# Cryptanalysis of the Multilinear Map over the Integers

Jung Hee Cheon<sup>1</sup>, Kyoohyung Han<sup>1</sup>, Changmin Lee<sup>1</sup>, Hansol Ryu<sup>1</sup>, Damien Stehlé<sup>2</sup>

Seoul National University (SNU), Republic of Korea
 ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, INRIA, UCBL), France.

**Abstract.** We describe a polynomial-time cryptanalysis of the (approximate) multilinear map of Coron, Lepoint and Tibouchi (CLT). The attack relies on an adaptation of the so-called *zeroizing* attack against the Garg, Gentry and Halevi (GGH) candidate multilinear map. Zeroizing is much more devastating for CLT than for GGH. In the case of GGH, it allows to break generalizations of the Decision Linear and Subgroup Membership problems from pairing-based cryptography. For CLT, this leads to a total break: all quantities meant to be kept secret can be efficiently and publicly recovered.

Keywords: Multilinear maps, graded encoding schemes.

### 1 Introduction

Cryptographic bilinear maps, made possible thanks to pairings over elliptic curves, have led to a bounty of exciting cryptographic applications. In 2002, Boneh and Silverberg [BS02] formalized the concept of cryptographic multilinear maps and provided two applications: a one-round key multi-party exchange protocol, and a very efficient broadcast encryption scheme. But these promising applications were only day-dreaming exercises, as no realization of such multilinear maps was known. This was changed about ten years later, as Garg, Gentry and Halevi proposed the first approximation to multilinear maps [GGH13a]. They introduced the concept of (approximate) graded encoding scheme as a variant of multilinear maps, and described a candidate construction relying on ideal lattices (which we will refer to as GGH in this work). Soon after, Coron, Lepoint and Tibouchi [CLT13] proposed another candidate construction of a graded encoding scheme, relying on a variant of the approximate greatest common divisor problem (CLT, for short).

The GGH and CLT constructions share similarities. Both are derived from a homomorphic encryption scheme (Gentry's scheme [Gen09] and the van Dijk et al. scheme [DGHV10], respectively). And both rely on some extra public data, called the zero-testing or extraction parameter, which allows to publicly decide whether the plaintext data hidden in a given encoding is zero, as long as the encoding is not the output of a too deep homomorphic evaluation circuit.

Graded encoding schemes serve as a basis to define presumably hard problems. These problems are then used as security foundations of cryptographic constructions. A major discrepancy between GGH and CLT is that some natural problems seem easy when instantiated with the GGH graded encoding scheme, and hard for CLT. Two such problems are subgroup membership (SubM) and decision linear (DLIN). Roughly speaking, SubM asks to distinguish between encodings of elements of a group and encodings of elements of a subgroup thereof. DLIN consists in determining whether a matrix of elements is singular, given as input encodings of those elements. Another similar discrepancy seems to exist between the asymmetric variants of GGH and CLT: the External Decision Diffie-Hellman (XDH) problem seems hard

for CLT but is easy for GGH. XDH is exactly DDH for one of the components of the asymmetric graded encoding scheme. These problems have been extensively used in the context of cryptographic bilinear maps [Sco02,BBS04,BGN05].

In the first public version of [GGH13a] (dated 29 Oct. 2012),<sup>3</sup> the GGH construction was thought to provide secure DLIN instantiation. It was soon realized that DLIN could be broken in polynomial-time. The attack consists in multiplying an encoding of some element m by an encoding of 0 and by the zero-testing parameter: this produces a small element (because the encoded value is  $m \cdot 0 = 0$ ), which happens to be a multiple of m. This zeroizing attack (also called weak discrete logarithm attack) is dramatic for SubM, DLIN and XDH. Fortunately, it does not seem useful against other problems, such as Graded Decision Diffie Hellman (GDDH), the adaptation of DDH to the graded encoding scheme setting. As no such attack was known for CLT, the presumed hardness of the CLT instantiations of SubM, DLIN and XDH was exploited as a security grounding for several cryptographic constructions [ABP14,Att14,BP13,BLMR13,GGHZ14a,GGHZ14b,GLW14,GLSW14,LMR14,Zha14,Zim14].

Main result. We describe a zeroizing attack on the CLT graded encoding scheme. It runs in polynomial-time, and allows to publicly compute all the parameters of the CLT scheme that were supposed to be kept secret.

Impact of the attack. The CLT candidate construction should be considered broken, unless the low-level encodings of 0 are not made public. At the moment, there does not remain any candidate multilinear map approximation for which any of SubM, DLIN and XDH is hard. Several recent cryptographic constructions cannot be realized anymore: this includes all constructions from [Att14,GGHZ14a,GGHZ14b,Zha14], the GPAKE construction of [ABP14] for more than 3 users, one of the two constructions of password hashing of [BP13], the alternative key-homomorphic PRF construction from [BLMR13], and the use of the latter in [LMR14].

Our attack heavily relies on the fact that low-level encodings of 0 are made publicly available. It is not applicable if these parameters are kept secret. They are used in applications to homomorphically re-randomize encodings, in order to "canonicalize" their distributions. A simple way to thwart the attack is to not make any low-level encoding of 0 public. This approach was used in [GGH<sup>+</sup>13b] and [BR13], for example. It seems that this approach can be used to secure the construction from [Zim14] as well.

**Related works.** A third candidate construction of a variant of graded encoding schemes was recently proposed in [GGH14]. In that scheme, no encoding of 0 is provided, as it would incur serious security issues (see [GGH14, Se. 4]).

Our attack was extended in [BWZ14,GHMS14] to settings in which no low-level encoding of 0 is available. The extensions rely on low-level encodings of elements corresponding to orthogonal vectors, and impact [GLW14,GLSW14].

After our attack was published, the draft [GGHZ14a] was updated, to propose a candidate immunization against our attack (see [GGHZ14a, Se. 6]).<sup>4</sup> Another candidate immunization was proposed in [BWZ14]. Both immunizations have been showed insecure in [CLT14a].

**Open problems.** A natural line of research is to extend the range of applications of graded encoding schemes for which the encodings of zero are not needed.

<sup>&</sup>lt;sup>3</sup> It can be accessed from the IACR eprint server.

<sup>&</sup>lt;sup>4</sup> The former version that was impacted by our attack can still be accessed from the IACR eprint server.

