# New Multilinear Maps over the Integers

Jean-Sébastien Coron[1], Tancrède Lepoint[2], and Mehdi Tibouchi[3]

[1] University of Luxembourg, jean-sebastien.coron@uni.lu
[2] CryptoExperts, tancrede.lepoint@cryptoexperts.com
[3] NTT Secure Platform Laboratories, tibouchi.mehdi@lab.ntt.co.jp

**Abstract.** In the last few years, cryptographic multilinear maps have proved their tremendous potential as building blocks for new constructions, in particular the first viable approach to general program obfuscation. After the first candidate construction by Garg, Gentry and Halevi (GGH) based on ideal lattices, a second construction over the integers was described by Coron, Lepoint and Tibouchi (CLT). However the CLT scheme was recently broken by Cheon et al.; the attack works by computing the eigenvalues of a diagonalizable matrix over $\mathbb{Q}$ derived from the multilinear map.

In this paper we describe a new candidate multilinear map over the integers. Our construction is based on CLT but with a new arithmetic technique that makes the zero-testing element non-linear in the encoding, which prevents the Cheon et al. attack. Our new construction is relatively practical as its efficiency is comparable to the original CLT scheme. Moreover the subgroup membership and decisional linear assumptions appear to hold in the new setting.

## 1 Introduction

**Multilinear maps.** Since the breakthrough construction of Garg, Gentry and Halevi [GGH13a], there has been a growing interest in *cryptographic multilinear maps*. They have spurred scores of new cryptographic applications. Chiefly among them is possibly the first proposed approach to general program obfuscation [GGH$^+$13b]. Currently only three candidate constructions are known. Shorty after the first candidate construction of multilinear maps based on ideal lattices [GGH13a] (which we will refer to as GGH), Coron, Lepoint and Tibouchi proposed a second construction over the integers (CLT) using the same general paradigm [CLT13]. Recently, Gentry, Gorbunov and Halevi proposed another multilinear maps in which the map is defined with respect to a directed acyclic graph [GGH15].

A straightforward application of multilinear maps is multipartite Diffie-Hellman key exchange with $\kappa + 1$ users, where $\kappa$ is the maximum level of the multilinear map scheme. Initially each user publishes a level-1 encoding of a random element while keeping a level-0 encoding of the same element private. Then each user can compute the product its level-0 by the product of the level-1 encodings of the other users. With $\kappa + 1$ users this gives a level-$\kappa$ encoding from which the same secret value can be extracted by all users. The security of the protocol relies on a new hardness assumption which is a natural extension of the Decisional Diffie-Hellman assumption.

**The CLT multilinear map over the integers.** We recall the multilinear maps scheme over the integers from [CLT13]. One generates $n$ secret primes $p_i$ and publishes $x_0 = \prod_{i=1}^{n} p_i$ (where $n$ is large enough to ensure security); one also generates $n$ small secret primes $g_i$ and a random secret integer $z$ modulo $x_0$. The message space is $R = \mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}$. A level-$k$ encoding of a vector $\boldsymbol{m} = (m_i) \in R$ is then an integer $c$ such that for all $1 \leqslant i \leqslant n$:

$$c \equiv \frac{r_i \cdot g_i + m_i}{z^k} \pmod{p_i} \tag{1}$$

for some small random integers $r_i$; the integer $c$ is therefore defined modulo $x_0$ by CRT. Encodings can then be added and multiplied modulo $x_0$, as long as the noise $r_i$ is such that $r_i \cdot g_i + m_i < p$ for each $i$. The multiplication of a level-$i$ encoding by a level-$j$ encoding gives an encoding at level $i + j$.

For level-$\kappa$ encodings one defines a zero-testing parameter $p_{zt}$ with:

$$p_{zt} = \sum_{i=1}^{n} h_i \cdot \left(z^\kappa \cdot g_i^{-1} \bmod p_i\right) \cdot \frac{x_0}{p_i} \bmod x_0$$

for some small integers $h_i$. Given a level-$\kappa$ encoding $c$ as in (1), as a zero-testing procedure one computes $\omega = p_{zt} \cdot c \bmod x_0$ which gives:

$$\omega = \sum_{i=1}^{n} h_i \cdot \left(r_i + m_i \cdot (g_i^{-1} \bmod p_i)\right) \cdot \frac{x_0}{p_i} \bmod x_0 \ . \tag{2}$$

If $m_i = 0$ for all $i$, since the $r_i$'s and $h_i$'s are small, we obtain that $\omega$ is small compared to $x_0$; this enables to test whether $c$ is an encoding of $\mathbf{0}$ or not. Moreover for non-zero encodings the leading bits of $\omega$ only depend on the $m_i$'s and not on the noise $r_i$; for level-$\kappa$ encodings this enables to extract a function of the $m_i$'s only, which eventually defines a degree-$\kappa$ multilinear map.

**Cheon et al. attack.** The CLT scheme above was completely broken by a recent attack from Cheon, Han Lee, Ryu and Stehlé [CHL+15]; the attack runs in polynomial time, and recovers all secret parameters. The attack works by computing the eigenvalues of a diagonalizable matrix over $\mathbb{Q}$ derived from the multilinear map. More precisely, when applying the zero-testing procedure to the product of two encodings $x$ and $x'$, where $x$ is an encoding of 0, the resulting $\omega$ in (2) can be seen as a diagonal quadratic form over $\mathbb{Z}$ in the CRT components $x \bmod p_i$ and $x' \bmod p_i$. By computing the values $\omega_{jk}$ of the quadratic form for $n^2$ product pairs of encodings $x_j \cdot x'_k$, one can then recover the coefficients of the quadratic form using eigendecomposition, which reveals all the secret $p_i$'s and completely breaks the scheme. We recall the attack in more details in Section 3.

**Tentative fixes.** Shortly after Cheon et al. attack, two independent approaches to fix the CLT scheme have been proposed on the Cryptology ePrint Archive, due to Garg, Gentry, Halevi and Zhandry on the one hand [GGHZ14, Sec. 7][4], and Boneh, Wu and Zimmerman on the other [BWZ14]. However, both countermeasures were shown to be insecure in [CLT14,CGH+15]. Indeed, although these countermeasures do not expose encodings of zero, the value $\omega$ from the zero-testing procedure can still be expressed as a quadratic form in the CRT components of encodings. As a result, they can both be broken by a variant of the original Cheon et al. attack. Further extensions of the Cheon et al. attack along those lines are presented in [GHMS14,CGH+15].

**Our new construction.** Our new construction keeps the same CLT encodings but departs from the two previous countermeasures by modifying the zero-testing procedure itself. Namely, we modify the definition of the zero-testing element $p_{zt}$ so that $\omega$ cannot be expressed as a quadratic form anymore. For this we use a new arithmetic technique that maps the $n$ CRT components $c \bmod p_i$ to some value modulo an independent integer $N$, so that the resulting $\omega$ in the zero-testing procedure depends on the CRT components in a non-linear way, rather than linearly as in (2).

The technique works as follows. Consider a level-$\kappa$ encoding $c$ as in (1); by the Chinese Remainder Theorem, we can write a relation of the form:

$$c = \sum_{i=1}^{n} \left(r_i + m_i \cdot (g_i^{-1} \bmod p_i)\right) \cdot u_i - a \cdot x_0 \tag{3}$$

over $\mathbb{Z}$ for some $a \in \mathbb{Z}$, where the $u_i$'s are the CRT coefficients corresponding to the primes $p_i$'s, and scaled by $g_i \cdot z^{-\kappa}$ for each $i$. Let $N$ be a large integer and let $p_{zt} \in \mathbb{Z}_N$. For the zero-testing procedure we compute $\omega = p_{zt} \cdot c \bmod N$ which gives from (3):

$$\omega \equiv \sum_{i=1}^{n} \left(r_i + m_i \cdot (g_i^{-1} \bmod p_i)\right) \cdot v_i - a \cdot v_0 \pmod{N} \tag{4}$$

where $v_i := p_{zt} \cdot u_i \bmod N$ and $v_0 := p_{zt} \cdot x_0 \bmod N$. Assume now that we can generate $p_{zt}$ and $N$ such that all the $v_i$'s are small compared to $N$, including $v_0$. Now if $m_i = 0$ for all $i$, since the $r_i$'s are small, the integer $a$ in (3) is also small, which implies that $\omega$ in (4) will also be small compared to $N$. This enables to test whether $c$ is an encoding of 0 or not. As previously for level-$\kappa$ encodings one can then extract a function of the $m_i$'s only, which gives a degree-$\kappa$ multilinear map. We show that such an element $p_{zt}$ can be efficiently generated for any large enough $N$, owing to the particular structure of the CRT coefficients $u_i$.

