






Efficient Protocols for Oblivious Linear Function Evaluation from Ring-LWE

Carsten Baum¹, Daniel Escudero¹, Alberto Pedrouzo-Ulloa², Peter Scholl¹, and Juan Ramón Troncoso-Pastoriza³

¹ Aarhus University, Aarhus, Denmark
{cbaum,escudero,peter.scholl}@cs.au.dk   

² University of Vigo, Vigo, Galicia, Spain
apedrouzo@gts.uvigo.es 

³ EPFL, Lausanne, Switzerland
juan.troncoso-pastoriza@epfl.ch 

Abstract. An oblivious linear function evaluation protocol, or OLE, is a two-party protocol for the function $f(x) = ax + b$, where a sender inputs the field elements a, b , and a receiver inputs x and learns $f(x)$. OLE can be used to build secret-shared multiplication, and is an essential component of many secure computation applications including general-purpose multi-party computation, private set intersection and more.

In this work, we present several efficient OLE protocols from the ring learning with errors (RLWE) assumption. Technically, we build two new passively secure protocols, which build upon recent advances in homomorphic secret sharing from (R)LWE (Boyle et al., Eurocrypt 2019), with optimizations tailored to the setting of OLE. We upgrade these to active security using efficient amortized zero-knowledge techniques for lattice relations (Baum et al., Crypto 2018), and design new variants of zero-knowledge arguments that are necessary for some of our constructions.

Our protocols offer several advantages over existing constructions. Firstly, they have the lowest communication complexity amongst previous, practical protocols from RLWE and other assumptions; secondly, they are conceptually very simple, and have just one round of interaction for the case of OLE where b is randomly chosen. We demonstrate this with an implementation of one of our passively secure protocols, which can perform more than 1 million OLEs per second over the ring \mathbb{Z}_m , for a 120-bit modulus m , on standard hardware.

1 Introduction

Oblivious linear function evaluation, or OLE, is a two-party protocol between a sender, with input $a, b \in \mathbb{F}$, and a receiver, who inputs $x \in \mathbb{F}$ and receives $y = ax + b$. OLE is an arithmetic generalization of oblivious transfer to a larger field \mathbb{F} , since OLE over \mathbb{F}_2 can be seen as equivalent to oblivious transfer on the messages z_0, z_1 by setting $a = z_0 + z_1$ and $b = z_0$, so the receiver learns $y = z_x$. Similarly to oblivious transfer, OLE can be used in constructions of secure two-party and multi-party computation, and is particularly useful for the setting of securely computing arithmetic circuits over \mathbb{F} [34,29,4,28], where OT tends to be less efficient. As well as general secure computation protocols, OLE can be used to carry out specific tasks like private set intersection [31], secure matrix multiplication and oblivious polynomial evaluation [41,44].

OLE can be constructed from a range of “public-key” type assumptions. In the simplest, folklore construction, the receiver encrypts its input x using a linearly homomorphic encryption scheme and gives this to the sender. Using the homomorphic properties of the scheme, the sender computes an encryption of $y = ax + b$ and sends this back to the receiver to decrypt. This approach can be instantiated with Paillier encryption or lattice-based encryption based on the learning with errors (LWE) [43] or RLWE assumptions [38], and has been implicitly used in several secure multi-party computation protocols [12,36,40]. There are also constructions of OLE from coding-theoretic assumptions [41,34,30] which mostly rely on the hardness of decoding Reed-Solomon codes in certain parameter regimes with a high enough noise rate. These constructions are asymptotically efficient, but so far have not been implemented in practice, to the best of our knowledge. For the special (and easier) case of *vector-OLE*, which is a large batch of many OLEs with the same input x from the receiver, there are efficient constructions

from more standard coding-theoretic assumptions over general codes, which also have good performance in practice [4,14,44,15].

Despite the fact there are many existing constructions of OLE, either implicit or explicit in the literature, very few of these works study the practical efficiency of OLE in its own right (except for the special case of vector-OLE). Instead, most of the aforementioned works either focus on the efficiency of higher-level primitives such as secure multi-party computation, or mainly discuss asymptotic efficiency rather than performance in practice. In this work, we advocate for the practical study of OLE as a *standalone primitive*. This has the benefits that it can be plugged into any higher-level application that needs it in a modular way, potentially simplifying analysis and security proofs compared with a more monolithic approach.

1.1 Our Contributions

We present and study new OLE protocols with security based on the ring learning with errors (RLWE) assumption, with passive and active security. Our passively secure protocols are very simple, consisting of just one message per party, and our most efficient variant achieves the lowest communication complexity of any practical (implemented) OLE protocol we are aware of, requiring around half the bandwidth of previous solutions. We add active security using zero-knowledge proofs, which have a low amortized complexity when performing a large number of OLEs, giving only a small communication overhead over the passive protocols. To adapt existing zero-knowledge proof techniques to our protocols, we have to make several modifications, and describe a new amortized proof of knowledge that can be used to show a batch of *secret-key* (R)LWE ciphertexts is well-formed (previous techniques only apply to *public-key* ciphertexts).