Publishing encodings of zero as well as a zero-test parameter can lead to damaging consequences (total break of CLT, weakness of SubM, DLIN and XDH for GGH). An impossibility result would be fascinating.

**Organization.** In Section 2, we recall the CLT scheme and the zeroizing attack against GGH. In Section 3, we present our attack on CLT.

#### 2 Preliminaries

**Notation.** We use  $a \leftarrow A$  to denote the operation of uniformly choosing an element a from a finite set A. We define  $[n] = \{1, 2, ..., n\}$ . We let  $\mathbb{Z}_q$  denote the ring  $\mathbb{Z}/(q\mathbb{Z})$ . For pairwise coprime integers  $p_1, p_2, ..., p_n$ , we define  $\mathsf{CRT}_{(p_1, p_2, ..., p_n)}(r_1, r_2, ..., r_n)$  (abbreviated as  $\mathsf{CRT}_{(p_i)}(r_i)$ ) as the unique integer in  $\left(-\frac{1}{2}\prod_{i=1}^n p_i, \frac{1}{2}\prod_{i=1}^n p_i\right]$  which is congruent to  $r_i \mod p_i$  for all  $i \in [n]$ . We use the notation  $[t]_p$  for integers t and p to denote the reduction of t modulo p into the interval (-p/2, p/2].

We use lower-case bold letters to denote vectors whereas upper-case bold letters are used to denote matrices. For matrix  $\mathbf{S}$ , we denote by  $\mathbf{S}^T$  the transpose of  $\mathbf{S}$ . We define  $\|\mathbf{S}\|_{\infty} = \max_i \sum_{j \in [n]} |s_{ij}|$ , where  $s_{ij}$  is the (i,j) component of  $\mathbf{S}$ . Finally we denote by  $\operatorname{diag}(a_1,\ldots,a_n)$  the diagonal matrix with diagonal coefficients equal to  $a_1,\ldots,a_n$ .

### 2.1 A Candidate Multilinear Map over the Integers

First, we briefly recall the Coron *et al.* construction. We refer to the original paper [CLT13] for a complete description.

The scheme relies on the following parameters.

- $\lambda$ : the security parameter
- $\kappa$ : the multilinearity parameter
- $\rho$ : the bit length of the randomness used for encodings
- $\alpha$ : the bit length of the message slots
- $\eta$ : the bit length of the secret primes  $p_i$
- n: the number of distinct secret primes
- $\tau$ : the number of level-1 encodings of zero in public parameters
- $\ell$ : the number of level-0 encodings in public parameters
- $\nu$ : the bit length of the image of the multilinear map
- $\beta$ : the bit length of the entries of the zero-test matrix H

Coron et al. suggested to set the parameters so that the following conditions are met:

- $\rho = \Omega(\lambda)$ : to avoid brute force attack (see also [LS14] for a constant factor improvement).
- $\alpha = \lambda$ : so that the ring of messages  $\mathbb{Z}_{g_1} \times \ldots \times \mathbb{Z}_{g_n}$  does not contain a small subring  $\mathbb{Z}_{g_i}$ .
- $n = \Omega(\eta \cdot \lambda)$ : to thwart lattice reduction attacks.
- $\ell \geq n \cdot \alpha + 2\lambda$ : to be able to apply the leftover hash lemma from [CLT13, Le. 1].
- $\tau \ge n \cdot (\rho + \log_2(2n)) + 2\lambda$ : to apply leftover hash lemma from [CLT13, Se. 4].
- $\beta = \Omega(\lambda)$ : to avoid the so-called gcd attack.
- $\eta \ge \rho_{\kappa} + \alpha + 2\beta + \lambda + 8$ , where  $\rho_{\kappa}$  is the maximum bit size of the random  $r_i$ 's a level- $\kappa$  encoding. When computing the product of  $\kappa$  level-1 encodings and an additional level-0 encoding, one obtains  $\rho_{\kappa} = \kappa \cdot (2\alpha + 2\rho + \lambda + 2\log_2 n + 2) + \rho + \log_2 \ell + 1$ .
- $\nu = \eta \beta \rho_f \lambda 3$ : to ensure zero-test correctness.

<sup>&</sup>lt;sup>5</sup> In fact, it seems that making the primes  $g_i$  public, equal, and  $\Omega(\kappa)$  may not lead to any specific attack [CLT14b].

Instance generation: (params,  $p_{zt}$ )  $\leftarrow$  InstGen $(1^{\lambda}, 1^{\kappa})$ . Set the scheme parameters as explained above. For  $i \in [n]$ , generate  $\eta$ -bit primes  $p_i$ ,  $\alpha$ -bit primes  $g_i$ , and compute  $x_0 = \prod_{i \in [n]} p_i$ . Sample  $z \leftarrow \mathbb{Z}_{x_0}$ . Let  $\Pi = (\pi_{ij}) \in \mathbb{Z}^{n \times n}$  with  $\pi_{ij} \leftarrow (n2^{\rho}, (n+1)2^{\rho}) \cap \mathbb{Z}$  if i = j, otherwise  $\pi_{ij} \leftarrow (-2^{\rho}, 2^{\rho}) \cap \mathbb{Z}$ . For  $i \in [n]$ , generate  $\mathbf{r}_i \in \mathbb{Z}^n$  by choosing randomly and independently in the half-open parallelepiped spanned by the columns of the matrix  $\Pi$  and denote by  $r_{ij}$  the j-th component of  $\mathbf{r}_i$ . Generate  $\mathbf{H} = (h_{ij}) \in \mathbb{Z}^{n \times n}$ ,  $\mathbf{A} = (a_{ij}) \in \mathbb{Z}^{n \times \ell}$  such that  $\mathbf{H}$  is invertible and  $\|\mathbf{H}^T\|_{\infty} \leq 2^{\beta}$ ,  $\|(\mathbf{H}^{-1})^T\|_{\infty} \leq 2^{\beta}$  and for  $i \in [n]$ ,  $j \in [\ell]$ ,  $a_{ij} \leftarrow [0, g_i)$ . Then define:

$$y = \mathsf{CRT}_{(p_i)}\left(\frac{r_i g_i + 1}{z}\right), \text{ where } r_i \leftarrow (-2^\rho, 2^\rho) \cap \mathbb{Z} \text{ for } i \in [n],$$

$$x_j = \mathsf{CRT}_{(p_i)}\left(\frac{r_{ij} g_i}{z}\right) \text{ for } j \in [\tau],$$

$$x_j' = \mathsf{CRT}_{(p_i)}(x_{ij}'), \text{ where } x_{ij}' = r_{ij}' g_i + a_{ij} \text{ and } r_{ij}' \leftarrow (-2^\rho, 2^\rho) \cap \mathbb{Z} \text{ for } i \in [n], j \in [\ell],$$

$$(\boldsymbol{p}_{zt})_j = \left[\sum_{i=1}^n \left[h_{ij} \cdot (z^\kappa \cdot g_i^{-1}]_{p_i} \cdot \prod_{i' \neq i} p_{i'}\right]_{x_0} \text{ for } j \in [n].$$

