On the Statistical Leak of the GGH13 Multilinear Map and some Variants

Léo Ducas, Alice Pellet–Mary

On the Statistical Leak of the GGH13 Multilinear Map and some Variants

Léo Ducas[1⋆] and Alice Pellet--Mary[2⋆⋆]

[1] Cryptology Group, CWI, Amsterdam, The Netherlands
[2] ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, INRIA, UCBL), France.

Abstract. At EUROCRYPT 2013, Garg, Gentry and Halevi proposed a candidate construction (later referred as GGH13) of cryptographic multilinear map (MMap). Despite weaknesses uncovered by Hu and Jia (EUROCRYPT 2016), this candidate is still used for designing obfuscators.
The naive version of the GGH13 scheme was deemed susceptible to averaging attacks, i.e., it could suffer from a statistical leak (yet no precise attack was described). A variant was therefore devised, but it remains heuristic. Recently, to obtain MMaps with low noise and modulus, two variants of this countermeasure were developed by Döttling et al. (EPRINT:2016/599).
In this work, we propose a systematic study of this statistical leak for all these GGH13 variants. In particular, we confirm the weakness of the naive version of GGH13. We also show that, among the two variants proposed by Döttling et al., the so-called conservative method is not so effective: it leaks the same value as the unprotected method. Luckily, the leak is more noisy than in the unprotected method, making the straightforward attack unsuccessful. Additionally, we note that all the other methods also leak values correlated with secrets.
As a conclusion, we propose yet another countermeasure, for which this leak is made unrelated to all secrets. On our way, we also make explicit and tighten the hidden exponents in the size of the parameters, as an effort to assess and improve the efficiency of MMaps.

Keywords: Cryptanalysis, Multilinear Maps, Statistical Leaks, Ideal Lattices.

1 Introduction

Since their introduction in cryptographic constructions by [Jou00], cryptographic bilinear maps, as provided by pairings on elliptic curves, have enabled the construction of more and more advanced cryptographic protocols, starting with the seminal Identity-Based Encryption scheme of Boneh and Franklin [BF01]. More abstractly, a group equipped with an efficient bilinear map, and on which some discrete-logarithm like problems are hard (such as the bilinear-Diffie-Hellmann problem), provides foundation for a whole branch of cryptography. A natural open question is whether it can be generalized to degrees higher than 2 while ensuring hardness of generalizations of the Diffie-Hellmann problem. Such hypothetical objects are referred to as *Cryptographic Multilinear Maps* (or, for short, MMaps).

In 2013, Garg, Gentry and Halevi [GGH13] proposed a candidate construction for MMaps related to ideal-lattices, yet without a clearly identified underlying hard lattice problem. It differs from the pairing case in the sense that elements in the low-level groups have no canonical representation, and

⋆ Supported by a Veni Innovational Research Grant from NWO under project number 639.021.645.
⋆⋆ Supported by an ERC Starting Grant ERC-2013-StG-335086-LATTAC.

that the representation is noisy. Yet, these differences are not too problematic on the functionality front.

On the security front, it rapidly turned out that this construction was insecure, at least in its original set-up. In particular, the natural one-round k-partite protocol based on this MMap was broken by the zeroizing attack of Hu and Jia [HJ16]: this construction fails to securely mimic the tripartite protocol of [Jou00]. More generally, the mere knowledge of a non-trivial representative of 0 tends to make constructions based on this MMap insecure. Orthogonally, it has been discovered that solving over-stretched versions of the NTRU problem (whose intractability is necessary for the security of the GGH MMap) was significantly easier than previously thought, due to the presence of an unusually dense sublattice [ABD16, CJL16, KF17], yet this can be compensated at the cost of increasing parameters. Also, due to recent algorithms for the Principal Ideal Problem [BS16, BEF+17] and Short generator recovery [CGS14, CDPR16], the GGH MMap can be broken[3] in quantum polynomial time, and classical subexponential time $\exp(\tilde{O}(\sqrt{n}))$, where n is the dimension of the used ring.

Nevertheless, this candidate MMap was still considered in a weaker form,[4] to attempt realizing a cryptographic Grail, namely, indistinguishability obfuscation (or, for short, iO). Several iO candidates were broken by attacks that managed to build low-level encodings of zero even if no such encodings were directly given (this is referred to as zeroizing attacks, see e.g. [CGH17]). To try to capture and prevent such attacks, a Weak MMap model was devised in [MSZ16, GMM+16].

Some iO constructions come with a security proof based on assumptions in the standard model [Lin16, Lin17, AS16], but cannot be securely instantiated with the GGH13 MMap as they require low-level encodings of 0. Others are proved secure in a non-standard model (the Generic MMap model [BGK+14, BR14] or the Weak MMap Model [GMM+16, DGG+16]). These models remain not fully satisfactory, as they imply Virtual-Black-Box Obfuscation [BR14, GMM+16], a provably impossible primitive [BGI+01]. The latest candidate of Lin and Tessaro [LT17] did escape these pitfalls by relying on pairings, but it required special Pseudo-Random Generators that were rapidly proved not to exist [LV17, BBKK17].

Statistical leaks in lattice-based cryptography. Early signature schemes based on lattices [GGH97, HPS01, HHGP+03] suffered from statistical leaks, which led to devastating attacks [GS02, NR06]. Those leaks can be fixed in a provably secure way using a *Gaussian Sampling* algorithm from Klein [Kle00], as proven in [GPV08]: the samples available to the adversary are made statistically independent from the secret key.

Similar leaks are a worry in the original construction of [GGH13], and therefore, a candidate countermeasure was developed, making use of Klein's sampling procedure. Nevertheless, no formal statement was made on what this countermeasure prevents: the countermeasure is heuristic. This particular countermeasure turned out to be a hassle when considering variants of the original scheme, as done in [DGG+16], which aims at reaching polynomially small errors and modulus — aiming at improving both efficiency and security of the GGH map, especially in the light of the dense sublattice attacks [ABD16, CJL16, KF17]. Two modified versions of [GGH13] are proposed in [DGG+16], a so-called conservative one, leading to quite efficient parameters, and a so-called aggressive one.

Ideally, one wishes to make provable statements about those four variants, as done in other contexts [GPV08]. Unfortunately, in the context of MMaps, it is not even clear what the statement

[3] The secret value h can be recovered exactly, allowing in particular to construct zero-tester at larger levels.
[4] Without providing any low-level encoding of 0, and keeping the order of the multilinear group secret.

should exactly be. The next best guarantee is a precise understanding of what can be done from a cryptanalytic point of view, as initiated in [GGH13].

The analysis of the leak of [GGH13] focuses on the covariance of products of encodings of zero. One can (informally) argue that this analysis captures all the information of the leak. Indeed, up to discretization, such a product is the product of several centered Gaussian distributions (non necessary spherical), and such a distribution is fully identified by its covariance. The countermeasure proposed in Section 6.4 of [GGH13] attempts to make this covariance proportional to the identity matrix (and therefore unrelated to all secrets) by sampling each element of the product according to a spherical distribution, that is a distribution whose covariance is proportional to the identity matrix. As we shall see, this attempt is unsuccessful, as one of the factor of the product (namely, the one related to the zero-testing parameter) is fixed. Obtaining several independent multiples of it, with covariance proportional to the identity matrix, then reveals an approximation of this factor.

Contributions. Our main contribution is to give a systematic study of the statistical leak in the GGH13 scheme and its variants, in a simple framework we define. We first suggest a common formalism that encompasses all the variants at hand, by parametrising the sampling procedure for encodings by an arbitrary covariance matrix. Following the nomenclature of [GGH13, DGG+16], except for the second one that had no clear name, we consider:

1. The simplistic method: the GGH MMap without countermeasure [GGH13, Sec. 4.1]. This method was only given for simplicity of exposition and was already highly suspected to be insecure;
2. The exponential method:[5] the GGH MMap with countermeasure [GGH13, Sec. 6.4];
3. The conservative method, proposed in [DGG+16] —which we partly revisit to tackle some of its limitations;
4. The aggressive method, proposed in [DGG+16] —we note that this method is specific to the iO construction of [DGG+16], and is not applicable to all constructions over the GGH MMap.

In order to formalize our study of the leak, we propose a simple setting of the GGH multilinear map. Indeed, due to the attacks in presence of encodings of zero, the exact set-up for the analysis of the leak in [GGH13] is not relevant anymore. We adjust their setting to not provide low-level encodings of zero directly. Still, some relations between encodings are needed for the MMaps to be non-trivial; to ensure that those relations do not allow zeroizing attacks, we provide a security proof in the weak multilinear map model of [MSZ16, GMM+16, DGG+16]. For ease of exposure, we restrict ourselves to degree $\kappa = 2$, yet our analysis easily extends to higher degrees.

Using this framework, we are able to analyse a particular averaging attack against the GGH multilinear map. On the one hand, our analysis shows that Method 3 leads to the same leak as Method 1. We also prove that with Method 1, a polynomial-time attack can be mounted using the leak. Interestingly, it does not require the Gentry-Szydlo algorithm [GS02], unlike the approach discussed in [GGH13, Sec. 6.3.2 and Sec. 7.6]. Nevertheless, we did not manage to extend the attack to Method 3: while the same quantity is statistically leaked, the number of samples remains too low for the attack to go through completely. On the other hand, we show that the statistical leak of Method 4 is similar to the one of Method 2: perhaps surprisingly the aggressive method seems more secure than the conservative one.

[5] The naming reflects the fact that this method leads to a modulus q which is exponential in the number ℓ of so-called *atoms*.

Finally, having built a better understanding of which information is leaked, we devise a countermeasure that we deem more adequate than all the above:

5. The compensation method.

This method is arguably simpler, and provides better parameters. More importantly, applying the same leakage attack than above, one only obtains a distribution whose covariance is independent of all secrets. We wish to clarify that this is in no way a formal statement of zero-knowledgedness. The statistical attacks considered in this work are set up in a minimalistic setting, and extensions could exist beyond this minimalistic setting. For example, one could explore what can be done by varying the zero-tested polynomial, or by keeping certain encodings fixed between several successful zero-tests.

As a secondary contribution, we also make explicit and tighten many hidden constants present in the previous constructions, in an effort to evaluate and improve the efficiency of GGH13-like MMaps.

Impact. This result may be useful in pursuit of an underlying hard problem on which one could based the GGH multilinear map. Indeed, we show here that it is possible to recover some information about secret elements, for all the previously proposed sampling methods. Hence, an underlying hard problem (or the security reduction) should capture this leak. This enables us to get a bit more insight into what could be (or could not be) an underlying hard problem for the GGH map. In that regard, finding such a hard underlying problem could be easier with our new method, since one specific leak has been sealed. Again, we *do not* claim that no other leaks exist.

Further, our analysis shows that the weak multilinear map model does not capture averaging attacks. This is not surprising, as the weak multilinear map model only allows to evaluate polynomials in the post-zero-test values, while we need to average on them for this attack. But proving that averaging cannot be achieved by evaluating polynomials is not so immediate. Interestingly, our results prove it. Indeed, using averaging techniques, we were able to mount a polynomial time attack against our setting when using the simplistic sampling method (Method 1), but we also proved that in the weak multilinear map model, no polynomial time attacks could be mounted. This proves that the weak multilinear map model does not capture averaging attacks.[6]

Finally, our new method severely decreases the length of encodings in the GGH13 multilinear map, which substentially contribute to their practical feasibility.

Outline of the article. In Section 2, we recall some mathematical background about cyclotomic number fields and statistics. We also describe the GGH multilinear map and precise the size of its parameters. In Section 3, we describe different sampling methods for the GGH multilinear map, which come from [GGH13] and [DGG+16], using a common formalism so as to factor the later analysis. We describe our simple setting and analyse the leak in Section 4. The security proof of this simple setting in the weak multilinear map model is postponed in Appendix B. Finally, we discuss the design of sampling methods in Section 5, and propose a design we deem more rational.

Acknowledgments. The authors are grateful to Alex Davidson, Nico Döttling and Damien Stehlé for helpful discussions.

[6] The precise component of the attack which is not captured by the weak multilinear map model is the rounding operation performed at the end.

2 Preliminaries

2.1 Mathematical Background

Rings. We denote by R the ring of integers $\mathbb{Z}[X]/(X^n+1)$ for some n which is a power of 2 and $K = \mathbb{Q}[X]/(X^n+1)$ its fraction field. We denote by $\sigma_j : K \to \mathbb{C}$, with $1 \leq j \leq n$, the complex embeddings of K in $\mathbb{C}$. We also denote $K_{\mathbb{R}} = \mathbb{R}[X]/(X^n+1)$ the topological closure of K. For $x \in K_{\mathbb{R}}$, we denote $x_i \in \mathbb{R}$ its i-th coefficient, so that $x = \sum_{i=0}^{n-1} x_i X^i$. For $g \in K$ (or even $K_{\mathbb{R}}$) we denote gR the ideal generated by g: $gR = \{gx | x \in R\}$. The complex conjugation over R and K is denoted $\bar{\cdot}$. It is the automorphism of R sending X to X^{-1}. We denote S the subring of $K_{\mathbb{R}}$ of symmetric elements, that is $S = \{x \in K_{\mathbb{R}} | x = \bar{x}\}$. We set S^+ the subset of symmetric positive elements of S, defined by $S^+ = \{x\bar{x} | x \in K_{\mathbb{R}}\}$. Alternatively, S is the completion of the real subfield of K, and S^+ is (the completion of) the set of elements of K whose embeddings are all non-negative reals. Note that S^+ is closed under addition, multiplication, division, but not under subtraction. The elements of S^+ also admit one and exactly one square root (resp. k-th root) in S^+, which we denote $\sqrt{\cdot}$ (resp. $\sqrt[k]{\cdot}$) . Finally, we call $x\bar{x} \in S^+$ the autocorrelation[7] of $x \in K_{\mathbb{R}}$, and note it $A(x)$. For $\Sigma \in S^+$ it holds that $A(\sqrt{\Sigma}) = \Sigma$. We also define equivalence over S^+ up to scaling by reals, and write $x \sim y$ for invertible elements $x, y \in S^+$ if $x = \alpha y$ for some positive real $\alpha > 0$. Let q be a prime congruent to 1 modulo $2n$. We denote by R_q the quotient ring $R/(qR)$. For $x \in R$, we denote by $[x]_q$ (or $[x]$ when there is no ambiguity) the coset of the element x in R_q. We will often lift back elements from R_q to R, in which case we may implicitly mean that we choose the representative with coefficients in the range $[-q/2, q/2]$. To avoid confusion, we will always note x^{-1} for the inversion in R_q, and keep the fraction symbols $1/x$ and $\frac{1}{x}$ for inversion in K and $K_{\mathbb{R}}$.

Geometry. Because we work in the ring $\mathbb{Z}[X]/(X^n+1)$, the canonical geometry of the coefficients embeddings is equivalent, up to scaling, to the geometry of the Minkowski embeddings. We stick with the former, following the literature on multilinear maps. More precisely, the inner product of two elements $x, y \in K$ is defined by $\langle x, y \rangle = \sum x_i y_i$. The Euclidean norm (or ℓ_2-norm) is defined by $\|x\| = \langle x, x \rangle$. The ℓ_∞-norm is noted $\|x\|_\infty = \max |x_i|$.

We recall the following inequalities:

$$\|xy\| \leq \sqrt{n} \cdot \|x\| \cdot \|y\| \tag{1}$$

$$\|x\|_\infty \leq \|x\| \leq \sqrt{n} \cdot \|x\|_\infty \tag{2}$$

$$\|x\|^2 \leq \|x\bar{x}\|_\infty \tag{3}$$

$$\|\bar{x}\| = \|x\| \text{ and } \|\bar{x}\|_\infty = \|x\|_\infty. \tag{4}$$

Statistics. We denote by $\Pr[E]$ the probability of an event E. For a random variable x over $K_{\mathbb{R}}$, we denote by $\mathbb{E}[x]$ the expectation of x, and by $\mathbb{V}[x] = \mathbb{E}[x\bar{x}] - \mathbb{E}[x]\mathbb{E}[\bar{x}]$ its variance. It should be noted that $\mathbb{V}[x] \in S^+$ for any random variable x over $K_{\mathbb{R}}$. A random variable x is said centered if $\mathbb{E}[x] = 0$, and isotropic if $\mathbb{V}[x] \sim 1$. We recall Hoeffding's inequality.

Theorem 1 (Hoeffding's inequality). *Let $Y_1, \cdots, Y_m$ be independent random variables in $\mathbb{R}$ with the same mean $\mu \in \mathbb{R}$ and such that $|Y_i| \leq B$ for all i's. Then for all $t > 0$,*

$$Pr\left[\left|\frac{1}{m}\sum_{i=0}^{m} Y_i - \mu\right| \geq t\right] < e^{-\frac{2mt^2}{B^2}}.$$

[7] In an algebraic context, this would be more naturally described as the norm of x relative to the maximal real subfield of K, yet for our purposes it is more adequate to use the vocabulary of statistics.

Hoeffding's inequality, as given above, applies to random variables in $\mathbb{R}$. In this article, we will be interested in random variables in R. We will then see our elements in R as vectors in $\mathbb{R}^n$ and apply Hoeffding's inequality coefficient-wise.

