# Two-Round Oblivious Linear Evaluation from Learning with Errors

Pedro Branco[1], Nico Döttling[2], and Paulo Mateus[1]

[1]IT, IST - University of Lisbon
[2]Helmholtz Center for Information Security (CISPA)

**Abstract**

Oblivious Linear Evaluation (OLE) is the arithmetic analogue of the well-known oblivious transfer primitive. It allows a sender, holding an affine function $f(x) = a + bx$ over a finite field or ring, to let a receiver learn $f(w)$ for a $w$ of the receiver's choice. In terms of security, the sender remains oblivious of the receiver's input $w$, whereas the receiver learns nothing beyond $f(w)$ about $f$. In recent years, OLE has emerged as an essential building block to construct efficient, reusable and maliciously-secure two-party computation.

We present a two-round vector OLE protocol over a prime field $\mathbb{F}_p$ based on the Learning with Errors (LWE) assumption. The protocol has statistical security against malicious receivers and computational security against semi-honest senders, and its first message supports polynomially many reuses. The proof combines efficient extraction of a short augmented lattice direction with Gaussian noise and uniform decoding-cell dither. An explicit quasi-polynomial-modulus instantiation preserves the constant code expansion of the prime-modulus case.

## 1 Introduction

Oblivious Linear Evaluation (OLE) is a cryptographic primitive between a sender and a receiver, where the sender inputs an affine function $f(x) = a + bx$ over a finite field $\mathbb{F}$, the receiver inputs an element $w \in \mathbb{F}$, and in the end the receiver learns $f(w)$. The sender remains oblivious of the receiver's input $w$ and the receiver learns nothing beyond $f(w)$ about $f$. OLE can be seen as a generalization of the well-known Oblivious Transfer (OT) primitive.[1] In fact, just as secure computation of *Boolean* circuits can be based on OT, secure computation of *arithmetic* circuits can be based on OLE [GMW87, IPS09, AIK11].

In recent years, OLE has emerged as one of the most promising avenues to realize efficient two-party secure computation in different settings [IPS09,

[1]It is easy to see that, if we consider the affine function $f : \{0,1\} \to \{0,1\}$ such that $f(b) = m_0 + b(m_1 - m_0)$, OLE trivially implements OT.

AIK11, ADI+17, DGN+17, BCGI18, HIMV19, CDI+19]. Interestingly, OLE has found applications, not just in the secure computation of generic functions, but also in specific tasks such as Private Set Intersection [GN19, GS19] or Machine Learning related tasks [MZ17, JVC18].

Other aspects that set OLE apart from OT are reusability, meaning that the first message of a protocol is reusable across multiple executions,[2] and the fact that even a semi-honest secure OLE can be used to realize maliciously secure two-party computation [HIMV19].

Although OLE secure against semi-honest adversaries is complete for maliciously-secure two-party computation [HIMV19], this comes at the cost of efficiency and, thus, is it always preferable to start with a maliciously-secure one. Moreover, some applications of OLE even ask specifically for a maliciously-secure one [GN19]. Given this state of affairs and the importance of OLE in constructing two-party secure computation protocols, we ask the following question:

*Can we build efficient and maliciously-secure two-round OLE protocols from (presumed) post-quantum hardness assumptions?*

## 1.1 Our Results

In this work, we partially answer the question above. Specifically, we present a simple, efficient and round-optimal protocol for OLE based on the hardness of the Learning with Errors (LWE) assumption [Reg05], which is conjectured to be post-quantum secure.

Before we start, we clarify what type of OLE we obtain. OLE comes in many flavors, one of the most useful being *vector OLE* where the sender inputs two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{F}^\ell$ and the receiver obtains a linear combination of them $\mathbf{z} = \mathbf{a} + w\mathbf{b} \in \mathbb{F}^\ell$ [BCGI18]. For simplicity, we just refer to this variant as OLE.

The plaintext field is $\mathbb{F} = \mathbb{Z}_p$ and the ciphertext modulus is $q = pQ$, where $p$ is prime and $Q \geq 1$ is arbitrary, including $Q = 1$. Section 4.3 gives an instantiation with quasi-polynomial $q$ and a constant-expansion linear code.

Our protocol implements the functionality in just two rounds and has the following properties:

- Our protocol (Section 4) for OLE achieves statistical security against a malicious receiver and computational semi-honest security against a corrupted sender based on LWE. Additionally, the receiver's first message can be reused across polynomially many executions with the same receiver input.
- Our main technical innovation is a new extraction technique which recovers a shortest nonzero pair $(\mathbf{f}, t) \in \mathbb{Z}^{n+1}$ below a prescribed length cutoff

[2] While two-party *reusable* non-interactive secure computation (NISC) is impossible in the OT-hybrid model [CDI+19], reusable NISC for general Boolean circuits is known to be possible in the (reusable) OLE-hybrid model assuming one-way functions [CDI+19]. The result stated above is meaningful only if we have access to a reusable two-round OLE protocol. The only efficient realizations of this primitive are based on the Decisional Composite Residuosity (DCR) and the Quadratic Residuosity assumptions [CDI+19].

satisfying $\mathbf{f} = \mathbf{u}\mathbf{A} + v\mathbf{a} + t\mathbf{z} \pmod q$. Here $\mathbf{A} \in \mathbb{Z}_q^{k \times n}$ and $\mathbf{a} \in \mathbb{Z}_q^n$ are provided in the CRS, and $\mathbf{z} \in \mathbb{Z}_q^n$ is the receiver's message. We provide an algorithm which solves this problem efficiently given a full trapdoor basis for the lattice $\Lambda_q^{\perp}((\mathbf{A}, \mathbf{a}))$. We believe that this contribution is of independent interest. In particular, our extractor leads to an efficient simulation strategy for our PVW-based OLE protocol even for super-polynomial moduli $q$.

## 1.2 Related Work and Comparison

In the following, we briefly review some proposals from prior work and compare them with our proposal. We consider schemes with simulation-based security under the corruption models indicated below. OLE can be trivially implemented using Fully/Somewhat Homomorphic Encryption (e.g., [JVC18]) but these solutions are usually just proven secure against semi-honest adversaries and it is unclear how to extend security against malicious adversaries without relying on generic approaches such as Non-Interactive Zero-Knowledge (NIZK) proofs.[3] OLE can also be trivially implemented using generic solutions for two-party secure computation (via OT) such as [Yao82]. However, these solutions fall short in achieving an *acceptable* level of efficiency.

The work of Döttling et al. [DKM12, DKMQ12] proposed an OLE protocol with unconditional security, in the stateful tamper-proof hardware model. The protocol takes only two rounds, however further interaction with the token is needed by the parties.

In [IPS09], a semi-honest protocol for oblivious multiplication was proposed, which can be easily extended to a OLE protocol. The protocol is based on noisy encodings. Based on the same assumption, [GNN17] proposed a maliciously-secure OLE protocol, which extends the techniques of [IPS09]. However, their protocol takes eight rounds of interaction.

Chase et al. [CDI+19] presented a round-optimal reusable OLE protocol based on the Decisional Composite Residuosity (DCR) and the Quadratic Residuosity (QR) assumptions. The protocol is maliciously-secure and, to the best of our knowledge, it is the most efficient protocol for OLE proposed so far. However, it is well-known that both the DCR and the QR problems are quantumly insecure.

Recently, two new protocols for OLE based on the Ring LWE assumption were presented in [BEPU+20, dCJV20]. Both protocols run in two rounds but the protocol of [BEPU+20] either requires a PKI or a setup phase, and the protocol of [dCJV20] is secure only against semi-honest adversaries.

We also remark that our protocol implements vector OLE where the sender's inputs are vectors over a field, as in [GNN17].

In Table 1, a brief comparison between several OLE protocols is presented.

[3] As an example consider the work of [CDI+19], where the Paillier cryptosystem is extended into an OLE protocol with malicious security and the construction is highly non-trivial.

| | Hardness Assumption | Setup Assumption | Rounds | Reusability | Security |
|---|---|---|---|---|---|
| [IPS09] | Noisy Encodings | OT | 3 | - | semi-honest |
| [DKM12] | - | Stateful tamper proof hardware | 2 | ✗ | malicious |
| [GNN17] | Noisy Encodings | OT | 8 | - | malicious |
| [CDI+19] | DCR & QR | CRS | 2 | ✓ | malicious |
| [BEPU+20] | RLWE | PKI/Setup | 2 | ✗ | malicious |
| [dCJV20] | RLWE | - | 2 | - | semi-honest |
| This work | LWE | CRS | 2 | ✓ | malicious receiver |

Table 1: Comparison between different OLE schemes.

### 1.3 Open Problems

Our protocol is secure against semi-honest senders and its first message is reusable. Extending it to malicious senders remains open in this work. Hence, the main open problem left in our work is the following: Can we construct a reusable maliciously-secure two-round OLE protocol based on the LWE assumption?

## 2 Technical Outline

We will now give a brief overview of our protocol. In abuse of notation, we drop the transposition operator for transposed vectors and always assume that vectors multiplied from the right side are transposed.

### 2.1 The PVW Protocol

Our starting point is the LWE-based oblivious transfer protocol of Peikert, Vaikuntanathan and Waters [PVW08], which is based on Regev's encryption scheme [Reg05]. Since our goal is to construct an OLE protocol, we will describe its public-key pattern as a $\mathbb{F}_2$ OLE rather than the standard OT functionality. The shared-randomness step below is our modification of this pattern; the original PVW protocol encrypts separately under the two branch keys. Assume for simplicity that the LWE modulus $q$ is even.

The PVW scheme uses a common reference string which consists of a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a vector $\mathbf{a} \in \mathbb{Z}_q^m$, which together syntactically form a Regev public key. Given the CRS $(\mathbf{A}, \mathbf{a})$, the receiver, whose input is a choice bit $b \in \{0, 1\}$ chooses a uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ and a $\mathbf{e} \in \mathbb{Z}_q^m$ from a (short) LWE error distribution, e.g. a discrete gaussian. The receiver now sets $\mathbf{z} = \mathbf{sA} + \mathbf{e} - b \cdot \mathbf{a}$. In other words, if $b = 0$ then $(\mathbf{A}, \mathbf{z})$ is a well-formed Regev public key, whereas if $b = 1$ then $(\mathbf{A}, \mathbf{z} + \mathbf{a})$ is a well-formed Regev public key.

The receiver now sends $\mathbf{z}$ to the sender who proceeds as follows. Say the sender's input are $v_0, v_1 \in \{0,1\}$. The sender now encrypts $v_0$ under the public key $(\mathbf{A}, \mathbf{z})$ and $v_1$ under $(\mathbf{A}, \mathbf{a})$ *using the same randomness* $\mathbf{r}$. Specifically, the sender chooses $\mathbf{r} \in \mathbb{Z}^m$ from a wide enough discrete gaussian, sets $\mathbf{c} = \mathbf{A}\mathbf{r}$, $c_0 = \mathbf{z}\mathbf{r} + \frac{q}{2}v_0$ and $c_1 = \mathbf{a}\mathbf{r} + \frac{q}{2}v_1$. Now the sender sends $(\mathbf{c}, c_0, c_1)$ back to the receiver. The receiver then computes and outputs $y = \lceil b \cdot c_1 + c_0 - \mathbf{s}\mathbf{c} \rfloor_2$. Here $\lceil \cdot \rfloor_2$ denotes the rounding operation with respect to $q/2$.