[4] We refer to the revised version of [GGHZ14] of November 12 2014, accessible on the Cryptology ePrint Archive.

**Security analysis.** By comparing equations (2) and (4), we see that the original CLT scheme is actually a particular case, with $N = x_0$ and $v_0 = 0$. Therefore the main difference in the new scheme is that $v_0 \neq 0$, which causes the value $\omega$ in (4) to depend on the integer $a$ in (3). But that integer $a$ depends on the CRT components $r_i$ in a non-linear way. As a result, it is no longer true that the value $\omega$ computed from encoding products $x_j \cdot x'_k$ can be expressed as a quadratic form in the CRT components of $x_j$ and $x'_k$, and the Cheon et al. attack is thus thwarted.

Another difference with the original CLT scheme is that we cannot publish $x_0 = \prod_{i=1}^{n} p_i$ anymore. Namely for encodings of 0 we get a small $\omega$ and therefore (4) holds over $\mathbb{Z}$. Therefore from $x_0$ one could compute $v_0 = p_{zt} \cdot x_0 \bmod N$ and apply the Cheon et al. attack modulo $v_0$ instead of over $\mathbb{Z}$. It is not a problem to keep $x_0$ private, however, as we can mimic the technique introduced by van Dijk et al. for their fully homomorphic encryption scheme over the integers [DGHV10] and approximate modular reduction by $x_0$ with a ladder of encodings of zero of increasing sizes.

We provide a detailed security analysis of our new construction in Section 3 (for the Cheon et al. attack and its variants) and Section 4 (for lattice attacks). We also explain why the subgroup membership (SubM) and decisional linear (DLIN) problems, which are known to be easy in the GGH scheme [GGH13a], seem to be hard in our new setting.

**Implementation.** We describe an implementation of our scheme, with a few optimizations. Instead of using a ladder of encodings of 0 at every level, we publish a small multiple $x'_0$ of $x_0$ so that intermediate encodings can be reduced modulo $x'_0$; only at the last level do we use a ladder of a few level-$\kappa$ encodings of 0. Additionally, to reduce the size of public parameters, we store only a small subset of the public elements needed for re-randomization and combine them pairwise to generate the full public parameters, as in [CLT13]; such an optimization was originally described in [GH11]. With these optimizations our scheme is relatively practical; for reasonable security parameters a multipartite Diffie-Hellman computation with 7 users requires about 30 seconds, with a public parameter size of roughly 6 GBytes; a proof-of-concept implementation is available at [Lep15].

## 2 New Multilinear Map over the Integers

In this section we define our new multilinear scheme. Our scheme is actually a *graded encoding scheme* (GES) as in previous works [GGH13a,CLT13]; we recall the notion of GES in the full version of this paper [CLT15]. As explained in introduction, our new multilinear map scheme keeps the same CLT encodings as given by (1), with two main differences:

1. The zero-testing parameter $\boldsymbol{p}_{zt}$ is computed differently, so that the CRT components modulo $p_i$ of a level-$\kappa$ encoding $c$ are mapped to some value

modulo an independent integer $N$, instead of modulo $x_0$. The resulting $\boldsymbol{\omega}$ in the zero-testing procedure then depends on those CRT components in a non-linear way, rather than linearly in the original CLT scheme, which prevents the Cheon et al. attack.

2. The integer $x_0 = \prod_{i=1}^{n} p_i$ is kept private. For re-randomization, this implies that we must slightly modify the proof of statistical indistinguishability. To reduce the size of intermediate encodings back to the size of $x_0$, we publish a ladder of encodings of 0. In Section 5 we describe a simple optimization with a public multiple $x_0'$ of $x_0$.

### 2.1 Scheme Description

**System parameters.** The system parameters are similar to the original CLT scheme. One first defines the security parameter $\lambda$ and the required multilinearity level $\kappa \leqslant \mathsf{poly}(\lambda)$. Based on $\lambda$ and $\kappa$, we choose:

- $n$: the vector dimension
- $\eta$: the bit-size of the primes $p_i$
- $\alpha$: the bit-size of the primes $g_i$
- $\rho$: the bit-size of the randomness used in encodings

and various other parameters that will be specified later. The constraints that these parameters must satisfy are described in Section 2.2. For integers $z$, $p$ we denote the reduction of $z$ modulo $p$ by $(z \bmod p)$ or $[z]_p$ with $-p/2 < [z]_p \leqslant p/2$. For integers $x_1, \ldots, x_n$ we denote $\mathsf{CRT}_{p_1,\ldots,p_n}(x_1, \ldots, x_n)$ the unique integer $x$ such that $x \equiv x_i \bmod p_i$ for all $1 \leqslant i \leqslant n$ and $0 \leqslant x < \prod_{i=1}^{n} p_i$.

As in the original CLT scheme a level-$k$ encoding of a vector $\boldsymbol{m} = (m_i)$ is an integer $c$ such that for all $1 \leqslant i \leqslant n$:

$$c \equiv \frac{r_i \cdot g_i + m_i}{z^k} \pmod{p_i} \tag{5}$$

where the $r_i$'s are $\rho$-bit random integers (specific to the encoding $c$), with the following secret parameters: the $p_i$'s are random $\eta$-bit prime integers, the $g_i$'s are random $\alpha$-bit primes, and the denominator $z$ is a random (invertible) integer modulo $x_0 = \prod_{i=1}^{n} p_i$. The integer $c$ is therefore defined by CRT modulo $x_0$, but as opposed to the original CLT scheme, $x_0$ is kept secret. We denote by $\gamma$ the size of $x_0$ in bits. As in the CLT scheme the domain is the ring $R = \mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}$, so that for $\boldsymbol{m} = (m_i) \in R$ the components $m_i$ are defined modulo $g_i$ for all $1 \leqslant i \leqslant n$.

**Instance generation:** $(\mathsf{pp}, \boldsymbol{p}_{zt}) \leftarrow \mathsf{instGen}(1^\lambda, 1^\kappa)$. Instance generation is similar to [CLT13], except for the generation of $\boldsymbol{p}_{zt}$; moreover $x_0$ is kept private. We generate $n$ secret random $\eta$-bit primes $p_i$ and compute $x_0 = \prod_{i=1}^{n} p_i$. We generate a random invertible integer $z$ modulo $x_0$. We generate $n$ random $\alpha$-bit prime integers $g_i$, and various other parameters that will be specified later.

We publish the parameters $(\mathsf{pp}, \boldsymbol{p}_{zt})$ with

$$\mathsf{pp} = \Big(n, \eta, \alpha, \rho, \beta, \tau, \ell, \mu, y, \{x_j'\}_{j=1}^{\ell}, \{X_j^{(k)}\}, \{x_j\}_{j=1}^{\tau}, \{\mathit{\Pi}_j\}_{j=1}^{n+1}, s\Big).$$

**Sampling level-zero encodings:** $c \leftarrow \mathsf{samp}(\mathsf{pp})$. Since the primes $p_i$'s in (5) must remain secret, the user cannot encode a vector $\boldsymbol{m} \in R$ by CRT directly from (5). Instead, as in [CLT13], a level-0 encoding $c$ is generated as a random subset sum of random level-0 encodings $x'_j$ from the public parameters. The only difference with [CLT13] is that the random subset-sum is computed over $\mathbb{Z}$ instead of modulo $x_0$, since $x_0$ is not public.

Therefore we publish as part as our instance generation a set of $\ell$ integers $x'_j$, where each $x'_j$ encodes at level-0 the column vector $\boldsymbol{a}_j \in \mathbb{Z}^n$ of a secret matrix $\boldsymbol{A} = (a_{ij}) \in \mathbb{Z}^{n \times \ell}$, where each component $a_{ij}$ is randomly generated in $[0, g_i) \cap \mathbb{Z}$. More precisely, using the CRT modulo $x_0$ we generate integers $x'_j$ such that:

$$1 \leqslant j \leqslant \ell, \qquad x'_j \equiv r'_{ij} \cdot g_i + a_{ij} \pmod{p_i} \tag{6}$$

where the $r'_{ij}$'s are randomly generated in $(-2^\rho, 2^\rho) \cap \mathbb{Z}$.

