Zeroizing without zeroes: Cryptanalyzing multilinear maps without encodings of zero

Preliminary Report

Craig Gentry* Shai Halevi* Hemanta K. Maji† Amit Sahai‡

Abstract

We extend the recent zeroizing attacks of Cheon et al. on multilinear maps to some settings where no encodings of zero below the maximal level are available. Some of the new attacks apply to the CLT scheme (resulting in total break) while others apply to the GGH scheme (resulting in a weak-DL attack).

Keywords: Cryptanalysis, Multilinear Maps.

*IBM Research.

†Department of Computer Science, and Center for Encrypted Functionalities, University of California, Los Angeles. hemanta.maji@gmail.com.

‡Department of Computer Science, and Center for Encrypted Functionalities, University of California, Los Angeles. amitsahai@gmail.com. Research supported in part from a DARPA/ONR PROCEED award, NSF Frontier Award 1413955, NSF grants 1228984, 1136174, 1118096, and 1065276, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11- 1-0389. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government.

1 Introduction

Recent candidates for approximate multilinear attacks [GGH13a, CLT13] have been shown to suffer from "zeroizing" attacks, where the presence of encodings of zero at levels below the zero-test level can be exploited to mount attacks [GGH13a, CHL+14]. Such attacks have been particularly devastating in the context of the CLT candidate [CHL+14].

However, some of the most prominent applications of multilinear maps, such as obfuscation [GGH+13b], do not require any encodings of zero below the zero-test level. (This setting is called the "Multilinear Jigsaw Puzzle" setting in [GGH+13b].) We could therefore optimistically hope that these zeroizing attacks do not apply to this setting.

In this work we show, however, that such the absence of low-level encoding of zero is not by itself a good enough defense against such zeroizing attacks. Specifically, we extend the recent attacks of Cheon et al. [CHL+14], and show how to use them even in some settings where no encodings of zero below the zero-test level are available.

We stress that so far we do not have a working attack on current obfuscation candidates, but our attacks call for renewed effort to either attack these candidates or understand the source of hardness that underlie their security. In particular in this note we also propose a generic attack model which is closer to reality (and in particular captures the essence of our GGH-related attacks), we hope that analysis in this new model would be useful in this effort.

2 The Attack in the CLT setting

The setting. We consider a setting of the CLT multilinear maps, where only certain parameters and encodings are provided to the adversary, and these *do not include* encodings of zero below the maximal level.

Let $p_1, \ldots, p_n$ be the secret primes numbers in the CLT scheme and let $x_0 = p_1 \cdots p_n$. This modulus x_0 is provided to the adversary, together with a single zero-testing parameter $\mathbf{p}_{\mathrm{zt}}$, which is recalled in greater detail below.

An encoding of $(m_1, \ldots, m_n)$ at level t according to this scheme, where $m_i \in [0, g_i)$, is represented by an $m \in \mathbb{Z}_{x_0}$ such that: $m = (r_i g_i + m_i) z^{-t} \mod p_i$, for all $i \in [n]$ and r_i being small random integer in the range $[-2^\rho, 2^\rho)$ and $z \in \mathbb{Z}_{x_0}$ is a secret parameter. This set of equations is represented tersely as: $m = \mathsf{CRT}_{(p_i)}(m'_i)$, where $m'_i = r_i g_i + m_i$. We sometime refer informally to m_i as "the content of the i'th slot" in the encoding.

Suppose the maximum level of supported multi-linearity (i.e. the zero-test level) is some $\kappa \in [3, n]$. Namely, we have a zero-testing parameter $\mathbf{p}_{\mathrm{zt}} = \sum_{i=1}^{n} h_i \cdot (z^\kappa g_i^{-1} \mod p_i) \cdot (x_0/p_i) \mod x_0$.