To see that this scheme is correct, note that

$$\begin{aligned} b \cdot c_1 + c_0 - \mathbf{s}\mathbf{c} &= b\mathbf{a}\mathbf{r} + b \cdot \frac{q}{2}v_1 + \mathbf{z}\mathbf{r} + \frac{q}{2}v_0 - \mathbf{s}\mathbf{A}\mathbf{r} \\ &= b\mathbf{a}\mathbf{r} + b \cdot \frac{q}{2}v_1 + (\mathbf{s}\mathbf{A} + \mathbf{e} - b\mathbf{a})\mathbf{r} + \frac{q}{2}v_0 - \mathbf{s}\mathbf{A}\mathbf{r} \\ &= \frac{q}{2}(bv_1 + v_0) + \mathbf{e}\mathbf{r}. \end{aligned}$$

Provided that $\mathbf{e}\mathbf{r}$ is shorter than $q/4$, we can conclude that $y = \lceil b \cdot c_1 + c_0 - \mathbf{s}\mathbf{c} \rfloor_2 = bv_1 + v_0$, where the last equality is over $\mathbb{F}_2$.

**Security.** Security against semi-honest senders follows routinely from the hardness of LWE. We will omit the discussion on security against malicious senders for now and focus on security against malicious receivers.

The basic issue here is that a malicious receiver may choose $\mathbf{z}$ not of the form $\mathbf{z} = \mathbf{s}\mathbf{A} + \mathbf{e} - b\mathbf{a}$ but rather arbitrarily.

In the original PVW protocol, security is established by showing that at least one of the two branch keys $(\mathbf{A}, \mathbf{z})$ and $(\mathbf{A}, \mathbf{z} + \mathbf{a})$ is *lossy*: an encryption under that key statistically hides its message. This uses the Smoothing Lemma [MR07] to connect the absence of short relations with statistical hiding. In order to simulate, we must determine which branch can be lossy and which choice bit to extract. For our shared-randomness modification, it is important that hiding be proved jointly with the other response coordinates; marginal uniformity of either mask alone would not suffice.

In [PVW08] this is achieved using a *lattice trapdoor* [GPV08, MP12] for $\mathbf{A}$, which allows to efficiently *decode* a point $\mathbf{x} \in \mathbb{Z}_q^m$ to the point in the row-span of $\mathbf{A}$ closest to $\mathbf{x}$ (given that $\mathbf{x}$ is sufficiently close to the row-span of $\mathbf{A}$). The extractor checks for short relations by going through the nonzero multiples $\alpha\mathbf{z}$ of $\mathbf{z}$, for $\alpha \in \mathbb{Z}_q \setminus \{0\}$, and testing their proximity to the row-span of $\mathbf{A}$. This identifies a branch whose lossiness can be guaranteed and hence an extracted choice bit. Below we replace this exhaustive search by an efficient extraction argument for the shared-randomness OLE response.

A severe drawback of this method is that the extractor must iterate over all nonzero $\alpha \in \mathbb{Z}_q$. Consequently, for the extractor to be efficient $q$ must be of polynomial size. A recent work of Quach [Qua20] devised an extraction method for superpolynomial modulus $q$ by using Hash Proof Systems (HPS). This approach modifies the underlying encryption using randomized bit-rounding, whose constant per-copy correctness is amplified by repetition. It does not retain the linear plaintext-mask identity on which our OLE construction relies.

## 2.2 An Oblivious Linear Evaluation Protocol based on PVW

We will now discuss our OLE modification of the PVW scheme. The basic idea is very simple: We will modify the underlying Regev encryption scheme to support a larger plaintext space, namely $\mathbb{Z}_{q_1}$ for a prime $q_1$ and exploit linear homomorphism over $\mathbb{Z}_{q_1}$, which will lead to an OLE over $\mathbb{Z}_{q_1}$.

Concretely, let $q = q_1 \cdot q_2$, where $q_1$ is prime and $q_2 \geq 1$ is any integer. We have the same CRS as in the PVW scheme, i.e. a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a random vector $\mathbf{a} \in \mathbb{Z}_q^m$. Now the receiver's input is a $x \in \mathbb{Z}_{q_1}$, and he computes $\mathbf{z}$ by $\mathbf{z} = \mathbf{sA} + \mathbf{e} - x \cdot \mathbf{a}$ (where $\mathbf{s}$ and $\mathbf{e}$ are as above). The sender's input is now a pair $v_0, v_1 \in \mathbb{Z}_{q_1}$, and the sender computes

$$\mathbf{c} = \mathbf{Ar}, \qquad c_0 = \mathbf{zr} + q_2 v_0 + U + g, \qquad c_1 = \mathbf{ar} + q_2 v_1.$$

Here $\mathbf{r}$ and the additional scalar $g$ have independent discrete Gaussian coordinates of width $\delta$, and $U$ is independently uniform on $q_2$ consecutive integers centered at zero. These two added noises will be useful for security; in particular, $U = 0$ when $q_2 = 1$. Given $(\mathbf{c}, c_0, c_1)$ the receiver computes $y = \lceil x \cdot c_1 + c_0 - \mathbf{sc} \rfloor_{q_1}$. Here $\lceil \cdot \rfloor_{q_1}$ is defined by $\lceil u \rfloor_{q_1} = \lceil u/q_2 \rfloor \bmod q_1$. We can establish the same cancellation as above:

$$\begin{aligned} x \cdot c_1 + c_0 - \mathbf{sc} &= x\mathbf{ar} + xq_2 v_1 + \mathbf{zr} + q_2 v_0 + U + g - \mathbf{sAr} \\ &= x\mathbf{ar} + xq_2 v_1 + (\mathbf{sA} + \mathbf{e} - x\mathbf{a})\mathbf{r} + q_2 v_0 + U + g - \mathbf{sAr} \\ &= q_2(xv_1 + v_0) + \mathbf{er} + U + g. \end{aligned}$$

Now, whenever $|\mathbf{er} + U + g| < q_2/2$, rounding gives $y = xv_1 + v_0$. Since $U$ fills a decoding cell, the probability of crossing its boundary is bounded by $(\mathbb{E}|\mathbf{er} + g| + 1)/q_2$, rather than solely by a Gaussian tail. For general parameters, including $q_2 = 1$, we use a linear code in place of scalar rounding.

A detailed description of the protocol is presented in Section 4.[4] The protocol described there directly implements *vector OLE*, instead of just OLE as presented above, by replacing $q_2 v_i$ with a linear encoding of the sender's input. Section 4.3 gives one constant-expansion code and a joint parameter instantiation for every factorization $q = q_1 q_2$.

**Security.** Security against semi-honest senders follows, just as above, routinely from the LWE assumption. But for superpolynomial moduli $q_1$ (which, in the OLE setting, is the case we are mostly interested in) we are seemingly at an impasse when it comes to proving security against malicious receivers: In this case, the PVW extractor is not efficient and the randomized bit-rounding in Quach's technique [Qua20] does not preserve the linear structure used above.

Consequently, we need to devise an alternative method of extracting the receiver's input $x$. The core idea of our extractor is surprisingly simple: While PVW choose the matrix $\mathbf{A}$ together with a lattice trapdoor, we will instead

[4] The formal construction writes $q_1 = p$, $q_2 = Q$, and uses dimensions $k, n$ in place of $n, m$.

choose the matrix $\mathbf{A}' = \begin{pmatrix} \mathbf{A} \\ \mathbf{a} \end{pmatrix}$ together with a lattice trapdoor $\mathbf{T} \in \mathbb{Z}^{m \times m}$, whose columns form a short full basis of $\Lambda_q^\perp(\mathbf{A}')$ and satisfy $\mathbf{A}'\mathbf{T} = 0 \bmod q$. This is possible as the vector $\mathbf{a}$ is also provided in the CRS.

How does this help us to extract a $\tilde{x} \in \mathbb{Z}_{q_1}$ from the malicious receiver's message $\mathbf{z}$? Let $\mathbf{z} \in \mathbb{Z}_q^m$ be arbitrary and consider short nonzero pairs $(\mathbf{d}, t) \in \mathbb{Z}^{m+1}$ satisfying

$$\mathbf{d} = \mathbf{s}\mathbf{A} + v\mathbf{a} + t\mathbf{z} \pmod q$$

for some $\mathbf{s} \in \mathbb{Z}_q^n$ and $v \in \mathbb{Z}_q$. The additional Gaussian $g$ ensures that the relevant smoothing condition concerns the length of the *pair* $(\mathbf{d}, t)$, not just $\mathbf{d}$. As we explain below, the trapdoor forces all such pairs below a suitable cutoff $B$ onto one rational line. Choose its generator. If $t$ is nonzero modulo $q_1$, set $x = v/t \bmod q_1$; if there is no short pair, or $q_1$ divides $t$, set $x = 0$.

The point of the uniform noise $U$ is to handle the latter case. For a nonzero coefficient modulo $q$ that is divisible by $q_1$,

$$\mathbb{E}\big[e^{2\pi i t U/q}\big] = 0.$$

Thus these short Fourier components vanish exactly. For every remaining short component, its coefficients satisfy $v = tx \bmod q_1$. A directional form of partial smoothing [BD18], proved in Lemma 12, then allows us to add the correlated shifts $(-q_2 x h, q_2 h)$ to $(c_0, c_1)$ for an independent uniform $h \in \mathbb{Z}_{q_1}$. The longer components are controlled by a Gaussian tail bound. Crucially, this assertion concerns the entire response jointly and does not require $\mathbf{A}\mathbf{r}$ to become uniform.

Then it holds that

$$\begin{aligned}
(\mathbf{c}, c_0, c_1) &= (\mathbf{A}\mathbf{r}, \mathbf{z}\mathbf{r} + U + g + q_2 v_0, \mathbf{a}\mathbf{r} + q_2 v_1) && (1)\\
&\approx_s (\mathbf{A}\mathbf{r}, \mathbf{z}\mathbf{r} + U + g + q_2 v_0 - q_2 x h, \mathbf{a}\mathbf{r} + q_2(v_1 + h)) && (2)\\
&\equiv (\mathbf{A}\mathbf{r}, \mathbf{z}\mathbf{r} + U + g + q_2(x v_1 + v_0) - q_2 x h', \mathbf{a}\mathbf{r} + q_2 h'). && (3)
\end{aligned}$$

In other words, we can simulate $(\mathbf{c}, c_0, c_1)$ given only $xv_1 + v_0$. In the above derivation, (2) is the directional smoothing statement, and (3) follows by the simple substitution $h' = h + v_1$, where $h' \in \mathbb{Z}_{q_1}$ is again uniformly random. This is the partial-smoothing viewpoint with auxiliary information: we randomize only the direction needed to hide the rest of the sender's input.

**Efficient Extraction** It remains to be discussed how we can *efficiently* recover $x$ from $\mathbf{z}$ given the lattice trapdoor $\mathbf{T}$ for $\Lambda_q^\perp(\mathbf{A}')$. We will recover the short relation $\mathbf{d} = \mathbf{s}'\mathbf{A}' + t\mathbf{z}$, where $\mathbf{s}' = (\mathbf{s}, v)$. Setting $\mathbf{f} = \mathbf{z}\mathbf{T} \bmod q$ we get

$$\mathbf{d}\mathbf{T} = (\mathbf{s}'\mathbf{A}' + t\mathbf{z})\mathbf{T} = t\mathbf{z}\mathbf{T} = t\mathbf{f} \pmod q.$$

Assuming that $(\mathbf{d}, t)$ is sufficiently short, it holds that $(\mathbf{d}', t) = (\mathbf{d}\mathbf{T}, t)$ is also short. We will now solve the equation system $t\mathbf{f} = \mathbf{d}'$, in which $\mathbf{f}$ is known, for $t$ and $\mathbf{d}'$. Write $\mathbf{f} = (f_1, \ldots, f_m)$ and $\mathbf{d}' = (d'_1, \ldots, d'_m)$. Then we get the equation system

$$-f_1 t + d'_1 = 0, \ldots, -f_m t + d'_m = 0 \pmod q.$$

Keeping $t$ as an additional short coordinate lets us solve these equations without dividing by any of the $f_j$, which need not be invertible.