To generate a level-0 encoding $c$, we first generate a random binary vector $\boldsymbol{b} = (b_j) \in \{0,1\}^\ell$ and output the level-0 encoding

$$c = \sum_{j=1}^{\ell} b_j \cdot x'_j \, .$$

From (6), this gives $c \equiv (\sum_{j=1}^{\ell} r'_{ij} b_j) \cdot g_i + \sum_{j=1}^{\ell} a_{ij} b_j \pmod{p_i}$; as required the output $c$ is a level-0 encoding:

$$c \equiv r_i \cdot g_i + m_i \pmod{p_i} \tag{7}$$

of some vector $\boldsymbol{m} = \boldsymbol{A} \cdot \boldsymbol{b} \in R$ which is a random subset-sum of the column vectors $\boldsymbol{a}_j$. We note that for such level-0 encodings we get $|r_i \cdot g_i + m_i| \leqslant \ell \cdot 2^{\rho+\alpha}$ for all $i$. As in [CLT13] by applying the leftover hash lemma over $R = \mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}$ the distribution of $\boldsymbol{m}$ can be made statistically close to uniform over $R$.

**Lemma 1 ([CLT13]).** *Let $c \leftarrow \mathsf{samp}(\mathsf{pp})$ and write $c \equiv r_i \cdot g_i + m_i \pmod{p_i}$. Assume $\ell \geqslant n \cdot \alpha + 2\lambda$. The distribution of $(\mathsf{pp}, \boldsymbol{m})$ is statistically close to the distribution of $(\mathsf{pp}, \boldsymbol{m}')$ where $\boldsymbol{m}' \leftarrow R$.*

As opposed to [CLT13] we cannot reduce $c$ modulo $x_0$; we only have the upper-bound $|c| \leqslant \ell \cdot 2^\gamma$, where $\gamma$ is the size of $x_0$ in bits. In the full version of this paper [CLT15], we show that instead of random sampling one can also publicly encode elements from the domain $R$, using a technique described in [BWZ14].

**Encoding at higher levels:** $c_k \leftarrow \mathsf{enc}(\mathsf{pp}, k, c)$. As in [CLT13], to allow encoding at higher levels, we publish as part of our instance-generation a level-one random encoding of $\mathbf{1}$, namely an integer $y$ such that:

$$y \equiv \frac{r_i \cdot g_i + 1}{z} \pmod{p_i}$$

for random $r_i \in (-2^\rho, 2^\rho) \cap \mathbb{Z}$; as previously the integer $y$ is computed by CRT modulo $x_0$. Given a level-0 encoding $c$ of $\boldsymbol{m} \in R$ as given by (7), we can then compute a level-1 encoding of the same $\boldsymbol{m}$ by computing over $\mathbb{Z}$:

$$c_1 = c \cdot y.$$

Namely we obtain as required:

$$c_1 \equiv \frac{r'_i \cdot g_i + m_i}{z} \pmod{p_i}$$

for some integers $r'_i$. From $|c| \leqslant \ell \cdot 2^\gamma$, we obtain $|c_1| \leqslant \ell \cdot 2^{2\gamma}$.

The difference with [CLT13] is that we cannot reduce $c_1$ modulo $x_0$. Instead we provide a ladder of level-1 encodings of zero $X_j^{(1)}$ of increasing size, so that the size of a level-1 encoding can be progressively reduced down to the size of $x_0$, as in the DGHV scheme [DGHV10, Sec. 3.3.1]. Specifically, for $j = 0, \ldots, \gamma + \lfloor \log_2 \ell \rfloor$, we set:

$$X_j^{(1)} = \mathsf{CRT}_{p_1,\ldots,p_n}\left([r_{1j} \cdot g_1/z]_{p_1}, \ldots, [r_{nj} \cdot g_n/z]_{p_n}\right) + q_j \cdot x_0$$

where $r_{ij} \leftarrow (-2^\rho, 2^\rho) \cap \mathbb{Z}$ and $q_j \leftarrow [2^{\gamma+j-1}/x_0, 2^{\gamma+j}/x_0) \cap \mathbb{Z}$.

We can then iteratively reduce the size of $c_1$ down to the size of $x_0$, first by $X^{(1)}_{\gamma+\lfloor \log_2 \ell \rfloor}$ and eventually by $X_0^{(1)}$. Since the size reduction is done bit-by-bit, at each step some integer $b_j \cdot X_j^{(1)}$ is subtracted from $c_1$, for $b_j \in \{0, 1\}$. Therefore the noise increases additively by at most $(\gamma + \lfloor \log_2 \ell \rfloor + 1) \cdot 2^\rho$ in absolute value. After reduction, the resulting encoding $\hat{c}_1$ will be such that

$$\hat{c}_1 \equiv (\hat{r}_i \cdot g_i + m_i)/z \pmod{p_i}, \tag{8}$$

with $|\hat{r}_i \cdot g_i + m_i| \leqslant \ell \cdot 2^{\rho+\alpha} \cdot 2^{\rho+\alpha} + (\gamma + \lfloor \log_2 \ell \rfloor + 1) \cdot 2^\rho \leqslant 2\ell \cdot 2^{2\rho+2\alpha}$ for all $i$.

More generally to generate a level-$k$ encoding we compute $c_k = c_0 \cdot y^k$, and the size of $c_k$ can be iteratively reduced after each multiplication by $y$ using ladders of similarly designed level-$k$ encodings $\{X_j^{(k')}\}_{j=0}^{\gamma+\lfloor \log_2 \ell \rfloor}$ for levels $k' = 1, \ldots, k$.

**Re-randomization:** $c' \leftarrow \mathsf{reRand}(\mathsf{pp}, k, \hat{c}_k)$. Our re-randomization procedure is similar to [CLT13] except that again we cannot reduce the encodings modulo $x_0$. We describe the re-randomization of encodings at level $k = 1$; the procedure can be easily adapted to randomize at level $k > 1$. We publish as part of our instance-generation a set of $n + 1$ integers $\Pi_j$:

$$1 \leqslant j \leqslant n + 1, \qquad \Pi_j = \sum_{i=1}^{n} \varpi_{ij} \cdot g_i \cdot u_i + \varpi_{n+1,j} \cdot x_0$$

where the $u_i$'s are appropriate CRT coefficients so that the $\Pi_j$'s are all level-1 random encodings of zero:

$$1 \leqslant j \leqslant n + 1, \qquad \Pi_j \equiv \frac{\varpi_{ij} \cdot g_i}{z} \pmod{p_i}.$$

Namely, we let for all $1 \leqslant i \leqslant n$:

$$u_i := \left( z^{-1} \cdot \left( \frac{x_0}{p_i} \right)^{-1} \bmod p_i \right) \cdot \frac{x_0}{p_i} \tag{9}$$

The matrix $\boldsymbol{\Pi} = (\varpi_{ij}) \in \mathbb{Z}^{(n+1)\times(n+1)}$ is a diagonally dominant matrix generated as follows: the non-diagonal entries are randomly and independently generated in $(-2^\rho, 2^\rho) \cap \mathbb{Z}$, while the diagonal entries are randomly generated in $((n+1)2^\rho, (n+2)2^\rho) \cap \mathbb{Z}$.

We also publish as part of our instance-generation a set of $\tau$ integers $x_j$:

$$1 \leqslant j \leqslant \tau, \qquad x_j = \sum_{i=1}^{n} r_{ij} \cdot g_i \cdot u_i + r_{n+1,j} \cdot x_0$$

so that each $x_j$ is a level-1 random encoding of zero:

$$1 \leqslant j \leqslant \tau, \qquad x_j \equiv \frac{r_{ij} \cdot g_i}{z} \pmod{p_i}$$

and where the column vectors of the matrix $\boldsymbol{X} = (r_{ij}) \in \mathbb{Z}^{(n+1)\times\tau}$ are randomly and independently generated in the half-open parallelepiped spanned by the columns of the previous matrix $\boldsymbol{\Pi}$; an algorithm to generate such $r_i$'s is described in [CLT13, App. E]; we obtain $|r_{ij} \cdot g_i| \leqslant 3n2^{\rho+\alpha}$ for all $i, j$.