Encodings needed for attack. Roughly speaking, we consider a setting where for every "slot" we have some encoding that contains zero in that slot, but we do not have any encoding of zero (in all the slots simultaneously), and moreover we cannot compute any such encoding below the zero-test level. Suppose we are provided with the following encodings:

1. Let $a_j \in \mathbb{Z}_{x_0}$, for $j \in [n]$, be such that: $a_j = \mathsf{CRT}_{(p_i)}\left(\frac{a'_{i,j}}{z}\right)$, where:

   - $a'_{i,j} = r_{a,i,j} \cdot g_i + \widehat{a}_{i,j}$,
   - $\widehat{a}_{i,j} \in [0, g_i)$, and
   - $\widehat{a}_{1,j} = 0$ and $\widehat{a}_{i,j} = 0$, for all $i > \kappa$.

   That is, a_j is level 1 encoding of $(0, \widehat{a}_{2,j}, \ldots, \widehat{a}_{\kappa,j}, 0, \ldots, 0)$.

   Remark. We note that in our setting, the 0s after $a_{\kappa,j}$ above are for padding. Our attack naturally extends to other cases where these 0s are replaced with similar structures.

2. Let $b_k \in \mathbb{Z}_{x_0}$, for $k \in [n]$, be such that: $b_k = \mathsf{CRT}_{(p_i)}\left(\frac{b'_{i,k}}{z}\right)$, where:

   - $b'_{i,k} = r_{b,i,k} \cdot g_i + \widehat{b}_{i,k}$,
   - $\widehat{b}_{i,k} \in [0, g_i)$, and
   - $\widehat{b}_{2,k} = 0$ and $\widehat{b}_{i,j} = 0$, for all $i > \kappa$.

   That is, b_k is level 1 encoding of $(\widehat{b}_{1,k}, 0, \widehat{b}_{3,k}, \ldots, \widehat{b}_{\kappa,k}, 0, \ldots, 0)$.

3. Let $c \in \mathbb{Z}_{x_0}$ be such that: $c = \mathsf{CRT}_{(p_i)}\left(\frac{c'_i}{z}\right)$, where:

   - $c'_i = r_{c,i} \cdot g_i + \widehat{c}_i$
   - $\widehat{c}_i \in [0, g_i)$, and
   - $\widehat{c}_3 = 0$ and $\widehat{c}_i = 0$, for all $i > \kappa$.

   That is, c is a level 1 encoding of $(\widehat{c}_1, \widehat{c}_2, 0, \widehat{c}_4, \ldots, \widehat{c}_\kappa, 0, \ldots, 0)$.

4. Let $\tilde{c} \in \mathbb{Z}_{x_0}$ be such that: $\tilde{c} = \mathsf{CRT}_{(p_i)}\left(\frac{\tilde{c}'_i}{z}\right)$, where:

   - $\tilde{c}'_i = r_{\tilde{c},i} \cdot g_i + \widehat{\tilde{c}}_i$
   - $\widehat{\tilde{c}}_i \in [0, g_i)$, and
   - $\widehat{\tilde{c}}_3 = 0$ and $\widehat{\tilde{c}}_i = 0$, for all $i > \kappa$.

   That is, c is a level 1 encoding of $(\widehat{\tilde{c}}_1, \widehat{\tilde{c}}_2, 0, \widehat{\tilde{c}}_4, \ldots, \widehat{\tilde{c}}_\kappa, 0, \ldots, 0)$.

5. Let $d_t \in \mathbb{Z}_{x_0}$, for $4 \leqslant t \leqslant \kappa$, be a level 1 encoding of $(\widehat{d}_{t,1}, \ldots, \widehat{d}_{t,t-1}, 0, \widehat{d}_{t,t+1}, \ldots, \widehat{d}_{t,\kappa}, 0, \ldots, 0)$. As above, we have $d_t = \mathsf{CRT}_{(p_i)}\left(d'_{t,i}/z\right)$, where $d'_{t,i} = r_{d,t,i} g_i + \widehat{d}_{t,i}$. We shall not need these parameters explicitly in our attack description and the proof presented below.