Output params =  $(n, \eta, \alpha, \rho, \beta, \tau, \ell, \nu, y, \{x_j\}, \{x'_j\}, \{\Pi_j\}, s)$  and  $p_{zt}$ . Here s is a seed for a strong randomness extractor, which is used for an "Extraction" procedure. We do not recall the latter as it is not needed to describe our attack.

Re-randomizing level-1 encodings:  $c' \leftarrow \text{reRand}(\text{params}, c)$ . For  $j \in [\tau], i \in [n]$ , sample  $b_j \leftarrow \{0, 1\}, b'_i \leftarrow [0, 2^{\mu}) \cap \mathbb{Z}$ , with  $\mu = \rho + \alpha + \lambda$ . Return  $c' = [c + \sum_{j \in [\tau]} b_j \cdot x_j + \sum_{i \in [n]} b'_i \cdot \Pi_i]_{x_0}$ . Note that this is the only procedure in the CLT multilinear map that uses the  $x_j$ 's.<sup>6</sup>

Adding and multiplying encodings:  $Add(c_1, c_2) = [c_1 + c_2]_{x_0}$  and  $Mul(c_1, c_2) = [c_1 \cdot c_2]_{x_0}$ .

**Zero-testing:** is Zero(params,  $p_{zt}, u_{\kappa}$ ) =  $^{?}$  0/1. Given a level- $\kappa$  encoding c, return 1 if  $||[p_{zt} \cdot c]_{x_0}||_{\infty} < x_0 \cdot 2^{-\nu}$ , and return 0 otherwise.

Coron et al. also described a variant where only one such  $(\mathbf{p}_{zt})_j$  is given out, rather than n of them (see [CLT13, Se. 6]). Our attack requires only one  $(\mathbf{p}_{zt})_j$ . In [GLW14, App. B.3], Gentry et al. described a variant of the above construction that aims at generalizing asymmetric cryptographic bilinear maps. Our attack can be adapted to that variant.

#### 2.2 Zeroizing Attack on GGH

As a warm-up before describing the zeroizing attack on CLT, we recall the zeroizing attack on GGH.

Garg et al. constructed the first approximation to multilinear maps, by using ideal lattices [GGH13a]. They used the polynomial ring  $R = \mathbb{Z}[x]/(x^n + 1)$  and a (prime) principal ideal  $\mathcal{I} = \langle \boldsymbol{g} \rangle \subseteq R$ , where  $\boldsymbol{g}$  is a secret short element. They also chose an integer parameter q and another random secret  $\boldsymbol{z} \in R_q = R/(qR)$ . Then one can encode an element of  $R/\mathcal{I}$ , via division by  $\boldsymbol{z}$  in  $R_q$ . More precisely, a level-i encoding of the coset  $\boldsymbol{e} + \mathcal{I}$  is an element of the form  $[\boldsymbol{c}/\boldsymbol{z}^i]_q$ , where  $\boldsymbol{c} \in \boldsymbol{e} + \mathcal{I}$  is short. By publishing a zero-testing parameter, any user can decide whether two elements encode the same coset or not.

<sup>&</sup>lt;sup>6</sup> This procedure can be adapted to higher levels  $1 < k \le \kappa$  by publishing appropriate quantities in params.

The zero-testing parameter is  $p_{zt} = [h \cdot z^{\kappa}/g]_q$ , where h is appropriately small. For a given level- $\kappa$  encoding  $u = [c/z^{\kappa}]_q$ , the quantity  $[u \cdot p_{zt}]_q = [h \cdot c/g]_q$  is small if and only if  $c \in \mathcal{I}$ , i.e., u is an encoding of zero.

The latter creates a weakness in the scheme, which enables to solve the Subgroup Membership (SubM) and the decision linear (DLIN) problems easily, by so-called "zeroizing" attack. It uses the property that an encoding of zero has small value when it is multiplied by the zero-testing parameter. In that case, the reduction modulo q is vacuous, and one can have equations over R (instead of  $R_q$ ) and compute some fixed multiples of secrets. The attack procedure can be summarized as follows (and refer the reader to [GGH13a] for a more detailed description). It relies on the following public parameters:

- $y = [a/z]_q$ , with  $a \in 1 + \mathcal{I}$  and a small, a level-1 encoding of 1,
- $x_j = [b_j g/z]_q$ , with  $b_j$  small, a level-1 encoding of 0,
- $p_{zt} = [hz^{\kappa}/g]_q$ , with  $h \in R$  appropriately small, the zero-testing parameter.

**Step 1:** Compute level- $\kappa$  encodings of zero and get the equations in R by multiplying by the zero-testing parameter.

Let  $u = d/z^t$  be a level-t encoding of some message  $d \mod \mathcal{I}$ . Then compute

$$egin{aligned} oldsymbol{f} &:= [oldsymbol{u} \cdot oldsymbol{x}_j \cdot oldsymbol{p}_{zt} \cdot oldsymbol{y}^{\kappa-t-1}]_q = \left[ rac{oldsymbol{d}}{oldsymbol{z}} \cdot rac{oldsymbol{b}_j \cdot oldsymbol{g}}{oldsymbol{z}} \cdot rac{oldsymbol{a}^{\kappa-t-1}}{oldsymbol{z}} 
ight]_q \ &= \underbrace{oldsymbol{d} \cdot oldsymbol{b}_j \cdot oldsymbol{h} \cdot oldsymbol{a}^{\kappa-t-1}}_{\ll q}. \end{aligned}$$

Note that the last term in the above equation consists of only small elements, so that the equality holds without modulus reduction by q. Therefore we can obtain various multiples of h (in R) for various u and  $x_i$ .