Given as input a (reduced) level-1 encoding $\hat{c}_1$ as given by Equation (8), we randomize $\hat{c}_1$ with a random subset-sum of the $x_j$'s and a linear combination of the $\Pi_j$'s, over $\mathbb{Z}$:

$$c_1' = \hat{c}_1 + \sum_{j=1}^{\tau} b_j \cdot x_j + \sum_{j=1}^{n+1} b_j' \cdot \Pi_j \tag{10}$$

where $b_j \leftarrow \{0,1\}$, and $b_j' \leftarrow [0, 2^\mu) \cap \mathbb{Z}$, where $\mu := \rho + \alpha + \lambda$. The following Lemma shows that as required the distribution of $c_1'$ is nearly independent of the input (as long as it encodes the same $\boldsymbol{m}$). This essentially follows from the "leftover hash lemma over lattices" of [CLT13, Sec. 4.2]; the proof is given in the full version of this paper [CLT15].

**Lemma 2.** *Let the encodings $c \leftarrow \mathsf{samp}(\mathsf{pp})$, $\hat{c}_1 \leftarrow \mathsf{enc}(\mathsf{pp}, 1, c)$, and $c_1'$ as given by (10). Write $c_1' \equiv (r_i \cdot g_i + m_i)/z \pmod{p_i}$ for all $1 \leqslant i \leqslant n$ and $r_{n+1} = (c_1' - \sum r_i \cdot g_i \cdot u_i)/x_0$, and define $\boldsymbol{r} = (r_1, \ldots, r_n, r_{n+1})^T$. If $2(\rho+\alpha+\lambda) \leqslant \eta$ and $\tau \geqslant (n+2)\cdot\rho + 2\lambda$, then the distribution of $(\mathsf{pp}, \boldsymbol{r})$ is statistically close to that of $(\mathsf{pp}, \boldsymbol{r'})$, where $\boldsymbol{r'} \in \mathbb{Z}^{n+1}$ is randomly generated in the half-open parallelepiped spanned by the column vectors of $2^\mu \boldsymbol{\Pi}$. Moreover we have $|r_i \cdot g_i + m_i| \leqslant 4n^2 \cdot 2^{2\rho+2\alpha+\lambda}$ for all $1 \leqslant i \leqslant n$.*

Finally, we can reduce the size of $c_1'$ down to the size of $x_0$ using the ladder $\{X_j^{(1)}\}$, and we obtain an encoding $\hat{c}_1'$. Writing $\hat{c}_1' \equiv (\hat{r}_i' \cdot g_i + m_i)/z \pmod{p_i}$, we obtain

$$|\hat{r}_i' \cdot g_i + m_i| \leqslant 4n^2 \cdot 2^{2\rho+2\alpha+\lambda} + (\gamma + \lfloor \log_2 \ell \rfloor + 1) \cdot 2^\rho \leqslant 5n^2 \cdot 2^{2\rho+2\alpha+\lambda}\,.$$

**Adding, negating and multiplying encodings.** As in [CLT13] we can add, negate and multiply encodings. The difference is that we do those operations over $\mathbb{Z}$ instead of modulo $x_0$. More precisely, given level-one encodings $v_j$ of vectors $\boldsymbol{m}_j \in \mathbb{Z}^n$ for $1 \leqslant j \leqslant \kappa$, with $v_j \equiv (r_{ij} \cdot g_i + m_{ij})/z \pmod{p_i}$, we compute over $\mathbb{Z}$:

$$v = \prod_{j=1}^{\kappa} v_j \,.$$

This gives:

$$v \equiv \frac{\prod\limits_{j=1}^{\kappa} (r_{ij} \cdot g_i + m_{ij})}{z^\kappa} \equiv \frac{r_i \cdot g_i + \left( \prod\limits_{j=1}^{\kappa} m_{ij} \right) \bmod g_i}{z^\kappa} \pmod{p_i}$$

for some integers $r_i \in \mathbb{Z}$. Hence we obtain a level-$\kappa$ encoding of the vector $\boldsymbol{m}$ obtained by componentwise product of the vectors $\boldsymbol{m}_j$, as long as the components do not wrap modulo $p_i$, that is $\prod_{j=1}^{\kappa}(r_{ij} \cdot g_i + m_{ij}) < p_i$ for all $i$. Then, using the ladder $X_j^{(\kappa)}$ one can reduce its size down to the size of $x_0$, at the cost of an additive increase in absolute value of the noise.

In multipartite Diffie-Hellman key exchange we compute the product of $\kappa$ level-1 encodings from $\mathsf{reRand}$ and one level-0 encoding from $\mathsf{samp}$, which gives from previous bounds for all $i$:

$$|r_i| \leqslant (6n^2 2^{2\rho+2\alpha+\lambda})^\kappa \cdot \ell \cdot 2^{\rho+1}$$

In Section 5 we describe an optimization in which we publish a multiple $x_0'$ of $x_0$; then all intermediate encodings can be reduced modulo $x_0'$, instead of using a ladder of encodings of zero; only at the last stage do we need a ladder of a few level-$\kappa$ encodings of zero.

**Zero testing.** $\mathsf{isZero}(\mathsf{pp}, \boldsymbol{p}_{zt}, c) \overset{?}{=} 0/1$. To prevent the Cheon et al. attack, we keep the same encoding as in (1) but we compute the $p_{zt}$ differently; this is the most important difference. Let $c$ be a level-$\kappa$ encoding. We assume $0 \leqslant c < x_0$, as a result of approximate modular reduction using a ladder of level-$\kappa$ encodings of 0. From (5) we can write by CRT:

$$c \equiv \sum_{i=1}^{n} \left( \frac{r_i \cdot g_i + m_i}{z^\kappa} \bmod p_i \right) \cdot \left( \left( \frac{x_0}{p_i} \right)^{-1} \bmod p_i \right) \cdot \frac{x_0}{p_i} \pmod{x_0}$$

$$c \equiv \sum_{i=1}^{n} \left( r_i + m_i \cdot g_i^{-1} \bmod p_i \right) \cdot \left( g_i \cdot z^{-\kappa} \cdot \left( \frac{x_0}{p_i} \right)^{-1} \bmod p_i \right) \cdot \frac{x_0}{p_i} \pmod{x_0}$$

Therefore we can write over the integers:

$$c = \sum_{i=1}^{n} \left( r_i + m_i \cdot g_i^{-1} \bmod p_i \right) \cdot u_i' - a \cdot x_0 \tag{11}$$

for some integer $a$, where the $u'_i$'s are the scaled CRT coefficients:

$$u'_i = \left( g_i \cdot z^{-\kappa} \cdot \left(\frac{x_0}{p_i}\right)^{-1} \bmod p_i \right) \cdot \frac{x_0}{p_i} \tag{12}$$

We generate a random prime integer $N$ of size $\gamma + 2\eta + 1$ bits. Using LLL in dimension 2, we obtain[5] pairs of nonzero integers $(\alpha_i, \beta_i)$ satisfying:

$$|\alpha_i| < 2^{\eta-1} \qquad |\beta_i| \leqslant \frac{4}{3} \cdot \frac{N}{2^{\eta-1}} < 2^{2-\eta} \cdot N \qquad \beta_i \equiv \alpha_i \cdot (u'_i/p_i) \pmod N.$$

We also generate as in [CLT13] an integer matrix $\boldsymbol{H} = (h_{ij}) \in \mathbb{Z}^{n \times n}$ such that $\boldsymbol{H}$ is invertible in $\mathbb{Z}$ and both $\|\boldsymbol{H}^T\|_\infty \leqslant 2^\beta$ and $\|(\boldsymbol{H}^{-1})^T\|_\infty \leqslant 2^\beta$, for some parameter $\beta$ specified later; here $\|\cdot\|_\infty$ is the operator norm on $n \times n$ matrices with respect to the $\ell^\infty$ norm on $\mathbb{R}^n$. A technique for generating such $\boldsymbol{H}$ is discussed in the full version of this paper [CLT15]. We then publish as part of our instance generation the following zero-testing vector $\boldsymbol{p}_{zt} \in \mathbb{Z}^n$:

$$(\boldsymbol{p}_{zt})_j = \sum_{i=1}^{n} h_{ij} \cdot \alpha_i \cdot p_i^{-1} \bmod N \tag{13}$$

To determine whether a level-$\kappa$ encoding $c$ is an encoding of zero or not, we compute the vector $\boldsymbol{\omega} = c \cdot \boldsymbol{p}_{zt} \bmod N$ and test whether $\|\boldsymbol{\omega}\|_\infty$ is small:

$$\mathsf{isZero}(\mathsf{pp}, \boldsymbol{p}_{zt}, c) = \begin{cases} 1 & \text{if } \|c \cdot \boldsymbol{p}_{zt} \bmod N\|_\infty < N \cdot 2^{-\nu} \\ 0 & \text{otherwise} \end{cases}$$

for some parameter $\nu$ specified later.