Note that given these encodings, it is not possible to create encodings of zero at any level below κ using the allowed addition and multiplication operations.

The attack. Given, these parameters we present our attach which allows us to factorize x_0 and thereby achieve a total break of CLT. The attack proceeds along the same lines as [CHL+14], but we set up the attack in a different manner to avoid the need for zero encodings, and our algebraic manipulations exploit commutativity within the CRT representation to yield a different final factorization of matrices, but in a manner that still allows to carry out the attack. We now give details of the attack.

We define $d := d_4 \cdots d_\kappa$, which is a level $\kappa - 3$ encoding of a message of form: $(\widehat{d_1}, \widehat{d_2}, \widehat{d_3}, 0, \ldots, 0)$. That is, $d = \mathsf{CRT}_{(p_i)}(d'_i)$ where $d'_i = r_{d,i} g_i + \widehat{d_i}$, $\widehat{d_i} = 0$ for all $i \neq 3$.

Let $\lambda_{j,k} := [a_j b_k d]_{x_0}$. Note that $\lambda_{j,k}$ is a level $\kappa - 1$ encoding of a message of form $(0, 0, \widehat{\lambda}_{j,k}, 0, \ldots, 0)$.

Recalling that the zero-testing parameter is $\mathbf{p}_{\mathrm{zt}} = \sum_{i=1}^{n} h_i \cdot (z^\kappa g_i^{-1} \mod p_i) \cdot p_{-i} \mod x_0$ (where $p_{-i} = x_0/p_i$), we can compute for all i, k:

$$\begin{aligned} w_{j,k}^{(c)} := [c \cdot \lambda_{j,k} \cdot \mathbf{p}_{\mathrm{zt}}]_{x_0} &= \sum_{i=1}^{n} h_i (c \lambda_{j,k} z^\kappa g_i^{-1} \mod p_i) p_{-i} \\ &= \sum_{i=1}^{n} c'_i a'_{i,j} b'_{i,k} \underbrace{h_i d'_i g_i^{-1} p_{-i}}_{=h'_i} \pmod{x_0} \end{aligned}$$

Below we denote $h'_i := [h_i d'_i g_i^{-1} p_{-i}]_{x_0}$ for $i \in [n]$.

Note that in the above equations, h'_1, h'_2 and h'_3 contain multiplicative factors of g_1^{-1}, g_2^{-1} and g_3^{-1}, respectively, but these cancel out with the factors of g_1, g_2, g_3 in $a'_{1,j}, b'_{2,k}, c'_3$, respectively (and similarly the g_i^{-1} factors for $i > 3$ cancel out with the g_i factors in the d'_i's). That is, we can re-write the above equation as:

$$\begin{aligned} w_{j,k}^{(c)} \;=\; & \underbrace{(a'_{1,j} g_1^{-1})}_{=a''_{1,j}} \cdot \underbrace{(h'_1 g_1)}_{=h''_1} \cdot c'_1 b'_{1,k} + a'_{2,j} \cdot \underbrace{(h'_2 g_2)}_{=h''_2} \cdot c'_2 \cdot \underbrace{(b'_{2,k} g_2^{-1})}_{=b''_{2,k}} + a'_{3,j} \cdot \underbrace{(h'_3 g_3)}_{=h''_3} \cdot \underbrace{(c'_3 g_3^{-1})}_{=c''_3} \cdot b'_{3,k} \\ & + \sum_{i=4}^{n} a'_{i,j} h'_i c'_i b'_{i,k} \pmod{x_0} \end{aligned}$$

Crucially, the equality $w_{j,k}^{(c)} = a''_{1,j} h''_1 c'_1 b'_{1,k} + a'_{2,j} h''_2 c'_2 b''_{2,k} + a'_{3,j} h''_3 c''_3 b'_{3,k} + \sum_{i>3} a'_{i,j} h'_i c'_i b'_{i,k}$ holds not just modulo x_0 but also over $\mathbb{Z}$, because both sides are smaller than x_0. To see that the right hand side consists only of small terms, recall that we have