Consequently, it is sufficient to recover the common coordinate $t$ and its associated $d'_j$ from these two-dimensional equations. To recover $t$, we rely on the fact that solving the Shortest Vector Problem (SVP) in a two-dimensional lattice can actually be done in polynomial time in the input bit length [LP94]. Consider the lattice $\Lambda_j = \Lambda_q^{\perp}(\mathbf{b}_j)$, where $\mathbf{b}_j = (-f_j, 1)$. First note that $(t, d'_j)$ is a short vector in $\Lambda_j$. Furthermore, notice that $\det(\Lambda_j) = q$ as the second component of $\mathbf{b}_j$ is 1 ($\mathbf{b}_j$ is *primitive*). Let $K \geq \max\{1, \|\mathbf{T}\|_{\mathrm{op}}\}$ and choose the cutoff so that $K^2B^2 < q/2$. Two projected vectors of length at most $KB$ have determinant divisible by $q$, but of magnitude smaller than $q$ by Hadamard's inequality. They must therefore be linearly dependent. A nonzero global short pair cannot have $t = 0$, since then $\mathbf{dT}$ would be zero over the integers. These observations also establish the common-line claim used above.

By applying an SVP solver, we are able to find a shortest nonzero vector $\mathbf{g}_j = (g_j^{(1)}, g_j^{(2)})$ in each $\Lambda_j$, oriented with $g_j^{(1)} > 0$. Whenever a global short pair exists, $t$ must be an integer multiple of $g_j^{(1)}$ for every $j$. Hence we first set $t_0$ to the least common multiple of $g_1^{(1)}, \ldots, g_m^{(1)}$, and set $d'_{0,j} = (t_0/g_j^{(1)})g_j^{(2)}$.

Having recovered $\mathbf{d}'_0$, we solve $\mathbf{d}_0\mathbf{T} = \mathbf{d}'_0$ over $\mathbb{Q}$ and multiply both $\mathbf{d}_0$ and $t_0$ by the least positive integer clearing the denominators. This gives the generator $(\mathbf{d}, t)$ of the line inside the augmented lattice, rather than merely a vector on the same rational line. Finally, given $(\mathbf{d}, t)$, we can efficiently find $\mathbf{s}' \in \mathbb{Z}_q^{n+1}$ using modular linear algebra to solve

$$\mathbf{s}'\mathbf{A}' = \mathbf{d} - t\mathbf{z} \pmod{q}.$$

We verify lattice membership and the length bound, and extract $x = s'_{n+1}/t \bmod q_1$ when $t$ is nonzero modulo $q_1$. If no sufficiently short pair is found, or $q_1$ divides $t$, we set $x = 0$ by default. Lemma 11 gives the details and the failure certificate. The algorithm runs in time polynomial in the dimensions and $\log q$, and applies to prime and composite $q$ alike.

Using the directional smoothing statement above we can finally argue that $(\mathbf{Ar}, \mathbf{zr} + U + g + q_2v_0, \mathbf{ar} + q_2v_1)$ only contains information about $xv_1 + v_0$, but, up to negligible statistical distance, leaks otherwise no information about $v_0, v_1$. The extracted $x$ depends only on the receiver's first message and the trapdoor, so it also suffices for polynomially many reuses of that message.

The smoothing cutoff $B$ is separate from the honest LWE error scale. For example, Section 4.3 takes $\delta = q^{3/4}$ and $B = \lfloor q^{1/3} \rfloor$, with polynomial trapdoor norm and $q = 2^{\Theta(\log^2 \lambda)}$. This leaves the Gaussian Fourier scale $q/\delta = q^{1/4}$ well below $B$, while still satisfying the determinant bound. With the linear code described there, the same parameters give negligible correctness error for every $q_1, q_2$, including $q_2 = 1$.

### 2.3 Applications to PVW OT

Note that by setting $q_1 = 2$ our OLE protocol realizes a bit OLE, and hence OT, with the same public-key pattern as the PVW protocol [PVW08]. Our new extraction mechanism allows the modulus $q$ to be superpolynomial. The added noises and the shared encryption randomness mean that this is a modified PVW-style construction, rather than exactly the original protocol.

### 2.4 The Remaining Challenge: Malicious Senders

The remaining issue is security against malicious senders. If $\mathbf{c}, c_0, c_1$ are honestly generated, the receiver recovers an affine function of $x$ except with negligible probability. However, for malformed responses the rounding or decoding operation can produce a non-affine function, or make abort depend on $x$. By the functionality of OLE in the ideal model we have to compute an affine function, so extracting candidate sender inputs alone does not resolve this issue.

An earlier version of this manuscript claimed UC security against malicious adversaries via a cut-and-choose compiler. Its security argument does not rule out input-dependent abort, and we do not establish this claim in the present revision. Repairing the compiler remains an open problem.

## 3 Preliminaries

Throughout this work, $\lambda$ denotes the security parameter and PPT stands for "probabilistic polynomial-time".

For a real or integer vector $\mathbf{x}$, $\|\mathbf{x}\| = \|\mathbf{x}\|_2$ denotes its Euclidean norm. For $\mathbf{x} \in \mathbb{Z}_q^n$, let $[\mathbf{x}]_q$ be its coordinatewise centered lift to $(-q/2, q/2]^n$ and interpret $\|\mathbf{x}\|$ as $\|[\mathbf{x}]_q\|_2$. For a real or integer matrix $\mathbf{M} \in \mathbb{R}^{r\times t}$ with columns $\mathbf{m}^{(i)}$, write $\|\mathbf{M}\| = \max_{i\in[t]} \|\mathbf{m}^{(i)}\|_2$ for its maximum column norm. The distinct notation $\|\mathbf{M}\|_{\mathsf{op}}$ denotes its Euclidean operator norm; in particular, $\|\mathbf{M}\|_{\mathsf{op}} \le \sqrt{t}\, \|\mathbf{M}\|$. Norms of matrices over $\mathbb{Z}_q$ refer to their centered integer lifts. For $\mathbf{b} \in \{0,1\}^k$, $wt(\mathbf{b})$ denotes its number of nonzero coordinates.

If $S$ is a (finite) set, we denote by $x \leftarrow\!\!\$\, S$ an element $x \in S$ sampled according to a uniform distribution. Moreover, we denote by $U(S)$ the uniform distribution over $S$. If $D$ is a distribution over $S$, $x \leftarrow\!\!\$\, D$ denotes an element $x \in S$ sampled according to $D$. If $\mathcal{A}$ is an algorithm, $y \leftarrow \mathcal{A}(x)$ denotes the output $y$ after running $\mathcal{A}$ on input $x$.

A negligible function $\mathsf{negl}(n)$ in $n$ is a function that vanishes faster than the inverse of any polynomial in $n$.

Given two distributions $D_1$ and $D_2$, we say that they are $\varepsilon$-statistically indistinguishable, denoted by $D_1 \approx_\varepsilon D_2$, if the statistical distance is at most $\varepsilon$.

For nonzero integers $i_1, \ldots, i_j$, $\mathsf{lcm}(i_1, \ldots, i_j)$ is the smallest positive integer divisible by every $i_h$. If any argument is zero, we use the convention $\mathsf{lcm}(i_1, \ldots, i_j) = 0$.

**Error-Correcting Codes.** We define Error-Correcting Codes (ECC). An ECC over $\mathbb{Z}_q$ is composed by the following algorithms $\mathsf{ECC}_{q',q,\ell,k,\delta} = (\mathsf{Encode}, \mathsf{Decode})$ such that: i) $\mathbf{c} \leftarrow \mathsf{Encode}(\mathbf{m})$ takes as input a message $\mathbf{m} \in \mathbb{Z}_{q'}^{\ell}$ and outputs a codeword $\mathbf{c} \in \mathbb{Z}_q^k$; ii) $\mathbf{m} \leftarrow \mathsf{Decode}(\tilde{\mathbf{c}})$ takes as input corrupted codeword $\tilde{\mathbf{c}} \in \mathbb{Z}_q^k$ and outputs a message $\mathbf{m} \in \mathbb{Z}_{q'}^{\ell}$ if $\|\tilde{\mathbf{c}} - \mathbf{c}\| \leq \delta$ where $\mathbf{c} \leftarrow \mathsf{Encode}(\mathbf{m})$. In this case, we say that $\mathsf{ECC}$ corrects up to $\delta$ errors. We say that $\mathsf{ECC}$ is linear if any linear combination of codewords of $\mathsf{ECC}$ is also a codeword of $\mathsf{ECC}$.

An example of such code is the primitive lattice of [MP12] which allows for efficient decoding and fulfills all the properties that we need. In this code, $q = q'$ and $\ell < k$.

Alternatively, if $\mathbf{m} \in \mathbb{Z}_{q'}^{\ell}$ for $q't = q$ for some $t \in \mathbb{N}$, we can use the encoding $\mathbf{c} = t \cdot \mathbf{m}$ which is usually used in lattice-based cryptography (e.g., [BPR12]). Decoding a corrupted codeword $\tilde{\mathbf{c}}$ works by rounding $\lceil \tilde{\mathbf{c}} \rfloor_{q'} = \lceil (1/t) \cdot \tilde{\mathbf{c}} \rfloor \mod q'$.

## 3.1 Universal Composability

UC-framework [Can01] allows to prove security of protocols even under arbitrary composition with other protocols. Let $\mathcal{F}$ be a functionality, $\pi$ a protocol that implements $\mathcal{F}$ and $\mathcal{Z}$ be a environment, an entity that oversees the execution of the protocol in both the real and the ideal worlds. Let $\mathsf{IDEAL}_{\mathcal{F},\mathsf{Sim},\mathcal{Z}}$ be a random variable that represents the output of $\mathcal{Z}$ after the execution of $\mathcal{F}$ with adversary $\mathsf{Sim}$. Similarly, let $\mathsf{REAL}^{\mathcal{G}}_{\pi,\mathcal{A},\mathcal{Z}}$ be a random variable that represents the output of $\mathcal{Z}$ after the execution of $\pi$ with adversary $\mathcal{A}$ and with access to the functionality $\mathcal{G}$.

A protocol $\pi$ *UC-realizes* $\mathcal{F}$ *in the* $\mathcal{G}$*-hybrid model* if for every PPT adversary $\mathcal{A}$ there is a PPT simulator $\mathsf{Sim}$ such that for all PPT environments $\mathcal{E}$, the distributions $\mathsf{IDEAL}_{\mathcal{F},\mathsf{Sim},\mathcal{Z}}$ and $\mathsf{REAL}^{\mathcal{G}}_{\pi,\mathcal{A},\mathcal{Z}}$ are computationally indistinguishable.

In this work, we only consider *static* adversaries. That is, parties involved in the protocol are corrupted at the beginning of the execution.

We now present the ideal functionalities that we will use in this work.

**CRS functionality.** This functionality generates a $\mathsf{crs}$ and distributes it between all the parties involved in the protocol. Here, we present the ideal functionality as in [PVW08].