Corollary 1 (Hoeffding's inequality in R). *Let $Y_1, \cdots, Y_m$ be independent random variables in R with the same mean $\mu \in K_{\mathbb{R}}$ and such that $\|Y_i\|_\infty \leq B$ for all i's. Let $\varepsilon > 0$, then*

$$Pr\left[\left\|\frac{1}{m}\sum_{i=0}^{m} Y_i - \mu\right\|_\infty \geq B\sqrt{\frac{\ln n - \ln \varepsilon}{2m}}\right] < \varepsilon.$$

Proof. For $1 \leq i \leq m$ and $0 \leq j \leq n-1$, define $Y_{i,j}$ to be the j-th coefficient of the variable $Y_i \in R$ and μ_j to be the j-th coefficient of μ. For a fixed j, the variables $Y_{i,j}$ (where only i varies) are independent random variables in $\mathbb{R}$ of mean μ_j. Moreover, as $\|Y_i\|_\infty \leq B$ for all i's, the coefficients $Y_{i,j}$ are also bounded by B. We can then apply Hoeffding's inequality (Theorem 1) to them. We obtain

$$\begin{aligned}
&\Pr\left[\left\|\frac{1}{m}\sum_{i=0}^{m} Y_i - \mu\right\|_\infty \geq B\sqrt{\frac{\ln n - \ln \varepsilon}{2m}}\right] \\
&= \Pr\left[\exists j : \left|\frac{1}{m}\sum_{i=0}^{m} Y_{i,j} - \mu_j\right|_\infty \geq B\sqrt{\frac{\ln n - \ln \varepsilon}{2m}}\right] \\
&\leq \sum_{j=0}^{n-1} \Pr\left[\left|\frac{1}{m}\sum_{i=0}^{m} Y_{i,j} - \mu_j\right|_\infty \geq B\sqrt{\frac{\ln n - \ln \varepsilon}{2m}}\right] \\
&< \sum_{j=0}^{n-1} e^{-\frac{2mB^2(\ln n - \ln \varepsilon)}{2B^2 m}} = \sum_{j=0}^{n-1} \frac{\varepsilon}{n} = \varepsilon.
\end{aligned}$$

We used the union bound and Hoeffding's inequality with $t = B\sqrt{\frac{\ln n - \ln \varepsilon}{2m}}$. □

Discrete Gaussians. For $\Sigma \in S^+$ and $x_0 \in K_{\mathbb{R}}$, we define the *Gaussian weight function* on $K_{\mathbb{R}}$ as

$$\rho_{\sqrt{\Sigma},x_0} : x \mapsto \exp\left(-\frac{1}{2}\left\|\frac{x - x_0}{\sqrt{\Sigma}}\right\|^2\right).$$

For any shifted ideal $I + c$, $I \subset K$, $c \in K_{\mathbb{R}}$, we define the *discrete Gaussian distribution* over $I + c$ of parameter $\sqrt{\Sigma}$, centered in x_0 by:

$$\forall x \in I + c,\ D_{I+c,\sqrt{\Sigma},x_0}(x) = \frac{\rho_{\sqrt{\Sigma},x_0}(x)}{\rho_{\sqrt{\Sigma},x_0}(I + c)}.$$

For concision, we write $D_{I+c,\sqrt{\Sigma}}$ instead of $D_{I+c,\sqrt{\Sigma},0}$ and $\rho_{\sqrt{\Sigma}}$ instead of $\rho_{\sqrt{\Sigma},0}$.

Theorem 2 (Reformulation of [GPV08, Thm 4.1.]). *There exists a PPT algorithm that given $g \in R$ and a parameter Σ such that $\|g/\sqrt{\Sigma}\| \leq o(1/\sqrt{\log n})$, outputs x from a distribution negligibly close to $D_{gR+c,\sqrt{\Sigma}}$.*

This reformulation simply relies on the identity $D_{gR+c,\sqrt{\Sigma}} = \frac{\sqrt{\Sigma}}{\sigma} \cdot D_{(gR+c)/\sqrt{\Sigma},\sigma}$. We also recall that, above the smoothing parameter [MR04], a discrete Gaussian resembles the continuous Gaussian, in particular it is almost centered at 0, and of variance almost Σ.

Lemma 1. *For any $g \in K$, $\Sigma \in S^+$, $c \in K_{\mathbb{R}}$ such that $\|g/\sqrt{\Sigma}\| \leq o(1/\sqrt{\log n})$, if $x \leftarrow D_{gR+c,\sqrt{\Sigma}}$, then $\|\mathbb{E}[x]\| \leq \varepsilon \cdot \|\sqrt{\Sigma}\|$ and $\|\mathbb{V}[x] - \Sigma\| \leq \varepsilon \cdot \|\Sigma\|$ for some negligible function $\varepsilon(n)$.*

The proof of this result, using [MR04, Lemma 4.2], can be found in Appendix A.

2.2 The GGH13 multilinear map

We describe in this section the GGH13 multilinear map [GGH13], in its asymmetric setting. The GGH13 multilinear map encodes elements of a ring of integers R, modulo a secret small element $g \in R$. More concretely, an authority generates the following parameters:

- an integer n which is a power of 2 (serving as the security parameter).
- a (small) element g in R. We note $I = gR$ the ideal generated by g in R.
- a (large) positive integer q such that $q \equiv 1 \mod 2n$. Originally, q was chosen exponentially large in n [GGH13], but variants were proposed for polynomially sized q [LSS14, DGG+16].
- ℓ invertible elements $[z_i] \in R_q^\times$, for $1 \leq i \leq \ell$, chosen uniformly at random in $R_q^\times$.
- a zero-testing parameter $[p_{zt}] = [hz^*g^{-1}]$ where $[z^*] = [\prod_{1 \leq i \leq \ell} z_i]$ and h is a random element in R, generated according to a Gaussian distribution of standard deviation approximately $\sqrt{q}$.

We detail in Section 2.2 the size of the parameters described above (we will chose them to ensure the correctness of the scheme). The elements n, q and p_{zt} are public while the parameters h, g and the z_i's are kept secret.

Encoding of an element. The GGH13 multilinear map allows to encode cosets of the form $a + I$ for some element a in R. Let $\boldsymbol{v} \in \{0,1\}^\ell$ be a vector of size ℓ. An encoding of the coset $a + I$ at level $\boldsymbol{v}$ is an element of R_q of the form

$$u = \left[(a + rg) \cdot z_{\boldsymbol{v}}^{-1}\right]$$

where $[z_{\boldsymbol{v}}] = [\prod_{i,\boldsymbol{v}[i]=1} z_i]$ and $a + rg$ is a small element in the coset $a + I$. We call $\boldsymbol{v}$ the level of the encoding. We abuse notation by saying that u is an encoding of a (instead of an encoding of the coset $a + I$).

An encoding generated by the authority is called a fresh encoding, by opposition to encodings that are obtained by adding or multiplying other encodings. The precise distribution of $a + rg$ for a fresh encoding will be a discrete Gaussian distribution over the coset $a + I$, but not necessarily a spherical one: $a + rg \leftarrow D_{a+I,\sqrt{\Sigma_{\boldsymbol{v}}}}$. The shape $\Sigma_{\boldsymbol{v}}$ of this Gaussian is essentially what distinguishes the variants that we will discuss in Section 3.

Adding and multiplying encodings. If u_1 and u_2 are two encodings of elements a_1 and a_2 at the same level $\boldsymbol{v}$ then $u_1 + u_2$ is an encoding of $a_1 + a_2$ at level $\boldsymbol{v}$.

If u_1 and u_2 are two encodings of elements a_1 and a_2 at levels $\boldsymbol{v}$ and $\boldsymbol{w}$ with $\boldsymbol{v}[i] \cdot \boldsymbol{w}[i] = 0$ for all $1 \leq i \leq \ell$, then $u_1 \cdot u_2$ is an encoding of $a_1 \cdot a_2$ at level $\boldsymbol{v} + \boldsymbol{w}$ (where the addition is the usual addition on vectors of size ℓ).

Zero-testing. We denote by $\boldsymbol{v}^* = (1, \ldots, 1)$ the maximum level of an encoding. The zero testing parameter allows us to test if an encoding u at level $\boldsymbol{v}^*$ is an encoding of zero, by computing

$$[w] = [u \cdot p_{zt}].$$

If w is small compared to q (the literature usually requires its coefficients to be less than $q^{3/4}$), then u is an encoding of zero. Otherwise, it is not.

Size of the parameters and correctness. We define Q such that $q = n^Q$ and L such that $\ell = n^L$ (the elements Q and L are not necessarily integers). The bounds below on the size of g and h come from [GGH13]. The secret generator g is sampled so that:

$$\|g\| = O(n), \quad \|1/g\| = O(n^2). \tag{5}$$

Remark. There seems to be some inconsistencies in [GGH13] about the size of g, which is on page 10 sampled with width $\sigma = \tilde{O}(\sqrt{n})$, while on page 13 the width σ is set to $\sqrt{n\lambda}$ to ensure the smoothing condition $\sigma \geq \eta_{2^{-\lambda}}(\mathbb{Z}^n)$ (where $\lambda = O(n)$ denote the security parameter). Yet, according to [MR04, Lemma 3.3], it holds that $\eta_{2^{-\lambda}}(\mathbb{Z}^n) \leq O(\sqrt{\lambda} + \log n)$, so $\sigma = O(\sqrt{n})$ is sufficient, and we do have $\|g\| \leq O(n)$ with overwhelming probability by [MR04, Lemma 4.4].

The numerator $c = a + rg$ of a fresh encoding of $a + I$ at level $\boldsymbol{v}$ is sampled such that

$$\|c\| = \Theta(n^{\gamma + \eta \cdot \|\boldsymbol{v}\|_1 + \nu L}), \tag{6}$$

where γ, η and ν are positive reals, and depend on the sampling method, such as the ones proposed in [DGG+16] (depending on the method, η and ν may be zero). We describe later the different sampling methods and the values of γ, η and ν associated to each method. When we do not need to focus on the dependence on $\|\boldsymbol{v}\|_1$ and L, we just call $E := \Theta(n^{\gamma + \eta \cdot \|\boldsymbol{v}\|_1 + \nu L})$ the bound above. For each sampling method described below, we choose this bound to be as small as possible under the specific constraints that will arise with the sampling method.

The mildly large element h is sampled so that

$$\|h\| = \Theta(\sqrt{nq}). \tag{7}$$

Remark. In the second variant proposed in [GGH13, Section 6.4] to try to prevent averaging attacks, the authors generate h according to a non spherical Gaussian distribution. However, as h is sampled only once, its distribution does not matter for the attack we analyze in this article. This is why we only specify here the size of h, and not its distribution.

In the following, we will be interested in the case where we are provided with fresh encodings at a somewhat high level, and we can create encodings at maximum level $\boldsymbol{v}^*$ by multiplying just a small number of fresh encodings (namely κ). This is what motivates the condition given here to ensure correctness of the zero-testing procedure. Correctness of zero-testing a homogeneous polynomial of degree κ, whose absolute sum of the coefficients is bounded by n^B and evaluated in fresh encodings, is guaranteed if $n^B \cdot \|\frac{h}{g} \prod_{i=1}^{\kappa} c_i\| \leq q^{3/4}$. It is then sufficient to have

$$B + \frac{\kappa + 1}{2} + \frac{Q + 1}{2} + 2 + \kappa(\gamma + \nu L) + \eta\ell \leq \frac{3}{4} Q. \tag{8}$$

The term $\frac{\kappa+1}{2}$ appears from applying inequality (1) $\kappa+1$ times. One should also note that $\sum_{i=1}^{\kappa}\|\boldsymbol{v}_i\|_1 = \|\boldsymbol{v}^*\|_1 = \ell$, because we can only zero test at level $\boldsymbol{v}^*$ (where $\boldsymbol{v}_i$ is the level of encoding c_i). More compactly, correctness holds if:

$$B + 3 + \kappa(1/2 + \gamma + \nu L) + \eta\ell \leq Q/4. \tag{9}$$

In our simple setting of the GGH multilinear map defined in Section 4.1, we will only query the zero-testing procedure on encodings of this form, with $\kappa = 2$ and $B = \log(m)/\log(n)$, for some constant m we will define later. Hence, taking $4 + 2\gamma + 2\nu L + \eta\ell + \log(m)/\log(n) \leq Q/4$ will be sufficient in our setting to ensure correctness of the zero-testing procedure.

Remark. We note that the bound $q^{3/4}$ for positive zero-tests is somewhat arbitrary and could very well be replaced by $q/4$, allowing to square-root the parameter q. Indeed, the probability of a false positive during zero-testing would remain as small as 2^{-n}. This would have a serious impact on concrete efficiency and security.

3 Sampling methods

We describe in this section different sampling methods that can be used to generate the fresh encodings of the GGH multilinear map and we give the values of γ, η and ν that correspond to these methods. As said above, we will be interested in cases where (at least some of) the fresh encodings have a somewhat high degree and we just have to multiply a constant number of them (say 2) to obtain an encoding at maximal level $\boldsymbol{v}^*$. We denote by $\mathcal{A}$ the set of "atoms", that is the set of levels $\boldsymbol{v} \in \{0,1\}^\ell$ at which we want to encode fresh encodings. In our simple setting of the GGH multilinear map (see Section 4.1 for a full description of our setting), we will chose $\mathcal{A}$ to be the set of levels $\boldsymbol{v} \in \{0,1\}^\ell$ that have weight exactly 1 or $\ell-1$, where the weight of $\boldsymbol{v}$ is the number of its non-zero coefficients. For all $\boldsymbol{v} \in \mathcal{A}$, we denote by $\tilde{\boldsymbol{v}} = \boldsymbol{v}^* - \boldsymbol{v}$ the complement of $\boldsymbol{v}$. We note that $\mathcal{A}$ is closed by complement.

In all the following sampling methods except the first one, one chooses a representative $z_{\boldsymbol{v}} \in R$ of $[z_{\boldsymbol{v}}] \in R_q$ for all $\boldsymbol{v} \in \mathcal{A}$. This representative will not necessarily be the canonical one, with coefficients in $[-q/2, q/2]$. Then, we will take $\Sigma_{\boldsymbol{v}} = \sigma_{\boldsymbol{v}}^2 z_{\boldsymbol{v}}\bar{z_{\boldsymbol{v}}}$, with $\sigma_{\boldsymbol{v}} = \Theta(n^2\|1/z_{\boldsymbol{v}}\|)$. Using Inequalities (3) and (4), we can see that $\|1/\sqrt{\Sigma_{\boldsymbol{v}}}\| \leq 1/\sigma_{\boldsymbol{v}} \cdot n^{1/4} \cdot \|1/z_{\boldsymbol{v}}\|$. Hence, with our choice of $\sigma_{\boldsymbol{v}}$ and the fact that $\|g\| = O(n)$, we obtain

$$\left\|\frac{g}{\sqrt{\Sigma_{\boldsymbol{v}}}}\right\| \leq \sqrt{n} \cdot \|g\| \cdot \left\|\frac{1}{\sqrt{\Sigma_{\boldsymbol{v}}}}\right\| = O\left(\frac{1}{n^{1/4}}\right) = o\left(\frac{1}{\sqrt{\log n}}\right).$$

We can therefore apply Theorem 2 to sample the numerators of fresh encodings at level $\boldsymbol{v}$, according to a Gaussian distribution of parameter $\Sigma_{\boldsymbol{v}}$. Using tail-cut of Gaussian distributions, we have that if c is the numerator of a fresh encoding, then $\|c\| \leq n\|\sqrt{\Sigma_{\boldsymbol{v}}}\| \leq n^{1.5}\sigma_{\boldsymbol{v}}\|z_{\boldsymbol{v}}\|$ with overwhelming probability. This means that we can take

$$E \leq \Theta(n^{3.5} \cdot \|1/z_{\boldsymbol{v}}\| \cdot \|z_{\boldsymbol{v}}\|). \tag{10}$$

Hence, in the following methods (except the simplistic one), we will focus on the size of $\|1/z_{\boldsymbol{v}}\| \cdot \|z_{\boldsymbol{v}}\|$ to get a bound on the value of E.

Remark. Inequality (10) above is not tight. We could at least improve it to $E \leq \Theta(n^{3+\varepsilon} \cdot \|1/z_{\boldsymbol{v}}\| \cdot \|z_{\boldsymbol{v}}\|)$ for any $\varepsilon > 0$, with the same reasoning. This ensures statistical closeness to the desired distribution up to $\exp(-n^{2\varepsilon})$. Considering that there are already classical attacks in time $\exp(\tilde{O}(\sqrt{n}))$ (namely, using [CDPR16, BEF+17] to recover h from the ideal hR), one may just choose $\varepsilon = 1/4$.