**Step 2:** From multiples of h, compute a basis of  $\langle h \rangle$ . Using a similar procedure, compute a basis of  $\langle h \cdot g \rangle$ , and hence a basis for  $\mathcal{I}$  (by dividing  $\langle h \cdot g \rangle$  by  $\langle h \rangle$ ).

SubM is as follows: Given a level-1 encoding  $u = [d/z]_q$ , assess whether  $d \in \langle g_1 \rangle$ , where  $g = g_1 \cdot g_2$  (note that in this context,  $\mathcal{I}$  is not a prime ideal). Using the above method, we can get  $f = d \cdot \Delta$  for some  $\Delta$  (which is unrelated to g). Taking the gcd of  $\langle f \rangle$  and  $\mathcal{I}$ , we easily solve the subgroup membership problem.

DLIN is as follows: Given level-t encodings  $C = (c_{ij})_{i,j \in [N]}$  of messages  $M = (m_{ij})_{i,j \in [N]}$  for some  $t < \kappa$  and  $N > \kappa/t$ , assess whether the rank of M (over the field  $R/\mathcal{I}$ ) is full or not. Using the above, we can compute  $M \cdot \Delta$  for some scalar  $\Delta \in R/\mathcal{I}$  which is unlikely to be 0. In that case, the matrices  $M \cdot \Delta$  and M have equal rank, and the problem is easy to solve.

## 3 A Zeroizing Attack on CLT

The first step of the attack is similar to that of the zeroizing attack of GGH. We compute many level- $\kappa$  encodings of zero and multiply them by the zero-testing parameter. Then we get matrix equations over  $\mathbb{Q}$  (not reduced modulo  $x_0$ ). By adapting the latter to CLT, one

 $<sup>^{7}</sup>$  If N is smaller than that, the problem is not interesting as it can always be solved efficiently using the zero-test parameter.

would obtain samples from the ideal  $\langle h_1, \ldots, h_n \rangle \subseteq \mathbb{Z}$ . Most of the time, it is the whole  $\mathbb{Z}$ , and the samples do not contain any useful information. Instead, we form matrix equations by using several  $x_i$ 's rather than a single one.

These equations share common terms. The second step of the attack is to remove some of these common terms by computing the ratio (over the rationals) between two such equations, and to extract the ratios of the CRT components of the underlying plaintexts by computing the eigenvalues.

The third step consists in recovering the  $p_i$ 's from these CRT components. Once the  $p_i$ 's are obtained, recovering the other secret quantities is relatively straightforward.

Now we give full details of each step.

## 3.1 Constructing Matrix Equations over $\mathbb{Z}$

Let  $t \leq \kappa - 1$ . Let c be a level-t encoding of  $(m_1^{(c)}, \ldots, m_n^{(c)})$ , i.e.,  $c = c_i/z^t \mod p_i$  and  $c_i = m_i^{(c)}$  for all  $i \in [n]$ . Then we can compute the following quantities using the public parameters (for  $j \in [\ell], k \in [\tau]$ ):

$$\begin{aligned} w_{jk} &:= \left[ c \cdot x_j' x_k \cdot y^{\kappa - t - 1} \cdot (\boldsymbol{p}_{zt})_1 \right]_{x_0} = \left[ \sum_{i=1}^n \left[ h_{i1} \cdot c \cdot x_j' x_k y^{\kappa - t - 1} z^{\kappa} g_i^{-1} \right]_{p_i} \cdot \frac{x_0}{p_i} \right]_{x_0} \\ &= \left[ \sum_{i=1}^n h_{i1} c_i x_{ij}' r_{ik} (r_i g_i + 1)^{\kappa - t - 1} \cdot \frac{x_0}{p_i} \right]_{x_0} \\ &= \left[ \sum_{i=1}^n x_{ij}' h_i' c_i r_{ik} \right]_{x_0}, \end{aligned}$$

where  $h'_i := h_{i1}(r_i g_i + 1)^{\kappa - t - 1} x_0 / p_i$  for  $i \in [n]$ .

Now, as c is a level-t encoding, then  $x'_j \cdot (c \cdot x_k \cdot y^{\kappa - t - 1})$  is a valid level- $\kappa$  Diffie-Hellman product (i.e., a product of one level-0 encoding and  $\kappa$  level-1 encodings). Further, it is an encoding of 0, as  $x_k$  is an encoding of 0. By design, we have that  $|w_{jk}|$  is much smaller than  $x_0$  (this may be checked by a tedious computation, but this is exactly how the correctness requirement for the zero-test parameter is derived). As a result, the equation  $w_{jk} = \sum_{i \in [n]} x'_{ij} h'_i c_i r_{ik}$  holds over the integers.

This equation can be rewritten as follows:

$$w_{jk} = (x'_{1j}, \dots, x'_{nj}) \cdot \mathsf{diag}(c_1, \dots, c_n) \cdot \mathsf{diag}(h'_1, \dots, h'_n) \cdot (r_{1k}, \dots, r_{nk})^T$$
.

By letting the index pair (j,k) vary in  $[n] \times [n]$ , we obtain a matrix equation involving the following matrix  $\mathbf{W}_c = (w_{jk}) \in \mathbb{Z}^{n \times n}$ .

$$\mathbf{W}_{c} = \begin{pmatrix} x'_{11} & \cdots & x'_{n1} \\ & \ddots & \\ x'_{1n} & \cdots & x'_{nn} \end{pmatrix} \begin{pmatrix} c_{1} & 0 \\ & \ddots & \\ 0 & c_{n} \end{pmatrix} \begin{pmatrix} h'_{1} & 0 \\ & \ddots & \\ 0 & h'_{n} \end{pmatrix} \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ & \ddots & \\ r_{n1} & \cdots & r_{nn} \end{pmatrix} \\
= \mathbf{X}' \qquad \operatorname{diag}(c_{1}, \dots, c_{n}) \operatorname{diag}(h'_{1}, \dots, h'_{n}) \qquad \mathbf{R}.$$
(1)

To build these equations, we need sufficiently many  $x_j'$ 's and  $x_k$ 's. Namely, we need  $\ell \geq n$  and  $\tau \geq n$ . The design conditions on  $\ell$  and  $\tau$  ensure that this is the case.