**$\mathcal{G}_{\mathsf{CRS}}$ functionality**

**Parameters:** An algorithm $\mathsf{D}$.

- Upon receiving $(\mathsf{sid}, \mathsf{P}_i, \mathsf{P}_j)$ from $\mathsf{P}_i$, $\mathcal{G}_{\mathsf{CRS}}$ runs $\mathsf{crs} \leftarrow \mathsf{D}(1^\kappa)$ and returns $(\mathsf{sid}, \mathsf{crs})$ to $\mathsf{P}_i$.
- Upon receiving $(\mathsf{sid}, \mathsf{P}_i, \mathsf{P}_j)$ from $\mathsf{P}_j$, $\mathcal{G}_{\mathsf{CRS}}$ returns $(\mathsf{sid}, \mathsf{crs})$ to $\mathsf{P}_j$.

**OT functionality.** Oblivious Transfer (OT) can be seen as a particular case of OLE. We show the ideal OT functionality below.

**$\mathcal{F}_{\mathsf{OT}}$ functionality**

**Parameters:** $\mathsf{sid} \in \mathbb{N}$ known to both parties.

- Upon receiving $(\mathsf{sid}, (M_0, M_1))$ from $\mathsf{S}$, $\mathcal{F}_{\mathsf{OT}}$ stores $(M_0, M_1)$ and ignores future messages from $\mathsf{S}$ with the same $\mathsf{sid}$;
- Upon receiving $(\mathsf{sid}, b \in \{0,1\})$ from $\mathsf{R}$, $\mathcal{F}_{\mathsf{OT}}$ checks if it has recorded $(\mathsf{sid}, (M_0, M_1))$. If so, it returns $(\mathsf{sid}, M_b)$ to $\mathsf{R}$ and $(\mathsf{sid}, \mathsf{receipt})$ to $\mathsf{S}$, and halts. Else, it sends nothing, but continues running.

**OLE functionality.** We now present the OLE functionality. This functionality involves two parties: the sender $\mathsf{S}$ and the receiver $\mathsf{R}$.

**$\mathcal{F}_{\mathsf{OLE}}$ functionality**

**Parameters:** $\mathsf{sid}, q, k \in \mathbb{N}$ and a finite field $\mathbb{F}$ known to both parties.

- Upon receiving $\big(\mathsf{sid}, (\mathbf{a}, \mathbf{b}) \in \mathbb{F}^k \times \mathbb{F}^k\big)$ from $\mathsf{S}$, $\mathcal{F}_{\mathsf{OLE}}$ stores $(\mathbf{a}, \mathbf{b})$ and ignores future messages from $\mathsf{S}$ with the same $\mathsf{sid}$;
- Upon receiving $(\mathsf{sid}, x \in \mathbb{F})$ from $\mathsf{R}$, $\mathcal{F}_{\mathsf{OLE}}$ checks if it has recorded $(\mathsf{sid}, (\mathbf{a}, \mathbf{b}))$. If so, it returns $(\mathsf{sid}, \mathbf{z} = \mathbf{a} + x\mathbf{b})$ to $\mathsf{R}$ and $(\mathsf{sid}, \mathsf{receipt})$ to $\mathsf{S}$, and halts. Else, it sends nothing but continues running.

**Batch OLE functionality.** Here we define a batch version of the functionality defined above. In this functionality, the receiver inputs several OLE inputs at the same time. The sender can then input an affine function together with an index corresponding to which input the receiver should receive the linear combination.

**$\mathcal{F}_{\mathsf{bOLE}}$ functionality**

**Parameters:** $\mathsf{sid}, q, k, k' \in \mathbb{N}$ and a finite field $\mathbb{F}$ known to both parties.

- Upon receiving $\Big(\mathsf{sid}, \{(\mathbf{a}_i, \mathbf{b}_i)\}_{i\in[k']} \in (\mathbb{F}^k \times \mathbb{F}^k)^{k'}\Big)$ from $\mathsf{S}$, $\mathcal{F}_{\mathsf{bOLE}}$ stores $\{(\mathbf{a}_i, \mathbf{b}_i)\}_{i\in[k']}$ and ignores future messages from $\mathsf{S}$ with the same $\mathsf{sid}$;
- Upon receiving $(\mathsf{sid}, \{x_i\}_{i\in[k']})$ from $\mathsf{R}$, where $x_i \in \mathbb{F}$, $\mathcal{F}_{\mathsf{bOLE}}$ checks if it has recorded $\big(\mathsf{sid}, \{(\mathbf{a}_i, \mathbf{b}_i)\}_{i\in[k']}\big)$. If so, it returns $\big(\mathsf{sid}, \{\mathbf{z}_i = \mathbf{a}_i + x_i\mathbf{b}_i\}_{i\in[k']}\big)$ to $\mathsf{R}$ and $(\mathsf{sid}, \mathsf{receipt})$ to $\mathsf{S}$, and halts. Else, it sends nothing but continues running.

## 3.2 Lattices and Hardness Assumptions

**Notation.** For an invertible matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$, the full-rank lattice with row basis $\mathbf{B}$ is $\Lambda(\mathbf{B}) = \{\mathbf{xB} : \mathbf{x} \in \mathbb{Z}^n\}$. When a matrix is specified as a column basis, its integer column span is intended instead. For a full-rank lattice $\Lambda \subseteq \mathbb{R}^n$, its dual is $\Lambda^* = \{\mathbf{x} \in \mathbb{R}^n : \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z} \text{ for every } \mathbf{y} \in \Lambda\}$. It satisfies $(\Lambda^*)^* = \Lambda$.

We denote by $\gamma\mathcal{B}$ the Euclidean ball of radius $\gamma$ centered at zero, namely $\gamma\mathcal{B} = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| \leq \gamma\}$.

A lattice $\Lambda$ is $q$-ary if $q\mathbb{Z}^n \subseteq \Lambda \subseteq \mathbb{Z}^n$. For every such lattice there is $\mathbf{A} \in \mathbb{Z}_q^{k \times n}$ such that

$$\Lambda = \Lambda_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^n : \mathbf{y} = \mathbf{xA} \pmod q \text{ for some } \mathbf{x} \in \mathbb{Z}_q^k\}.$$

Define $\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^n : \mathbf{A}\mathbf{y}^T = 0 \pmod q\}$. Then $\Lambda_q(\mathbf{A})^* = q^{-1}\Lambda_q^\perp(\mathbf{A})$.

For $s > 0$, define the unnormalized Gaussian weight $\rho_s(\mathbf{x}) = \exp(-\pi\|\mathbf{x}\|_2^2/s^2)$ and, for a countable set $S$, $\rho_s(S) = \sum_{\mathbf{x} \in S} \rho_s(\mathbf{x})$. When $0 < \rho_s(S) < \infty$, the discrete Gaussian $D_{S,s}$ assigns probability $\rho_s(\mathbf{x})/\rho_s(S)$ to each $\mathbf{x} \in S$.

For $\varepsilon > 0$, the smoothing parameter $\eta_\varepsilon(\Lambda)$ of a lattice $\Lambda$ is the least real number $\sigma > 0$ such that $\rho_{1/\sigma}(\Lambda^* \setminus \{0\}) \leq \varepsilon$ [MR07].

**Useful Lemmata.** The following lemmas are well-known results on discrete Gaussians over lattices.

**Lemma 1** ([Ban93])**.** *Let $\sigma > 0$ and $\mathbf{x} \leftarrow\!\!\$\, D_{\mathbb{Z}^n,\sigma}$. Then we have that*

$$\Pr\left[\|\mathbf{x}\| \geq \sigma\sqrt{n}\right] \leq \mathsf{negl}(n)\,.$$

The next lemma is a consequence of the smoothing lemma [MR07] and it tells us that $\mathbf{Ae}^T$ is uniform, when $\mathbf{e}$ is sampled from a discrete Gaussian for a proper choice of parameters.

**Lemma 2** ([GPV08])**.** *Let $q \geq 2$ and let $\mathbf{A} \in \mathbb{Z}_q^{k \times n}$ have columns generating $\mathbb{Z}_q^k$, equivalently a right inverse over $\mathbb{Z}_q$. Let $\varepsilon \in (0, 1/2)$ and $\sigma \geq \eta_\varepsilon(\Lambda_q^\perp(\mathbf{A}))$. For $\mathbf{e} \leftarrow\!\!\$\, D_{\mathbb{Z}^n,\sigma}$ and independently uniform $\mathbf{u} \leftarrow\!\!\$\, \mathbb{Z}_q^k$,*

$$\mathbf{Ae}^T \bmod q \approx_{2\varepsilon} \mathbf{u}^T.$$

The following intermediate-lattice statement isolates the Fourier-analytic principle behind partial smoothing with auxiliary leakage [BD18]. We state and prove the precise form used here.

**Lemma 3** (Intermediate-lattice smoothing)**.** *Let $\Lambda_0 \subseteq \Lambda_1 \subseteq \mathbb{Z}^n$ be full-rank lattices and $X \leftarrow\!\!\$\, D_{\mathbb{Z}^n,\sigma}$, where $\sigma > 0$. Independently choose a uniform coset $\bar{U} \in \Lambda_1/\Lambda_0$ and any representative $U$ of that coset. Then*

$$\Delta\big(X \bmod \Lambda_0, (X + U) \bmod \Lambda_0\big) \leq \frac{\rho_{1/\sigma}(\Lambda_0^* \setminus \Lambda_1^*)}{2\rho_{1/\sigma}(\mathbb{Z}^n)}.$$

*The distribution modulo $\Lambda_0$ does not depend on the representative chosen for $\bar{U}$.*

*Proof.* Let $P$ be the distribution of $X \bmod \Lambda_0$. The characters of $\mathbb{Z}^n/\Lambda_0$ are indexed by $y \in \Lambda_0^*/\mathbb{Z}^n$. Poisson summation gives their Fourier coefficients

$$\widehat{P}(y) = \frac{\rho_{1/\sigma}(\mathbb{Z}^n + y)}{\rho_{1/\sigma}(\mathbb{Z}^n)}.$$

Averaging over shifts from $\Lambda_1/\Lambda_0$ preserves exactly the characters indexed by $\Lambda_1^*/\mathbb{Z}^n$ and removes the others. Fourier inversion and the triangle inequality therefore bound twice the statistical distance by

$$\sum_{y \in (\Lambda_0^*/\mathbb{Z}^n) \setminus (\Lambda_1^*/\mathbb{Z}^n)} |\widehat{P}(y)| = \frac{\rho_{1/\sigma}(\Lambda_0^* \setminus \Lambda_1^*)}{\rho_{1/\sigma}(\mathbb{Z}^n)}.$$

The equality uses the disjoint cosets $\mathbb{Z}^n + y$ in the indicated dual lattice difference. □

Recall Hadamard's inequality.

**Theorem 1** (Hadamard's inequality)**.** *Let $\Lambda \subseteq \mathbb{R}^n$ be a lattice and let $\mathbf{e}_1, \ldots, \mathbf{e}_n$ be a basis of $\Lambda$. Then it holds that*

$$\det(\Lambda) \leq \prod_{i=1}^{n} \|\mathbf{e}_i\|.$$

The following two lemmas give us an upper-bound on and the value of the determinant of a two-dimensional lattice $\Lambda_q^\perp(\mathbf{a})$ for $\mathbf{a} \in \mathbb{Z}_q^2$.