1. $a''_{1,j} = [a'_{1,j} g_1^{-1}]_{x_0} = r_{a,1,j}$, for $j \in [n]$ (since the a_j's have 0 in their first slots).
2. $b''_{2,k} = [b'_{2,k} g_2^{-1}]_{x_0} = r_{b,2,k}$, for $k \in [n]$ (since the b_k's have 0 in their second slots).
3. $c''_3 = [c'_3 g_3^{-1}]_{x_0} = r_{c,3}$ (since c's has 0 in its third slot).
4. $h''_3 = [h'_i g_i]_{x_0} = h_i d'_i p_{-1}$ for $i \in \{1, 2, 3\}$ (by definition of h'_i).
5. For $3 < i \leqslant n$, we have $h'_i = h_i r_{d,i} p_{-i}$ (since d has zeros inall the slots 4 and up).

Let $C^{(c)} := \mathsf{diag}(c'_1, c'_2, c''_3, c'_4, \ldots, c'_n)$. Let $H := \mathsf{diag}(h''_1, h''_2, h''_3, h'_4, \ldots, h'_n)$. Now, we define the matrix $W^{(c)} = \left(w^{(c)}_{j,k}\right)_{j,k\in[n]}$. Then, we can write the following matrix equations over $\mathbb{Z}$:

$$W^{(c)} = \underbrace{\begin{pmatrix} a''_{1,1} & a'_{2,1} & & a'_{n,1} \\ \vdots & & \ddots & \\ a''_{1,n} & a'_{2,n} & & a'_{n,n} \end{pmatrix}}_{=A} \times C^{(c)} \times H \times \underbrace{\begin{pmatrix} b'_{1,1} & \cdots & b'_{1,n} \\ b''_{2,1} & \cdots & b''_{2,n} \\ b'_{3,1} & & b'_{3,n} \\ & \ddots & \\ b'_{n,1} & & b'_{n,n} \end{pmatrix}}_{=B}$$

We can use the same procedure to obtain a similar matrix $W^{(\tilde{c})}$, and we have the matrix equations $W^{(\tilde{c})} = A \times C^{(\tilde{c})} \times H \times B$, also over $\mathbb{Z}$. With high probability over the randomness, both $W^{(c)}$ and $W^{(\tilde{c})}$ are invertible over $\mathbb{Q}$. We now compute the matrix $W = W^{(c)}W^{(\tilde{c})^{-1}}$ over $\mathbb{Q}$ and its eigenvalues. We have:

$$\begin{aligned} W &= (AC^{(c)}HB) \times (AC^{(\tilde{c})}HB)^{-1} \\ &= A \times C^{(c)} \times (C^{(\tilde{c})})^{-1} \times A^{-1} = A \times \begin{pmatrix} c'_1/\tilde{c}'_1 & & 0 \\ & \ddots & \\ 0 & & c'_n/\tilde{c}'_n \end{pmatrix} \times A^{-1}, \end{aligned}$$

so the eigenvalues of W are all the rational numbers $c'_n/\tilde{c}'_n$ (which are distinct whp). Let $\alpha_i/\beta_i = c'_i/\tilde{c}'_i$ be a simplified fraction (i.e. where α_i, β_i are relatively prime), then on one hand we have $\beta_i c_i - \alpha_i \tilde{c}_i = 0$, and on the other hand for all $i \neq i$ we have $\beta_i c_{i'} - \alpha_i \tilde{c}_{i'} \neq 0 \pmod{p_{i'}}$ whp. Hence $\beta_i c - \alpha_i \tilde{c}$ will have non-trivial gcd with x_0, namely p_i. Running over all eigenvalues of W we can recover all the prime factors of x_0.

3 The Attack in the GGH setting

Given the GGH "weak discrete logarithm" attacks [GGH13a], previous works (e.g., [GGH$^+$13b]) employ some counter-measures to avoid providing encodings of zero. In this section we point out that these countermeasures may be insufficient in some cases.