Note that the only component in the right hand side of Equation (1) that depends on c is  $diag(c_1, \ldots, c_n)$ : the matrices X', R and  $diag(h'_1, \ldots, h'_n)$  are independent of c.

## 3.2 Breaking into the CRT Decomposition

We now take t = 0, and instantiate Equation (1) twice, with  $c = x'_1$  and  $c = x'_2$ . We obtain, for  $j \in \{1, 2\}$ :

$$W_i := X' \cdot \mathsf{diag}(x'_{1i}, \dots, x'_{ni}) \cdot \mathsf{diag}(h'_1, \dots, h'_n) \cdot R.$$

We can then compute (over  $\mathbb{Q}$ ):

$$\boldsymbol{W}_1 \cdot \boldsymbol{W}_2^{-1} = \boldsymbol{X}' \cdot \operatorname{diag}\left(\frac{x'_{11}}{x'_{12}}, \dots, \frac{x'_{n1}}{x'_{n2}}\right) \boldsymbol{X}'^{-1}.$$

In the latter, we need that  $W_2$  is invertible. Below, we will also need that  $W_1$  is invertible. We argue here that we may assume this is the case. We prove it for  $W_1$ . Note first that the  $x'_{i1}$ 's and the  $h'_{i}$ 's are all non-zero, with overwhelming probability. Note that by design, the matrix  $(r_{ij})_{i\in[n],j\in[\tau]}$  has rank n (see [CLT13, Se. 4]). The same holds for the matrix  $(x'_{ij})_{i\in[n],j\in[\ell]}$  (see [CLT13, Le. 1]). As we can compute the rank of a  $W_c \in \mathbb{Z}^{t\times t}$  obtained by using an  $X' \in \mathbb{Z}^{t\times n}$  and an  $R \in \mathbb{Z}^{n\times t}$  obtained by respectively using a t-subset of the  $x'_{j}$ 's and a t-subset of the  $x_{j}$ 's, without loss of generality we may assume that our  $X', R \in \mathbb{Z}^{n\times n}$  are non-singular. The cost of finding such a pair (X', R) is bounded as  $\widetilde{\mathcal{O}}((\tau+\ell)\cdot(n^{\omega}\log x_0))=\widetilde{\mathcal{O}}(\kappa^{\omega+3}\lambda^{2\omega+6})$ , with  $\omega\leq 2.38$  (assuming all parameters are set smallest possible so that the bounds of Subsection 2.1 hold). Here we used the fact that the rank of a matrix  $A\in\mathbb{Z}^{n\times n}$  may be computed in time  $\widetilde{\mathcal{O}}(n^{\omega}\log\|A\|_{\infty})$  (see [Sto09]). This dominates the overall cost of the attack.

As X' is non-singular, we obtain that the  $x'_{i1}/x'_{i2}$ 's are the eigenvalues (over  $\mathbb{Q}$ ) of  $W_1 \cdot W_2^{-1}$ . These may be computed in polynomial-time from  $W_1 \cdot W_2^{-1}$  (e.g., by factoring the characteristic polynomial). We hence obtain the  $x'_{i1}/x'_{i2}$ 's, for all  $i \in [n]$ , possibly in a permuted order. We write the fraction  $x'_{i1}/x'_{i2}$  as  $x''_{i1}/x''_{i2}$ , with co-prime  $x''_{i1}$  and  $x''_{i2}$ . At this stage, we have the  $(x''_{i1}, x''_{i2})$ 's at hand, for all  $i \in [n]$ . For each of these pairs, we compute:

$$\gcd(x_{i1}'' \cdot x_2' - x_{i2}'' \cdot x_1', x_0).$$

The prime  $p_i$  is a common factor of both  $x''_{i1} \cdot x'_2 - x''_{i2} \cdot x'_1$  and  $x_0$ . As all the other factors of  $x_0$  are huge, there is a negligible probability that the gcd is not exactly  $p_i$ : another  $p_j$  divides  $x''_{i1} \cdot x'_2 - x''_{i2} \cdot x'_1$  if and only if  $x'_{i1} \cdot x'_{j2} = x'_{i2} \cdot x'_{j1}$ .

#### 3.3 Disclosing all the Secret Quantities

At this stage, we know all the  $p_i$ 's.

Let  $j \in [\tau]$ . We have  $x_j/y = r_{ij}g_i/(r_ig_i+1) \mod p_i$ . As the numerator and denominator are coprime and very small compared to  $p_i$ , they can be recovered by rational reconstruction. We hence obtain  $r_{ij}g_i$  for all j. The gcd of the  $(r_{ij}g_i)$ 's reveals  $g_i$ . As a result, we can also recover all the  $r_{ij}$ 's and  $r_i$ 's.

As  $x_1 = r_{i1}g_i/z \mod p_i$  and as the numerator is known, we can recover  $z \mod p_i$  for all i, and hence  $z \mod x_0$ . The  $h_{ij}$ 's can then be recovered as well. So can the  $r'_{ij}$ 's and  $a_{ij}$ 's.

Acknowledgments. The authors thank Michel Abdalla, Jean-Sébastien Coron, Shai Halevi, Adeline Langlois, Tancrède Lepoint, Benoît Libert, Alon Rosen, Gilles Villard and Joe Zimmerman for helpful discussions. The first four author were supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2014R1A2A 1A11050917). The last author was supported by the ERC Starting Grant ERC-2013-StG-335086-LATTAC.