**Lemma 4.** *Let $q \in \mathbb{N}$, $B \in \mathbb{R}$ and $\mathbf{a} \in \mathbb{Z}_q^2$ such that $\mathbf{a} \neq 0$. Let $\mathbf{e}, \mathbf{e}' \in \mathbb{Z}^2$ such that $\mathbf{e}, \mathbf{e}' \in \Lambda_q^\perp(\mathbf{a})$, $\|\mathbf{e}\|, \|\mathbf{e}'\| < B$ and $\mathbf{e}, \mathbf{e}'$ are linearly independent over $\mathbb{Z}$. Then $\det\left(\Lambda_q^\perp(\mathbf{a})\right) \leq B^2$.*

*Proof.* The independent vectors $\mathbf{e}, \mathbf{e}'$ form a basis of the full-rank sublattice $L' = \mathbb{Z}\mathbf{e} + \mathbb{Z}\mathbf{e}' \subseteq \Lambda_q^\perp(\mathbf{a})$. Its index is a positive integer, so

$$\det(\Lambda_q^\perp(\mathbf{a})) \leq \det(L') \leq \|\mathbf{e}\| \, \|\mathbf{e}'\| < B^2,$$

where the second inequality is Hadamard's inequality. □

We will need the following simple Definition and Lemma.

**Definition 1.** *Let $q \geq 2$. A vector $\mathbf{a} \in \mathbb{Z}_q^n$ is* primitive *if the map $\mathbf{x} \mapsto \langle \mathbf{a}, \mathbf{x} \rangle$ from $\mathbb{Z}_q^n$ to $\mathbb{Z}_q$ is surjective, equivalently if its coordinates generate $\mathbb{Z}_q$ as an ideal. Thus every $z \in \mathbb{Z}_q$ has such a representation.*

**Lemma 5.** *Let $q \geq 2$ and let $\mathbf{a} \in \mathbb{Z}_q^n$ be primitive. Then $\det(\Lambda_q^\perp(\mathbf{a})) = q$.*

*Proof.* The homomorphism $\phi : \mathbb{Z}^n \to \mathbb{Z}_q$ defined by $\phi(\mathbf{x}) = \langle \mathbf{a}, \mathbf{x} \rangle \bmod q$ is surjective and has kernel $\Lambda_q^\perp(\mathbf{a})$. It induces a bijection $\mathbb{Z}^n/\Lambda_q^\perp(\mathbf{a}) \cong \mathbb{Z}_q$. Consequently $\det(\Lambda_q^\perp(\mathbf{a})) = [\mathbb{Z}^n : \Lambda_q^\perp(\mathbf{a})] = q$. □

In dimension two, a shortest nonzero lattice vector can be found efficiently.

**Lemma 6** ([LP94])**.** *Let $q \geq 2$ and let $\Lambda \subseteq \mathbb{Z}^2$ be a $q$-ary lattice. There is an algorithm $\mathsf{SolveSVP}$ that, given an integer basis of $\Lambda$, outputs a shortest nonzero vector $\mathbf{e} \in \Lambda$. It runs in time polynomial in the bit length of the supplied basis; for a basis with $O(\log q)$-bit entries this is $\mathsf{poly}(\log q)$ bit operations.*

The next lemma states that every sufficiently short vector of $\Lambda_q^\perp(\mathbf{a})$ is an integer multiple of a shortest nonzero vector of that lattice.

**Lemma 7.** *Let $q \geq 2$, $0 < B < \sqrt{q}$, and let $\mathbf{a} \in \mathbb{Z}_q^2$ be primitive. Let $\mathbf{e}$ be a shortest nonzero vector of $\Lambda_q^\perp(\mathbf{a})$. If $\|\mathbf{e}\| < B$, then every $\mathbf{e}' \in \Lambda_q^\perp(\mathbf{a})$ with $\|\mathbf{e}'\| < B$ is of the form $\mathbf{e}' = t\mathbf{e}$ for an integer $t$.*

*Proof.* Suppose toward a contradiction that $\mathbf{e}, \mathbf{e}'$ are linearly independent. Lemma 4 gives $\det(\Lambda_q^\perp(\mathbf{a})) \leq B^2$.

By Lemma 5, that determinant equals $q$, contradicting $B^2 < q$. Thus $\mathbf{e}'$ lies on the real line spanned by $\mathbf{e}$. The intersection of this line with $\Lambda_q^\perp(\mathbf{a})$ is a rank-one lattice. Its generator must be $\mathbf{e}$ up to sign: otherwise it would give a shorter nonzero vector in $\Lambda_q^\perp(\mathbf{a})$. It follows that $\mathbf{e}' = t\mathbf{e}$ for some $t \in \mathbb{Z}$. □

**The LWE Assumption.** The Learning with Errors assumption was first presented in [Reg05]. The assumption roughly states that it should be hard to solve a set linear equations by just adding a little noise to it.

**Definition 2** (Learning with Errors)**.** *Let $q, k \in \mathbb{N}$, where $k \in \mathsf{poly}(\lambda)$, let $n = \mathsf{poly}(k \log q)$, and let $\beta \in \mathbb{R}$. The $\mathsf{LWE}_{k,\beta,q}$ assumption states that the ensembles*

$$\{(\mathbf{A}, \mathbf{sA}+\mathbf{e}) : \mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{k \times n}, \mathbf{s} \leftarrow_\$ \mathbb{Z}_q^k, \mathbf{e} \leftarrow_\$ D_{\mathbb{Z}^n,\beta}\} \quad \textit{and} \quad \{(\mathbf{A}, \mathbf{y}) : \mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{k \times n}, \mathbf{y} \leftarrow_\$ \mathbb{Z}_q^n\}$$

*are computationally indistinguishable.*

Regev proved in [Reg05] that there is a (quantum) worst-case to average-case reduction from some problems on lattices which are believed to be hard even in the presence of a quantum computer.

**Trapdoors for Lattices.** Recent works [GPV08, MP12] have presented trapdoors functions based on the hardness of LWE.

**Lemma 8** ([GPV08, MP12])**.** *For every $q \geq 2$ and suitable $n = O(k \log q + \lambda)$, there is a polynomial-time algorithm $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{TdGen}(1^\lambda, n, k, q)$ with the following properties:*

- *$\mathbf{A} \in \mathbb{Z}_q^{k \times n}$ has a right inverse over $\mathbb{Z}_q$ and has a marginal distribution statistically close to uniform.*
- *$\mathbf{T} \in \mathbb{Z}^{n \times n}$ is invertible over $\mathbb{Q}$, and its columns form a full basis of $\Lambda_q^\perp(\mathbf{A})$. There is a known bound $K = \mathrm{poly}(k, \log q, \lambda)$ such that $\|\mathbf{T}\|_{\mathsf{op}} \leq K$ except on a small failure event. The sum of this failure probability and the preceding statistical distance can be made at most $\nu_{\mathsf{Td}} \leq 2^{-\lambda}$.*

- *On the norm event, for every* $\mathbf{s} \in \mathbb{Z}_q^k$ *and* $\mathbf{e} \in \mathbb{Z}^n$ *satisfying* $\|\mathbf{e}\|_2 < q/(2K)$*, a polynomial-time algorithm* $\mathsf{Invert}$*, given* $\mathbf{A}, \mathbf{T}$ *and* $\mathbf{y} = \mathbf{sA} + \mathbf{e} \bmod q$*, recovers both* $\mathbf{e}$ *and* $\mathbf{s}$*.*

The standard statistical trapdoor construction and conversion to a full kernel basis provide the first two properties for arbitrary moduli $q \geq 2$; they do not require $q$ to be prime. The trapdoor quality here is the stated bound on the full basis's operator norm. The inversion assertion follows directly from this interface. Since $\mathbf{AT} = 0 \pmod q$, we have $\mathbf{yT} = \mathbf{eT} \pmod q$. Moreover, $\|\mathbf{eT}\|_\infty \leq K\|\mathbf{e}\|_2 < q/2$, so the centered lift is exactly $[\mathbf{yT}]_q = \mathbf{eT}$ over the integers. Compute

$$\mathbf{e} = [\mathbf{yT}]_q \mathbf{T}^{-1}$$

over the rationals and then recover the unique $\mathbf{s}$ by solving $\mathbf{sA} = \mathbf{y} - \mathbf{e} \pmod q$, using a right inverse of $\mathbf{A}$.

The same interface gives a deterministic lower bound on nonzero vectors of the row lattice. If $0 \neq \mathbf{w} \in \Lambda_q(\mathbf{A})$, then $\mathbf{wT} \in q\mathbb{Z}^n$ and $\mathbf{wT} \neq 0$, because $\mathbf{T}$ is invertible. Consequently

$$q \leq \|\mathbf{wT}\|_2 \leq K\|\mathbf{w}\|_2, \qquad \text{so} \qquad \|\mathbf{w}\|_2 \geq q/K.$$

# 4 Oblivious Linear Evaluation for General Ciphertext Moduli

We give one construction for a prime plaintext modulus $p$ and a ciphertext modulus $q = pQ$, where $Q \geq 1$ is any integer. No coprimality condition is needed. In particular, $Q = 1$ is the prime-ciphertext-modulus case. We retain the encoding into $Q\mathbb{Z}_p \subseteq \mathbb{Z}_q$ from the technical outline, and add two independent noises to the sender's first scalar response. Their roles are different: a uniform decoding-cell dither cancels certain characters exactly, while a Gaussian controls the coefficient of the receiver's message in every significant short relation.

## 4.1 Protocol

For an integer $Q \geq 1$, put

$$I_Q = \{-\lfloor Q/2 \rfloor, \ldots, \lceil Q/2 \rceil - 1\}.$$

Thus $|I_Q| = Q$ and $I_1 = \{0\}$. We use a public efficiently generated encoding and decoder $(\mathsf{Enc}, \mathsf{Dec})$, with

$$\mathsf{Enc} : \mathbb{Z}_p^{\ell'} \longrightarrow (Q\mathbb{Z}_p)^\ell \subseteq \mathbb{Z}_q^\ell, \qquad \mathsf{Enc}(v + xw) = \mathsf{Enc}(v) + x\mathsf{Enc}(w) \pmod q.$$

Here $x \in \mathbb{Z}_p$ acts through any integer representative; the equality is well defined because $p\mathsf{Enc}(w) = 0 \pmod q$. We require the following noise-specific correctness guarantee. With probability at least $1 - \varepsilon_{\mathsf{dec}}$ over code generation and independent

$$e \leftarrow\!\!\$\, D_{\mathbb{Z}^n,\beta}, \quad R \leftarrow\!\!\$\, D_{\mathbb{Z}^{n\times\ell},\delta}, \quad g \leftarrow\!\!\$\, D_{\mathbb{Z}^\ell,\delta}, \quad U \leftarrow\!\!\$\, I_Q^\ell,$$

the identity

$$\mathsf{Dec}(\mathsf{Enc}(v) + eR + g + U \bmod q) = v \quad \text{holds for every } v \in \mathbb{Z}_p^{\ell'}. \tag{4}$$

The entries of $R$ are independent discrete Gaussians. Section 4.3 constructs one constant-expansion code and decoder satisfying this guarantee for both prime and composite $q$.