We have implemented and benchmarked our most efficient passively secure protocol, and show it can compute more than 1 million OLEs per second on a standard laptop, over a ≈ 120 -bit ring \mathbb{Z}_m where m is the product of two CPU word-sized primes. The communication cost per OLE is around 4 elements of \mathbb{Z}_m per party, and the amortized complexity of our actively secure protocol is almost the same, when computing a large enough number of OLEs. This is almost half the communication cost of previous protocols based on RLWE, and less than 25% of the cost of an actively secure protocol based on oblivious transfer and noisy Reed-Solomon encodings [30].

1.2 Outline

In Section 1.3 below, we present an overview of the main techniques in our constructions. We then describe some preliminaries in Section 2. Section 3 contains our OLE protocols based on public-key RLWE encryption, which only require a standard public key infrastructure as a setup assumption. In Section 4, we present more efficient protocols which reduce communication using secret-key encryption, and a more specialized setup assumption. Then, in Section 5, we present details on the zero-knowledge arguments which are used to make the previous protocols actively secure. Finally, in Section 6, we analyze the concrete efficiency of our solutions, compare this with previous OLE protocols, and present implementation results for our most efficient passively secure protocol.

1.3 Techniques

Our protocols construct a symmetric variant of OLE, where one party, Alice, inputs a field element $u \in \mathbb{F}$, the other party, Bob, inputs $v \in \mathbb{F}$, and the parties receive random values α and β (respectively) such that $\alpha + \beta = u \cdot v$. This can easily be used to construct an OLE by having the sender, say Alice, one-time-pad encrypt her additional input using α , allowing Bob to correct his output accordingly. In this formulation, OLE is also equivalent to producing an additive secret-sharing of the product of two private inputs; this type of secret-shared multiplication is an important building block in multi-party computation protocols, for instance in constructing Beaver multiplication triples [10]. In our protocols, we first create OLEs over a large polynomial

ring $\mathcal{R}_m = \mathbb{Z}_m[X]/(X^N + 1)$, which comes from the RLWE assumption, and then convert each OLE over \mathcal{R}_m to a batch of N OLEs over \mathbb{Z}_m , for some prime modulus m , using packing techniques from homomorphic encryption [45].

Our point of departure is the recent *homomorphic secret sharing* scheme by Boyle et al. [18], based on LWE or RLWE. Homomorphic secret sharing is a form of secret sharing in which shares can be computed upon non-interactively, such that the parties end up with an *additive* secret sharing of the result of the computation. HSS was first constructed under the DDH assumption [17] and variants of threshold and multi-key fully homomorphic encryption [27], followed by the more efficient lattice-based construction of [18], which supports homomorphic computation of branching programs (or, “restricted multiplication” circuits where every multiplication gate must involve at least one input wire). Note that any “public-key” type two-party HSS scheme that supports multiplication leads to a simple OLE protocol: each party sends a secret-sharing of its input, then both parties multiply the shares to obtain an additive share of the product.

Efficient OLE from a public-key setup. Our first construction can be seen as taking the HSS scheme of Boyle et al. and optimizing it for the specific functionality of OLE. When plugging in their scheme to perform OLE, a single share from one party consists of two RLWE ciphertexts: one encrypting the message, and one encrypting a function of the secret key, which is needed to perform the multiplication. Our first observation is that, in the setting of OLE where we have two parties who each have one of the inputs to be multiplied, we can reduce this to just *one ciphertext per party*, where Alice sends an encryption of her input u multiplied by a secret key, and Bob sends an encryption of his input. Both of these ciphertexts, including the one dependent on the secret key, can be created from a standard public-key infrastructure-like setup where Alice and Bob have each others’ RLWE public keys, thanks to a weak KDM security property of the scheme. This gives a communication complexity of two \mathcal{R}_q elements per party, for a RLWE ciphertext modulus q , to create a single ring-OLE over \mathcal{R}_m . We can also obtain a further saving by sending one party’s ciphertext at a smaller modulus $p < q$.

Reducing communication with a dedicated setup. Our second protocol considers a different setup assumption, where the parties are assumed to have access to a single OLE over \mathcal{R}_q , which gives them secret shares of the product of two RLWE secret keys. With this, we are able to replace the public-key RLWE ciphertexts from the previous protocol with *secret-key* ciphertexts, which can be of size just one ring element instead of two. This cuts the overall communication in half, and also reduces computational costs.