#### References

- [ABP14] M. Abdalla, F. Benhamouda, and D. Pointcheval. Disjunctions for hash proof systems: New constructions and applications. *IACR Cryptology ePrint Archive*, 2014:483, 2014.
- [Att14] N. Attrapadung. Fully secure and succinct attribute based encryption for circuits from multi-linear maps. IACR Cryptology ePrint Archive, 2014:772, 2014.
- [BBS04] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Proc. of CRYPTO*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004.
- [BGN05] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. In Proc. of TCC, volume 3378 of LNCS, pages 325–341. Springer, 2005.
- [BLMR13] D. Boneh, K. Lewi, H. Montgomery, and A. Raghunathan. Key homomorphic PRFs and their applications. In *Proc. of CRYPTO*, pages 410–428. Springer, 2013.
- [BP13] F. Benhamouda and D. Pointcheval. Verifier-based password-authenticated key exchange: New models and constructions. *IACR Cryptology ePrint Archive*, 2013:833, 2013.
- [BR13] Z. Brakerski and G. N. Rothblum. Obfuscating conjunctions. In Proc. of CRYPTO, volume 8043 of LNCS, pages 416–434. Springer, 2013.
- [BS02] D. Boneh and A. Silverberg. Applications of multilinear forms to cryptography. Contemporary Mathematics, 324:71–90, 2002.
- [BWZ14] D. Boneh, D. J. Wu, and J. Zimmerman. Immunizing multilinear maps against zeroizing attacks. IACR Cryptology ePrint Archive, 2014:930, 2014.
- [CLT13] J.-S. Coron, T. Lepoint, and M. Tibouchi. Practical multilinear maps over the integers. In Proc. of CRYPTO, pages 476–493. Springer, 2013.
- [CLT14a] J.-S. Coron, T. Lepoint, and M. Tibouchi. Cryptanalysis of two candidate fixes of multilinear maps over the integers. *IACR Cryptology ePrint Archive*, 2014:975, 2014.
- [CLT14b] J.-S. Coron, T. Lepoint, and M. Tibouchi. Personal communication. 2014.
- [DGHV10] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *Proc. of EUROCRYPT*, volume 6110 of *LNCS*, pages 24–43. Springer, 2010.
- [Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In Proc. of STOC, pages 169–178. ACM, 2009.
- [GGH13a] S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. In Proc. of EUROCRYPT, volume 7881 of LNCS, pages 1–17. Springer, 2013.
- [GGH<sup>+</sup>13b] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *Proc. of FOCS*, pages 40–49. IEEE Computer Society Press, 2013.
- [GGH14] C. Gentry, S. Gorbunov, and S. Halevi. Graded multilinear maps from lattices. IACR Cryptology ePrint Archive, 2014:645, 2014.
- [GGHZ14a] S. Garg, C. Gentry, S. Halevi, and M. Zhandry. Fully secure attribute based encryption from multilinear maps. Cryptology ePrint Archive, Report 2014/622, 2014.
- [GGHZ14b] S. Garg, C. Gentry, S. Halevi, and M. Zhandry. Fully secure functional encryption without obfuscation. Cryptology ePrint Archive, Report 2014/666, 2014.
- [GHMS14] C. Gentry, S. Halevi, H. K. Maji, and A. Sahai. Zeroizing without zeroes: Cryptanalyzing multilinear maps without encodings of zero. IACR Cryptology ePrint Archive, 2014:929, 2014.
- [GLSW14] C. Gentry, A. B. Lewko, A. Sahai, and B. Waters. Indistinguishability obfuscation from the multilinear subgroup elimination assumption. IACR Cryptology ePrint Archive, 2014:309, 2014.
- [GLW14] C. Gentry, A. B. Lewko, and B. Waters. Witness encryption from instance independent assumptions. In Proc. of CRYPTO, pages 426–443. Springer, 2014.
- [LMR14] K. Lewi, H. W. Montgomery, and A. Raghunathan. Improved constructions of PRFs secure against related-key attacks. In Proc. of ACNS, volume 8479 of LNCS, pages 44–61. Springer, 2014.
- [LS14] H. T. Lee and J. H. Seo. Security analysis of multilinear maps over the integers. In *Proc. of CRYPTO*, pages 224–240. Springer, 2014.
- [Sco02] M. Scott. Authenticated ID-based key exchange and remote log-in with simple token and PIN number. *IACR Cryptology ePrint Archive*, 2002:164, 2002.
- [Sto05] A. Storjohann. The shifted number system for fast linear algebra on integer matrices. *J. Complexity*, 21(4):609–650, 2005.
- [Sto09] A. Storjohann. Integer matrix rank certification. In Proc. of ISSAC, pages 333–340. ACM, 2009.
- [Zha14] M. Zhandry. Adaptively secure broadcast encryption with small system parameters. IACR Cryptology ePrint Archive, 2014:757, 2014.
- [Zim14] J. Zimmerman. How to obfuscate programs directly. IACR Cryptology ePrint Archive, 2014:776, 2014.

## A The Subgroup Membership and Decision Linear Problems

In this appendix, we introduce a method to solve the Subgroup Membership (SubM) and Decision Linear (DLIN) problems without using the secret primes  $p_i$ . Its computational complexity seems larger than computing secret primes  $p_i$ . We opted to give the attack anyway, as one could imagine a way to repair the Coron *et al.* multilinear map that may make the finding of the  $p_i$ 's hard, but for which SubM and DLIN might still be easy to solve.

We start this section by defining the SUbM and DLIN problems associated to the Coron et al. multilinear map. We then describe how to solve these problems in polynomial time. The attack procedure consists of two steps. First, in Section A.1, we show how to recover  $\prod_i g_i$ . It is a common procedure for solving the SubM and DLIN. Next, in Sections A.2 and A.3, we use that quantity to recognize valid instances of SubM and DLIN.

Let  $G = \mathbb{Z}_{g_1} \times \ldots \times \mathbb{Z}_{g_n}$  and  $G_i$  be the subgroup of order  $g_i$  obtained by forcing the components of the other  $\mathbb{Z}_{g_j}$ 's to be zero. For index set  $I \subseteq [n]$ , we denote  $G_I = \prod_{i \in I} G_i$ . We let  $\mathsf{enc}_1(\mathfrak{t})$  denote a properly generated level-1 encoding of  $\mathfrak{t} \in G$ . For integers L, N > 0, we let  $\mathsf{Rk}_i(\mathbb{Z}_N^{L \times L})$  denote the set of  $L \times L$  matrices over  $\mathbb{Z}_N$  of rank i. If N is a product of primes, we define the rank of a matrix as the maximum of the ranks of the matrices obtained by reduction modulo all the prime divisors of N.