**Construction 1** (Dithered OLE)**.** *Let $p$ be prime, $Q \geq 1$, $q = pQ$, and $k, n, \ell, \ell' \in \mathbb{N}$.*

$\mathsf{GenCRS}(1^\lambda)$**.** *Sample $A \leftarrow\!\!\$\, \mathbb{Z}_q^{k\times n}$ and $a \leftarrow\!\!\$\, \mathbb{Z}_q^n$ independently. Generate the public code and publish $\mathsf{crs} = (A, a, \mathsf{Enc}, \mathsf{Dec})$. Write $A_0 = \binom{A}{a} \in \mathbb{Z}_q^{(k+1)\times n}$.*

$\mathsf{R}_1(\mathsf{crs}, x \in \mathbb{Z}_p)$**.** *Sample $s \leftarrow\!\!\$\, \mathbb{Z}_q^k$ and $e \leftarrow\!\!\$\, D_{\mathbb{Z}^n,\beta}$, and send*

$$z = sA + e - xa \pmod q.$$

*Retain $(s, x)$, using the same representative of $x$ in both receiver steps.*

$\mathsf{S}(\mathsf{crs}, (v_0, v_1), z)$**.** *For $v_0, v_1 \in \mathbb{Z}_p^{\ell'}$, independently sample $R, g, U$ as above and send*

$$C = AR, \qquad t_0 = zR + \mathsf{Enc}(v_0) + U + g, \qquad t_1 = aR + \mathsf{Enc}(v_1) \pmod q.$$

$\mathsf{R}_2(\mathsf{crs}, (s, x), (C, t_0, t_1))$**.** *Output $\mathsf{Dec}(xt_1 + t_0 - sC \bmod q)$.*

The new randomness changes neither the number of rounds nor the response length. It must be included in the sender's random tape in any subsequent application that opens or recomputes sender randomness.

## 4.2 Analysis

For $s > 0$ write $\rho_s(S) = \sum_{w\in S} \exp(-\pi\|w\|_2^2/s^2)$. For a matrix $N$ over $\mathbb{Z}_q$, use the row-lattice convention

$$\Lambda_q(N) = \{w \in \mathbb{Z}^d : w \bmod q \in \mathrm{rowspan}_{\mathbb{Z}_q}(N)\},$$
$$\Lambda_q^\perp(N) = \{h \in \mathbb{Z}^d : Nh = 0 \pmod q\}.$$

We use operator norms explicitly below, rather than the column norm used in some preliminary statements.

**Gaussian estimates.**

**Lemma 9** (Absolute Gaussian tail)**.** *For $s, B > 0$ and $d \geq 1$,*

$$\rho_s(\mathbb{Z}^d \setminus B\mathcal{B}_2^d) \leq \exp\left(-\frac{\pi B^2}{2s^2}\right)(1+\sqrt{2}s)^d.$$

*For $X \leftarrow\!\!\$\, D_{\mathbb{Z}^d,s}$ with $s \geq 1$, $\Pr[\|X\|_2 > 2s\sqrt{d}] \leq e^{-\Omega(d)}$ and $\mathbb{E}\|X\|_2^2 \leq ds^2$.*

*Proof.* Outside the ball, split the Gaussian exponent into two equal terms, bound one by $\exp(-\pi B^2/(2s^2))$, and sum the other over $\mathbb{Z}^d$. The estimate $\rho_s(\mathbb{Z}) \leq 1+s$ follows by comparison with the Gaussian integral and gives the displayed inequality. For the normalized tail, Poisson summation gives $\rho_s(\mathbb{Z}) \geq s$; thus the same bound at $B = 2s\sqrt{d}$ is at most $[e^{-2\pi}(\sqrt{2}+1)]^d$. For the second moment, differentiating $\rho_s(\mathbb{Z}) = s\rho_{1/s}(\mathbb{Z})$ shows $\mathbb{E}[X_1^2] \leq s^2/(2\pi) \leq s^2$; coordinates are independent. □

**The trapdoor interface.** The simulator generates $(A_0, T)$ such that $A_0$ has a right inverse over $\mathbb{Z}_q$, its marginal is within $\nu_{\mathsf{Td}}$ of uniform, and the columns of the invertible integer matrix $T$ form a full basis of $\Lambda_q^\perp(A_0)$. We require a known bound

$$\max\{1, \|T\|_{\mathsf{op}}\} \leq K.$$

Standard trapdoor generation provides this interface for arbitrary $q$, with $n = O((k+1)\log q + \lambda)$ and polynomial $K$ [GPV08, MP12]. In particular, the statistical instantiation of the gadget construction and its conversion to a full kernel basis do not require $q$ prime. Failure probabilities for the trapdoor norm are included in $\nu_{\mathsf{Td}}$.

The full-basis condition implies the equivalence

$$wT = 0 \pmod q \quad \Longleftrightarrow \quad w \bmod q \in \mathrm{rowspan}(A_0). \tag{5}$$

Indeed, the reduction of $\Lambda_q^\perp(A_0)$ is the entire kernel of $A_0$ over $\mathbb{Z}_q$, whose character annihilator is the row span. The right-inverse condition also makes row-space coefficients unique.

**Lemma 10** (One short augmented direction)**.** *Fix $A_0, T, K$ as above and any $z \in \mathbb{Z}_q^n$. Put*

$$M_z = \begin{pmatrix} A & 0 \\ a & 0 \\ z & 1 \end{pmatrix}, \qquad \mathcal{L}_z = \Lambda_q(M_z).$$

*If $B \geq 1$ and $K^2B^2 < q/2$, all nonzero vectors of $\mathcal{L}_z$ of length at most $B$ lie on a single rational line. Every such vector $(f,t)$ has $0 < |t| < q$.*

*Proof.* Let $y = zT \bmod q$. A vector $(f,t) \in \mathcal{L}_z$ satisfies $fT = ty \pmod q$. Its image $(fT, t)$ under $\mathrm{diag}(T, 1)$ has length at most $KB$. If $t = 0$, every coordinate of $fT$ is a multiple of $q$ of magnitude less than $q$, so $fT = 0$ over $\mathbb{Z}$ and hence

$f = 0$. Also $|t| \le B < q$. For two nonzero short vectors $(f, t), (f', t')$, every determinant

$$(fT)_i t' - (f'T)_i t$$

is a multiple of $q$, and its magnitude is at most $K^2B^2 < q$ by Hadamard's inequality applied to the two projected vectors. It is therefore zero. Since $t, t' \neq 0$, the transformed vectors, and then the original vectors, are rationally proportional. □

**Lemma 11** (Efficient augmented-direction extraction)**.** *Under Lemma 10, there is an algorithm polynomial in $n, \log q$ and the bit lengths of its inputs that returns $\perp$ exactly when $\mathcal{L}_z$ has no nonzero vector of length at most $B$. Otherwise it returns the generator $(f_*, t_*)$ of the intersection of its short rational line with $\mathcal{L}_z$, with $t_* > 0$, together with the unique $u_*, v_*$ satisfying*

$$f_* = u_* A + v_* a + t_* z \pmod{q}.$$

*Every vector of $\mathcal{L}_z$ of length at most $B$ is an integer multiple of this generator.*

*Proof.* Compute $y = zT \bmod q$. For each $i \in [n]$, solve exact SVP in

$$L_i = \{(a, b) \in \mathbb{Z}^2 : b - y_i a = 0 \pmod{q}\},$$

a determinant-$q$ lattice. Two-dimensional lattice reduction takes polynomial time even for composite $q$ [LP94]. Write a shortest nonzero vector as $(a_i, b_i)$ and orient it with $a_i > 0$. Return $\perp$ if this is impossible or if its length exceeds $KB$. Set

$$h_0 = \operatorname{lcm}_{i \in [n]} a_i, \qquad h_i = (h_0/a_i) b_i, \qquad w = (h_1, \ldots, h_n) T^{-1}.$$

Let $c$ be the least positive integer clearing the denominators of $w$, and set $(f_*, t_*) = (cw, ch_0)$. Return $\perp$ if its length exceeds $B$; otherwise recover $u_*, v_*$ by solving $f_* - t_* z = u_* A + v_* a \pmod{q}$. Check this relation before returning. Linear equations over $\mathbb{Z}_q$ can be solved by integer Smith normal form; the algorithm never requires a factorization of $q$.

For completeness, suppose $(f, t)$ is a nonzero vector of length at most $B$, oriented with $t > 0$. The vector $(t, (fT)_i)$ is a nonzero member of $L_i$ of length at most $KB$. Its determinant with the chosen shortest vector is divisible by $q$ and has magnitude at most $K^2B^2 < q$. Consequently it is an integer multiple of that shortest lattice vector. Here one must use an actual shortest vector in $L_i$, not divide it by its ambient coordinate gcd: the latter operation need not preserve lattice membership. Thus $a_i$ divides $t$ for every $i$, so $t = jh_0$ and $fT = j(h_1, \ldots, h_n)$ for some positive integer $j$. Integrality of $f = jw$ implies $c \mid j$. Therefore $(f_*, t_*)$ divides $(f, t)$ over $\mathbb{Z}$ and is no longer than it.

Conversely, $h_i = h_0 y_i \pmod{q}$ implies $(f_* - t_* z)T = 0 \pmod{q}$, so (5) proves membership in $\mathcal{L}_z$. Every lattice vector on this line has its projected pairs in $L_i$; the same divisibility and denominator argument shows it is an integer multiple of $(f_*, t_*)$. This proves the generator claim and the failure certificate. All least common multiples and rational linear algebra operations have polynomial bit complexity. □

**Dithered directional smoothing.** The following corollary isolates the distributional statement used by the protocol. Its proof uses Fourier expansion and Poisson summation, as in lattice smoothing, and retains all response coordinates jointly.

**Lemma 12** (Dithered directional smoothing)**.** *Let $p$ be prime, $q = pQ$, and assume the preceding trapdoor conditions, $B \geq 1$, and $K^2B^2 < q/2$. There is an efficiently extractable $x \in \mathbb{Z}_p$, depending only on $(A, a, z, T)$, such that, for independent $r \leftarrow\!\!\$\, D_{\mathbb{Z}^n,\delta}$, $g \leftarrow\!\!\$\, D_{\mathbb{Z},\delta}$, $U \leftarrow\!\!\$\, I_Q$, and $h \leftarrow\!\!\$\, \mathbb{Z}_p$,*

$$(Ar, ar, zr + g + U) \approx_{\varepsilon_{\mathsf{sm}}} (Ar, ar + Qh, zr + g + U - Qxh),$$

*where all response coordinates are reduced modulo $q$ and*

$$\varepsilon_{\mathsf{sm}} = \frac{\rho_{q/\delta}(\mathbb{Z}^{n+1} \setminus B\mathcal{B}_2^{n+1})}{2\rho_{q/\delta}(q\mathbb{Z}^{n+1})}. \tag{6}$$

*The assertion is simultaneous for all $z$, for every fixed valid trapdoor instance. For $\ell$ independent columns the error is at most $\ell\varepsilon_{\mathsf{sm}}$.*

*Proof.* Run Lemma 11. If it returns $\perp$ or a generator with $t_* = 0 \pmod p$, set $x = 0$. Otherwise set $x = v_* t_*^{-1} \pmod p$; this is the only modular division needed.

Consider characters indexed by $(u, v, t) \in \mathbb{Z}_q^k \times \mathbb{Z}_q^2$, in the response order $(Ar, ar, zr + g + U)$. Write $s = q/\delta$. Poisson summation on $\mathbb{Z}^{n+1}$ gives their Fourier coefficients as

$$\widehat{P}(u, v, t) = \phi_Q(t) \frac{\rho_s((uA + va + tz, t) + q\mathbb{Z}^{n+1})}{\rho_s(q\mathbb{Z}^{n+1})}, \qquad \phi_Q(t) = \frac{1}{Q} \sum_{j \in I_Q} e^{2\pi i t j/q}. \tag{7}$$

The cosets are independent of the chosen integer lifts. The elementary geometric sum shows

$$\phi_Q(t) = 0 \quad \text{if } t \neq 0 \pmod q \text{ and } t = 0 \pmod p.$$

This identity does not use $\gcd(p, Q) = 1$.