3.1 The simplistic method

The simplistic method consists in always choosing $\Sigma_{\boldsymbol{v}} \sim 1$, independently of $\boldsymbol{v}$ and $z_{\boldsymbol{v}}$. This is done by applying Klein's algorithm [Kle00], and requires for correctness [GPV08, Thm 4.1] that $\Sigma_{\boldsymbol{v}} = \sigma^2$ for a positive scalar $\sigma \in \mathbb{R}$, where $\sigma \geq \|g\| \cdot \omega(\sqrt{\log n})$. So by taking $\sigma = \Theta(n^{1+\varepsilon})$ with $\varepsilon > 0$, one may have $E = \Theta(\sqrt{n}\sigma) = \Theta(n^{1.5+\varepsilon})$, that is $\gamma = 1.5 + \varepsilon$ and $\eta = \nu = 0$.

This method was deemed subject to averaging attacks and hence less secure than the following one in [GGH13], but the authors claim that their attack attempts failed because all recovered elements were larger that $\sqrt{q}$, and that averaging attacks would need super-polynomially many elements.[8] We explicit an attack, and will show that this attack is possible even for exponential q, as long as E^κ remains polynomial: in other words, the presence of the mildly large factor h (of size $\sqrt{q}$) can be circumvented.

3.2 The exponential method

We present here the countermeasure of [GGH13, Sec. 6.4], generalized to multi-dimensional universe, as done in [DGG+16, Sec. 2.1]. For $1 \leq i \leq \ell$, set z_i to be the canonical representative of $[z_i]$ in R (with coefficients in the range $[-q/2, q/2]$). Using rejection sampling when choosing z_i, assume that $\|z_i\| \cdot \|1/z_i\| \leq Z$; this is efficient for Z as small as $n^{5/2}$ using [DGG+16], and can even be improved to $Z = n^{3/2}$ using Lemma 3 below and its corollary.

For $\boldsymbol{v}$ in $\mathcal{A}$, set $z_{\boldsymbol{v}} = \prod z_i^{v_i}$ over R. Recall that Inequality (10) gives us: $E \leq \Theta(n^{3.5}\|1/z_{\boldsymbol{v}}\| \cdot \|z_{\boldsymbol{v}}\|)$. But we have $\|z_{\boldsymbol{v}}\| \leq n^{(\|v\|_1-1)/2} \prod_{i \in \boldsymbol{v}} \|z_i\|$ and $\|1/z_{\boldsymbol{v}}\| \leq n^{(\|v\|_1-1)/2} \prod_{i \in \boldsymbol{v}} \|1/z_i\|$. Hence we can take

$$E = \Theta(n^{2.5+\|v\|_1} \cdot Z^{\|v\|_1}) = \Theta(n^{2.5+2.5\|v\|_1}).$$

This means that we have $\gamma = 2.5, \eta = 2.5$ and $\nu = 0$.

Correctness is guaranteed for $q \geq n^{\Omega(\ell)}$ (because $\eta \neq 0$), and because ℓ is much larger than the constant degree κ in [DGG+16], this is not a satisfying solution, as we aim at decreasing q to polynomial. Two alternatives (conservative and aggressive) are therefore developed in [DGG+16].

3.3 The conservative method [DGG+16]

The first alternative suggested is to do as above, but reducing the $z_{\boldsymbol{v}}$ modulo q, that is, set $z_{\boldsymbol{v}}$ to be the representative of $[\prod z_i^{v_i}]$ with coefficients in $[-q/2, q/2]$. One then ensures, by rejection of all the z_i's together, that $\|z_{\boldsymbol{v}}\| \cdot \|1/z_{\boldsymbol{v}}\| \leq n^{2.5}$ for all $\boldsymbol{v} \in \mathcal{A}$. This leads to $E = \Theta(n^{3.5} \cdot n^{2.5}) = \Theta(n^6)$ (i.e. $\gamma = 6,\ \eta = \nu = 0$) and therefore allows correctness for q as small as $n^{O(\kappa)}$, which is polynomial for constant degree κ.

Using [DGG+16, Lemma 8] restated below, the authors conclude that this method is quite inefficient because for the above bound to hold simultaneously for all $\boldsymbol{v} \in \mathcal{A}$ with good probability requires increasing n together with ℓ. Indeed, using Lemma 2, we can bound the probability that one of the $z_{\boldsymbol{v}}$ does not satisfy $\|z_{\boldsymbol{v}}\| \cdot \|1/z_{\boldsymbol{v}}\| \leq n^{2.5}$ by $2|\mathcal{A}|/n = 4\ell/n$. So if we want this probability to be small (say less than $1/2$) in order for the sampling procedure to be efficient, we should increase n with ℓ.

[8] Recall that the original proposal was setting E and therefore q to be super-polynomial even for bounded degree ℓ because of the drowning technique for publicly sampling encodings. Since then, attacks using encodings of zero [HJ16, CGH+15, MSZ16] have restricted encodings to be private, allowing polynomially large E.

Lemma 2 (Lemma 8 from [DGG+16]). *Let $[z]$ be chosen uniformly at random in R_q and z be its canonical representative in R (i.e. with coefficients in $[-q/2, q/2]$). Then it holds that*

$$Pr\left[\|1/z\| \geq n^2/q\right] \leq 2/n.$$

In the following section, we revisit the conservative method by generalizing this lemma.

3.4 The conservative method revisited

In the following lemma, we introduce an extra degree of freedom c compared to the lemma of [DGG+16], but also improve the upper bound from $O(n^{1-c})$ to $O(n^{1-2c})$.

Lemma 3. *Let $[z]$ be chosen uniformly at random in R_q and z be its representative with coefficients between $-q/2$ and $q/2$. Then, for any $c \geq 1$, it holds that*

$$Pr\left[z = 0 \vee \|1/z\| \geq n^c/q\right] \leq 4/n^{2c-1}.$$

Corollary 2. *Let $[z]$ be chosen uniformly at random in $R_q^\times$ and z be its representative with coefficients between $-q/2$ and $q/2$. Then, for any $c \geq 1$, it holds that*

$$Pr\left[\|1/z\| \geq n^c/q\right] \leq 8/n^{2c-1}.$$

We can use this corollary to compute the probability that one of the $z_{\boldsymbol{v}}$ does not satisfy $\|1/z_{\boldsymbol{v}}\| \leq n^c/q$ when the $[z_i]$'s are independent and chosen uniformly at random in $R_q^\times$. Indeed, the $[z_{\boldsymbol{v}}]$'s are uniform in $R_q^\times$ because they are product of uniform invertible elements, and, by union bound, we have

$$\begin{aligned}\Pr\left[\exists \boldsymbol{v} \in \mathcal{A} \text{ s.t. } \|1/z_{\boldsymbol{v}}\| > n^c/q\right] &\leq \sum_{\boldsymbol{v}\in\mathcal{A}} \Pr\left[\|1/z_{\boldsymbol{v}}\| > n^c/q\right] \\ &\leq \frac{8|\mathcal{A}|}{n^{2c-1}}.\end{aligned}$$

If we want this probability to be less than $1/2$, in order to re-sample all the z_i's only twice on average, we should take

$$|\mathcal{A}| \leq \frac{n^{2c-1}}{16}. \tag{11}$$

But we also have $\|z_{\boldsymbol{v}}\| \leq \sqrt{n}\|z_{\boldsymbol{v}}\|_\infty \leq \sqrt{n}q$, hence $\|1/z_{\boldsymbol{v}}\| \cdot \|z_{\boldsymbol{v}}\| \leq n^{c+0.5}$. In order to minimize E, we wish to minimize c, under (11). By taking the minimal value of c that satisfies this constraint, and recalling that $|\mathcal{A}| = 2\ell$, we obtain

$$E = \Theta(n^{4.5+L/2}).$$

This means that $\gamma = 4.5$, $\nu = 0.5$ and $\eta = 0$. This conservative method revisited is the same as the original one, except that we improve on the encodings size bound E.[9] In the following, we will then only focus on the conservative method revisited and not on the original one.

[9] We also change a bit the point of view by fixing n first and then obtaining an upper bound on ℓ (which will appear because $\nu \neq 0$ in E), while the authors of [DGG+16] first fix ℓ and then increase n consequently.

Proof (Proof of Lemma 3). The proof of this lemma uses the same ideas as the one of [SS13, Lemma 4.1], but here, the element z is sampled uniformly modulo q instead of according to a Gaussian distribution. Let $[z]$ be chosen uniformly at random in R_q and z be its representative with coefficients between $-q/2$ and $q/2$. Recall that we denote $\sigma_j : K \to \mathbb{C}$ the complex embeddings of K in $\mathbb{C}$, with $1 \leq j \leq n$. We know that the size of z is related to the size of its embeddings. Hence, if we have an upper bound on the $|\sigma_j(1/z)|$, we also have an upper bound on $\|1/z\|$. Moreover, the σ_j's are morphisms, so $\sigma_j(1/z) = 1/\sigma_j(z)$, and it suffices to have a lower bound on $|\sigma_j(z)|$.

Let $j \in \{1, \cdots, n\}$, there exists a primitive $2n$-th root of unity ζ such that

$$\sigma_j(z) = \sum_{i=0}^{n-1} a_i \zeta^i,$$

where the a_i's are the coefficients of z, and so are sampled uniformly and independently between $-q/2$ and $q/2$. As ζ is a primitive 2^k-th root of unity for some k, there exists i_0 such that $\zeta^{i_0} = I$, where I is a complex square root of -1. So we can write

$$\sigma_j(z) = a_0 + I a_{i_0} + \tilde{z},$$

for some $\tilde{z} \in \mathbb{C}$ that is independent of a_0 and a_{i_0}. Now, we have that

$$\begin{aligned}
\Pr\left[|\sigma_j(z)| < \frac{q}{n^c}\right] &= \Pr\left[a_0 + I a_{i_0} \in B(-\tilde{z}, \frac{q}{n^c})\right] \\
&\leq \frac{\mathrm{Vol}(B(-\tilde{z}, \frac{q}{n^c}))}{q^2} \\
&\leq \frac{4}{n^{2c}},
\end{aligned}$$

where $B(-\tilde{z}, q/n^c)$ is the ball centered in $-\tilde{z}$ of radius q/n^c. A union bound yields that

$$\Pr\left[\exists j,\ |\sigma_j(z)| < \frac{q}{n^c}\right] \leq n \cdot \frac{4}{n^{2c}} = \frac{4}{n^{2c-1}}.$$

Which in turns implies

$$\Pr\left[\forall j,\ \left|\sigma_j\left(\frac{1}{z}\right)\right| \leq \frac{n^c}{q}\right] \geq 1 - \frac{4}{n^{2c-1}}.$$

To complete the proof, we use the fact that for cyclotomic fields of power-of-two order, we have $\|1/z\| \leq \max_j(|\sigma_j(1/z)|)$. This gives the desired result. □

Proof (Proof of Corollary 2). First, note that sampling $[z]$ uniformly in $R_q^\times$ is the same as sampling $[z]$ uniformly in R_q and re-sampling it until $[z]$ is invertible. We denote by $U(R_q)$ (resp. $U(R_q^\times)$) the uniform distribution in R_q (resp. $R_q^\times$). We then have that

$$\Pr_{[z] \leftarrow U(R_q^\times)}[\|1/z\| \geq n^c/q] = \Pr_{[z] \leftarrow U(R_q)}[\|1/z\| \geq n^c/q \mid [z] \in R_q^\times].$$

But using the definition of conditional probabilities, we can rewrite

$$\Pr_{[z] \leftarrow U(R_q)}[\|1/z\| \geq n^c/q \mid [z] \in R_q^\times] = \frac{\Pr_{[z] \leftarrow U(R_q)}[[z] \in R_q^\times \text{ and } \|1/z\| \geq n^c/q]}{\Pr_{[z] \leftarrow U(R_q)}[[z] \in R_q^\times]}.$$

The numerator of this fraction is less than $\Pr_{[z]\leftarrow U(R_q)}[\|1/z\| \geq n^c/q]$, which is less than $\frac{4}{n^{2c-1}}$ using Lemma 3. And at least half of the elements of R_q are invertible (if q is prime, we can even say that the proportion of non invertible elements is at most n/q, because $q \equiv 1 \mod 2n$). Hence, $\Pr_{[z]\leftarrow U(R_q)}[[z] \in R_q^\times] \geq 1/2$ and we obtain the desired result

$$\Pr_{[z]\leftarrow U(R_q^\times)}[\|1/z\| \geq n^c/q] \leq \frac{8}{n^{2c-1}}.$$

□

3.5 The aggressive method

This aggressive method was proposed by Döttling et al. in [DGG+16] in order to instantiate the GGH multilinear map for their obfuscator. This method cannot be used for any set of atoms $\mathcal{A}$, as it relies on the fact that the levels at which we encode fresh encodings have a specific structure. Indeed, for each $\boldsymbol{v} \in \mathcal{A}$, we have either $[z_{\boldsymbol{v}}] = [z_i]$ for some $i \in \{1, \cdots, \ell\}$ or $[z_{\boldsymbol{v}}] = [z^* \cdot z_i^{-1}]$. Using this remark, the secret $[z_i]$'s are generated in the following way.

For i from 1 to ℓ do:

- *sample a uniformly random invertible element $[z_i]$ in R_q. Let z_i be the representative of $[z_i]$ in R with coefficients between $-q/2$ and $q/2$, and $\widetilde{z_i}$ be the representative of $[z_i^{-1}]$ in R with coefficients between $-q/2$ and $q/2$.*
- *until both following conditions are satisfied, re-sample $[z_i]$:*

$$\|1/z_i\| \leq n^3/q \tag{12}$$
$$\|1/\widetilde{z_i}\| \leq n/q. \tag{13}$$

- *if $i = \ell$, we also re-sample $[z_i]$ until this third condition is met*

$$\|1/z^*\| \leq n/q, \tag{14}$$

where z^* is the representative of $[\prod_{1\leq i\leq \ell} z_i]$ with its coefficients between $-q/2$ and $q/2$.

Remark. As we sample the $[z_i]$'s from $i = 1$ to ℓ, when we generate $[z_\ell]$ all other $[z_i]$'s are already fixed, so we can define $[z^*]$.

Note that with this method, we re-sample each z_i an expected constant number of times, independently of ℓ. Indeed, all $[z_i]$'s for $i \leq \ell - 1$ are sampled independently. And the two conditions we want are satisfied except with probability at most $\frac{8}{n}$ for each condition (using Corollary 2 with $[z_i]$ and $[z_i^{-1}]$ that are uniform in $R_q^\times$ and with $c = 3$ or $c = 1$). So, applying a union bound, the probability that we have to re-sample $[z_i]$ is at most $\frac{16}{n}$, which is less than $1/2$ if $n \geq 32$. The idea is the same for $[z_\ell]$ except that we also want $\|1/z^*\|$ to be small. But all $[z_i]$ for $i < \ell$ are already fixed, so $[z^*]$ only depends on $[z_\ell]$ and is uniform in $R_q^\times$. Hence this last condition is also satisfied except with probability $\frac{8}{n}$ from Corollary 2. And the probability that the three conditions are met for $[z_\ell]$ is at least $1/2$ as long as $n \geq 48$.

To conclude, if $n \geq 48$, the procedure described above will sample each $[z_i]$ at most twice in average, independently of the choice of ℓ. So we can choose ℓ arbitrarily large and the sampling procedure will take time $O(\ell) \cdot \text{poly}(n)$.

It remains to choose our representative $z_{\boldsymbol{v}} \in R$ of $[z_{\boldsymbol{v}}] \in R_q$ and to get a bound on $\|1/z_{\boldsymbol{v}}\| \cdot \|z_{\boldsymbol{v}}\|$ for all $\boldsymbol{v} \in \mathcal{A}$, in order to get the value of E. We will show that $\|z_{\boldsymbol{v}}\| \cdot \|1/z_{\boldsymbol{v}}\| \leq n^4$ for some choice of the representative $z_{\boldsymbol{v}}$ we detail below.

First case. If $\boldsymbol{v}$ has weight 1, that is $[z_{\boldsymbol{v}}] = [z_i]$ for some i, then we take $z_{\boldsymbol{v}} = z_i$. With our choice of $[z_i]$, we have that $\|1/z_{\boldsymbol{v}}\| \leq n^3/q$. And as $\|z_{\boldsymbol{v}}\|$ has its coefficients between $-q/2$ and $q/2$ we have that $\|z_{\boldsymbol{v}}\| \leq \sqrt{n}q$ and hence $\|z_{\boldsymbol{v}}\| \cdot \|1/z_{\boldsymbol{v}}\| \leq n^{3.5} \leq n^4$.

Second case. If $\boldsymbol{v}$ has weight $\ell - 1$, then there exists $i \in \{1, \cdots, \ell\}$ such that $[z_{\boldsymbol{v}}] = [z^* \cdot z_i^{-1}]$. We choose as a representative of $[z_{\boldsymbol{v}}]$ the element $z_{\boldsymbol{v}} = z^* \cdot \widetilde{z_i} \in R$, with z^* and $\widetilde{z_i}$ as above (with coefficients between $-q/2$ and $q/2$). We then have

$$\|1/z_{\boldsymbol{v}}\| = \|1/z^* \cdot 1/\widetilde{z_i}\| \leq \sqrt{n} \cdot \|1/z^*\| \cdot \|1/\widetilde{z_i}\| \leq n^{2.5}/q^2.$$

Further, we have that $\|z_{\boldsymbol{v}}\| = \|z^* \cdot \widetilde{z_i}\| \leq \sqrt{n} \cdot \sqrt{n}q \cdot \sqrt{n}q = n^{1.5}q^2$. This finally gives us

$$\|z_{\boldsymbol{v}}\| \cdot \|1/z_{\boldsymbol{v}}\| \leq n^4.$$

To conclude, this method gives us

$$E = \Theta(n^{7.5}).$$

This means that $\gamma = 7.5$ and both η and ν are zero.