**Definition 1.** (The Subgroup Membership Problem) SubM is as follows. Given  $\lambda$  and  $\kappa$ , generate params and  $p_{zt}$  using InstGen and  $\{\operatorname{enc}_1(\mathfrak{g}_i): i \in [\ell]\}$  where the  $\mathfrak{g}_i$ 's are uniformly and independently sampled in a strict subgroup  $G_I$  of G, with  $\ell$  sufficiently large so that the  $\mathfrak{g}_i$ 's generate  $G_I$  with overwhelming probability. Given params,  $p_{zt}$ ,  $\{\operatorname{enc}_1(\mathfrak{g}_i): i \in [\ell]\}$  and  $u = \operatorname{enc}_1(\mathfrak{m})$ , determine whether  $\mathfrak{m}$  is sampled uniformly in  $G_I$  or in G.

In [GLW14], Gentry et al. provide a framework to prove the security of witness encryption schemes. They use computational assumptions involving graded encodings to prove security of their witness encryption scheme. Another important application of multilinear maps is a construction of secure indistinguishability obfuscation. In [GLSW14], Gentry et al. provide the first construction of indistinguishability obfuscation which is secure under an instance independent computational assumption, the so-called Multilinear Subgroup Elimination Assumption.

These works rely on computational assumptions involving the CLT multilinear maps that are variants of the SubM problem of Definition 1. Instead of distinguishing an encoding of a uniform element in the full ring from an encoding of a uniform element in a sub-ring, they require to distinguish encodings of uniform elements in two different sub-rings. The attack described below can be readily adapted to break these variants of SubM.

**Definition 2.** (L-Decisional Linear Problem) L-DLIN is as follows. Given  $\lambda$  and  $\kappa$ , generate params and  $\mathbf{p}_{zt}$  using InstGen. Define  $N = \prod_i g_i$ . Given params and  $\mathbf{p}_{zt}$ , the goal is to distinguish between the distributions

$$\{(\mathsf{enc}_1(m_{ij}))_{i,j}\}_{(m_{ij})_{i,j} \leftarrow \mathsf{Rk}_{L-1}(\mathbb{Z}_N^{L \times L})} \quad and \quad \{(\mathsf{enc}_1(m'_{ij}))_{i,j}\}_{(m'_{ij})_{i,j} \leftarrow \mathsf{Rk}_L(\mathbb{Z}_N^{L \times L})}.$$

One of the constructions of [BLMR13] relies on the assumption that L-DLIN as defined above is hard. In one of the constructions of [ABP14], the authors rely on the following particular case. The problem is as follows. The algorithm is given params and  $p_{zt}$ , as well as  $\{\mathsf{enc}_1(a_i)\}_{i\in[L]}$  and  $\{\mathsf{enc}_1(a_ib_i)\}_{i\in[L]}$  for some uniform and independent  $a_1,\ldots,a_L,b_1,\ldots,b_L\in\{\mathsf{enc}_1(a_ib_i)\}_{i\in[L]}$ 

G. It is also given  $\operatorname{enc}_1(m)$ , and it has to assess whether m is uniformly and independently sampled in G or whether  $m = b_1 + \ldots + b_L$ . This can be restated as a special case of Definition 2, by noting that it requests to assess whether the matrix just below is full-rank.

$$\begin{pmatrix} a_1b_1 \ a_1 \ 0 \ \dots \ 0 \\ a_2b_2 \ 0 \ a_2 \dots \ 0 \\ \vdots \\ a_Lb_L \ 0 \ 0 \dots a_L \\ m \ 1 \ 1 \dots \ 1 \end{pmatrix}$$

## A.1 Step 1: Computing $\prod_i g_i$

The main step in the attack is to get  $\prod_i g_i$  from (params,  $p_{zt}$ ). It may be admissible to assume that the  $g_i$ 's are public, in which case computing  $\prod_i g_i$  is trivial. If for some reason the  $g_i$ 's have to stay secret, one must set their bit-sizes as  $\Omega(\lambda^2)$ , so that they cannot be recovered by combining the approach described below with the elliptic curve factorization algorithm.

Similarly to computing  $w_{jk}$  in the Section 3.1, we compute  $w'_{jk} := \left[x_j \cdot x_k \cdot y^{\kappa-2} \cdot (\boldsymbol{p}_{zt})_1\right]_{x_0}$ ,  $w'_{jk,i} := \left[x_i \cdot x_j \cdot x_k \cdot y^{\kappa-3} \cdot (\boldsymbol{p}_{zt})_1\right]_{x_0}$  and obtain a matrix

$$\begin{aligned} & \boldsymbol{W}_y' = \boldsymbol{R} \cdot \mathsf{diag}(r_1 g_1 + 1, \dots, r_n g_n + 1) \cdot \mathsf{diag}(h_1', \dots, h_n') \cdot \boldsymbol{R}^T. \\ & \boldsymbol{W}_i' = \boldsymbol{R} \cdot \mathsf{diag}(r_{i1} g_1, \dots, r_{in} g_n) \cdot \mathsf{diag}(h_1', \dots, h_n') \cdot \boldsymbol{R}^T. \end{aligned}$$

We can get a multiple of  $\prod_i g_i$  by taking a ratio of gcd's of determinants of appropriate subsets of  $\{W'_1, \ldots, W'_m, W'_y\}$ :

$$\frac{\gcd(\det \boldsymbol{W}'_1, \dots, \det \boldsymbol{W}'_m)}{\gcd(\det \boldsymbol{W}'_1, \dots, \det \boldsymbol{W}'_m, \det \boldsymbol{W}'_y)} = \frac{\gcd(\prod_i r_{i1}, \dots, \prod_i r_{im})}{\gcd(\prod_i r_{i1}g_i, \dots, \prod_i r_{im}g_i, \prod_i (r_ig_i + 1))} \cdot \prod_i g_i$$

$$= \Delta \cdot \prod_i g_i,$$

for some integer  $\Delta$ . By Lemma 1, the integer  $\Delta$  is (2n)-smooth with probability > 0.9. We eliminate it by trial division by all integers  $\leq 2n$ . This costs  $\widetilde{\mathcal{O}}((\kappa + \lambda)^2 \lambda^5)$  bit operations. This is dominated by the cost of the operations described in Sections 3.1 and 3.2, which is  $\widetilde{\mathcal{O}}((\kappa + \lambda)^{\omega+3}\lambda^{2\omega+6})$ .

**Lemma 1 (Heuristic).** Let  $r_{ij}$  be a random integer for  $i \in [n], j \in [m]$  with  $m \geq s \log(2n)$  for some positive integer s. Then  $\gcd(\prod_i r_{i1}, \ldots, \prod_i r_{im})$  is 2n-smooth with probability  $\geq \zeta(s)^{-1}$ , which  $is \geq 0.9$  when  $s \geq 4$ .