Namely for a level-$\kappa$ ciphertext $c$ we obtain from (11):

$$\begin{aligned}(\boldsymbol{\omega})_j = (c \cdot \boldsymbol{p}_{zt} \bmod N)_j &= \sum_{i=1}^{n} h_{ij} \cdot \alpha_i \cdot p_i^{-1} \cdot c \bmod N \\ &= \sum_{i=1}^{n} h_{ij} \cdot \alpha_i \cdot p_i^{-1} \cdot \left( \sum_{k=1}^{n} \left(r_k + m_k \cdot g_k^{-1} \bmod p_k\right) \cdot u'_k - a \cdot x_0 \right) \bmod N\end{aligned}$$

which gives:

$$\begin{aligned}(\boldsymbol{\omega})_j = &\sum_{i=1}^{n} h_{ij} \cdot \Big( \left(r_i + m_i \cdot g_i^{-1} \bmod p_i\right) \cdot \beta_i + \\ &\alpha_i \cdot \sum_{k=1,\, k \neq i}^{n} \left(r_k + m_k \cdot g_k^{-1} \bmod p_k\right) \cdot \frac{u'_k}{p_i} - a \cdot \alpha_i \cdot \frac{x_0}{p_i} \Big) \bmod N\end{aligned} \tag{14}$$

[5] More precisely, we apply Legendre reduction to the 2-dimensional lattice generated by the rows of $\begin{pmatrix} \lceil N/B^2 \rceil & u'_i/p_i \bmod N \\ 0 & N \end{pmatrix}$, where $B = (3/4)^{1/4} 2^{\eta-1}$. The shortest vector is of the form $(\alpha_i \lceil N/B^2 \rceil, \beta_i)$.

Recall that $\alpha_i$ is at most $\eta - 1$ bits, therefore $\alpha_i \cdot u'_k/p_i$ has size at most $\eta - 1 + \gamma - (\eta - 1) = \gamma$ bits; the integer $\alpha_i \cdot x_0/p_i$ has size also at most $\gamma$ bits; moreover $\beta_i$ is at most $|N| - \eta + 1$ bits. Therefore in Equation (14) the integers $\beta_i$, $\alpha_i \cdot u_k/p_i$ and $\alpha_i \cdot x_0/p_i$ are all small compared to $N$. This implies that if $m_i = 0$ for all $1 \leqslant i \leqslant n$, then $\omega_j$ will be small compared to $N$, when the $r_i$'s are small enough, i.e. a limited number of additions/multiplications on encodings has been performed. Conversely if $m_i \neq 0$ for some $i$ we show that $\|\boldsymbol{\omega}\|_\infty$ must be large. This shows the correctness of our zero-testing procedure. More precisely we prove the following lemma in the full version of this paper [CLT15].

**Lemma 3.** *Let $n$, $\eta$, $\alpha$ and $\beta$ be as in our parameter setting. Let $\rho_f$ be such that $\alpha + \log_2 n < \rho_f \leqslant \eta - 2\beta - 2\alpha - \lambda - 8$, and let $\nu = \eta - \rho_f - \beta - \lambda - 3 \geqslant 2\alpha + \beta + 5$. Let $c$ be such that $c \equiv (r_i \cdot g_i + m_i)/z^\kappa \pmod{p_i}$ for all $1 \leqslant i \leqslant n$, where $0 \leqslant m_i < g_i$ for all $i$. Let $\boldsymbol{r} = (r_i)_{1 \leqslant i \leqslant n}$ and assume that $\|\boldsymbol{r}\|_\infty < 2^{\rho_f}$. If $\boldsymbol{m} = 0$ then $\|\boldsymbol{\omega}\|_\infty < 2^{-\nu-\lambda} \cdot N$. Conversely if $\boldsymbol{m} \neq 0$ then $\|\boldsymbol{\omega}\|_\infty > 2^{-\nu+2} \cdot N$.*

**Extraction.** $sk \leftarrow \mathsf{ext}(\mathsf{pp}, \boldsymbol{p}_{zt}, u_\kappa)$. This part is essentially the same as in [GGH13a]. To extract a random function of the vector $\boldsymbol{m}$ encoded in a level-$\kappa$ encoding $c$, we multiply $c$ by the zero-testing parameter $\boldsymbol{p}_{zt}$ modulo $N$, collect the $\nu$ most significant bits of each of the $n$ components of the resulting vector, and apply a strong randomness extractor (using the seed $s$ from $\mathsf{pp}$):

$$\mathsf{ext}(\mathsf{pp}, \boldsymbol{p}_{zt}, c) = \mathsf{Extract}_s\big(\mathsf{msbs}_\nu(c \cdot \boldsymbol{p}_{zt} \bmod N)\big)$$

where $\mathsf{msbs}_\nu$ extracts the $\nu$ most significant bits of the result.

Namely if two encodings $c$ and $c'$ encode the same $\boldsymbol{m} \in \mathbb{Z}^n$ then from Lemma 3 we have $\|(c - c') \cdot \boldsymbol{p}_{zt} \bmod N\|_\infty < N \cdot 2^{-\nu-\lambda}$, and therefore we expect that $\boldsymbol{\omega} = c \cdot \boldsymbol{p}_{zt} \bmod N$ and $\boldsymbol{\omega'} = c' \cdot \boldsymbol{p}_{zt} \bmod N$ agree on their $\nu$ most significant bits, and therefore extract to the same value.

Conversely if $c$ and $c'$ encode different vectors then by Lemma 3 we must have $\|(c - c') \cdot \boldsymbol{p}_{zt} \bmod N\|_\infty > N \cdot 2^{-\nu+2}$, and therefore the $\nu$ most significant bits of the corresponding $\boldsymbol{\omega}$ and $\boldsymbol{\omega'}$ must be different. This implies that for random $\boldsymbol{m} \in R = \mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}$ the min-entropy of $\mathsf{msbs}_\nu(c \cdot \boldsymbol{p}_{zt} \bmod N)$ when $c$ encodes $\boldsymbol{m}$ is at least $\log_2 |R| \geqslant n(\alpha - 1)$. Therefore we can use a strong randomness extractor to extract a nearly uniform bit-string of length $\lfloor \log_2 |R| \rfloor - \lambda$.

This concludes the description of our new multilinear encoding scheme.

*Remark 1.* By comparing equations (2) and (4) we see that the original CLT scheme is a particular case with $N = x_0$ and $\alpha_i = 0$ for all $1 \leqslant i \leqslant n$. Therefore the main difference of our construction is that it incorporates the additional term $a$, which depends on the $r_i$'s in a non-linear way; this is to prevent the Cheon et al. attack (see Section 3).

## 2.2 Setting the Parameters

The constraints on the system parameters are similar to [CLT13].

- The bit-size $\rho$ of the randomness used for encodings must satisfy $\rho = \Omega(\lambda)$ to avoid brute force attack on the noise. The improved attacks from [CN12] and [LS14] both have complexity $\tilde{\mathcal{O}}(2^{\rho/2})$, but with a large overhead, so in practice we can take $\rho = \lambda$.
- The bit-size $\alpha$ of the primes $g_i$ must be large enough so that the order of the group $R = \mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}$ does not contain small prime factors (see the full version of this paper [CLT15]). One can take $\alpha = \lambda$.
- The parameter $n$ must be large enough to thwart lattice-based attacks on the encodings, namely $n = \omega(\eta \log \lambda)$; see Section 4.
- The number $\ell$ of level-0 encodings $x'_j$ for $\mathsf{samp}$ must satisfy $\ell \geqslant n \cdot \alpha + 2\lambda$ in order to apply the leftover hash lemma; see Lemma 1.
- The number $\tau$ of level-1 encodings $x_j$ must satisfy $\tau \geqslant (n+2) \cdot \rho + 2\lambda$ in order to apply the leftover hash lemma over lattices; see Lemma 2.
- As a conservative security precaution, we take $\beta = 3\lambda$ (see the full version of this paper [CLT15]).
- The bit-size $\eta$ of the primes $p_i$ must satisfy $\eta \geqslant \rho_f + 2\alpha + 2\beta + \lambda + 8$, where $\rho_f$ is the maximum bit size of the randoms $r_i$ a level-$\kappa$ encoding (see Lemma 3). When computing the product of $\kappa$ level-1 encodings and an additional level-0 encoding (as in a multipartite Diffie-Hellman key exchange with $\kappa + 1$ users), one obtains $\rho_f = \kappa \cdot (2\rho + 2\alpha + \lambda + 2\log_2 n + 3) + \rho + \log_2 \ell + 1$ (see previous Section).
- We set $\nu = \eta - \rho_f - \lambda - \beta - 3$ for the number of most significant bits to extract (see Lemma 3).