Remark. For all methods with $\Sigma_{\boldsymbol{v}} \sim z_{\boldsymbol{v}}\bar{z_{\boldsymbol{v}}}$ (i.e. all methods except the simplistic one), if $c \leftarrow D_{I+a,\sqrt{\Sigma_{\boldsymbol{v}}}}$ is sampled using a Gaussian distribution of standard deviation $\sqrt{\Sigma_{\boldsymbol{v}}}$, we can rewrite $c = c^* z_{\boldsymbol{v}}$ with $c^* \leftarrow D_{\frac{I+a}{z_{\boldsymbol{v}}},\sigma_{\boldsymbol{v}}}$ for some $\sigma_{\boldsymbol{v}} \in \mathbb{R}$. Note that c^* is now a following a spherical Gaussian distribution but its support depends on $z_{\boldsymbol{v}}$. In addition to this remark, one can observe that in all the methods described above, there exists a real σ such that $\sigma_{\boldsymbol{v}}\sigma_{\tilde{\boldsymbol{v}}} = \sigma$ for all $\boldsymbol{v} \in \mathcal{A}$ (in fact, $\sigma_{\boldsymbol{v}}$ only depends on the weight of $\boldsymbol{v}$ in all the methods above). This means that for every fresh encodings $[c_{\boldsymbol{v}} z_{\boldsymbol{v}}^{-1}]$ and $[c_{\tilde{\boldsymbol{v}}} z_{\tilde{\boldsymbol{v}}}^{-1}]$ at level $\boldsymbol{v}$ and $\tilde{\boldsymbol{v}}$ generated independently, we have an element $c^* \in K$, following an isotropic distribution[10] of variance σ^2 such that $c_{\boldsymbol{v}} c_{\tilde{\boldsymbol{v}}} = c^* z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}}$ in R. Again, we note that the support of c^* depends on $z_{\boldsymbol{v}}$ and $z_{\tilde{\boldsymbol{v}}}$, but as σ is larger than the smoothing parameter, this has no influence on the variance of c^* (by Lemma 1).

A summary of the different values of γ, η and ν for the different sampling methods can be found in Table 1 (Section 4.3).

4 Averaging attack

4.1 Our simple setting of the GGH multilinear map

To study the leakage of the GGH multilinear map, we need to make reasonable assumptions on what is given to the adversary. It has been shown in [HJ16] that knowing low level encodings of zero for the GGH13 multilinear map leads to zeroizing attacks that completely break the scheme. So our setting should not provide any, yet we will provide enough information for some zero-tests to pass. To this end, we will prove our setting to be secure in the weak multilinear map model, which supposedly prevents zeroizing attacks.

[10] c^* is isotropic as it is the product of two independent isotropic Gaussian variables.

This setting is inspired by the use of multilinear maps in current candidate obfuscator constructions, and more precisely the low noise candidate obfuscator of [DGG+16]. Yet, for easier analysis, we tailored this setting to the bare minimum. We will assume the degree of the multilinear map to be exactly $\kappa = 2$, and will provide the attacker with elements that pass zero-test under a known polynomial. The restriction $\kappa = 2$ can easily be lifted but it would make the exposition of the model and the analysis of the leak less readable.

More precisely, we fix a number $m > 1$ of monomials, and consider the homogeneous degree-2 polynomial:

$$H(x_1, y_1, \ldots, x_m, y_m) = \sum x_i y_i.$$

Recall that we chose the set of "atoms" $\mathcal{A}$ to be the set of levels $\boldsymbol{v} \in \{0,1\}^\ell$ that have weight exactly 1 or $\ell - 1$, where the weight of $\boldsymbol{v}$ is the number of its non-zero coefficients. For all $\boldsymbol{v} \in \mathcal{A}$, we let $\tilde{\boldsymbol{v}} = \boldsymbol{v}^* - \boldsymbol{v}$ (we say that $\tilde{\boldsymbol{v}}$ is the complement of $\boldsymbol{v}$). We assume that for each $\boldsymbol{v} \in \mathcal{A}$ of weight 1, the authority reveals encodings $u_{\boldsymbol{v},1}, \ldots, u_{\boldsymbol{v},m}$ at level $\boldsymbol{v}$ of random values $a_{\boldsymbol{v},1}, \ldots, a_{\boldsymbol{v},m}$ modulo I, and encodings $u_{\tilde{\boldsymbol{v}},1}, \ldots, u_{\tilde{\boldsymbol{v}},m}$ at level $\tilde{\boldsymbol{v}}$ of random values $a_{\tilde{\boldsymbol{v}},1}, \ldots, a_{\tilde{\boldsymbol{v}},m}$ modulo I, under the only constraint that

$$H(a_{\boldsymbol{v},1}, a_{\tilde{\boldsymbol{v}},1}, \ldots, a_{\boldsymbol{v},m}, a_{\tilde{\boldsymbol{v}},m}) = 0 \bmod I.$$

We remark that generating almost uniform values $a_{\cdot,\cdot}$ under the constraint above is easily done, by choosing all but one of them at random, and setting the last one to

$$a_{\tilde{\boldsymbol{v}},m} = -a_{\boldsymbol{v},m}^{-1} \sum_{i=1}^{m-1} a_{\boldsymbol{v},i} a_{\tilde{\boldsymbol{v}},i} \bmod I.$$

In the weak multilinear map model [MSZ16, GMM+16, DGG+16], we can prove that an attacker that has access to this simple setting of the GGH multilinear map cannot recover a multiple of the secret element g, except with negligible probability. The definition of the weak multilinear map model and the proof that an attacker cannot recover a multiple of g can be found in Appendix B.[11] This weak multilinear-map model was used to prove security of candidate obfuscators in [GMM+16, DGG+16], as it is supposed to capture zeroizing attacks, like the ones of [MSZ16, CGH17]. In the weak multilinear map model, recovering a multiple of g is considered to be a successful attack. This is what motivates our proof that no polynomial time adversary can recover a multiple of g in our simple setting, under this model.

4.2 Analysis of the leaked value

We describe in this section the information we can recover using averaging attacks, depending on the sampling method. We will see that depending on the sampling method, we can recover an approximation of $A(z^*h/g)$, or an approximation of $A(h/g)$ or even the exact value of $A(h/g)$. In order to unify notation, we introduce the leak $\mathfrak{L}$, which will refer to $A(z^*h/g)$ or $A(h/g)$ depending the method. We explain below what is the value of $\mathfrak{L}$ for the different methods, and how we can recover an approximation of it. In the case of the simplistic method, we also explain how we can recover the exact value of $\mathfrak{L}$ from its approximation and how to use it to create a zero-testing parameter at level $2\boldsymbol{v}^*$.

[11] We postpone the proof in appendix as the idea is the same as in [GMM+16, DGG+16], in a much simpler context (this is based on a generalized version of the Schwartz-Zippel lemma from [MSZ16]).

Statistical leakage. Let $\boldsymbol{v} \in \mathcal{A}$ be of weight 1. We denote by $[u_{\boldsymbol{v}}]$ the encoding $[H(u_{\boldsymbol{v},1}, u_{\tilde{\boldsymbol{v}},1}, \ldots, u_{\boldsymbol{v},m}, u_{\tilde{\boldsymbol{v}},m})]$. Recall that we have $[u_{i,\boldsymbol{v}}] = [c_{i,\boldsymbol{v}} z_{\boldsymbol{v}}^{-1}]$, where $c_{i,\boldsymbol{v}} = a_{i,\boldsymbol{v}} + r_{i,\boldsymbol{v}} g$ for some $r_{i,\boldsymbol{v}} \in R$. So using the definition of H and the fact that $[u_{\boldsymbol{v}}]$ passes the zero test, we can rewrite

$$\begin{aligned}
[u_{\boldsymbol{v}} p_{zt}] &= [H(c_{\boldsymbol{v},1}, c_{\tilde{\boldsymbol{v}},1}, \ldots, c_{\boldsymbol{v},m}, c_{\tilde{\boldsymbol{v}},m})(z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}})^{-1} \cdot z^* h g^{-1}] \\
&= [H(c_{\boldsymbol{v},1}, c_{\tilde{\boldsymbol{v}},1}, \ldots, c_{\boldsymbol{v},m}, c_{\tilde{\boldsymbol{v}},m}) \cdot h g^{-1}] \\
&= H(c_{\boldsymbol{v},1}, c_{\tilde{\boldsymbol{v}},1}, \ldots, c_{\boldsymbol{v},m}, c_{\tilde{\boldsymbol{v}},m}) \cdot h/g.
\end{aligned}$$

Note that the product of the last line is in R, as it is a product of small elements compared to q. Also, the first term is a small multiple of g so we can divide by g. We denote by $w_{\boldsymbol{v}} \in R$ the value above (i.e., the representative of $[u_{\boldsymbol{v}} p_{zt}]$ with coefficients in $[-q/2, q/2]$). The term h/g of the product is fixed, but the first factor $H(c_{\boldsymbol{v},1}, c_{\tilde{\boldsymbol{v}},1}, \ldots, c_{\boldsymbol{v},m}, c_{\tilde{\boldsymbol{v}},m})$ depends on $\boldsymbol{v}$: we can average over it. We now analyze this first factor, depending on the method we choose for generating the fresh encodings of the GGH map. We will denote by $Y_{\boldsymbol{v}}$ the random variable $H(c_{\boldsymbol{v},1}, c_{\tilde{\boldsymbol{v}},1}, \ldots, c_{\boldsymbol{v},m}, c_{\tilde{\boldsymbol{v}},m})$.

By definition of the polynomial H, we know that $Y_{\boldsymbol{v}} = \sum c_{i,\boldsymbol{v}} c_{i,\tilde{\boldsymbol{v}}}$. Moreover, all the $c_{i,\boldsymbol{v}}$ are independent when i or $\boldsymbol{v}$ vary. So the $c_{i,\boldsymbol{v}} c_{i,\tilde{\boldsymbol{v}}}$ are centered random variables of variance $\Sigma_{\boldsymbol{v}} \Sigma_{\tilde{\boldsymbol{v}}}$ (observe that the variance of a product of independent centered variables is the product of their variances) and $Y_{\boldsymbol{v}}$ is a centered random variable of variance $m \Sigma_{\boldsymbol{v}} \Sigma_{\tilde{\boldsymbol{v}}}$ (recall that H is a sum of m monomials). We now consider several cases, depending on the choice of $\Sigma_{\boldsymbol{v}}$.

Case 1 (the simplistic method). In this case, we have $\Sigma_{\boldsymbol{v}} = \sigma^2$ for all $\boldsymbol{v} \in \mathcal{A}$, for some $\sigma \in \mathbb{R}$. This means that the $Y_{\boldsymbol{v}}$ are centered isotropic random variables with the same variance. Let us call $\mu := \mathbb{E}[A(Y_{\boldsymbol{v}})] = m\sigma^2 \in \mathbb{R}^+$ this variance. If we compute the empirical mean of the $A(Y_{\boldsymbol{v}})$, this will converge to μ and we can bound the speed of convergence using Hoeffding's inequality. Going back to the variables $w_{\boldsymbol{v}} = Y_{\boldsymbol{v}} \cdot h/g$, we have that $\mathbb{E}[A(w_{\boldsymbol{v}})] = \mu \cdot A(h/g)$ for some μ in $\mathbb{R}^+$. Furthermore, all the $A(w_{\boldsymbol{v}})$, with $\boldsymbol{v}$ of weight 1, are independent variables with the same mean, so we can apply Hoeffding's inequality.

Case 2 (the conservative method). In this case, we chose $\Sigma_{\boldsymbol{v}} \sim z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}}$. We do not know the variance of the $Y_{\boldsymbol{v}}$ (because the $z_{\boldsymbol{v}}$ are secret) but we will be able to circumvent this difficulty, by averaging over the $z_{\boldsymbol{v}}$'s.

First, using the remark we made at the end of Section 3, we have that $Y_{\boldsymbol{v}} = \sum c_{i,\boldsymbol{v}} c_{i,\tilde{\boldsymbol{v}}} = \sum c^*_{i,\boldsymbol{v}} z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}}$, with the $c^*_{i,\boldsymbol{v}}$ being independent centered isotropic random variables with the same variance $\sigma^2 \in \mathbb{R}^+$. Hence, we can rewrite $Y_{\boldsymbol{v}} = X_{\boldsymbol{v}} z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}}$ with $X_{\boldsymbol{v}}$ a centered isotropic variable of variance $m\sigma^2$ (which is independent of $\boldsymbol{v}$). Unlike the previous case, we now have some $z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}}$ that contribute in Y_v. However, we will be able to remove them again by averaging. Indeed, even if all the $z_{\boldsymbol{v}}$ satisfy $[z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}}] = [z^*]$ in R_q, this is not the case in R. For our analysis, let us treat the $z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}}$ as random variables in R, that are isotropic and independent when $\boldsymbol{v}$ varies. We will call $\mu_z := \mathbb{E}[A(z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}})]$ their variance. Recall that as the $z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}}$ are isotropic, μ_z is in $\mathbb{R}^+$. While the independence assumption may be technically incorrect, experiments confirm that the empirical mean $\mathbb{E}[A(z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}})]$ does indeed converge to some $\mu_z \in \mathbb{R}^+$ as the number of sample grows, and more precisely it seems to converge as $\mu_z \cdot (1+\varepsilon)$ where $\varepsilon \in K_{\mathbb{R}}$ satisfies $\|\varepsilon\|_\infty = \tilde{O}(\sqrt{1/|\mathcal{A}|})$, as predicted by the Hoeffding bound (results of the experiments are plotted in Appendix C).

Assuming that the $X_{\boldsymbol{v}}$ are independent of the $z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}}$,[12] we finally obtain

$$\mathbb{E}[A(Y_{\boldsymbol{v}})] = \mathbb{E}[A(X_{\boldsymbol{v}})]\mathbb{E}[A(z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}})] = m\sigma^2 \mu_z.$$

We denote by $\mu = m\sigma^2\mu_z$ this value. As in the previous case, the variables $A(w_{\boldsymbol{v}})$ are independent (when $\boldsymbol{v}$ has weight 1) and have the same mean

$$\mathbb{E}\left[A(w_{\boldsymbol{v}})\right] = \mu \cdot A(h/g),$$

with $\mu \in \mathbb{R}^+$.

Case 3 (the exponential and aggressive methods). In these methods, we can again write $Y_{\boldsymbol{v}} = X_{\boldsymbol{v}} z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}}$ with $X_{\boldsymbol{v}}$ a centered isotropic variable of variance $m\sigma^2$ for some $\sigma \in \mathbb{R}^+$, independent of $\boldsymbol{v}$. However, unlike the previous case, the $z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}}$ are not isotropic variables anymore and therefore the z's do not "average-out".

In the exponential method, the identity $z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}} = z^*$ holds over R (where $z^* = \prod_i z_i \in R$ is a representative of $[z^*]$), hence, $z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}}$ is constant when $\boldsymbol{v}$ varies, and we have

$$\mathbb{E}\left[A(w_{\boldsymbol{v}})\right] = \mu \cdot A(hz^*/g),$$

for some scalar $\mu \in \mathbb{R}^+$.

In the aggressive method, we have $z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}} = z^* \cdot \widetilde{z}_i \cdot z_i$ for some $1 \leq i \leq \ell$, with z^* the representative of $[z^*]$, z_i the representative of $[z_i]$ and $\widetilde{z}_i$ the representative of $[z_i^{-1}]$ with coefficients in $[-q/2, q/2]$. The element z^* is fixed, but, as in the conservative case, we can see the $\widetilde{z}_i \cdot z_i$ as isotropic variables. Assuming they are independent, we then have $\mathbb{E}\left[A(z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}})\right] = \mu_z A(z^*)$ for some scalar $\mu_z \in \mathbb{R}^+$. And we again have

$$\mathbb{E}\left[A(w_{\boldsymbol{v}})\right] = \mu \cdot A(hz^*/g),$$

for some scalar $\mu \in \mathbb{R}^+$.

Conclusion on the average. To conclude, we have argued that in all methods,

$$\mathbb{E}\left[A(w_{\boldsymbol{v}})\right] = \mu \cdot \mathfrak{L}$$

for some scalar $\mu \in \mathbb{R}^+$, where the leaked variable $\mathfrak{L}$ depends on the sampling method in the following way:

- $\mathfrak{L} = A(h/g)$ for the simplistic and the conservative methods.
- $\mathfrak{L} = A(hz^*/g)$ for the exponential and the aggressive methods.

