly, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2015, pp. 439–467.
[26] S. Garg, C. Gentry, S. Halevi, Candidate multilinear maps from ideal lattices, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2013, pp. 1–17.
[27] J.-S. Coron, T. Lepoint, M. Tibouchi, Practical multilinear maps over the integers, in: Annual Cryptology Conference, Springer, 2013, pp. 476–493.
[28] C. Gentry, S. Gorbunov, S. Halevi, Graph-induced multilinear maps from lattices, in: Theory of Cryptography Conference, Springer, 2015, pp. 498–527.
[29] J.-S. Coron, T. Lepoint, M. Tibouchi, New multilinear maps over the integers, in: Annual Cryptology Conference, Springer, 2015, pp. 267–286.
[30] S. Halevi, Graded encoding, Variations on a scheme, in: IACR Cryptology EPrint Archive, 2015, URL: http://eprint.iacr.org/2015/866.
[31] D. Apon, N. Döttling, S. Garg, P. Mukherjee, Cryptanalysis of indistinguishability obfuscations of circuits over ggh13, in: 44th International Colloquium on Automata, Languages, and Programming, Vol. 80, ICALP 2017, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017, pp. 1–16.
[32] J.H. Cheon, K. Han, C. Lee, H. Ryu, D. Stehlé, Cryptanalysis of the multilinear map over the integers, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2015, pp. 3–12.
[33] 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, in: Annual Cryptology Conference, Springer, 2015, pp. 247–266.
[34] Z. Brakerski, C. Gentry, S. Halevi, T. Lepoint, A. Sahai, M. Tibouchi, Cryptanalysis of the quadratic zero-testing of ggh, in: IACR Cryptology EPrint Archive, 2015, URL: http://eprint.iacr.org/2015/845.
[35] Y. Hu, H. Jia, Cryptanalysis of GGH map, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2016, pp. 537–565.
[36] J.-S. Coron, M.S. Lee, T. Lepoint, M. Tibouchi, Cryptanalysis of GGH15 multilinear maps, in: Annual International Cryptology Conference, Springer, 2016, pp. 607–628.
[37] J.H. Cheon, P.-A. Fouque, C. Lee, B. Minaud, H. Ryu, Cryptanalysis of the new CLT multilinear map over the integers, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2016, pp. 509–536.
[38] M. Albrecht, S. Bai, L. Ducas, A subfield lattice attack on overstretched NTRU assumptions, in: Annual International Cryptology Conference, Springer, 2016, pp. 153–178.
[39] J.H. Cheon, J. Jeong, C. Lee, An algorithm for NTRU problems and cryptanalysis of the GGH multilinear map without a low-level encoding of zero, LMS J. Comput. Math. 19 (A) (2016) 255–266.
[40] P. Kirchner, P.-A. Fouque, Revisiting lattice attacks on overstretched NTRU parameters, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2017, pp. 3–26.
[41] E. Miles, A. Sahai, M. Zhandry, Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over GGH13, in: Annual International Cryptology Conference, Springer, 2016, pp. 629–658.
[42] Y. Chen, C. Gentry, S. Halevi, Cryptanalyses of candidate branching program obfuscators, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2017, pp. 278–307.
[43] M.R. Albrecht, A. Davidson, E. Larraia, Notes on GGH13 without the presence of ideals, in: IMA International Conference on Cryptography and Coding, Springer, 2017, pp. 135–158.
[44] M.R. Albrecht, A. Davidson, E. Larraia, A. Pellet-Mary, Notes on GGH13 without the presence of ideals, in: IACR Cryptology EPrint Archive, 2017, URL: http://eprint.iacr.org/2017/906.
[45] D. Micciancio, O. Regev, Worst-case to average-case reductions based on Gaussian measures, SIAM J. Comput. 37 (1) (2007) 267–302.
[46] S. Garg, P. Mukherjee, A. Srinivasan, Obfuscation without the Vulnerabilities of Multilinear Maps, in: IACR Cryptology EPrint Archive, 2016, URL: http://eprint.iacr.org/2016/390.
[47] E. Miles, A. Sahai, M. Zhandry, Secure obfuscation in a weak multilinear map model: A simple construction secure against all known attacks, in: IACR Cryptology ePrint Archive, 2016, URL: http://eprint.iacr.org/2016/588.