Achieving active security. To obtain security against active corruptions, we need to ensure that both parties’ RLWE ciphertexts are correctly generated, in particular, that the small “error” polynomials used as encryption randomness were generated correctly (and not too large). For a public key RLWE encryption, this boils down to proving knowledge of a short vector $\mathbf{s} \in \mathbb{Z}_q^n$, such that $\mathbf{A}\mathbf{s} = \mathbf{c}$ where \mathbf{A}, \mathbf{c} are public values defined by the RLWE public key and ciphertext, respectively. In practice, we do not know efficient methods of proving the above statement. Instead, we can obtain good *amortized* efficiency when proving knowledge for *many* such relations of the form

$$\mathbf{A}\bar{\mathbf{s}}_i = \mathbf{c}_i \tag{1}$$

for the same matrix \mathbf{A} , where now the secret $\bar{\mathbf{s}}_i$ may have slightly larger coefficients than the original secret \mathbf{s}_i . This overhead is known as the *soundness slack* parameter, and comes from the fact that a dishonest prover can sometimes make the proof succeed even when \mathbf{s}_i is slightly larger than the claimed bound. Efficient amortized proofs for (1) have been given in several works [37,24,7,22], most recently with a communication overhead that is *independent* of the number of relations being proven [5].

Proving correctness of a batch of public-key RLWE ciphertexts can be essentially done by proving a batch of relations of the form in (1), allowing use of these efficient amortized proofs.

To achieve active security in our public-key OLE protocol, we use a slightly modified version of the proof from [5], by allowing different size bounds to be proven for different components of \mathbf{s}_i . This gives us tighter parameters for the encryption scheme.

On the other hand, for our second protocol, things are not so straightforward. To see why, recall that a batch of secret-key RLWE ciphertexts have the form:

$$(a_i, a_i \cdot s + e_i + (q/p) \cdot x_i) \quad (2)$$

Here, a_i is a random element in the polynomial ring $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^N + 1)$, e_i is a small error value in \mathcal{R}_q , and $s \in \mathcal{R}_q$ is the secret key. We want to prove that both s and e_i have small coefficients.

The problem is, since a_i is different for each ciphertext, these cannot be expressed in the form of (1), since they are not linear in a fixed public value. This was *not* the case for the public-key setting, where every ciphertext is linear in the fixed public key; here, by switching to a secret-key encryption scheme to improve efficiency, we can unfortunately no longer apply the amortization techniques of [5].

Furthermore, there is a second obstacle, since we now have a special preprocessing phase which gives out shares of $s_A \cdot s_B$, for the parties' RLWE secret keys s_A and s_B . These must be the *same* secret keys that are used to produce the encryptions, and to ensure this, we also have to tie these together into the ZK proof statement.

To work around these issues, we perform two steps. Firstly, we modify the preprocessing so that each party gets a *commitment* to its secret key, under a suitable homomorphic commitment scheme (which can also be based on lattices [8]). We then design a new proof of knowledge, which proves knowledge of short (s, e_i, x_i) satisfying (2) with similar amortized efficiency to the proof from [5] for (1). Our proof simultaneously guarantees the secret s is the same s that was committed to in the preprocessing, leveraging the homomorphic properties of the commitment scheme.

2 Preliminaries

In this section we introduce some preliminaries and notation we will use. As basic notation, we write $\boldsymbol{\alpha} \star \boldsymbol{\beta}$ to denote the component-wise product of the vectors $\boldsymbol{\alpha}, \boldsymbol{\beta}$.

2.1 Rings & Rounding

Let q be an odd integer and $N = 2^r$ be a power of two. We define the ring $\mathcal{R} := \mathbb{Z}[X]/\langle X^N + 1 \rangle$ as well as $\mathcal{R}_q = \mathcal{R}/\langle q \rangle$ as the reduction of the polynomials of \mathcal{R} modulo q . Representing the coefficients of $f \in \mathcal{R}_q$ uniquely by its representatives from $[-(q-1)/2, (q-1)/2]$ we define $\|f\|_\infty$ as the largest norm of any coefficient of f when considered over the above interval. We define by $\mathcal{U}(R)$ the uniform distribution over the finite set R and furthermore let $S_\beta = \{x \in \mathcal{R} \mid \|x\|_\infty \leq \beta\}$.

We now introduce the computational problems we use over \mathcal{R}_q , which are Ring-LWE, Module-LWE and Module-SIS. We use Ring-LWE in our basic OLE protocols, while Module-LWE and -SIS are used for our zero-knowledge argument with homomorphic commitments.