Now, using the fact that the random variables $A(w_{\boldsymbol{v}})$ are independent for different $\boldsymbol{v} \in \mathcal{A}$ of weight 1, we can compute their empirical mean and Hoeffding's inequality will allow us to bound the distance to the theoretical mean. In the following we assume that we know μ. [13]

[12] We can view the variables $c^*_{i,\boldsymbol{v}}$ as being independent of the variables $z_{\boldsymbol{v}}$ because the standard deviation of the Gaussian distribution is larger than the smoothing parameter (see Lemma 1).

[13] The value of the scalar μ can be obtained from the parameters of the multilinear maps. If we do not want to analyze the multilinear map, we can guess an approximation of μ with a sufficiently small relative error, by an exhaustive search.

Relative error of the leakage. Compute

$$W = \frac{2}{|\mathcal{A}|} \sum_{\substack{\boldsymbol{v}\in\mathcal{A} \\ \boldsymbol{v} \text{ of weight } 1}} A(w_{\boldsymbol{v}})$$

the empirical mean of the random variables $A(w_{\boldsymbol{v}})$. This is an approximation of $\mu \cdot \mathfrak{L}$. We know that the coefficients of the random variable $w_{\boldsymbol{v}}$ are less than q, so the coefficients of $A(w_{\boldsymbol{v}})$ are less that nq^2. By applying Hoeffding's inequality in R (Corollary 1) with $\varepsilon = 1/n$, $B = nq^2$ and $m = |\mathcal{A}|/2$, we have that $\|W - \mu \cdot \mathfrak{L}\|_\infty < \frac{nq^2\sqrt{2\ln n}}{\sqrt{|\mathcal{A}|}}$ (except with probability at most $1/\mathrm{n}$). As the coefficients of $\mu\mathfrak{L}$ are of the order of nq^2, we have a relative error $\delta < \sqrt{2\ln n/|\mathcal{A}|}$ for each coefficient of $\mu\mathfrak{L}$. As μ is known, this means that we know $\mathfrak{L}$ with a relative error at most $\sqrt{2\ln n/|\mathcal{A}|}$. [14]

Unfortunately, we cannot directly recover the exact value of $\mathfrak{L}$ because its coefficients are not integers. When $\mathfrak{L} = A(hz^*/g)$, i.e. for the exponential and aggressive methods, we do not know how to use this approximation of $\mathfrak{L}$ to recover the exact value of $\mathfrak{L}$.[15] When $\mathfrak{L} = A(h/g)$, i.e. for the simplistic and conservatives methods, we can circumvent this difficulty. The idea is to transform our approximation of $\mathfrak{L}$ into an approximation of an element $r \in R$, with coefficients that are integers of logarithmic bit-size. Indeed, if we have an approximation of r with error less that $1/2$ we can round its coefficients and recover the exact value of r. And we can get such an approximation using a polynomial number of samples because the coefficients we want to recover have logarithmic bit-size. This is what we explain in next subsection. Unfortunately, we will see that for the conservative method, the number of samples we need to be able to round r to its exact value is not compatible with the constraint we had on $|\mathcal{A}|$ for being able to generate the $z_{\boldsymbol{v}}$.

From the leakage to a complete attack against the GGH map. In this section, we explain how we can recover the exact value of $A(h/g)$, when $\mathfrak{L} = A(h/g)$ and we have enough samples. We then show how we can use this exact value to construct a zero-testing parameter at level $2\boldsymbol{v}^*$.

Recovering $\mathfrak{L}$ exactly when $\mathfrak{L} = A(h/g)$. In the following, we assume that we have an approximation of $A(h/g)$ with relative error $\delta < \sqrt{2\ln n/|\mathcal{A}|}$ and we want to recover the exact value of $A(h/g)$. Let u be any encoding at level $\boldsymbol{v}^*$ that passes the zero test (we can take u to be one of the $[u_{\boldsymbol{v}}] = [H(u_{\boldsymbol{v},1}, u_{\tilde{\boldsymbol{v}},1}, \ldots, u_{\boldsymbol{v},m}, u_{\tilde{\boldsymbol{v}},m})]$). We have that $[u \cdot p_{zt}] = c \cdot h/g \in R$ for some small multiple c of g. In particular, the coefficients of c are somehow small[16] and are integers. Using our approximation W of $\mu \cdot A(h/g)$ with relative error δ plus the fact that we know μ and $c \cdot h/g$, we can recover an approximation of $A(c)$ with relative error at most $\delta \cdot n^2$ by computing $A(c \cdot h/g) \cdot \mu \cdot W^{-1}$.

The coefficients of $A(c)$ are integers and are less than $m^2n^2E^4$. Indeed, $c = H(c_{\boldsymbol{v},1}, c_{\tilde{\boldsymbol{v}},1}, \ldots, c_{\boldsymbol{v},m}, c_{\tilde{\boldsymbol{v}},m})$ for some $\boldsymbol{v}$ and we have $\|c_{\boldsymbol{v},i}\| \leq E$ for all $\boldsymbol{v}$'s and i's. So we know that $\|c\| \leq mn^{1/2}E^2$ and we get the desired bound on $\|A(c)\|_\infty$. Hence, if we have an approximation of the coefficients of $A(c)$ with relative error at most $\frac{1}{2m^2n^2E^4}$, the absolute error is less that $1/2$ and we can round the coefficients to recover $A(c)$ exactly. We can then recover $A(h/g)$ exactly by computing $A(c \cdot h/g)/A(c)$.

[14] Again, if we do not know μ, we can guess an approximation of μ with relative error at most $\sqrt{2\ln n/|\mathcal{A}|}$ (so that it has no influence on our approximation of $\mathfrak{L}$), with an exhaustive search.

[15] Note that if we recover the exact value of $A(hz^*/g)$, then its denominator is a multiple of g and this is considered as a success of the attacker in the weak multilinear map model.

[16] Recall that q may be exponentially large but we assumed that the numerator of a top level encoding remains polynomial in n.

Putting together the conditions we got on the parameters, we have $\delta < \sqrt{\frac{2\ln n}{|\mathcal{A}|}}$ and we want $\delta \cdot n^2 < \frac{1}{2m^2n^2E^4}$ to be able to recover $A(c)$. This is satisfied if $\sqrt{\frac{2\ln n}{|\mathcal{A}|}} < \frac{1}{2m^2n^4E^4}$, i.e., $|\mathcal{A}| > 8E^8m^4n^8\ln n$.

To conclude, if $|\mathcal{A}| > 8E^8m^4n^8\ln n$, we can recover $A(g/h) \in K$ exactly.[17] In Section 4.3, we compare this constraint to the ones we had for the samplings methods. We will see that for the simplistic method, our constraints are compatible, so we can perform the attack. But this is not the case with the conservative method.

Using $A(h/g)$ to create a zero testing parameter at a forbidden level. We present here a possible way of using the recovered value $A(h/g)$. Note that in current obfuscation model (for instance the weak multilinear map model of [GMM+16] or [DGG+16]), recovering $A(h/g)$ is already considered as a success for the attacker. Indeed, its denominator is a multiple of $A(g) = g\bar{g}$ so in particular we have recovered a multiple of g, which is considered as a success of the attacker in these models.[18] Moreover, even if we do not consider that recovering a multiple of g is bad news, we present here a way of using $A(h/g)$ to create a zero-testing parameter at a higher level than $\boldsymbol{v}^*$ (here we create a zero-testing parameter at level $2\boldsymbol{v}^*$).

First, note that the complex conjugation $\bar{\cdot}$ in R is compatible with R_q. Indeed, let $c, r \in R$, we have $\overline{c+qr} = \bar{c} + \overline{qr} = \bar{c} + q\bar{r}$ (because $\bar{\cdot}$ is $\mathbb{R}$-linear). So $\overline{c+qr} \equiv \bar{c} \bmod q$ and we can define the operation $\bar{\cdot}$ in R_q by $\overline{[r]} = [\bar{r}]$. We will use this to construct our zero-testing parameter. Let again $[u]$ be an encoding of zero at level $\boldsymbol{v}^*$ and write $[u] = [c \cdot (z^*)^{-1}]$ where c is a small multiple of g. Compute

$$\begin{aligned} p'_{zt} &= [\bar{u} \cdot p_{zt}^2 \cdot \overline{p_{zt}} \cdot A(h/g)^{-1}] \\ &= \left[\frac{\bar{c}}{\bar{z^*}} \cdot \frac{(z^*)^2h^2}{g^2} \cdot \frac{\bar{z^*}\bar{h}}{\bar{g}} \cdot \frac{g\bar{g}}{h\bar{h}}\right] \\ &= \left[\frac{(z^*)^2 \cdot (h\bar{c})}{g}\right]. \end{aligned}$$

As $h\bar{c}$ is small compared to q, this gives us a zero-testing parameter at level $2\boldsymbol{v}^*$.

4.3 Noise analysis of the leakage

We sum up in this section the leakage that we can obtain and with which precision, depending on the sampling methods presented in Section 3.

The simplistic method. In this method, we have $\mathfrak{L} = A(h/g)$. Recall that in this case, we can recover the exact value of $\mathfrak{L}$ if $\ell > 4E^8m^4n^8\ln n$ (using the fact that $|\mathcal{A}| = 2\ell$). But in this method, we had $E = O(n^{1.5+\varepsilon})$, for any $\varepsilon > 0$. Hence, taking $\ell = \Theta(n^{20+8\varepsilon}m^4\ln n)$ satisfies the conditions for generating the parameters plus our condition $\ell > 4E^8m^4n^8\ln n$. To conclude, when using the simplistic method with some choice of the parameters, we can recover the exact value $A(h/g)$ and

[17] Note that this bound does not depends on q but only on E. This is why our attack still works even if q is exponential in n, as long as E remains polynomial in n.

[18] For this to be true, we need h and g to be co-prime. But as the ideal $\langle g\rangle$ is prime, this will be true unless h is a multiple of g. And the case where h is a multiple of g is not a problem, as we can easily recover multiples of h (and so multiples of g).

use it to construct a forbidden zero-testing parameter at level $2\boldsymbol{v}^*$. Note that recovering $A(h/g)$ also means that we recovered a multiple of g. However, we proved that in the weak multilinear map model, no polynomial time attacker could recover a multiple of g. This proves that the averaging attack described above is not captured by the weak multilinear map model.

Remark. For this sampling method, as $\Sigma_{\boldsymbol{v}} \sim 1$, we do not need to average over the $\boldsymbol{v}$, so we could also have $\ell = 2$ as long as we have enough samples for each $\boldsymbol{v}$.

The exponential method. In this method, we have $\mathfrak{L} = A(z^*h/g)$. We can recover an approximation of $\mu\mathfrak{L}$ with relative error at most $\sqrt{\frac{2\ln n}{|\mathcal{A}|}}$. We do not know if it is possible to recover $\mathfrak{L}$ exactly.

The conservative method revisited. In this method, we have $\mathfrak{L} = A(h/g)$, we can recover an approximation of $\mu\mathfrak{L}$ with relative error at most $\sqrt{\frac{2\ln n}{|\mathcal{A}|}}$ according to our heuristic analysis. While the independence condition between the $A(z_{\boldsymbol{v}}z_{\tilde{\boldsymbol{v}}})$ for applying Hoeffding's bound may not be satisfied, we show that this rate of convergence seems correct in practice in Appendix C.

Recall that if $\ell > 4E^8m^4n^8\ln n$, then we can recover $A(h/g)$ exactly. But for the sampling method to work, we need to take $E = \Theta(n^{4.5}\sqrt{\ell})$. Hence, the condition $\ell > 4E^8m^4n^8\ln n$ can be rewritten

$$\ell > \Theta(n^{44}\ell^4m^4\ln n).$$

This condition cannot be satisfied, so we cannot have enough samples for our attack when using this sampling method. And all we get is an approximation of $\mu A(h/g)$. Nevertheless, the only thing that prevents the full attack is the size of the parameters we have to chose in order to be able to generate the fresh encodings. This is far from the kind of protection that was intended.

The aggressive method. In this method, we have $\mathfrak{L} = A(z^*h/g)$. We can recover an approximation of $\mu\mathfrak{L}$ with relative error at most $\sqrt{\frac{2\ln n}{|\mathcal{A}|}}$. We do not know if it is possible to recover $\mathfrak{L}$ exactly.

4.4 Concusion

We give in Table 1 a summary of the parameters used for the different sampling methods, and of the resulting leakage. The column 'constraints' specifies possible constraints on the parameters or on the atoms set $\mathcal{A}$, that arise when using this sampling method. Recall that due to the correctness bound (9), there is always a constraint on the modulus q, so we do not mention it in the column 'constraints'. This constraint on q can be obtained from the columns γ, η and ν, using the formula $\log q \geq 4\log(n)(3 + \kappa/2 + \kappa\gamma + \kappa\nu L + \eta\ell) + 4\log(m)$.

We have seen that the leak obtained in the conservative method is the same as the one of the unprotected scheme (the simplistic method). However, in the case of the conservative method, the number of available samples is not sufficient to complete the attack, as it is the case in the simplistic method. This limitation on the number of samples comes from some constraints in the sampling procedure and seems a bit accidental, we do not find this version of the countermeasure fully satisfactory.

We can also question the security of the other methods (exponential and aggressive), which leak an approximation of $A(hz^*/g)$, related to secret values. More precisely, one could wonder whether this noisy leak could be combined with the knowledge of $p_{zt} = [hz^*g^{-1}]$ to mount an attack. As this problem does not look like any traditional (ideal) lattice problem, we fail to conclude beyond

Sampling method	γ	η	ν	leak $\mathfrak{L}$	full attack?	constraints
Simplistic [GGH13]	$1.5+\varepsilon$	0	0	$A(h/g)$	yes	none
Exponential [GGH13]	2.5	2.5	0	$A(z^*h/g)$	no	none
Conservative [DGG+16]	6	0	0	$A(h/g)$	no	$n \geq 4\ell$
Conservative (revisited)	4.5	0	0.5	$A(h/g)$	no	none
Aggressive [DGG+16]	7.5	0	0	$A(z^*h/g)$	no	structure of $\mathcal{A}$
Compensation (Sec. 5)	$1.5+1/\kappa+\varepsilon$	0	0	1	no	none

Table 1. Summary of the leak analysis, depending on the sampling method. This includes our new method, sketched in Section 5. We recall that, according to correctness bound (9), the modulus q must satisfy $\log q \geq 4\log(n)(\text{cste}+\kappa/2+\kappa\gamma+\kappa\nu L+\eta\ell)+4\log(m)$.

reasonable doubt that it should be intractable. We would find more rational to make the leak unrelated to secret parameters. In the following section, we propose such a design, which is simple, and leads to better parameters.

5 The compensation method

In this section, we propose a new sampling method which is designed so that the leak $\mathfrak{L}$ that an attacker can recover by using the averaging attack described above, reveals no information about secret parameters of the GGH map. Nevertheless, we note that even if the attack described above does not apply directly to this method, other averaging attacks may be able to leak secret information. An idea could be to fix some encodings and average over the others.

Discussion on design. We have seen that choosing different covariance parameters $\Sigma_{\boldsymbol{v}}$ at different levels $\boldsymbol{v}$ can in fact make the leak *worse*, as the attacker can choose to average them out. We also remark that the parameters $[z_{\boldsymbol{v}}]$ can be *publicly re-randomized* without affecting anything else, in particular without affecting the covariance $\Sigma_{\boldsymbol{v}}$ of the numerator of the encodings. Indeed, we can choose random invertible elements $[\hat{z}_i] \in R_q^\times$, and apply the following transformation to all encodings $e_{\boldsymbol{v}}$ at level $\boldsymbol{v}$, as well as to the zero-testing parameter $[p_{zt}]$:

$$[e_{\boldsymbol{v}}] \mapsto \left[\prod_{i\in\boldsymbol{v}} \hat{z}_i^{-1}\right] \cdot [e_{\boldsymbol{v}}], \quad [p_{zt}] \mapsto \left[\prod_{i\in\boldsymbol{v}^\star} \hat{z}_i\right] [p_{zt}].$$

This means that the relation between the covariance $\Sigma_{\boldsymbol{v}}$ and the denominators $z_{\boldsymbol{v}}$ can be publicly undone while maintaining functionality.

The compensation method. We therefore proceed to set $\Sigma_{\boldsymbol{v}} = \Sigma$ for all levels $\boldsymbol{v}$, and to choose Σ independently of the $z_{\boldsymbol{v}}$. Doing so, we observe that the leak $\mathfrak{L}$ will generically be:

$$\mathfrak{L} \sim \Sigma^\kappa \cdot A(h/g). \tag{15}$$

We then choose $\Sigma \sim A(g/h)^{1/\kappa}$, ensuring $\mathfrak{L} \sim 1$: the leak is made constant, unrelated to any secret. We insist nevertheless that, as the previous methods, this method comes with no formal security argument. We also warned that we have not thoroughly explored more general leakage attacks, varying the zero-tested polynomials or keeping some encodings fixed.