*Proof.* Our heuristic assumption is that each  $r_{ij}$  is divisible by a prime p > 2n with probability  $\leq 1/p$ , for all p's. First, we observe that for each j, the integer  $\prod_i r_{ij}$  is divisible by p with probability  $\leq 1 - (1 - 1/p)^n \leq n/p$ . Then the probability that  $\gcd(\prod_i r_{i1}, \ldots, \prod_i r_{im})$  is divisible by p is  $\leq (n/p)^m$ . As a result, the gcd is 2n-smooth with probability at least

$$\prod_{p>2n} (1 - (n/p)^m) \ge \prod_{p>2n} (1 - 1/p^s) = \zeta(s)^{-1} \prod_{p\le 2n} (1 - 1/p^s)^{-1} \ge \zeta(s)^{-1}.$$

Here the first inequality comes from  $(n/p)^m \leq (n/2n)^m = (1/2)^m \leq 1/p^s$  for  $m \geq s \log p$ . The equality is Euler's identity for the Riemann zeta function. The latter is decreasing and  $\zeta(4)^{-1} > 0.9$ . This completes the proof.

## A.2 Solving the CLT SubM Problem

We compute  $w'_{ik}$  with  $c = u = \mathsf{enc}_1(\mathfrak{m})$ :

$$\boldsymbol{W}_{u}' = \boldsymbol{R} \cdot \operatorname{diag}(r_{1}g_{1} + x_{1}, \dots, r_{n}g_{n} + x_{n}) \cdot \operatorname{diag}(h'_{1}, \dots, h'_{n}) \cdot \boldsymbol{R}^{T},$$

with  $x_i \in \mathbb{Z}_{g_i}$  for all i. The attack consists in computing  $\gcd(\det \mathbf{W}'_u, \prod_i g_i)$ .

If  $\mathfrak{m}$  is uniformly sampled in G, then we expect  $n/2^{\alpha}$  of the  $x_i$ 's to be zero. Hence, in that case, we have  $\log \gcd(\det \mathbf{W}'_u, \prod_i g_i) \approx \alpha n/2^{\alpha}$ . For the original setting of  $\alpha = \lambda$ , this is essentially 0.

If  $\mathfrak{m}$  is uniformly sampled in  $G_I$ , then all the  $x_i$ 's for  $i \notin I$  are zero, and we expect  $(n-|I|)/2^{\alpha}$  of the others to be zero. Hence, in that case, we have  $\log \gcd(\det \mathbf{W}'_u, \prod_i g_i) \approx \alpha |I| + \alpha (n-|I|)/2^{\alpha}$ .

## A.3 Solving the CLT DLIN Problem

As we have seen, we may assume that  $\prod_i g_i$  is known. In DLIN, we are given a matrix of level-1 encodings  $A = (a_{jk})_{j,k}$ . We write  $a_{jk} = (r_i^{(j,k)}g_i + m_i)/z \mod p_i$ . We define  $\mathbf{M}_i = (m_i^{(j,k)})_{(j,k)} \in \mathbb{Z}_{g_i}^{L \times L}$  for all  $i \in [n]$ .

We compute matrices  $W'_{a_{jk}} \in \mathbb{Z}^{n \times n}$  for all  $a_{jk}$ . We define

$$m{F} = egin{pmatrix} m{W'}_{a_{11}} & m{W'}_{a_{12}} & \dots & m{W'}_{a_{1L}} \ m{W'}_{a_{21}} & m{W'}_{a_{22}} & \dots & m{W'}_{a_{2L}} \ dots & & dots \ m{W'}_{a_{L1}} & m{W'}_{a_{L2}} & \dots & m{W'}_{a_{LL}} \end{pmatrix} \in \mathbb{Z}^{nL imes nL}.$$

We compute the determinant of F. It satisfies the following equation.

$$\det(\mathbf{F}) = \det(\mathbf{R})^{2L} \cdot (\prod_i h_i')^L \cdot \det egin{pmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} & \dots & \mathbf{B}_{1,L} \\ \mathbf{B}_{2,1} & \mathbf{B}_{2,2} & \dots & \mathbf{B}_{2,L} \\ \vdots & & \vdots \\ \mathbf{B}_{L,1} & \mathbf{B}_{L,2} & \dots & \mathbf{B}_{L,L} \end{pmatrix},$$

where  $\boldsymbol{B}_{j,k} = \operatorname{diag}(r_1^{(j,k)} \cdot g_1 + m_1^{(j,k)}, \cdots, r_n^{(j,k)} \cdot g_n + m_n^{(j,k)})$  for all j,k. Let  $\Delta = \det(\boldsymbol{R})^{2L} \cdot (\prod_i h_i')^L$ . We have  $\det \boldsymbol{F} = \Delta \cdot \prod_i \det \boldsymbol{Q}_i$ , where  $\boldsymbol{Q}_i = (r_i^{(j,k)} \cdot g_i + m_i^{(j,k)})_{(j,k)} = \boldsymbol{M}_i \mod g_i$  for all  $i \in [n]$ .

To distinguish between the instances of DLIN, we compute  $\det \mathbf{F}$  and check whether it is divisible by  $\prod_i g_i$ . If  $(m_{jk})_{j,k}$  is of full rank, the determinant of  $\mathbf{M}_i$  is nonzero for some i. Hence  $\det \mathbf{F}$  cannot be multiple of  $\prod_i g_i$ . If  $(m_{jk})_{j,k}$  is not of full rank, then  $\det \mathbf{M}_i = 0$  for all i. Hence  $\det \mathbf{F}$  is a multiple of  $\prod_i g_i$ .

The total bit-complexity of the attack is  $\widetilde{\mathcal{O}}((\kappa + \lambda)^{\omega+3}\lambda^{2\omega+6} + (\kappa + \lambda)^{\omega+3}L^{\omega+1}\lambda^{2\omega+5})$ . Here we used the fact that the determinant of a matrix  $\mathbf{A} \in \mathbb{Z}^{n \times n}$  may be computed in time  $\widetilde{\mathcal{O}}(n^{\omega} \log ||A||_{\infty})$  (see [Sto05]).