### 2.3 Security of Our Construction

As in the original CLT scheme [CLT13] and in the GGH scheme [GGH13a] the security of our construction does not seem to be reducible to more classical assumptions, such as for example the Approximate-GCD problem. To prove the security of the one-round $(\kappa + 1)$-way Diffie-Hellman key exchange protocol, as in [GGH13a] one must therefore make the assumption that solving the Graded DDH problem (GDDH) is hard in our scheme; see the full version of this paper [CLT15].

## 3 Cheon et al. Attack

The goal of this section is to argue that the Cheon et al. attack [CHL$^+$15] is prevented in our new construction.

### 3.1 Attack Description

We first recall the Cheon et al. attack against the original CLT scheme. This attack makes use of low-level encodings of 0: if such encodings are made public, one can recover in polynomial time all secret parameters. In the CLT scheme such encodings of 0 are used for the rerandomization procedure, therefore the Cheon et al. attack leads to a complete break of CLT.

In the following we describe a slight simplification of [CHL$^+$15] in which only a single ciphertext $c$ is used instead of two ciphertexts $c_0$ and $c_1$; this enables to obtain as eigenvalues directly the CRT components of $c$, instead of the ratios of the CRT components of $c_0$ and $c_1$. For simplicity we assume $\kappa = 2$; the attack is easily extended to any $\kappa > 2$. Let $c$ be a level-0 encoding with $c \equiv c_i \pmod{p_i}$. Let $x$ be a level-1 encoding with $x \equiv x_i/z \pmod{p_i}$, and let $x'$ be a level-1 encoding of 0 with $x' \equiv r'_i \cdot g_i/z \pmod{p_i}$. Let $c'$ be the level-$\kappa$ product encoding

$$c' = x \cdot c \cdot x' \bmod x_0$$

From $c' \equiv x_i \cdot c_i \cdot r'_i \cdot g_i \cdot z^{-2} \pmod{p_i}$, we obtain by CRT:

$$c' \equiv \sum_{i=1}^{n} x_i \cdot c_i \cdot r'_i \cdot u_i \pmod{x_0} \tag{15}$$

with the CRT coefficients:

$$u_i = \left( g_i \cdot z^{-2} \cdot \left( \frac{x_0}{p_i} \right)^{-1} \bmod p_i \right) \cdot \frac{x_0}{p_i}$$

In the original CLT scheme, the zero-testing parameter $p_{zt}$ is given by

$$p_{zt} = \sum_{i=1}^{n} h_i \cdot \left( z^2 \cdot g_i^{-1} \bmod p_i \right) \cdot \frac{x_0}{p_i} \bmod x_0$$

Using $p_{zt} \cdot u_i \equiv h_i \cdot x_0/p_i \pmod{x_0}$ for all $1 \leqslant i \leqslant n$, we obtain from (15):

$$\omega = [p_{zt} \cdot c']_{x_0} = \sum_{i=1}^{n} x_i \cdot c_i \cdot r'_i \cdot h_i \cdot x_0/p_i \tag{16}$$

where the last equality holds over $\mathbb{Z}$ because $c'$ is an encoding of 0.

More generally, let $x_j$ be level-1 encodings with $x_j \equiv x_{ij}/z \pmod{p_i}$, and let $x'_k$ be a level-1 encodings of 0 with $x'_k \equiv r'_{ik} \cdot g_i/z \pmod{p_i}$. One can therefore compute for $1 \leqslant j, k \leqslant n$:

$$\omega_{jk} = [(x_j \cdot c \cdot x'_k) \cdot p_{zt}]_{x_0} \tag{17}$$

which gives as previously:

$$\omega_{jk} = \sum_{i=1}^{n} x_{ij} \cdot c_i \cdot r'_{ik} \cdot h_i \cdot x_0/p_i \tag{18}$$

over the integers. We note that $\omega_{jk}$ is a diagonal quadratic form over $\mathbb{Z}$ in the $x_{ij}$'s and the $r'_{ik}$'s. By spanning $1 \leqslant j, k \leqslant n$, one can construct a matrix $\boldsymbol{W_c} = (\omega_{jk})_{1 \leqslant j,k \leqslant n}$ such that

$$\boldsymbol{W_c} = \boldsymbol{X} \times \boldsymbol{C} \times \boldsymbol{R}, \tag{19}$$

where $\boldsymbol{C} = \mathsf{diag}(c_1, c_2, \ldots, c_n)$, $\boldsymbol{X} = (x_{ij} \cdot h_i \cdot x_0/p_i)_{1 \leqslant j,i \leqslant n}$ and $\boldsymbol{R} = (r'_{ik})_{1 \leqslant i,k \leqslant n}$.

We perform the same computation with $c = 1$ in (17); one can therefore compute a matrix $\boldsymbol{W_1}$ such that $\boldsymbol{W_1} = \boldsymbol{X} \times \boldsymbol{I} \times \boldsymbol{R}$, where $\boldsymbol{I}$ is the $n \times n$ identity matrix. Finally, one can publicly compute:

$$\boldsymbol{W} = \boldsymbol{W_c} \cdot \boldsymbol{W_1}^{-1} = \boldsymbol{X} \times \boldsymbol{C} \times \boldsymbol{X}^{-1}\,.$$

Since $\boldsymbol{C}$ is a diagonal matrix, by computing the eigenvalues of $\boldsymbol{W}$ one can recover the $c_i$'s, and then the $p_i$'s. Finally, Cheon et al. describe how to recover all the other secret values in [CHL+15].

**Extension.** A similar attack applies against two independent approaches to fix the CLT scheme, [GGHZ14, Sec. 7] and [BWZ14], proposed shortly after the Cheon et al. attack. Namely, although the two countermeasures do not expose encodings of zero, the value $\omega$ from the zero-testing procedure can still be expressed as a diagonal quadratic form in the CRT components of encodings, as in Equation (18), hence the two countermeasure can be broken by the same technique; we refer to [CLT14] for a description of the modified attacks.

### 3.2 Non-Applicability of Cheon et al. Attack

In this section we explain why the above attack does not apply against our new scheme. As previously we let $x$ be a level-1 encoding with $x \equiv x_i/z \pmod{p_i}$, and let $x'$ be a level-1 encoding of 0 with $x' \equiv r'_i \cdot g_i/z \pmod{p_i}$. We consider as previously the level-$\kappa$ product encoding, with $\kappa = 2$:

$$c' = x \cdot c \cdot x'$$

Here we cannot reduce $c'$ modulo $x_0$ since $x_0$ is kept private; instead we must use a ladder of level-2 encodings of zero. Let $c''$ be the resulting encoding, with $0 \leqslant c'' < x_0$; we obtain:

$$c'' \equiv c' + \frac{s_i \cdot g_i}{z^2} \pmod{p_i}$$

for some integers $s_i$ of size roughly $\rho$ bits. Therefore instead of (15) we obtain over the integers:

$$c'' = \sum_{i=1}^{n} (x_i \cdot c_i \cdot r'_i + s_i) \cdot u_i - a \cdot x_0 \tag{20}$$

for some integer $a$. Using the new definition of $p_{zt} \in \mathbb{Z}_N$, and letting $v_i = p_{zt} \cdot u_i \bmod N$ for all $1 \leqslant i \leqslant n$ and $v_0 = p_{zt} \cdot x_0 \bmod N$, we obtain from (20):

$$\omega = [p_{zt} \cdot c'']_N = \sum_{i=1}^{n} (x_i \cdot c_i \cdot r'_i + s_i) \cdot v_i - a \cdot v_0 \tag{21}$$

where as previously the last equality holds over $\mathbb{Z}$.