It remains to see how short one can efficiently sample encodings following this choice. To get tighter bounds, we look at the conditioning number (or distortion) $\delta(\sqrt{\Sigma}) = \frac{\max(\sigma_i(\sqrt{\Sigma}))}{\min(\sigma_i(\sqrt{\Sigma}))}$, where σ_i runs over all embeddings. One easily verifies the following properties:

$$\delta(A(x)) = \delta(x)^2 \tag{16}$$

$$\delta(x^k) = \delta(x)^{|k|} \quad \text{for any } k \in \mathbb{R}, \tag{17}$$

$$\delta(xy) \leq \delta(x)\delta(y). \tag{18}$$

If a variable $x \in K_{\mathbb{R}}$ has independent continuous Gaussian coefficients of parameter 1, then its embeddings are (complex) Gaussian variables of parameter $\Theta(\sqrt{n})$, and it holds with constant probability that

$$\forall i, \quad \Omega(1) \leq |\sigma_i(x)| \leq O(\sqrt{n \log n}). \tag{19}$$

Indeed, the right inequality follows from classic tail bounds on Gaussian. For the left inequality, consider that $|\sigma_i(x)| \geq \max(|\Re(\sigma_i(x))|, |\Im(\sigma_i(x))|)$, where both the real and imaginary parts are independent Gaussian of parameter $\Theta(\sqrt{n})$: each part will be smaller than $\Theta(1)$ with probability at most $1/\sqrt{2n}$. By independence, $|\sigma_i(x)| \leq \Theta(1)$ holds with probability at most $1/2n$ for each i, and one may conclude by the union bound.

By scaling (and plausibly ignoring discreteness issues since g and h are sampled above the smoothing parameter of $\mathbb{Z}^n$) we can therefore assume, using rejection sampling over h and g, that $\delta(g), \delta(h) \leq O(\sqrt{n \log n})$, and therefore

$$\delta(\sqrt{\Sigma}) = \delta(A(g/h))^{1/2\kappa} \leq (\delta(g)\delta(h))^{1/\kappa} \leq O(n \log n)^{1/\kappa}.$$

This allows to scale Σ so that:

- $\|g/\sqrt{\Sigma}\| \leq o(1/\sqrt{\log n})$, so that we can sample efficiently via Theorem 2.
- $E = \sqrt{n} \cdot \|g\| \cdot \delta(\sqrt{\Sigma}) \cdot \omega(\sqrt{\log n}) = O(n^{1.5+1/\kappa+\varepsilon})$: the size of the numerators of the encodings is barely worse than in the simplistic method, and significantly better than in all other methods.

References

ABD16. Martin R. Albrecht, Shi Bai, and Léo Ducas. A subfield lattice attack on overstretched NTRU assumptions - cryptanalysis of some FHE and graded encoding schemes. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 153–178. Springer, Heidelberg, August 2016.

AS16. Prabhanjan Ananth and Amit Sahai. Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. Cryptology ePrint Archive, Report 2016/1097, 2016. http://eprint.iacr.org/2016/1097.

BBKK17. Boaz Barak, Zvika Brakerski, Ilan Komargodski, and Pravesh K. Kothari. Limits on low-degree pseudorandom generators (or: Sum-of-squares meets program obfuscation). Cryptology ePrint Archive, Report 2017/312, 2017. http://eprint.iacr.org/2017/312.

BEF+17. Jean-François Biasse, Thomas Espitau, Pierre-Alain Fouque, Alexandre Gélin, and Paul Kirchner. Computing generator in cyclotomic integer rings. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 60–88. Springer, 2017.

BF01. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, Heidelberg, August 2001.

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

BGK+14. Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 221–238. Springer, Heidelberg, May 2014.

BR14. Zvika Brakerski and Guy N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In Yehuda Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 1–25. Springer, Heidelberg, February 2014.

BS16. Jean-François Biasse and Fang Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 893–902. Society for Industrial and Applied Mathematics, 2016.

CDPR16. Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. Recovering short generators of principal ideals in cyclotomic rings. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 559–585. Springer, Heidelberg, May 2016.

CGH+15. Jean-Sébastien Coron, Craig Gentry, Shai Halevi, Tancrède Lepoint, Hemanta K. Maji, Eric Miles, Mariana Raykova, Amit Sahai, and Mehdi Tibouchi. Zeroizing without low-level zeroes: New MMAP attacks and their limitations. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 247–266. Springer, Heidelberg, August 2015.

CGH17. Yilei Chen, Craig Gentry, and Shai Halevi. Cryptanalyses of candidate branching program obfuscators. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 278–307. Springer, 2017.

CGS14. Peter Campbell, Michael Groves, and Dan Shepherd. Soliloquy: A cautionary tale. In *ETSI 2nd Quantum-Safe Crypto Workshop*, pages 1–9, 2014.

CJL16. Jung Hee Cheon, Jinhyuck Jeong, and Changmin Lee. An algorithm for ntru problems and cryptanalysis of the ggh multilinear map without a low level encoding of zero. *Mh*, 1:0, 2016.

DGG+16. Nico Döttling, Sanjam Garg, Divya Gupta, Peihan Miao, and Pratyay Mukherjee. Obfuscation from low noise multilinear maps. Cryptology ePrint Archive, Report 2016/599, 2016. http://eprint.iacr.org/2016/599.

GGH97. Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 112–131. Springer, Heidelberg, August 1997.

GGH13. Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 1–17. Springer, Heidelberg, May 2013.

GMM+16. Sanjam Garg, Eric Miles, Pratyay Mukherjee, Amit Sahai, Akshayaram Srinivasan, and Mark Zhandry. Secure obfuscation in a weak multilinear map model. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B: 14th Theory of Cryptography Conference, Part II*, volume 9986 of *Lecture Notes in Computer Science*, pages 241–268. Springer, Heidelberg, October / November 2016.

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 197–206. ACM Press, May 2008.

GS02. Craig Gentry and Michael Szydlo. Cryptanalysis of the revised NTRU signature scheme. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 299–320. Springer, Heidelberg, April / May 2002.

HHGP+03. Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte. NTRUSIGN: Digital signatures using the NTRU lattice. In Marc Joye, editor, *Topics in Cryptology – CT-RSA 2003*, volume 2612 of *Lecture Notes in Computer Science*, pages 122–140. Springer, Heidelberg, April 2003.

HJ16. Yupu Hu and Huiwen Jia. Cryptanalysis of GGH map. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 537–565. Springer, Heidelberg, May 2016.

HPS01. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NSS: An NTRU lattice-based signature scheme. In Birgit Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 211–228. Springer, Heidelberg, May 2001.

Jou00. Antoine Joux. A one round protocol for tripartite diffie–hellman. In *International Algorithmic Number Theory Symposium*, pages 385–393. Springer, 2000.

KF17. Paul Kirchner and Pierre-Alain Fouque. Revisiting lattice attacks on overstretched ntru parameters. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 3–26. Springer, 2017.

Kle00. Philip N. Klein. Finding the closest lattice vector when it's unusually close. In David B. Shmoys, editor, *11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 937–941. ACM-SIAM, January 2000.

Lin16. Huijia Lin. Indistinguishability obfuscation from constant-degree graded encoding schemes. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 28–57. Springer, Heidelberg, May 2016.

Lin17. Huijia Lin. Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGs. In *Annual International Cryptology Conference*, pages 599–629. Springer, 2017.

LSS14. Adeline Langlois, Damien Stehlé, and Ron Steinfeld. GGHLite: More efficient multilinear maps from ideal lattices. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 239–256. Springer, Heidelberg, May 2014.

LT17. Huijia Lin and Stefano Tessaro. Indistinguishability obfuscation from bilinear maps and blockwise local prgs. Cryptology ePrint Archive, Report 2017/250, 2017. http://eprint.iacr.org/2017/250.

LV17. Alex Lombardi and Vinod Vaikuntanathan. On the non-existence of blockwise 2-local prgs with applications to indistinguishability obfuscation. Cryptology ePrint Archive, Report 2017/301, 2017. http://eprint.iacr.org/2017/301.

MR04. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th Annual Symposium on Foundations of Computer Science*, pages 372–381. IEEE Computer Society Press, October 2004.

MSZ16. Eric Miles, Amit Sahai, and Mark Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over GGH13. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 629–658. Springer, Heidelberg, August 2016.

NR06. Phong Q. Nguyen and Oded Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 271–288. Springer, Heidelberg, May / June 2006.

SS13. Damien Stehlé and Ron Steinfeld. Making NTRUEncrypt and NTRUSign as secure as standard worst-case problems over ideal lattices. Cryptology ePrint Archive, Report 2013/004, 2013. http://eprint.iacr.org/2013/004.

A Proof of Lemma 1

We recall here Lemma 1 and provide a proof of it, using a result of [MR04].

Lemma 1. *For any $g \in K$, $\Sigma \in S^+$, $c \in K_{\mathbb{R}}$ such that $\|g/\sqrt{\Sigma}\| \leq o(1/\sqrt{\log n})$, if $x \leftarrow D_{gR+c,\sqrt{\Sigma}}$, then $\|\mathbb{E}[x]\| \leq \varepsilon \cdot \|\sqrt{\Sigma}\|$ and $\|\mathbb{V}[x] - \Sigma\| \leq \varepsilon \cdot \|\Sigma\|$ for some negligible function $\varepsilon(n)$.*

Proof. This result follows from [MR04, Lemma 4.2], that we recall here:

Lemma 4 (Lemma 4.2 from [MR04]). *For any n-dimensional lattice Λ, point $c \in \mathbb{R}^n$, unit vector u, and reals $0 < \varepsilon < 1$, $s \geq 2\eta_\varepsilon(\Lambda)$,*

$$\left|\mathbb{E}_{x \leftarrow D_{\Lambda,s,c}}[\langle x - c, u\rangle]\right| \leq \sqrt{2\pi}\frac{\varepsilon s}{1-\varepsilon}$$

$$\left|\mathbb{E}_{x \leftarrow D_{\Lambda,s,c}}[\langle x - c, u\rangle^2] - s^2\right| \leq 2\pi\frac{\varepsilon s^2}{1-\varepsilon}$$

where $\eta_\varepsilon(\Lambda)$ is the smoothing parameter of the lattice Λ.[19]

We cannot use this lemma directly to prove our Lemma 1, as we have a standard deviation $\Sigma \in S^+$ that might not be a real and we are sampling in $gR + c$ which is not a lattice. But we can easily see that $D_{gR+c,\sqrt{\Sigma}} = c + \sqrt{\Sigma} \cdot D_{\Lambda,1,-\frac{c}{\sqrt{\Sigma}}}$, where Λ is the lattice $\frac{gR}{\sqrt{\Sigma}}$. We can then apply Lemma 4 to $D_{\Lambda,1,-\frac{c}{\sqrt{\Sigma}}}$ (this is a Gaussian distribution over a lattice, with standard deviation a scalar). If we prove that

$$\|\mathbb{E}[D_{\Lambda,1,-\frac{c}{\sqrt{\Sigma}}}] + \frac{c}{\sqrt{\Sigma}}\| \leq \mathrm{negl}(n) \tag{20}$$

$$\|\mathbb{V}[D_{\Lambda,1,-\frac{c}{\sqrt{\Sigma}}}] - 1\| \leq \mathrm{negl}(n) \tag{21}$$

then, using the properties of the mean and the variance, we will obtain the desired result of Lemma 1.

Let's apply Lemma 4 to the distribution $D_{\Lambda,1,-c/\sqrt{\Sigma}}$. We have $s = 1$ in the lemma, which gives us the constraint $\eta_\varepsilon(\Lambda) \leq 1/2$. So to get the best possible bound in the lemma, we want to minimize ε under this constraint. We will use the following upper-bound on $\eta_\varepsilon(\Lambda)$ (see for instance [MR04, Lemma 3.3]).

$$\eta_\varepsilon(\Lambda) \leq \sqrt{\frac{\log(2n(1+1/\varepsilon))}{\pi}}\lambda_n(\Lambda).$$

Recall that our lattice Λ is the ideal $\frac{g}{\sqrt{\Sigma}}R \subset K_{\mathbb{R}}$, hence we have $\lambda_n(\Lambda) = \|g/\sqrt{\Sigma}\| = o(1/\sqrt{\log n})$ by hypothesis. In order to get the best possible bound, we take ε such that

$$\sqrt{\frac{\log(2n(1+1/\varepsilon))}{\pi}}\lambda_n(\Lambda) = 1/2.$$

Using the fact that $\lambda_n(\Lambda) = o(1/\sqrt{\log n})$, we obtain that $\varepsilon \leq \frac{1}{n^c}$ for any constant c, i.e. $\varepsilon = \mathrm{negl}(n)$. We can then apply Lemma 4 to obtain Inequalities (20) and (21) (using the fact that $\varepsilon = \mathrm{negl}(n)$ and $s = 1$ in the lemma). This achieves the proof of Lemma 1. □

[19] Note that we do not use the same definition for ρ_{Σ,x_0} as the authors of [MR04], this is why we have some 2π that appears in the bound.

B Security proof of our setting of the GGH map in the weak multilinear map model

In this section, we first recall what is the weak multilinear map model (mentioned first in [MSZ16] and then used in [GMM+16] and in [DGG+16]). Then we prove that the setting we defined in Section 4.1 is secure in the weak multilinear map model, for some notion of security we define here.

B.1 The weak multilinear map model

The idea of the weak multilinear map model ([MSZ16], [GMM+16], [DGG+16]) is to limit the power of the attacker by not giving it the encoded values directly. Instead, an oracle $\mathcal{M}$ keeps a table with the encoded values and allows the attacker to perform only some operations on these encoded values. More formally, an encoded element is a couple $(a, \boldsymbol{v})$, with $a \in R/gR$ and $\boldsymbol{v} \in \{0,1\}^\ell$. Recall that we denote by $\boldsymbol{v}^*$ the vector $(1, 1, \cdots, 1)$.We can perform the following arithmetic operations on the encoded elements:

- **Addition/subtraction.** For any $\boldsymbol{v}$, we have $(a, \boldsymbol{v}) \pm (b, \boldsymbol{v}) = (a \pm b, \boldsymbol{v})$.
- **Multiplication.** If $\boldsymbol{v}$ and $\boldsymbol{w}$ are such that $\boldsymbol{v}[i]\boldsymbol{w}[i] = 0$ for all $i \in [\ell]$, then we have $(a, \boldsymbol{v}) \cdot (b, \boldsymbol{w}) = (a \cdot b, \boldsymbol{v} + \boldsymbol{w})$.
- **Scalar multiplication.** For any $\boldsymbol{v} \in \{0,1\}^\ell$, $a \in R/gR$ and $\alpha \in R$, we have $\alpha \cdot (a, \boldsymbol{v}) = (\alpha \cdot a, \boldsymbol{v})$.

The oracle $\mathcal{M}$ implements the following interfaces.

Initialization. The oracle first initializes the parameters. It sets n to be a power of 2, defines $R = \mathbb{Z}[X]/(X^n + 1)$ and samples g an element of R. The size of the parameters is the same as the one we described in Section 2.2. The oracle $\mathcal{M}$ then receives a set of r couples $(a, \boldsymbol{v})$ to encode. It creates a table T in which it stores the couples $(a, \boldsymbol{v})$ together with a handle h_i it generates, which is independent of the encoded value a but reveals the level of the encoding $\boldsymbol{v}$. Finally, the oracle outputs the handles h_i, for $1 \leq i \leq r$. The oracle $\mathcal{M}$ also creates a table T' for post-zero-test values, that is empty for the moment. This interface has to be called before the other ones, and any attempt to call this procedure more than once will fail.

Operations on encodings. Given two handles h_1, h_2 and an operation $\circ \in \{+, -, \cdot\}$, the oracle first checks whether the handles h_1 and h_2 are in its table. If one of them is not in the table, then in returns $\perp$. Otherwise, let $(a_1, \boldsymbol{v}_1)$ and $(a_2, \boldsymbol{v}_2)$ be the encoded elements associated to these handles. If $\circ \in \{+, -\}$, $\mathcal{M}$ checks whether $\boldsymbol{v}_1 = \boldsymbol{v}_2$. If this is not the case, $\mathcal{M}$ outputs $\perp$. Otherwise, it creates a new entry in its table, with the encoded value $(a_1 \circ a_2, \boldsymbol{v}_1)$ and a new handle h and it outputs h. If $\circ = \cdot$, then $\mathcal{M}$ checks whether $\boldsymbol{v}_1[i]\boldsymbol{v}_2[i] = 0$ for all $i \in [\ell]$. If this is not the case, it outputs $\perp$, otherwise it creates a new entry in its table with the encoded value $(a_1 \cdot a_b, \boldsymbol{v}_1 + \boldsymbol{v}_2)$ and a new handle h and it outputs h.

Multiplication by an element of R. Given a handle h and an element $\alpha \in R$, the oracle $\mathcal{M}$ first checks whether h is in its table T. If it is not, $\mathcal{M}$ outputs $\perp$. Otherwise, let $(a, \boldsymbol{v})$ be the corresponding encoded value. The oracle creates a new entry in its table, with encoded value $(\alpha a, \boldsymbol{v})$ and a new handle h'. Then it outputs h'.