3.1 Counter-Measures to Zeroing Attacks

To avoid providing encoding of zero in the public parameters, it was suggested in some previous work to replace the "bare encoding" of the values of interest by encoding matrices related to these values (e.g. have the desired value as an eigenvalue). For concreteness, in this note we consider the "ordered setting" in which the public parameters include an encoding of various values and we know ahead of time the order in which these values can be multiplied.[1] The attack below easily extends to other settings (in fact it may become a little easier).

[1] This is the case, for example, in the application for branching-program obfuscation, where these values of interest are the randomized matrices.

In more detail, we will assume a GGH-like scheme with plaintext space R/gR for some ring R and element $g \in R$, and ciphertext space R_q for some integer q. We encode elements relative to subsets of $[k] = \{1, \ldots, k\}$, and use $[k]$ itself as our zero-test level. Suppose that our application calls for publishing encoding of elements relative to the singleton sets $\{i\} \subset [k]$ and only needs to multiply elements in the natural order (with encoding relative to $\{1\}$ on the left, followed by encodings relative to $\{2\}$, $\{3\}$, etc.). Further, suppose that we need to re-randomize encodings (at least) relative to the singletons $\{1\}$ and $\{2\}$, so we would like to provide encodings of zero relative to these singletons.

Trying to protect against "zeroizing attacks," we could consider the following solution: we choose a random small vector $\mathbf{s}^* \in R^n$ and invertible matrices $T_0, T_1, \ldots, T_k \in R_q^{n \times n}$ (for some n). We then encode $\alpha \in R/gR$ relative to $\{i\}$ by choosing a random small matrix A^* such that $\mathbf{s}^* \times A^* = \alpha \mathbf{s}^* \pmod{gR}$, and publishing the matrix

$$A^{\{i\}} = \left[T_{i-1} \times A^* \times T_i^{-1}\right]_q .$$

For the purpose of zero-test we choose another mid-size random vector $\mathbf{t}^* \in R^n$ and publish the two vectors $\mathbf{s} = [g^{-1}\mathbf{s}^* \times T_0^{-1}]_q$ and $\mathbf{t} = [T_k \times \mathbf{t}^*]_q$. It is easy to see that this provides the functionality of a graded-encoding scheme, where a level-$[k]$ encoding $A^{[k]}$ can be tested for zero by checking that

$$\left\| \left[\mathbf{s} \times A^{[k]} \times \mathbf{t}\right]_q \right\| \ll q.$$

On the other hand, it seems hard to obtain a native GGH-encoding of zero even if we are given matrices $A^{\{i\}}$ that encode zero, so we could naively hope that the zeroizing attacks from [GGH13a] do not apply.

Where are the z_i's? Note that the denominators z_i of the native GGH encoding are absent from the description above, since they are absorbed into the matrices T_i. One can instead use $A^{\{i\}} = \left[z_i^{-1} \cdot T_{i-1} \times A^* \times T_i^{-1}\right]_q$. and multiply $\mathbf{s}$ by $z = \prod_i z_i$, and everything else in this section would remain unchanged (except the notations which would become a little more cumbersome by the need to specify all these z_i's). Indeed it is easy to see that this does not change the scheme at all. Similarly, the h element from GGH is implicitly defined as $h = \langle \mathbf{t}^*, \mathbf{s}^* \rangle$, which is indeed a mid-size element.

3.2 The Updated Weak-DL Attack

Unfortunately, we show that the counter-measures from above are insufficient to thwart a "weak-discrete-logarithm" attack in the setting above. Specifically, assume that we are given the following encoding matrices:

- Many level-$\{1\}$ encoding of zero, $A_j^{\{1\}} = \left[T_0 \times A_j^* \times T_1^{-1}\right]_q$, $j = 1, 2, \ldots$, s.t. $\mathbf{s}^* A_j^* = \mathbf{0} \pmod{gR}$.
- Many level-$\{2\}$ encoding of zero, $B_j^{\{2\}} = \left[T_1 \times B_j^* \times T_2^{-1}\right]_q$, $j = 1, 2, \ldots$, s.t. $\mathbf{s}^* B_j^* = \mathbf{0} \pmod{gR}$.