Now comparing equalities (16) and (21), we see that we obtain two additional terms: the $s_i$'s and the integer $a$. The $s_i$'s come from reducing $c'$ with the ladder of level-$\kappa$ encodings of 0, so that eventually $0 \leqslant c'' < x_0$; therefore the $s_i$'s depend on $x \cdot c \cdot x'$ in a non-linear way. Similarly the integer $a$ in (21), which is the quotient of the division of $\sum_{i=1}^{n} (x_i \cdot c_i \cdot r'_i + s_i) \cdot u_i$ by $x_0$, depends on the $x_i \cdot c_i \cdot x'_i$ in a non-linear way. Therefore, if we apply Cheon et al. attack, we do not obtain a quadratic form as in (18) anymore.

More precisely, we can let as previously $x_j$ be level-1 encodings with $x_j \equiv x_{ij}/z \pmod{p_i}$, and let $x'_k$ be a level-1 encodings of 0 with $x'_k \equiv r'_{ik} \cdot g_i/z \pmod{p_i}$. As previously for all $1 \leqslant j,k \leqslant n$, we can compute the product encodings $c'_{jk} = x_j \cdot c \cdot x'_k$ and we let $c''_{jk}$ be the encodings obtained after reducing $c'_{jk}$ such that $0 \leqslant c''_{jk} < x_0$, using the ladder of level-$\kappa$ encodings of zero. This gives:

$$\omega_{jk} = [p_{zt} \cdot c''_{jk}]_N = \sum_{i=1}^{n} (x_{ij} \cdot c_i \cdot r'_{ik} + s_{ijk}) \cdot v_i - a_{jk} \cdot v_0 \tag{22}$$

for integers $s_{ijk}$ and $a_{jk}$. Compared to (18), we see that the previous equation has two additional terms $s_{ijk}$ and $a_{jk}$. As previously we can write:

$$\boldsymbol{W_c} = \boldsymbol{X} \times \boldsymbol{C} \times \boldsymbol{R} + \boldsymbol{S} - \boldsymbol{A} \cdot v_0 \tag{23}$$

for some matrices $\boldsymbol{S}$ and $\boldsymbol{A}$. However we see that the previous attack does not apply, because of the additional terms $\boldsymbol{S}$ and $\boldsymbol{A} \cdot v_0$. Namely if as previously we perform the same computation with $c = 1$, we obtain:

$$\boldsymbol{W_1} = \boldsymbol{X} \times \boldsymbol{I} \times \boldsymbol{R} + \boldsymbol{S'} - \boldsymbol{A'} \cdot v_0 \tag{24}$$

but as opposed to the CLT scheme we cannot get a simple expression for $\boldsymbol{W} = \boldsymbol{W_c} \times \boldsymbol{W_1}^{-1}$. More generally, as opposed to the CLT case, it seems difficult to extract useful information about $\boldsymbol{C}$ from the matrices $\boldsymbol{W_c}$ and $\boldsymbol{W_1}$, since in equations (23) and (24) all terms $\boldsymbol{X}$, $\boldsymbol{R}$, $\boldsymbol{S}$, $\boldsymbol{S'}$, $\boldsymbol{A}$, $\boldsymbol{A'}$ and $v_0$ are unknown.

*Remark 2.* If we do not reduce $c'_{jk}$ with the ladder of encodings, the $s_{ijk}$ terms disappear but the integers $a_{jk}$ becomes too large and (22) does not hold over $\mathbb{Z}$ anymore. The equation still holds modulo $N$, however there is still the additional term $a_{jk}$ that prevents the Cheon et al. attack.

### 3.3 Attack with Known $\boldsymbol{x_0}$.

In this section we describe an extension of the Cheon et al. attack against our scheme when $x_0$ is known; this explains why $x_0$ must be kept secret in our scheme.

When $x_0$ is known, we can reduce the previous ciphertexts $c'_{jk}$ modulo $x_0$, and therefore the $s_{ijk}$ terms in (22) disappear. Moreover $v_0 = [p_{zt} \cdot x_0]_N$ is known. Therefore we can compute the $\boldsymbol{W_c}$ matrix as previously, and we obtain from (23) with $\boldsymbol{S} = 0$:

$$\boldsymbol{W_c} = \boldsymbol{X} \times \boldsymbol{C} \times \boldsymbol{R} \bmod v_0$$

which is the same equation as (19) in the original attack except that it holds modulo $v_0$ instead of over $\mathbb{Z}$.

Therefore we can apply the Cheon et al. attack modulo $v_0$ instead of over $\mathbb{Z}$. If $v_0$ is prime, one can recover the eigenvalues of $\boldsymbol{W} = \boldsymbol{W_c} \cdot \boldsymbol{W_1}^{-1} \bmod v_0$ by factoring the characteristic polynomial modulo $v_0$, which reveals the $c_i$'s as previously. If a prime $p$ can be extracted from $v_0$, one can still apply the attack modulo $p$ and recover the $c_i$'s modulo $p$; for large enough $p$ this reveals the $c_i$'s; alternatively for sufficiently many such primes $p$, the $c_i$'s could be recovered by CRT.

Actually the attack also works even if $v_0$ is hard to factor and no prime can be extracted. Namely the eigenvalues $c_i$'s are small, so to recover the roots of the characteristic polynomial one can use Coppersmith's first theorem for finding small roots of polynomial equations modulo an integer of unknown factorization [Cop97]. Namely Coppersmith's bound applies: with a modulus $v_0$ of size roughly $\gamma$ bits and a characteristic polynomial of degree $n$, the roots have size only roughly $\rho$ bits, with $\rho \ll \eta \simeq \gamma/n$.

### 3.4 Attack for Small Multiple of $x_0$

In Section 5 we describe an optimization with a known multiple $x'_0 = q \cdot x_0$, in order to avoid the ladder of encodings of 0. Here we show that we cannot take a too small multiple $x'_0$, otherwise the attacker can compute:

$$v'_0 = [p_{zt} \cdot x'_0]_N = [p_{zt} \cdot q \cdot x_0]_N = q \cdot v_0 \bmod N$$

where, as in Section 3.3, we let $v_0 := p_{zt} \cdot x_0 \bmod N$. If the prime $q$ is small enough then the previous equation holds over the integers, and the attacker obtains $v'_0 = q \cdot v_0$. Therefore the attacker can possibly extract a few primes from $v'_0$ and therefore from $v_0$. Letting $b$ be a divisor of $v_0$, one could then apply the Cheon et al. attack modulo $b$ instead of modulo $v_0$ and recover all secret parameters. Therefore one should make sure that $q \cdot v_0$ is greater than $N$. Letting $\eta_q$ be the bitsize of $q$, this gives the condition $\eta_q + \gamma \geqslant \gamma + 2\eta + 1$. Therefore we can take $\eta_q = 2\eta + \lambda$.

### 3.5 The Subgroup Membership and Decision Linear Problems

In the full version of this paper [CLT15] we also explain why the subgroup membership (SubM) and decisional linear (DLIN) problems, which are known to be easy in the GGH scheme [GGH13a], seem to be hard in our new setting.

## 4 Lattice Attacks

### 4.1 Lattice Attack on the Encodings

The first attack considered in [CLT13] against the original CLT scheme was based on computing a short basis for the lattice of vectors orthogonal modulo $x_0$ to

$\boldsymbol{x} = (x_j)_{1 \leqslant j \leqslant t}$, where the $x_j$'s are level-0 encodings of zero [CLT13, Sec. 5.1]. If the reduced basis vectors are short enough, they can reveal the noise values of the $x_j$'s and hence break the scheme.

The attack does not apply directly to our modified scheme, because $x_0$ is now secret, and it is therefore no longer possible to compute a basis for the lattice of vectors orthogonal to $\boldsymbol{x}$ modulo $x_0$. However, we can also mount the attack using the lattice $\boldsymbol{x}^{\perp}$ of vectors orthogonal to $\boldsymbol{x}$ over $\mathbb{Z}$, or the lattice of vectors orthogonal to $\boldsymbol{x}$ modulo some multiple $x_0'$ of $x_0$ when using the optimization suggested in Section 5 below.

Just as in [CLT13, Sec. 5.1], though, the complexity of these extended attacks remains exponential in $n$; it is in fact slightly worse, because the new lattice has slightly longer vectors for a given choice of the lattice dimension $t$. In particular, the complexity lower bound of $2^{\Omega(\gamma/\eta^2)}$ applies *a fortiori*. The attack is therefore defeated by letting $n = \omega(\eta \log \lambda)$.