Zhengjun Jing received his Ph.D. in Information and Security from Nanjing University of Posts and Telecommunications in 2015. Since 2016, he has been an Associate Professor in the Department of Computer Engineering, Jiangsu University of Technology. His interests are in the cryptanalysis, cloud computing security and design of cryptography.

Chunsheng Gu is a professor in the School of Computer Engineering, Jiangsu University of Technology. He received his Ph.D. Degree from University of Science and Technology of China in 2005. His research interests include cryptanalysis and lattice-based cryptography.

Yong Li He received the M.S. degree in the Architecture of Computer Systems from the Jilin University in 2004. He is currently pursuing a PhD degree with Jilin University, China. He is an associate professor with the College of Computer Science and Engineering at the Changchun University of Technology, China. Now, he is a visiting scholar with Macquarie University, Sydney, Australia. His research interests include Federated Learning, Edge-computing, Information Security and Privacy-preserving.

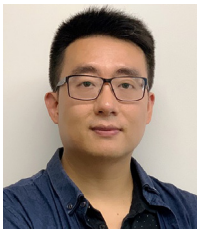
Mengshi Zhang obtained Ph.D. degree in Electrical and Computer Engineering from the University of Texas at Austin in August 2019 and received his BS degree in Electronic Engineering from Tsinghua University in July 2014. His research interests include fault localization, program repair, machine-learning-oriented software engineering and IoT systems.

Guangquan Xu is a Ph.D. and full professor at the Tianjin Key Laboratory of Advanced Networking (TANK), College of Intelligence and Computing, Tianjin University, China. He received his Ph.D. degree from Tianjin University in March 2008. He is an IET Fellow, members of the CCF and IEEE. His research interests include cyber security and trust management.

Alireza Jolfaei received the Ph.D. degree in Applied Cryptography from Griffith University, Gold Coast, Australia. He is the Program Leader of Master of IT in Cyber Security in the Department of Computing at Macquarie University, Sydney, Australia. Alireza has previously worked as an Assistant Professor at Federation University Australia and also at Temple University in Philadelphia, USA. His current research areas include Cyber and Cyber-Physical Systems Security. He has authored over 100 peer-reviewed articles on topics related to cybersecurity. He has received multiple awards for Academic Excellence, University Contribution, and Inclusion and Diversity Support. He received the prestigious IEEE Australian council award for his research paper published in the IEEE Transactions on Information Forensics and Security.

Peizhong Shi He was born in Jiangsu Province, P.R. China, in 1982. He received his PhD degree from the School of Computer Science and Engineering of Southeast University, Nanjing, Jiangsu Province, P.R. China, in 2014. He is currently working at Jiangsu University of Technology, Changzhou, Jiangsu Province, P.R. China. His current research interests include wireless sensor networks, cross-layer design, guarantee of QoS and distributed computing. Currently, he is in charge of the National Natural Science Foundation of China (NSFC).

Chenkai Tan is currently a master student with the Automobile and Traffic Engineering, Jiangsu University of Technology. His current research interest is blockchain-based VANETs application.



Xi Zheng, PhD in Software Engineering from UT Austin USA; Chief Solution Architect for Menulog Australia, now director of Intelligent systems research center (itseg.org), deputy director of software engineering, global engagement, and assistant professor in Software Engineering at Macquarie University. Specialised in Service Computing, IoT Security and Reliability Analysis. Published more than 80 high quality publications in top journals and conferences. Awarded multiple best papers in leading conferences and a few highly competitive funding including ARC LP and Data61 CRP as the Lead Investigator.