Zero-test query. Given a handle h, the oracle $\mathcal{M}$ checks whether h is in its table. If not $\mathcal{M}$ outputs $\perp$. Otherwise let $(a, \boldsymbol{v})$ be the associated encoded value. If $\boldsymbol{v} \neq \boldsymbol{v}^*$, then $\mathcal{M}$ outputs $\perp$. Otherwise, the oracle checks whether a is a multiple of g. If it is, then $\mathcal{M}$ creates a new entry in its post-zero-test table T', with value $a/g \in R$ and with a new handle h', then it outputs h'. If a is not a multiple of g, then $\mathcal{M}$ outputs $\perp$.

Post-zero-test query. Given a polynomial p of degree polynomial in n,[20] and a bunch of handles $h_1, \cdots, h_t$, the oracle checks whether all handles h_i are in its post-zero-test table T'. If it is not the case, or if p is the identically zero polynomial, it outputs $\perp$. Otherwise, let r_i be the value corresponding to the handle h_i in T'. The oracle computes $p(r_1, \cdots, r_t)$. If this is a non zero multiple of g, then the adversary outputs "WIN", otherwise it outputs $\perp$.

The adversary wins the game if it manages to have the oracle output "WIN" after a post-zero-test query.

Definition 1 (Security of our setting). *We say that our setting of the GGH multilinear map is secure in the weak multilinear map model if any polynomial time adversary has negligible probability to make the oracle output "WIN", when the oracle is initialized with the elements $a_{\boldsymbol{v},i}$ defined in Section 4.1.*

We will now prove that our simple setting defined in Section 4.1 is secure in the weak multilinear map model, for the definition of security given above. This proof does not depend on the sampling method chosen, as long as it has enough min-entropy. As all the sampling methods described in this article have enough min-entropy, our setting will be secure in the weak multilinear map model, independently of the sampling method chosen.

B.2 Mathematical tools

Definition 2. *Let Y be a random variable with values in a set S, the **guessing probability** of the variable Y is*

$$\max_{s \in S} \Pr(Y = s)$$

*and the **min-entropy** of Y is defined by*

$$H_\infty(Y) = -\log(\max_{s \in S} \Pr(Y = s)).$$

For multiple random variables $Y_1, \cdots, Y_k$, we let $p_i(s_1, \cdots, s_{i-1})$ be the guessing probability of Y_i conditioned on $Y_j = s_j$ for $j < i$. Then, define $p_i = \mathbb{E}_{X_1, \cdots, X_{i-1}}[p_i(X_1, \cdots, X_{i-1})]$ to be the expectation of the $p_i(s_1, \cdots, s_{i-1})$. Finally, we denote by $p_{\max}(Y_1, \cdots, Y_k) = \max_i p_i$ the maximum of the p_i's.

For the proof of our theorem, we will use the improved Schwartz-Zippel lemma of [MSZ16, Section 5.1]. The classical Schwartz-Zippel lemma needs independent and uniform random variables, while this version allows us to use random variables that might be correlated and non uniform, as long as they have enough min-entropy. The statement of the lemma is the following:

[20] This restriction on the degree of p may seem a bit unnatural, but this is needed for the proof, and it was already used (and discussed) in [MSZ16].

Lemma 5 (Improved Schwartz-Zippel lemma from [MSZ16]). *Let $\mathbb{F}$ be a field, $k > 0$ be an integer and $P \in \mathbb{F}[X_1, \cdots, X_k]$ be a polynomial of degree at most d. Let $Y_1, \cdots, Y_k$ be random variables in $\mathbb{F}$ (that might be correlated and non uniform). Then we have*

$$\Pr_{Y_1,\cdots,Y_k}[P(Y_1, \cdots, Y_k) = 0] \leq d \cdot p_{\max}(Y_1, \cdots, Y_k).$$

B.3 Security proof

The main idea of the security proof is to use Schwartz-Zippel lemma to show that, except with negligible probability, the adversary will never be able to create encodings that passes the zero-test, except for linear combinations of the ones that are made public in the setting. Then, it is easy to show that the adversary cannot use these encodings to create a multiple of g. This kind of proof was already used to prove security of obfuscators in the weak multilinear map model [GMM+16, DGG+16]. But as our setting is simpler than a candidate obfuscator, the proof will also be easier.

First, we recall and precise how we generate the encodings in our simple setting of the GGH map. Let D_a be a distribution over the elements of R that are invertible modulo g, with min-entropy at least n when reduced modulo g (i.e., for all $x \in (R/gR)^\times$, we have $\Pr_{y \leftarrow D_a}(y = x \mod g) \leq 2^{-n}$). And let D_r be a distribution over R with min-entropy at least n.[21]

For all $\boldsymbol{v} \in \mathcal{A}$ of weight 1, sample m elements $a_{\boldsymbol{v},i} \leftarrow D_a$ independently (with $1 \leq i \leq m$) and $m-1$ elements $a_{\tilde{\boldsymbol{v}},1}, \cdots, a_{\tilde{\boldsymbol{v}},m-1} \leftarrow D_a$ independently. Then, sample $r_{\boldsymbol{v}} \leftarrow D_r$ and let

$$a_{\tilde{\boldsymbol{v}},m} = -\widehat{a}_{\boldsymbol{v},m} \sum_{i=1}^{m-1} a_{\boldsymbol{v},i} a_{\tilde{\boldsymbol{v}},i} + r_{\boldsymbol{v}} g$$

where $\widehat{a}_{\boldsymbol{v},m}$ is an element in R such that $\widehat{a}_{\boldsymbol{v},m} a_{\boldsymbol{v},m} = 1 \mod g$, i.e. $\widehat{a}_{\boldsymbol{v},m}$ is a representative of the inverse of $a_{\boldsymbol{v},m}$ modulo g (chosen arbitrarily). We will initialize the oracle $\mathcal{M}$ with these elements $(a_{\boldsymbol{v},i}, \boldsymbol{v})$, for $\boldsymbol{v} \in \mathcal{A}$.

With the notations above, we have that for all $\boldsymbol{v} \in \mathcal{A}$ of weight 1,

$$H(a_{\boldsymbol{v},1}, a_{\tilde{\boldsymbol{v}},1}, \ldots, a_{\boldsymbol{v},m}, a_{\tilde{\boldsymbol{v}},m}) = \sum_{i=0}^{m} a_{\boldsymbol{v},i} a_{\tilde{\boldsymbol{v}},i} = (r'_{\boldsymbol{v}} + a_{\boldsymbol{v},m} r_{\boldsymbol{v}}) g$$

for some $r'_{\boldsymbol{v}}$ that depends on the $a_{\boldsymbol{v},i}$'s and $a_{\tilde{v},j}$'s (with $i \leq m$ and $j \leq m-1$) but not on $r_{\boldsymbol{v}}$. Hence, as $r_{\boldsymbol{v}}$ has min-entropy at least n and R is an integral domain, we have

$$H(a_{\boldsymbol{v},1}, a_{\tilde{\boldsymbol{v}},1}, \ldots, a_{\boldsymbol{v},m}, a_{\tilde{\boldsymbol{v}},m}) = \tilde{r}_{\boldsymbol{v}} g \tag{22}$$

for some $\tilde{r}_{\boldsymbol{v}}$ with min-entropy at least n. Moreover, knowing the $\tilde{r}_{\boldsymbol{w}}$ for $\boldsymbol{w} \neq \boldsymbol{v}$ ($\boldsymbol{w}$ of weight 1) does not decrease this min-entropy (because $r_{\boldsymbol{v}}$ is independent of the $\tilde{r}_{\boldsymbol{w}}$). Hence, we have that

$$p_{\max}(\{\tilde{r}_{\boldsymbol{v}}\}_{\boldsymbol{v} \in \mathcal{A} \text{ of weight } 1}) \leq 2^{-n}. \tag{23}$$

[21] In the GGH multilinear map, the distribution D_a should be a Gaussian distribution (whose shape depends on the sampling method). This has no importance for our proof, so we make no assumption about it here.

Theorem 3. *Assume we initialize the oracle $\mathcal{M}$ with the couples $(a_{\boldsymbol{v},i}, \boldsymbol{v})$ defined above, for $\boldsymbol{v} \in \mathcal{A}$ and $1 \leq i \leq m$. Then, for any PPT adversary $\mathfrak{A}$ interacting with the oracle $\mathcal{M}$, the probability that $\mathfrak{A}$ manages to make $\mathcal{M}$ output "WIN" is negligible in n. In other words, our simple setting of the GGH multilinear map is secure in the weak multilinear map model (see Definition 1).*

Remark. The conditions on the distributions D_a and D_r are satisfied by all the sampling methods described in this article. Hence, the theorem prove that our simple setting of the GGH multilinear map is secure in the weak multilinear map model, independently of the sampling method chosen (among the ones describes in this article).

Proof. For simplicity of notation, we will sometime index the elements of $\mathcal{A}$ by $\boldsymbol{v_1}, \cdots, \boldsymbol{v_{2\ell}}$.

In this proof, we will merge the arithmetic queries on the encodings and the zero-testing queries by saying that the adversary $\mathfrak{A}$ directly sends a polynomial p to the oracle. Then $\mathcal{M}$ performs the arithmetic operations on the encodings that correspond to the polynomial p (if they are relevant) and apply the zero-testing procedure on the result.

The idea of the proof is to show that the only encodings that $\mathfrak{A}$ can query that will pass the zero-testing procedure are linear combinations of elements of the form $H(a_{\boldsymbol{v},1}, a_{\tilde{\boldsymbol{v}},1}, \ldots, a_{\boldsymbol{v},m}, a_{\tilde{\boldsymbol{v}},m})$ (all other polynomials in the $a_{\boldsymbol{v},i}$'s will fail to pass the zero test with high probability). Then, each zero-test on a $H(a_{\boldsymbol{v},1}, a_{\tilde{\boldsymbol{v}},1}, \ldots, a_{\boldsymbol{v},m}, a_{\tilde{\boldsymbol{v}},m})$ will result in a handle of a random element (because of the randomness contained in $r_{\boldsymbol{v}}$), and all these elements will be independent. Hence the adversary has negligible probability of finding a polynomial that annihilate them.

Lemma 6. *Let P be a polynomial in the variables $(X_{\boldsymbol{v},i})_{\{\boldsymbol{v}\in\mathcal{A}, 1\leq i\leq m\}}$ generated by the attacker $\mathfrak{A}$ such that $P(a_{\boldsymbol{v_1},1}, \cdots, a_{\boldsymbol{v_1},m}, a_{\boldsymbol{v_2},1}, \cdots, a_{\boldsymbol{a_{2\ell}},m}) = 0 \mod g$. Then, with overwhelming probability, we have*

$$P(X_{\boldsymbol{v_1},1}, \cdots, X_{\boldsymbol{v_1},m}, X_{\boldsymbol{v_2},1}, \cdots, X_{\boldsymbol{v_{2\ell}},m}) = \sum_{\substack{\boldsymbol{v}\in\mathcal{A} \\ \|\boldsymbol{v}\|_1=1}} \alpha_{\boldsymbol{v}} H(X_{\boldsymbol{v},1}, X_{\tilde{\boldsymbol{v}},1}, \cdots, X_{\boldsymbol{v},m}, X_{\tilde{\boldsymbol{v}},m})$$

for some scalars $\alpha_{\boldsymbol{v}} \in R$.

This lemma means that the only encodings that will pass the zero test that the attacker can create are the $H(u_{\boldsymbol{v},1}, u_{\tilde{\boldsymbol{v}},1}, \cdots, u_{\boldsymbol{v},m}, u_{\tilde{\boldsymbol{v}},m})$ and linear combination of them. As zero-testing linear combinations of encodings that pass the zero test does not provide more information that what was revealed by zero-testing the original encodings, we will assume in the following that the adversary makes zero testing queries for $H(u_{\boldsymbol{v},1}, u_{\tilde{\boldsymbol{v}},1}, \cdots, u_{\boldsymbol{v},m}, u_{\tilde{\boldsymbol{v}},m})$ for all $\boldsymbol{v} \in \mathcal{A}$ of weight 1, and that they are the only queries that pass the zero test.

Recall that the numerator of $H(u_{\boldsymbol{v},1}, u_{\tilde{\boldsymbol{v}},1}, \cdots, u_{\boldsymbol{v},m}, u_{\tilde{\boldsymbol{v}},m})$ is of the form $H(a_{\boldsymbol{v},1}, a_{\tilde{\boldsymbol{v}},1}, \cdots, a_{\boldsymbol{v},m}, a_{\tilde{\boldsymbol{v}},m}) = \tilde{r}_{\boldsymbol{v}} g$ (see Equality 22). Hence, after all its zero-test queries, the attacker $\mathfrak{A}$ gets handles to the values $\tilde{r}_{\boldsymbol{v}}$ for all $\boldsymbol{v} \in \mathcal{A}$ of weight 1. These are the only post-zero-test handles the attacker obtains. These handles map to random elements $\tilde{r}_{\boldsymbol{v}}$ that may not be independent, but that have a lot of (conditioned) min-entropy. Hence, it is very unlikely that the attacker creates a non zero polynomial that annihilates these random values. More formally, let P be a polynomial of degree $d = \text{poly}(n)$ queried by the attacker in the post-zero-test phase. Then, using using the improved Schwartz-Zippel lemma of [MSZ16] (Lemma 5) in K, we have that

$$\Pr[P(\{\tilde{r}_{\boldsymbol{v}}\}_{\boldsymbol{v}\in\mathcal{A} \text{ of weight } 1}) = 0] \leq d \cdot 2^{-n} = \text{negl}(n)$$

using the fact that $p_{\max}(\{\tilde{r}_{\boldsymbol{v}}\}_{\boldsymbol{v}\in\mathcal{A} \text{ of weight } 1}) \leq 2^{-n}$ (see Inequality (23).

Hence, the attacker has negligible probability of creating a non zero polynomial P, of degree polynomial in n, that annihilates the post-zero-test handles. This concludes the proof of our theorem. □

Proof (Proof of Lemma 6).

Step 1. First, let $\boldsymbol{v} \in \mathcal{A}$ be of weight 1 and let P be a polynomial in the variables $\{X_{\boldsymbol{v},1}, X_{\tilde{\boldsymbol{v}},1}, \cdots, X_{\boldsymbol{v},m}, X_{\tilde{\boldsymbol{v}},m}\}$ (and not in all $X_{\boldsymbol{w},i}$ for $\boldsymbol{w} \in \mathcal{A}$ and $i \leq m$) such that $P(a_{\boldsymbol{v},1}, a_{\tilde{\boldsymbol{v}},1}, \cdots, a_{\boldsymbol{v},m}, a_{\tilde{\boldsymbol{v}},m}) = 0 \mod g$. We will show that $P = \alpha H \mod g$ for some $\alpha \in R/gR$. Note that as we are only interested (for the moment) in equalities modulo g, we will assume that our polynomial P has coefficients in R/gR (which is a field as gR is a prime ideal). We will also see the $a_{\boldsymbol{v},i}$ as elements of R/gR.

Using the fact that the polynomial $P(u_{\boldsymbol{v},1}, u_{\tilde{\boldsymbol{v}},1}, \cdots, u_{\boldsymbol{v},m}, u_{\tilde{\boldsymbol{v}},m})$ is a valid encoding at level $\boldsymbol{v}^*$, we know that

$$P = P_1(X_{\boldsymbol{v},1}, X_{\tilde{\boldsymbol{v}},1}, \cdots, X_{\boldsymbol{v},m}) + X_{\tilde{\boldsymbol{v}},m} P_2(X_{\boldsymbol{v},1}, X_{\boldsymbol{v},2}, \cdots, X_{\boldsymbol{v},m})$$

for some polynomials P_1 of degree 2 and P_2 of degree 1. We cannot apply the Schwartz-Zippel lemma to P because the variable $a_{\tilde{\boldsymbol{v}},m} \mod g$ is completely determined by the other variables that appears in the polynomial. So we first introduce a new polynomial that does not depend on $X_{\tilde{\boldsymbol{v}},m}$ before applying the Schwartz-Zippel lemma.