### 4.2 Lattice Attack against $\boldsymbol{p_{zt}}$

From $x_0 = \prod_{i=1}^n p_i$ and $(\boldsymbol{p}_{zt})_j = \sum_{i=1}^n h_{ij} \cdot \alpha_i \cdot p_i^{-1} \bmod N$, we obtain:

$$x_0 \cdot (\boldsymbol{p}_{zt})_j = \sum_{i=1}^{n} h_{ij} \cdot \alpha_i \cdot \frac{x_0}{p_i} \bmod N.$$

Now $x_0$ is of size $\gamma$ bits, and the right-hand side of this congruence, which we denote by $w_j$, is bounded above by $n 2^{\beta+\gamma}$: they are both small compared to $N$. Therefore, if we consider a vector $\boldsymbol{p}$ formed by a subset of the $(\boldsymbol{p}_{zt})_j$'s, say $\boldsymbol{p} = \big((\boldsymbol{p}_{zt})_j\big)_{1 \leqslant j \leqslant t} \in \mathbb{Z}^t$, it may be possible to recover $\boldsymbol{w} = (w_j)_{1 \leqslant j \leqslant t}$ as a short vector in the lattice generated by $\boldsymbol{p}$ and $N\mathbb{Z}^t$, and obtain $x_0$ accordingly.

We describe the attack in more details in the full version of this paper [CLT15]. We show that the lattice attack has a complexity lower bound of $2^{\Omega(n/\eta)} = 2^{\Omega(\gamma/\eta^2)}$, just as in Section 4.1. Thus, this attack is thwarted by our choice of parameters.

In the full version of this paper [CLT15], we consider three other lattice attacks on the zero-testing parameter $p_{zt}$, which are variants of the lattice attacks considered in [CLT13, Sec. 5.2, 5.3 and 5.4]. We show that they are also thwarted by our choice of parameters.

## 5 Optimizations and Implementation

In this section we describe an implementation of our new multilinear map scheme in the one-round $(\kappa+1)$-way Diffie-Hellman key exchange protocol; we recall the protocol in the full version of this paper [CLT15], following [BS03,GGH13a]. We use the following optimizations, described in details in the full version of this paper [CLT15]:

1. Integer $p_{zt}$: as in [CLT13] we use a single integer $p_{zt}$ instead of a vector $\boldsymbol{p}_{zt}$ with $n$ components, as this is enough for Diffie-Hellman key exchange. Moreover the integer $N$ can be generated as the product of large enough prime integers, instead of being prime.
2. Known multiple of $x_0$: we publish a multiple $x'_0 = q \cdot x_0$ of $x_0$, so that all intermediate encodings can be reduced modulo $x'_0$, instead of using a ladder of encodings of 0 at each level.
3. Quadratic re-randomization: as in [CLT13] we only store a small subset of encodings which are later combined pairwise to generate the full set of encodings. This implies that the randomization of encodings becomes heuristic only. We describe a slightly more efficient variant.

**Parameters and timings.** We have implemented a one-round $(\kappa + 1)$-way Diffie-Hellman key exchange protocol with $\kappa + 1 = 7$ users, in C++ using the GMP library [Gt14] to perform operations on large integers and fplll [ACPS] for LLL. We provide our concrete parameters and the resulting timings in Table 1, for security parameters ranging from 52 to 80 bits. As in [CLT13], for a security level $\lambda$ we expect that the best attack requires at least $2^\lambda$ clock cycles. The timings of Table 1 show that the implementation of our scheme improves upon the implementation in [CLT13], especially for the Setup phase.

| Instantiation | $\lambda$ | $\kappa$ | $n$ | $\eta$ | $\Delta$ | $\rho$ | $\gamma = n \cdot \eta$ | pk size | Setup | Publish | KeyGen |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Small | 52 | 6 | 540 | 1679 | 23 | 52 | $0.9 \cdot 10^6$ | 27 MB | 5.9 s | 0.10 s | 0.17 s |
| Medium | 62 | 6 | 2085 | 1989 | 45 | 62 | $4.14 \cdot 10^6$ | 175 MB | 36 s | 0.33 s | 1.06 s |
| Large | 72 | 6 | 8250 | 2306 | 90 | 72 | $19.0 \cdot 10^6$ | 1.2 GB | 583 s | 2.05 s | 6.17 s |
| Extra | 80 | 6 | 25305 | 2619 | 159 | 85 | $66.3 \cdot 10^6$ | 6.1 GB | 4528 s | 7.8 s | 23.9 s |

**Table 1.** Parameters and timings to instantiate a *one-round* 7*-way Diffie-Hellman key exchange protocol* with $\kappa = 6$, $\ell = 2\lambda$ and $\alpha, \beta, \nu = \lambda$ on a 16-core computer (Intel Xeon E7-8837 at 2.67GHz). Setup was run in parallel on the 16 cores, while the other steps ran on a single core. Publish and KeyGen timings are per party.

## References

[ACPS] M. Albrecht, D. Cadé, X. Pujol, and D. Stehlé. fplll-4.0, a floating-point LLL implementation. Available at `http://perso.ens-lyon.fr/damien.stehle`.

[BS03] Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324:71–90, 2003.

[BWZ14] Dan Boneh, David J. Wu, and Joe Zimmerman. Immunizing multilinear maps against zeroizing attacks. Cryptology ePrint Archive, Report 2014/930, 2014. `http://eprint.iacr.org/`.

[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 attacks on multilinear maps and their limitations. In Rosario Gennaro and Matthew Robshaw, editors, *CRYPTO*, LNCS. Springer, 2015. To appear.

[CHL$^+$15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT*, volume 9056 of *LNCS*, pages 3–12. Springer, 2015.

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In Ran Canetti and Juan A. Garay, editors, *CRYPTO*, volume 8042 of *LNCS*, pages 476–493. Springer, 2013.

[CLT14] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Cryptanalysis of two candidate fixes of multilinear maps over the integers. Cryptology ePrint Archive, Report 2014/975, 2014. `http://eprint.iacr.org/`.

[CLT15] Jean-Sebastien Coron, Tancrede Lepoint, and Mehdi Tibouchi. New multilinear maps over the integers. Cryptology ePrint Archive, Report 2015/162, 2015. `http://eprint.iacr.org/`. Full version of this paper.

[CN12] Yuanmi Chen and Phong Nguyen. Faster algorithms for approximate common divisors: Breaking fully-homomorphic-encryption challenges over the integers. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, volume 7237 of *LNCS*, pages 502–519. Springer, 2012.

[Cop97] Don Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *J. Cryptology*, 10(4):233–260, 1997.

[DGHV10] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Henri Gilbert, editor, *EUROCRYPT*, volume 6110 of *LNCS*, pages 24–43. Springer, 2010.

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT*, volume 7881 of *LNCS*, 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 *FOCS*, pages 40–49. IEEE Computer Society, 2013.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC*, volume 9015 of *LNCS*, pages 498–527, 2015.

[GGHZ14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mark Zhandry. Fully secure functional encryption without obfuscation. Cryptology ePrint Archive, Report 2014/666, 2014. `http://eprint.iacr.org/`.

[GH11] Craig Gentry and Shai Halevi. Implementing Gentry's fully-homomorphic encryption scheme. In Kenneth Paterson, editor, *EUROCRYPT*, volume 6632 of *LNCS*, pages 129–148. Springer, 2011.

[GHMS14] Craig Gentry, Shai Halevi, Hemanta K. Maji, and Amit Sahai. Zeroizing without zeroes: Cryptanalyzing multilinear maps without encodings of zero. Cryptology ePrint Archive, Report 2014/929, 2014. `http://eprint.iacr.org/`.

[Gt14] Torbjörn Granlund and the GMP development team. *GNU MP: The GNU Multiple Precision Arithmetic Library*, 6.0.0 edition, 2014. `http://gmplib.org/`.

[Lep15] Tancrède Lepoint. Proof-of-concept implementation of the "new" multilinear maps over the integers, 2015. `https://github.com/tlepoint/new-multilinear-maps`.

[LS14] Hyung Tae Lee and Jae Hong Seo. Security analysis of multilinear maps over the integers. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO*, volume 8616 of *LNCS*, pages 224–240. Springer, 2014.