Averaging over the shifts $(0, Qh, -Qxh)$ keeps exactly the characters with $v - xt = 0 \pmod p$. We show that every other character is either killed by $\phi_Q$ or has no lift of length at most $B$. If such a short lift exists, it is a nonzero integer multiple of $(f_*, t_*)$. Its last coordinate is nonzero modulo $q$ by Lemma 10. When $p \mid t_*$, the corresponding dither multiplier vanishes. Otherwise its row coefficients satisfy $v - xt = 0 \pmod p$ by uniqueness and additivity of those coefficients. Both alternatives prove the claim.

The rows of $M_z$ are independent over $\mathbb{Z}_q$: a right inverse for $A_0$ and the last coordinate 1 make $M_z$ surjective. Hence distinct characters in (7) correspond to disjoint lattice cosets. Fourier inversion and the triangle inequality bound the total variation distance from the averaged distribution by half the sum of the absolute values of the removed Fourier coefficients. Using $|\phi_Q| \leq 1$ and the preceding claim bounds this sum by the ambient Gaussian tail in (6). A hybrid over independent columns gives the last assertion. □

**Correctness and security.**

**Theorem 2** (Correctness)**.** *If* (4) *holds, Construction 1 outputs* $v_0 + xv_1$ *except with probability* $\varepsilon_{\mathsf{dec}}$*.*

*Proof.* Linearity of the encoding gives

$$xt_1 + t_0 - sC = \mathsf{Enc}(v_0 + xv_1) + eR + g + U \pmod q.$$

Apply (4). □

**Theorem 3** (Security)**.** *Assume uniform-secret decision LWE for* $(k, n, q, \beta)$*, the stated trapdoor interface,* $B \geq 1$*, and* $K^2B^2 < q/2$*. Construction 1 has computational receiver privacy against a semi-honest sender and statistical sender privacy against a malicious receiver, with simulation error at most*

$$\nu_{\mathsf{Td}} + \ell\varepsilon_{\mathsf{sm}},$$

*apart from negligible sampling errors. Together with Theorem 2, these properties realize the vector OLE functionality over* $\mathbb{Z}_p$ *in the CRS model against a statically corrupted malicious receiver or a statically corrupted semi-honest sender.*

*Proof.* Receiver privacy follows by replacing $sA + e$ by a uniform vector under LWE. Then $z = sA + e - xa$ is independent of $x$, even given $(A, a)$, and the sender's prescribed computation preserves indistinguishability.

For a malicious receiver, the simulator generates $(A_0, T)$ using the trapdoor algorithm and generates the public code honestly. On receiving $z$, before learning anything about the sender's inputs, it extracts the value $x$ from Lemma 12 and submits it to the ideal functionality. Given $y = v_0 + xv_1 \in \mathbb{Z}_p^{\ell'}$, it samples fresh $R, g, U$ and $h \leftarrow_{\$} \mathbb{Z}_p^{\ell}$, and returns

$$C = AR, \qquad t_1 = aR + Qh, \qquad t_0 = zR + g + U + \mathsf{Enc}(y) - xQh \pmod q.$$

For every fixed sender input pair, this is precisely the real response averaged over independent shifts $(0, Qh_j, -Qxh_j)$ in each column: translate $h$ by the unique $\bar{v}_1 \in \mathbb{Z}_p^{\ell}$ with $Q\bar{v}_1 = \mathsf{Enc}(v_1)$. Lemma 12 therefore gives distance at most $\ell\varepsilon_{\mathsf{sm}}$. The simulated CRS contributes $\nu_{\mathsf{Td}}$. No good-CRS event depending on an adversarial residual, and no receiver-message rank test, is required. □

The extraction statement is deterministic for every receiver message on a valid simulated CRS. Consequently polynomially many responses to the same first message use the same extracted $x$; their errors add by hybrids. Likewise a polynomial number of sessions sharing the CRS only multiplies the response error, while the CRS error is charged once. This does not assert the earlier multi-input batch extension or security against a malicious sender.

## 4.3 Parameter Instantiation and Comparison

We give one asymptotic parameter family and one constant-rate code that work for every factorization $q = pQ$, where $p$ is prime and $Q \geq 1$ is an integer. Neither primality of $Q$ nor coprimality of $p$ and $Q$ is needed. The case $Q = 1$ retains the full plaintext field $\mathbb{Z}_q$ of the prime-modulus construction.

**Choice of parameters.** Let

$$k = 8\lambda, \qquad m = k + 1, \qquad L_q = \lceil (\log_2 \lambda)^2 \rceil,$$

and choose $q = pQ$ with $\log_2 q = \Theta(L_q)$. Such moduli have polynomial bit length. For example, one may choose a prime $p = q$ in $[2^{L_q}, 2^{L_q+1})$ and $Q = 1$. Alternatively, for any specified polynomially bounded prime $p$, take $Q = \lceil 2^{L_q}/p \rceil$. Prime powers are also allowed: for fixed $p$, take $q = p^{\lceil L_q / \log_2 p \rceil}$ and $Q = q/p$. These examples have $q = 2^{\Theta(\log^2 \lambda)}$ and can be generated in expected polynomial time; only the first example requires generating a prime of quasi-polynomial magnitude.

Use the trapdoor interface of Theorem 3 with

$$n = \Theta(m \log q), \qquad K \geq \max\{1, \|T\|_{\mathrm{op}}\}, \qquad K = \mathrm{poly}(\lambda), \qquad \nu_{\mathsf{Td}} \leq 2^{-8\lambda}.$$

Here the bound on $K$ holds on the trapdoor event specified in that interface. The error $\nu_{\mathsf{Td}} \leq 2^{-8\lambda}$ is obtained by using internal security parameter $8\lambda$ in Lemma 8; this leaves the displayed asymptotic bounds unchanged. Let $\ell' = \mathrm{poly}(\lambda)$ and set

$$\beta = \lambda, \qquad \delta = q^{3/4}, \qquad s_{\mathsf{sm}} = q/\delta = q^{1/4}, \qquad B = \lfloor q^{1/3} \rfloor.$$

The honest error scale is $\beta\sqrt{n} = \mathrm{poly}(\lambda)$; it is independent of the substantially larger dual cutoff $B$. For sufficiently large $\lambda$,

$$\frac{(BK)^2}{q} \leq K^2 q^{-1/3} = o(1).$$

Consequently the extraction and short-direction conditions, including $BK < \sqrt{q/2}$ as required in Theorem 3, hold.

The absolute Gaussian tail from Lemma 9 satisfies

$$\begin{aligned} E_B &:= \rho_{s_{\mathsf{sm}}}(\mathbb{Z}^{n+1} \setminus B\mathcal{B}_2^{n+1}) \\ &\leq \exp\left(-\frac{\pi B^2}{2s_{\mathsf{sm}}^2}\right)(1 + \sqrt{2}s_{\mathsf{sm}})^{n+1} \\ &\leq \exp\bigl(-\tfrac{\pi}{8}q^{1/6} + O(n \log q)\bigr) \leq 2^{-8\lambda}. \end{aligned}$$

The last two inequalities hold for sufficiently large $\lambda$: $B \geq q^{1/3}/2$, whereas $q^{1/6}$ grows faster than every polynomial in $\lambda$. The statistical sender-security error, including the hybrid over $\ell$ columns and the trapdoor approximation, is therefore negligible. Receiver privacy assumes LWE for this displayed uniform-secret, quasi-polynomial-modulus parameter family. We do not claim a worst-case lattice reduction for that particular regime.

**One code for all factorizations.** Fix $L_0 = 32$, put $\ell = L_0\ell'$, and sample $a_{\mathsf{code}}$ uniformly from $\mathbb{Z}_p^{L_0} \setminus \{0\}$, once for the entire code. Encode a scalar $v \in \mathbb{Z}_p$ as

$$\mathsf{Enc}_{\mathrm{block}}(v) = Qva_{\mathsf{code}} \pmod q,$$

and encode vectors by applying this map to each coordinate. The code is $\mathbb{Z}_p$-linear, has rate $1/L_0$ over its plaintext alphabet, and takes values in $(Q\mathbb{Z}_p)^\ell$. In particular, $\mathsf{Enc}(v_0 + xv_1) = \mathsf{Enc}(v_0) + x\mathsf{Enc}(v_1)$ for $x \in \mathbb{Z}_p$.

For each received block, take its centered integer lift and solve exact CVP in the full-rank lattice

$$\mathcal{L}_{\mathsf{code}} = Q\big(p\mathbb{Z}^{L_0} + \mathbb{Z}[a_{\mathsf{code}}]_p\big) = q\mathbb{Z}^{L_0} + Q\mathbb{Z}[a_{\mathsf{code}}]_p.$$

Fix a deterministic rule for CVP ties. If $b$ is the resulting closest lattice vector, reduce $b/Q$ modulo $p$ and recover its coefficient of $a_{\mathsf{code}}$ using any nonzero coordinate of $a_{\mathsf{code}}$. A lattice basis is obtained by Hermite normal form from $(qI_{L_0} \mid Q[a_{\mathsf{code}}]_p)$. Exact CVP is polynomial in the input bit length in the fixed dimension $L_0$ [MV10], so code generation, encoding, and decoding are polynomial time. The same decoder is used below in both parameter ranges.

**Correctness when $p$ is small.** Suppose first that $p \le q^{1/8}$. This case does not require a random minimum-distance property of the code. Every lattice vector belongs to $Q\mathbb{Z}^{L_0}$. Thus, if coordinatewise rounding to $Q\mathbb{Z}$ recovers a transmitted codeword uniquely, that codeword is also the unique closest point of $\mathcal{L}_{\mathsf{code}}$.

Write the receiver's pre-decoding word as

$$\mathsf{Enc}(v_0 + xv_1) + U + w \pmod q, \qquad w = eR + g,$$

where the coordinates of $U$ are independent and uniform on the centered interval $I_Q$ of $Q$ consecutive integers. For every fixed integer offset $w_j$, at most $|w_j| + 1$ of the $Q$ possible positions of $U_j$ fail to lie strictly inside the correct rounding cell. This bound charges boundary ties as failures, including when $Q$ is even. A union bound, first conditional on $w$ and then averaged over $w$, gives

$$\Pr[\text{decoding failure}] \le \frac{\mathbb{E}\|eR + g\|_1 + \ell}{Q}. \tag{8}$$

Reduction modulo $q$ does not change this sufficient event, because $q\mathbb{Z}^{L_0} \subseteq \mathcal{L}_{\mathsf{code}}$.

For completeness, if $X \leftarrow\!\$\, D_{\mathbb{Z},s}$ and $s \ge 1$, then

$$\mathbb{E}\exp\big(\pi X^2/(2s^2)\big) = \frac{\rho_{\sqrt{2}s}(\mathbb{Z})}{\rho_s(\mathbb{Z})} \le 1 + \sqrt{2}.$$

Here we used $\rho_s(\mathbb{Z}) \ge s$ and $\rho_{\sqrt{2}s}(\mathbb{Z}) \le 1 + \sqrt{2}s$. Jensen's inequality implies $\mathbb{E}X^2 \le s^2$. The coordinates of $e$, each column $r_j$ of $R$, and $g$ are independent and centered. Therefore

$$\mathbb{E}|(er_j) + g_j| \le \sqrt{\mathbb{E}((er_j) + g_j)^2} \le \delta\sqrt{n\beta^2 + 1},$$

and hence

$$\Pr[\text{decoding failure}] \le \frac{\ell\delta\sqrt{n\beta^2+1}+\ell}{Q} \le \mathrm{poly}(\lambda)q^{-1/8} = \mathrm{negl}(\lambda).$$

This bound holds for every nonzero choice of $a_{\mathsf{code}}$.