We define the polynomial $Q \in (R/gR)[X_{\boldsymbol{v},1}, X_{\tilde{\boldsymbol{v}},1}, \cdots, X_{\boldsymbol{v},m}]$ (note that $X_{\tilde{\boldsymbol{v}},m}$ does not appear in Q) by

$$\begin{aligned} Q(X_{\boldsymbol{v},1}, X_{\tilde{\boldsymbol{v}},1}, \cdots, X_{\boldsymbol{v},m}) =& X_{\boldsymbol{v},m} P_1(X_{\boldsymbol{v},1}, X_{\tilde{\boldsymbol{v}},1}, \cdots, X_{\boldsymbol{v},m}) \\ & - \left(\sum_{i=1}^{m-1} X_{\boldsymbol{v},i} X_{\tilde{\boldsymbol{v}},i}\right) P_2(X_{\boldsymbol{v},1}, X_{\boldsymbol{v},2}, \cdots, X_{\boldsymbol{v},m}). \end{aligned}$$

Using the fact that $a_{\tilde{\boldsymbol{v}},m} = -a_{\boldsymbol{v},m}^{-1} \sum_{i=1}^{m-1} a_{\boldsymbol{v},i} a_{\tilde{\boldsymbol{v}},i} \mod g$, we have that

$$Q(a_{\boldsymbol{v},1}, a_{\tilde{\boldsymbol{v}},1}, \cdots, a_{\boldsymbol{v},m}) = a_{\boldsymbol{v},m} P(a_{\boldsymbol{v},1}, a_{\tilde{\boldsymbol{v}},1}, \cdots, a_{\boldsymbol{v},m}, a_{\tilde{\boldsymbol{v}},m}) = 0 \mod g.$$

But the variables $a_{\boldsymbol{v},1}, a_{\tilde{\boldsymbol{v}},1}, \cdots, a_{\boldsymbol{v},m}$ are drown from D_a independently with guessing probability at most 2^{-n} (even when reduced modulo g), and the degree of Q is at most 3. So using Schwartz-Zippel lemma (Lemma 5) in R/gR, we have that Q should be the zero polynomial, except with negligible probability. In the following, we will then assume that $Q = 0$. This means that we have the equality between polynomials $P_1 = X_{\boldsymbol{v},m}^{-1} \left(\sum_{i=1}^{m-1} X_{\boldsymbol{v},i} X_{\tilde{\boldsymbol{v}},i}\right) P_2$. Hence, we can re-write

$$\begin{aligned} P &= P_2 \left(X_{\boldsymbol{v},m}^{-1} \left(\sum_{i=1}^{m-1} X_{\boldsymbol{v},i} X_{\tilde{\boldsymbol{v}},i}\right) + X_{\tilde{\boldsymbol{v}},m} \right) \\ &= P_2 X_{\boldsymbol{v},m}^{-1} \cdot H(X_{\boldsymbol{v},1}, X_{\tilde{\boldsymbol{v}},1}, \cdots, X_{\boldsymbol{v},m}, X_{\tilde{\boldsymbol{v}},m}). \end{aligned}$$

As we know that P is a polynomial and $X_{\boldsymbol{v},m}$ does not divide H, this means that it divides P_2. But P_2 is of degree 1, hence we conclude that $P = \alpha H \mod g$ for some scalar $\alpha \in R$.

Step 2 . Now, let P be a polynomial in all the variables $X_{\boldsymbol{v},i}$ for $\boldsymbol{v} \in \mathcal{A}$ and $i \leq m$ such that $P((a_{\boldsymbol{v},i})_{\boldsymbol{v}\in\mathcal{A},i\leq m}) = 0 \mod g$. We will prove by induction on ℓ (recall that ℓ is the number of $\boldsymbol{v} \in \mathcal{A}$ of weight 1) that $P(X_{\boldsymbol{v_1},1}, \cdots, X_{\boldsymbol{v_1},m}, X_{\boldsymbol{v_2},1}, \cdots, X_{\boldsymbol{v_{2\ell}},m}) = \sum_{\boldsymbol{v}\in\mathcal{A} \text{ of weight } 1} \alpha_{\boldsymbol{v}} H(X_{\boldsymbol{v},1}, X_{\tilde{\boldsymbol{v}},1}, \cdots, X_{\boldsymbol{v},m}, X_{\tilde{\boldsymbol{v}},m}) \mod g$.

The case $\ell = 1$ was already done above (this is exactly step 1). Assume then that $\ell > 1$. Let $\boldsymbol{v}_1 \in \mathcal{A}$ be of weight 1 and assume without loss of generality that $\boldsymbol{v}_2 = \tilde{\boldsymbol{v}}_{\boldsymbol{1}}$. We define the polynomial $\tilde{P}$ in the variables $\{X_{\boldsymbol{v}_1,i}, X_{\boldsymbol{v}_2,j}\}_{i,j\leq m}$ to be the polynomial P where $X_{\boldsymbol{v}_j,i}$ is evaluated at $a_{\boldsymbol{v}_j,i}$ for all $j \geq 3$ and $i \leq m$, i.e.,

$$\tilde{P}(X_{\boldsymbol{v}_1,1}, X_{\boldsymbol{v}_1,2}, \cdots, X_{\boldsymbol{v}_2,m}) = P(X_{\boldsymbol{v}_1,1}, \cdots, X_{\boldsymbol{v}_2,m}, a_{\boldsymbol{v}_3,1}, \cdots, a_{\boldsymbol{v}_{2\ell},m}).$$

By hypothesis, we have that $\tilde{P}(a_{\boldsymbol{v}_1,1}, a_{\boldsymbol{v}_1,1}, \cdots, a_{\boldsymbol{v}_2,m}) = 0 \mod g$. But by step 1, this means that $\tilde{P} = \alpha H \mod g$ for some $\alpha \in R$. Using the structure of the levels of the encodings, we then know that

$$\begin{aligned} P(X_{\boldsymbol{v}_1,1}, \cdots, X_{\boldsymbol{v}_{2\ell,m}}) =& \alpha H(X_{\boldsymbol{v}_1,1}, \cdots, X_{\boldsymbol{v}_2,m}) + \sum_{\boldsymbol{i}=1}^{m} P_i(X_{\boldsymbol{v}_3,1}, \cdots, X_{\boldsymbol{v}_{2\ell},m}) X_{\boldsymbol{v}_1,i} \\ &+ T(X_{\boldsymbol{v}_3,1}, \cdots, X_{\boldsymbol{v}_{2\ell},m}) \end{aligned}$$

for some polynomials P_i and T in R/gR such that $P_i(a_{\boldsymbol{v}_3,1}, \cdots, a_{\boldsymbol{v}_{2\ell},m}) = 0$ and $T(a_{\boldsymbol{v}_3,1}, \cdots, a_{\boldsymbol{v}_{2\ell},m}) = 0$. By induction hypothesis, we then know that the polynomials P_i ans T are linear combination of the polynomial H evaluated at different $X_{\boldsymbol{v},i}$. But then, if P_i is non zero, $P_i(u_{\boldsymbol{v}_3,1}, \cdots, u_{\boldsymbol{v}_{2\ell},m}) u_{\boldsymbol{v}_1,i}$ is an encoding at level $\boldsymbol{v}^* + v_1$ which is not an admissible level. Hence, we have that $P_i = 0$ for all i and, by induction hypothesis on T, we obtain the desired result.

Step 3. We have proven that for all polynomial P that the adversary can query, which passes the zero test, then with overwhelming probability we have

$$P(X_{\boldsymbol{v_1},1}, \cdots, X_{\boldsymbol{v_{2\ell}},m}) = \sum_{\substack{\boldsymbol{v}\in\mathcal{A} \\ \|\boldsymbol{v}\|_1=1}} \alpha_{\boldsymbol{v}} H(X_{\boldsymbol{v},1}, X_{\tilde{\boldsymbol{v}},1}, \cdots, X_{\boldsymbol{v},m}, X_{\tilde{\boldsymbol{v}},m}) + gT$$

for some polynomial T in R. It remains to show that $T = 0$, except with negligible probability. Observe that given a polynomial of the form $\sum_{\substack{\boldsymbol{v}\in\mathcal{A} \\ \|\boldsymbol{v}\|_1=1}} \alpha_{\boldsymbol{v}} H(X_{\boldsymbol{v},1}, X_{\tilde{\boldsymbol{v}},1}, \cdots, X_{\boldsymbol{v},m}, X_{\tilde{\boldsymbol{v}},m}) + gT$ with $T \neq 0$, one can recover a multiple of g. Indeed, in $\sum_{\substack{\boldsymbol{v}\in\mathcal{A} \\ \|\boldsymbol{v}\|_1=1}} \alpha_{\boldsymbol{v}} H(X_{\boldsymbol{v},1}, X_{\tilde{\boldsymbol{v}},1}, \cdots, X_{\boldsymbol{v},m}, X_{\tilde{\boldsymbol{v}},m})$, all monomials $X_{\boldsymbol{v},i} X_{\tilde{\boldsymbol{v}},i}$ have the same coefficient $\alpha_{\boldsymbol{v}}$ when i varies and $\boldsymbol{v}$ is fixed. This means that we can recover a multiple of g by computing the difference of two such coefficient in our polynomial $\sum_{\substack{\boldsymbol{v}\in\mathcal{A} \\ \|\boldsymbol{v}\|_1=1}} \alpha_{\boldsymbol{v}} H(X_{\boldsymbol{v},1}, X_{\tilde{\boldsymbol{v}},1}, \cdots, X_{\boldsymbol{v},m}, X_{\tilde{\boldsymbol{v}},m}) + gT$ (at least one should be non zero if $T \neq 0$). Hence, if the adversary queries such a polynomials, it knows a multiple of g. But the handles output by the oracle reveal nothing about g and g is chosen with sufficiently many min-entropy, hence we will show that the adversary cannot create a multiple of g except with negligible probability.

The idea is that while the attacker performs zero-test queries for polynomials of the form $\sum_{\substack{\boldsymbol{v}\in\mathcal{A} \\ \|\boldsymbol{v}\|_1=1}} \alpha_{\boldsymbol{v}} H(X_{\boldsymbol{v},1}, X_{\tilde{\boldsymbol{v}},1}, \cdots, X_{\boldsymbol{v},m}, X_{\tilde{\boldsymbol{v}},m})$ (without the term gT) and queries that do not pass the

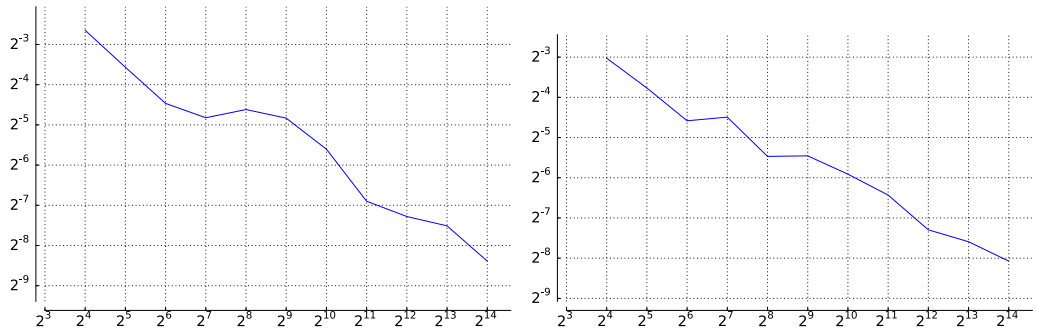
zero-test, it does not learn enough information to obtain a multiple of g with non negligible probability. Hence, it cannot ask for a polynomial of the form $\sum_{\|\boldsymbol{v}\|_1=1,\boldsymbol{v}\in\mathcal{A}} \alpha_{\boldsymbol{v}} H(X_{\boldsymbol{v},1}, X_{\tilde{\boldsymbol{v}},1}, \cdots, X_{\boldsymbol{v},m}, X_{\tilde{\boldsymbol{v}},m}) + gT$ with $T \neq 0$. Assume then that the attacker only queries polynomials of the form $\sum_{\|\boldsymbol{v}\|_1=1,\boldsymbol{v}\in\mathcal{A}} \alpha_{\boldsymbol{v}} H(X_{\boldsymbol{v},1}, X_{\tilde{\boldsymbol{v}},1}, \cdots, X_{\boldsymbol{v},m}, X_{\tilde{\boldsymbol{v}},m})$ and polynomials that do not pass the zero test. We prove that all other possible choices of g, except a negligible fraction of them, would have led to the same answers of the oracle. And hence, the adversary cannot guess the value of g except with negligible probability.

The queries on polynomials of the form $\sum_{\|\boldsymbol{v}\|_1=1,\boldsymbol{v}\in\mathcal{A}} \alpha_{\boldsymbol{v}} H(X_{\boldsymbol{v},1}, X_{\tilde{\boldsymbol{v}},1}, \cdots, X_{\tilde{\boldsymbol{v}},m})$ do not leak any information because for all values of g, we know that they should pass the zero test (and the handle that is generated when a query passes the zero test is independent of the choice of g). The queries on polynomials that do not pass the zero test may leak some information, but this leak will be negligible compared to the entropy of g. Let $c_1, \cdots, c_k$ be the numerators of all encodings that the adversary queried and that did not pass the zero test. Then, any g that is not a divisor of $c_1 c_2 \cdots c_k$ would have given the same answers. So the number of "bad" choices of g is at most the number of divisors of $c_1 c_2 \cdots c_k$. But, denoting by $\mathcal{N}$ the algebraically norm of elements in K, we have that $\mathcal{N}(x) > 2$ if $x \in R$ is non invertible, and if $x|y$, then $\mathcal{N}(x)|\mathcal{N}(y)$. Hence, the number of non invertible divisors of $c_1 c_2 \cdots c_k$ is at most $\log_2(\mathcal{N}(c_1 c_2 \cdots c_k))$. This is polynomial in n. Indeed, the element $c_1 c_2 \cdots c_k$ was computed by a polynomial-time attacker, so its coefficients are bounded by 2^{n^c} for some constant c (the attacker has to write this element with a polynomial number of bits). To conclude, there are only a polynomial number of "bad" g (i.e., a polynomial number of non invertible elements that divide $c_1 c_2 \cdots c_k$). But there is an exponential number of possible g when we generate the parameters of our multilinear map. Hence, the attacker has negligible chance to be able to guess a multiple of g if it knows nothing about it. This achieves the proof of our Lemma 6. □

C Experiments

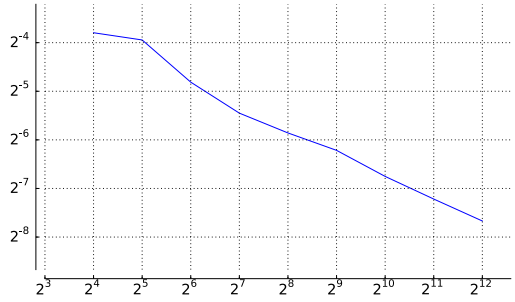
In this section we provide experimental data confirming the heuristic analysis of our attack against the conservative method. More precisely, following this sampling method, we study the empirical mean of $A(z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}})$, and the rate of convergence. We computed $\frac{1}{|\mathcal{A}|}\sum_{\boldsymbol{v}\in\mathcal{A}} A(z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}})$ for several values of n, q and $|\mathcal{A}|$ and wrote it as $\mu_z(1+\varepsilon)$, with $\varepsilon \in K_{\mathbb{R}}$. We plotted $\|\varepsilon\|_\infty$ as a function of $|\mathcal{A}|$ in log-log scale, see Figure 1.

We observe that $\frac{1}{|\mathcal{A}|}\sum_{\boldsymbol{v}\in\mathcal{A}} A(z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}})$ indeed converges to a constant μ_z in $\mathbb{R}^+$. Furthermore, we observe a slope of about $-1/2$ in log-log scale, confirming that the convergence is as fast as what would be given by the Hoeffding bound if the variables were indeed independent: $\|\varepsilon\|_\infty = f(n)/\sqrt{|\mathcal{A}|}$. In fact, it even seems that the function f is decreasing rather than slowly increasing (Hoeffding bound gives $f(n) \leq O(\sqrt{\log n})$). The modulus q seems to have no effect on the relative precision.

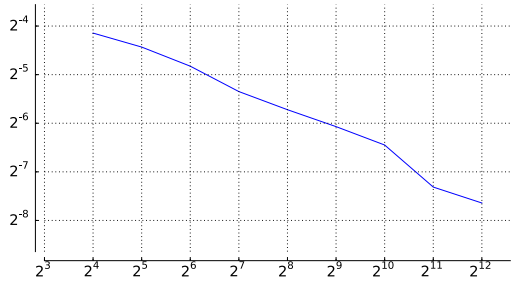


$n = 64,\ q = 12289$

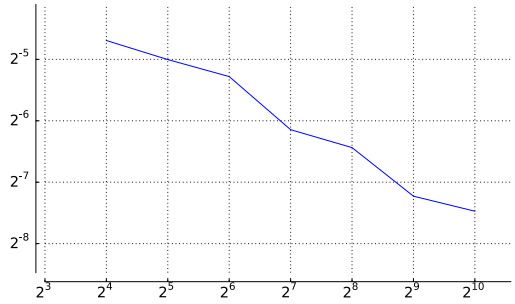
$n = 64,\ q = 1437697$



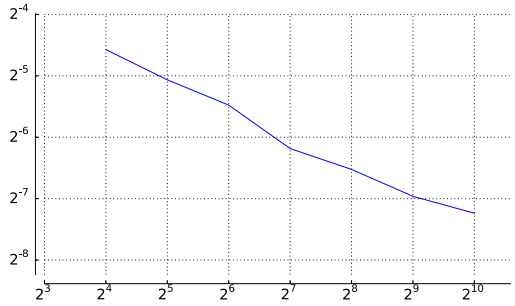
$n = 256,\ q = 12289$



$n = 256,\ q = 1437697$



$n = 1024,\ q = 12289$



$n = 1024,\ q = 1437697$

Fig. 1. Relative precision $\|\varepsilon\|_\infty$ of the empirical mean $\frac{1}{|\mathcal{A}|}\sum_{\boldsymbol{v}\in\mathcal{A}} A(z_{\boldsymbol{v}} z_{\tilde{\boldsymbol{v}}}) = \mu_z(1+\varepsilon)$ (vertical axis) as a function of $|\mathcal{A}|$ (horizontal axis).