- A level-$\{2\}$ encoding of one, $C^{\{2\}} = \left[T_1 \times C^* \times T_2^{-1}\right]_q$ s.t. $\mathbf{s}^* C_j^* = \mathbf{s}^* \pmod{gR}$.
- A level-$\{2\}$ encoding of an unknown element, $D^{\{2\}} = \left[T_1 \times D^* \times T_2^{-1}\right]_q$ s.t. $\mathbf{s}^* D_j^* = \delta \mathbf{s}^* \pmod{gR}$ for some small scalar $\delta \in R$.
- For $S = \{3, \ldots, k\}$, many level-S encoding of nonzero elements, $E_j^S = \left[T_2 \times E_j^* \times T_k^{-1}\right]_q$, $j = 1, 2, \ldots$, s.t. $\mathbf{s}^* E_j^* = \varepsilon_j \mathbf{s}^* \pmod{gR}$ for small scalars $\varepsilon_j \in R$, $\varepsilon_j \notin gR$.

We now show an attack that recovers the coset $\delta + gR$.

3.3 The Attack

Observe that since the $A_j^{\{1\}}$'s are encoding of zero then they cancel the term g^{-1} in the vector $\mathbf{s}$. Hence there exist small vectors $\mathbf{u}_j \in R^n$ such that we have the equality $\mathbf{s}^* A_j^* = g \cdot \mathbf{u}_j$ in R, and therefore $\left[\mathbf{s} \times A_j^{\{1\}} \times T_1\right]_q = \mathbf{s}^* A_j^* / g = \mathbf{u}_j$. Let us also denote below $\mathbf{v}_j = \left[T_2 \times E_j^S \times \mathbf{t}\right]_q$ and notice that $\mathbf{v}_j$ is the small vector $\mathbf{v}_j = E_j^* \mathbf{t}^*$ (equality in R). Then for all j_1, j_2, j_3 we get

$$\begin{aligned}
w_{j_1,j_2,j_3} &=: \left[\mathbf{s} \times A_{j_1}^{\{1\}} \times B_{j_2}^{\{2\}} \times E_{j_2}^S \times \mathbf{t}\right]_q = \mathbf{u}_{j_1} B_{j_2}^* \mathbf{v}_{j_3} \\
x_{j_1,j_3} &=: \left[\mathbf{s} \times A_{j_1}^{\{1\}} \times C^{\{2\}} \times E_{j_2}^S \times \mathbf{t}\right]_q = \mathbf{u}_{j_1} C^* \mathbf{v}_{j_3} \\
y_{j_1,j_3} &=: \left[\mathbf{s} \times A_{j_1}^{\{1\}} \times D^{\{2\}} \times E_{j_2}^S \times \mathbf{t}\right]_q = \mathbf{u}_{j_1} D^* \mathbf{v}_{j_3}.
\end{aligned}$$

with equalities over R.

Let us now denote by U the $n \times n$ matrix over R with the $\mathbf{u}_j$'s as rows and by V the $n \times n$ matrix with the $\mathbf{v}_j$'s as columns, and also define the $n \times n$ matrices $W_1, W_2, \ldots, X$, and Y via:

$$W_j[j_1, j_3] = w_{j_1,j,j_3}, \quad X[j_1, j_3] = x_{j_1,j_3}, \text{ and } Y[j_1, j_3] = y_{j_1,j_3}.$$

Then by definition we have

$$W_j = U \times B_j^* \times V, \quad X = U \times C^* \times V, \text{ and } Y = U \times D^* \times V,$$

again with equalities over R. Computing the determinant of all these matrices and taking the GCD, we get w.h.p. the determinant of $U \times V$, and by dividing it out we get the determinant of all the B_j^*'s. Since $\mathbf{s}^* \times B_j^* \in gR$ for all j, then $det(B_j^*)$ is divisible by g for all j. Hence w.h.p. the scalars $det(B_j^*)$ span the ideal gR, so we can compute this ideal and its order, let us denote this order by p and we assume that p is a prime integer.