**Correctness when $p$ is large.** Now suppose $p > q^{1/8}$, and put

$$\xi = \frac{Qp^{4/5}}{8}.$$

Before conditioning $a_{\mathsf{code}}$ to be nonzero, the vector $va_{\mathsf{code}}$ is uniform in $\mathbb{Z}_p^{L_0}$ for each fixed $v \neq 0$. The number of centered integer vectors with norm at most $2\xi/Q$ is at most $(4\xi/Q+1)^{L_0}$. A union bound over $v \neq 0$ and conditioning on $a_{\mathsf{code}} \neq 0$ therefore give, for sufficiently large $p$,

$$\begin{aligned}\eta_{\mathsf{code}} &:= \Pr\big[\text{a nonzero codeword has centered norm} \le 2\xi\big] \\ &\le \frac{p(4\xi/Q+1)^{L_0}}{(1-p^{-L_0})p^{L_0}} \le 2p^{1-L_0/5} = 2p^{-27/5} \le 2q^{-27/40}.\end{aligned}$$

On the complementary event, $\lambda_1(\mathcal{L}_{\mathsf{code}}) > 2\xi$: nonzero vectors that reduce to zero modulo $q$ have norm at least $q > 2\xi$, and all other vectors lift nonzero codewords. Exact CVP consequently corrects every integer error of norm at most $\xi$ in a block. A global error of norm at most $\xi$ restricts to an error of norm at most $\xi$ in each block. There is only one bad-code event, because all blocks use the same $a_{\mathsf{code}}$.

Gaussian tails imply, except with probability $(2\ell+1)e^{-\Omega(n)}$,

$$\|e\|_2 \le 2\beta\sqrt{n}, \qquad \max_j \|r_j\|_2 \le 2\delta\sqrt{n}, \qquad \max_j |g_j| \le 2\delta\sqrt{n}.$$

On this event,

$$\|U + eR + g\|_2 \le \frac{Q\sqrt{\ell}}{2} + \sqrt{\ell}\big(4\beta\delta n + 2\delta\sqrt{n}\big).$$

Both terms are smaller than $\xi$ with asymptotically vanishing relative size. Indeed,

$$\frac{Q\sqrt{\ell}/2}{\xi} = 4\sqrt{\ell}\,p^{-4/5} \le \mathrm{poly}(\lambda)q^{-1/10}, \qquad \frac{\mathrm{poly}(\lambda)\delta}{\xi} \le \mathrm{poly}(\lambda)q^{-1/20},$$

where the second inequality uses $p \le q$ and $Qp^{4/5} = qp^{-1/5} \ge q^{4/5}$. Thus Theorem 2 applies, and the total correctness error in this range is at most $2q^{-27/40} + (2\ell+1)e^{-\Omega(n)}$, which is negligible.

**Comparison with the prime-modulus construction.** When $Q = 1$, the interval $I_Q$ is $\{0\}$, so there is no uniform dither. The code above is exactly the same kind of full-field, constant-rate code used in the prime-modulus construction. The extra scalar Gaussian changes the standard-deviation bound for a coordinate of the honest error from at most $\delta\|e\|_2$ to at most $\delta\sqrt{\|e\|_2^2 + 1}$. The choices $\beta = \lambda$, $\delta = q^{3/4}$, polynomial dimensions, and constant code rate therefore incur no asymptotic exponent loss.

For $Q > 1$, full-cell dither has a real correctness cost: the raw scalar rounding error is controlled by the ratio of the honest noise magnitude to $Q$, as in (8), rather than just by a Gaussian tail beyond the cell boundary. Thus a merely polynomial gap between $Q$ and the noise scale need not give negligible error without additional coding. In the parameter family above, the small-$p$ gap is already superpolynomial, and for large $p$ the same constant-dimensional Euclidean code absorbs the dither. No increase in the displayed modulus exponent, Gaussian-width exponent, or code dimension is required. The ratio $\log q / \log p$ between ambient and plaintext symbol lengths is the existing cost of the subgroup encoding; it is not introduced by the repair.

# AI Disclosure

GPT 5.6 Sol and GPT 6 Astra were used to assist in revising this manuscript after You Lyu brought a parameter-instantiability issue to the authors' attention.

# Acknowledgment

The authors would like to thank You Lyu for bringing up a parameter-instantiability problem in an earlier version of this manuscript published at PKC 2022; A misapplication of the partial smoothing lemma led to a parameter conflict which made the resulting scheme uninstantiable.

Pedro Branco thanks the support from DP-PMI and FCT (Portugal) through the grant PD/BD/135181/2017. This work is supported by Security and Quantum Information Group of Instituto de Telecomunicações, by the Fundação para a Ciência e a Tecnologia (FCT) through national funds, by FEDER, COMPETE 2020, and by Regional Operational Program of Lisbon, under UIDB/50008/2020.

# References

[ADI+17] Benny Applebaum, Ivan Damgård, Yuval Ishai, Michael Nielsen, and Lior Zichron. Secure arithmetic computation with constant computational overhead. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 223–254, Santa

Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany.

[AIK11] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. How to garble arithmetic circuits. In Rafail Ostrovsky, editor, *52nd Annual Symposium on Foundations of Computer Science*, pages 120–129, Palm Springs, CA, USA, October 22–25, 2011. IEEE Computer Society Press.

[Ban93] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(4):625–636, 1993.

[BCGI18] Elette Boyle, Geoffroy Couteau, Niv Gilboa, and Yuval Ishai. Compressing vector OLE. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018: 25th Conference on Computer and Communications Security*, pages 896–912, Toronto, ON, Canada, October 15–19, 2018. ACM Press.

[BD18] Zvika Brakerski and Nico Döttling. Two-message statistically sender-private OT from LWE. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018: 16th Theory of Cryptography Conference, Part II*, volume 11240 of *Lecture Notes in Computer Science*, pages 370–390, Panaji, India, November 11–14, 2018. Springer, Heidelberg, Germany.

[BEPU+20] Carsten Baum, Daniel Escudero, Alberto Pedrouzo-Ulloa, Peter Scholl, and Juan Ramón Troncoso-Pastoriza. Efficient protocols for oblivious linear function evaluation from ring-lwe. In Clemente Galdi and Vladimir Kolesnikov, editors, *Security and Cryptography for Networks*, pages 130–149, Cham, 2020. Springer International Publishing.

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 719–737, Cambridge, UK, April 15–19, 2012. Springer, Heidelberg, Germany.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Computer Science*, pages 136–145, Las Vegas, NV, USA, October 14–17, 2001. IEEE Computer Society Press.

[CDI+19] Melissa Chase, Yevgeniy Dodis, Yuval Ishai, Daniel Kraschewski, Tianren Liu, Rafail Ostrovsky, and Vinod Vaikuntanathan. Reusable non-interactive secure computation. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part III*, volume 11694 of *Lecture Notes in*

*Computer Science*, pages 462–488, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

[dCJV20] Leo de Castro, Chiraag Juvekar, and Vinod Vaikuntanathan. Fast vector oblivious linear evaluation from ring learning with errors. Cryptology ePrint Archive, Report 2020/685, 2020. `https://eprint.iacr.org/2020/685`.

[DGN+17] Nico Döttling, Satrajit Ghosh, Jesper Buus Nielsen, Tobias Nilges, and Roberto Trifiletti. TinyOLE: Efficient actively secure two-party computation from oblivious linear function evaluation. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 2263–2276, Dallas, TX, USA, October 31 – November 2, 2017. ACM Press.

[DKM12] Nico Döttling, Daniel Kraschewski, and Jörn Müller-Quade. Statistically secure linear-rate dimension extension for oblivious affine function evaluation. In Adam Smith, editor, *ICITS 12: 6th International Conference on Information Theoretic Security*, volume 7412 of *Lecture Notes in Computer Science*, pages 111–128, Montreal, QC, Canada, August 15–17, 2012. Springer, Heidelberg, Germany.

[DKMQ12] Nico Döttling, Daniel Kraschewski, and Jörn Müller-Quade. David & Goliath oblivious affine function evaluation - asymptotically optimal building blocks for universally composable two-party computation from a single untrusted stateful tamper-proof hardware token. Cryptology ePrint Archive, Report 2012/135, 2012. `https://eprint.iacr.org/2012/135`.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred Aho, editor, *19th Annual ACM Symposium on Theory of Computing*, pages 218–229, New York City, NY, USA, May 25–27, 1987. ACM Press.

[GN19] Satrajit Ghosh and Tobias Nilges. An algebraic approach to maliciously secure private set intersection. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 154–185, Darmstadt, Germany, May 19–23, 2019. Springer, Heidelberg, Germany.

[GNN17] Satrajit Ghosh, Jesper Buus Nielsen, and Tobias Nilges. Maliciously secure oblivious linear function evaluation with constant overhead. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624

of *Lecture Notes in Computer Science*, pages 629–659, Hong Kong, China, December 3–7, 2017. Springer, Heidelberg, Germany.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 197–206, Victoria, BC, Canada, May 17–20, 2008. ACM Press.

[GS19] Satrajit Ghosh and Mark Simkin. The communication complexity of threshold private set intersection. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 3–29, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

[HIMV19] Carmit Hazay, Yuval Ishai, Antonio Marcedone, and Muthuramakrishnan Venkitasubramaniam. LevioSA: Lightweight secure arithmetic computation. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019: 26th Conference on Computer and Communications Security*, pages 327–344. ACM Press, November 11–15, 2019.

[IPS09] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Secure arithmetic computation with no honest majority. In Omer Reingold, editor, *Theory of Cryptography*, pages 294–314, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[JVC18] Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. GAZELLE: A low latency framework for secure neural network inference. In William Enck and Adrienne Porter Felt, editors, *USENIX Security 2018: 27th USENIX Security Symposium*, pages 1651–1669, Baltimore, MD, USA, August 15–17, 2018. USENIX Association.

[LP94] Mody Lempel and Azaria Paz. An algorithm for finding a shortest vector in a two-dimensional modular lattice. *Theoretical Computer Science*, 125(2):229 – 241, 1994.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718, Cambridge, UK, April 15–19, 2012. Springer, Heidelberg, Germany.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.

[MV10] Daniele Micciancio and Panagiotis Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. In Leonard J. Schulman, editor, *42nd Annual ACM Symposium on Theory of Computing*, pages 351–358, Cambridge, MA, USA, June 5–8, 2010. ACM Press.

[MZ17] Payman Mohassel and Yupeng Zhang. SecureML: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy*, pages 19–38, San Jose, CA, USA, May 22–26, 2017. IEEE Computer Society Press.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 554–571, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Heidelberg, Germany.

[Qua20] Willy Quach. UC-secure OT from LWE, revisited. In Clemente Galdi and Vladimir Kolesnikov, editors, *Security and Cryptography for Networks*, pages 192–211, Cham, 2020. Springer International Publishing.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 84–93, Baltimore, MA, USA, May 22–24, 2005. ACM Press.

[Yao82] Andrew C. Yao. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 160–164, 1982.