Next, we consider the univariate matrix polynomial $M(z) = z \cdot X - Y = U \times (z \cdot C^* - D^*) \times V$. This is a matrix with linear polynomials (over R) in all its entries, and note that evaluating this matrix polynomial at $z = \delta$ would give us a matrix $M(\delta)$ such that $\mathbf{s}^* \times M(\delta) = \mathbf{0} \pmod{gR}$ and therefore $det(M(\delta)) = 0 \pmod{gR}$.

Since $M(z)$ has linear polynomials (over R) in all its entries, then its determinant is a degree-n polynomial over R, and let us denote this polynomial by $P(x) := det(M(x))$. Knowing the

ideal gR and its order p, we can reduce all the coefficients of $P(X)$ modulo gR, using Z_p as the canonical representation of gR. This yields a degree-n polynomial $Q(X)$ over Z_p such that $Q(X) = P(X) \pmod{gR}$. By the above we know that $P(\delta) = 0 \pmod{gR}$, which means that if δ^* is the representative of δ in Z_q then $Q(\delta^*) = 0 \pmod{p}$. We can therefore factor the polynomial Q over Z_p, and recover δ^* as one of its roots, which would give us the coset $\delta^* + gR = \delta + gR$.

3.4 Comments

We note that the attack above can of course be mounted to recover the coset of any encoding, not just at level-$\{2\}$. Also if the matrix C was an encoding of an arbitrary non-zero element γ (rather than an encoding of 1) then the only difference would be that we would recover the coset up to multiplication by the fixed factor $\gamma \bmod p$. We also note that the zero-encoding A_j need not be at level-$\{1\}$, this was only done to simplify the notations.

4 A Refined Generic Model

The attacks that we sketched above point to the inadequacy of the generic graded-encoding model as used in recent work. Indeed these attacks are highly algebraic and yet they are not captured by that generic model. The main difference is that in the generic graded-encoding model the zero-test returns just a 0/1 bit, whereas in the GGH/CLT schemes this test returns a full ring element.

We therefore propose to augment this generic model as follows: In addition to the standard interfaces in the graded-encoding model (with some plaintext space $R' = R/gR$, which we assume is a field), we will now also have a black-box-field over the same space, except that we cannot directly obtain handles to this black-box field. Instead, the zero-test would serve as a translation device, letting us move things from the graded-encoding oracle to the black-box-field oracle.

In more detail, we would have the usual oracles to sample/encode elements in the graded-encoding scheme and to add and multiply them with the usual semantics. However, with each encoded element the graded-encoding oracle will also associate a "random element of R" from the appropriate coset. Namely, with each encoded value α the oracle will also have an associated $r_\alpha \in R$ and the intended semantics is that we use $\alpha + g \cdot r_\alpha$ to represent the coset $\alpha + gR$. The oracle keeps track of the representatives via the addition and multiplication operations of the graded-encoding scheme, by adding and multiplying the representatives in the ring R.

Then, if the zero-test is called on an encoding of zero with representative $g \cdot r \in R$, then in addition to the bit 1 the oracle will also give us a handle to an encoding of $r + gR$ in the black-box field. Namely, if we call it on an element which is divisible by g then it will divide by g and move it to the black-box field.

References

[CHL+14] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. Cryptology ePrint Archive, Report 2014/906, 2014. http://eprint.iacr.org/.

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 476–493. Springer, 2013.

[GGH13a] 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, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2013.

[GGH+13b] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 40–49. IEEE Computer Society, 2013.