Definition 1 (Ring-LWE). *Let \mathcal{R}_q be a ring as defined above, $n \in \mathbb{N}^+$ and $\sigma \in \mathbb{R}^+$. Let \mathcal{D}_σ be a discrete Gaussian distribution over \mathcal{R}_q with standard deviation σ , and \mathcal{D}_{sk} be some secret key distribution over \mathcal{R}_q . We say that an algorithm \mathcal{A} has advantage ϵ in solving the $\text{RLWE}_{n,\sigma,\mathcal{D}_{\text{sk}}}$ problem if*

$$\begin{aligned} & \left| \Pr[b = 1 \mid \mathbf{a} \leftarrow \mathcal{U}(R_q^n), \mathbf{e} \leftarrow \mathcal{D}_\sigma^n, s \leftarrow \mathcal{D}_{\text{sk}}, b \leftarrow \mathcal{A}(\mathbf{a}, \mathbf{a} \cdot s + \mathbf{e})] \right. \\ & \left. - \Pr[b = 1 \mid \mathbf{a} \leftarrow \mathcal{U}(R_q^n), \mathbf{u} \leftarrow \mathcal{U}(R_q^n), b \leftarrow \mathcal{A}(\mathbf{a}, \mathbf{u})] \right| \geq \epsilon \end{aligned}$$

Definition 2 (Module-LWE). Let \mathcal{R}_q be a ring as defined above, $n, k \in \mathbb{N}^+$. The $\text{MLWE}_{n,k,\beta}$ problem asks to distinguish the distribution $[\mathbf{I}_n \ \mathbf{A}'] \cdot \mathbf{y}$ for a short \mathbf{y} , from the uniform distribution when given \mathbf{A}' . We say that an algorithm \mathcal{A} has advantage ϵ in solving the $\text{MLWE}_{n,k,\beta}$ problem if

$$\left| \Pr[b = 1 \mid \mathbf{A}' \leftarrow \mathcal{U}(R_q^{n \times (k-n)}) \wedge \mathbf{y} \leftarrow \mathcal{U}(S_\beta^k) \wedge b \leftarrow \mathcal{A}(\mathbf{A}', [\mathbf{I}_n \ \mathbf{A}'] \cdot \mathbf{y})] \right. \\ \left. - \Pr[b = 1 \mid \mathbf{A}' \leftarrow \mathcal{U}(R_q^{n \times (k-n)}) \wedge \mathbf{u} \leftarrow \mathcal{U}(R_q^n) \wedge b \leftarrow \mathcal{A}(\mathbf{A}', \mathbf{u})] \right| \geq \epsilon$$

The Module-LWE problem is widely believed to be hard for polynomial-time distinguishers when \mathbf{y} is sampled from a discrete gaussian distribution over \mathcal{R} with large enough standard deviation. The version of Module-LWE we give here has a more aggressive error distribution, but is often used in practice.

A related well-known problem is called Module-SIS.

Definition 3 (Module-SIS). Let \mathcal{R}_q be a ring as defined above and $n, k \in \mathbb{N}^+$. The $\text{MSIS}_{n,k,\beta}$ problem asks to find a short vector \mathbf{y} with $\|\mathbf{y}\|_\infty \leq \beta$ satisfying $[\mathbf{I}_n \ \mathbf{A}'] \cdot \mathbf{y} = \mathbf{0}^n$ when given a random \mathbf{A}' . We say that an algorithm \mathcal{A} has advantage ϵ in solving the $\text{MSIS}_{n,k,\beta}$ problem if

$$\Pr \left[\mathbf{y} \in S_\beta^k \wedge [\mathbf{I}_n \ \mathbf{A}'] \cdot \mathbf{y} = \mathbf{0}^n \mid \mathbf{A}' \leftarrow \mathcal{U}(R_q^{n \times (k-n)}) \wedge \mathbf{0} \neq \mathbf{y} \leftarrow \mathcal{A}(\mathbf{A}') \right] \geq \epsilon.$$

Rounding. We define by $\lfloor f \rfloor_p$ the scaling of each coefficient of f by p/q over the reals and then rounding to the nearest integer in $[-(p-1)/2, (p-1)/2]$ respectively. A simple but useful result we will use throughout our protocols is the following.

Lemma 1. Let $p|q$, $\mathbf{x} \leftarrow \mathcal{R}_q^n$ and $\mathbf{y} = \mathbf{x} + \mathbf{e} \bmod q$ for some $\mathbf{e} \in \mathcal{R}_q^n$ with $\|\mathbf{e}\|_\infty < B < q/p$. Then $\Pr[\lfloor \mathbf{y} \rfloor_p \neq \lfloor \mathbf{x} \rfloor_p \bmod p] \leq \frac{2npNB}{q}$

Proof. It suffices to consider the case $n = 1$ and $R_q = \mathbb{Z}_q$, since, by the union bound, the general case can be obtained by multiplying the probability obtained in this particular case by $N \cdot n$.

Let $x \in \mathcal{R}_q$ be uniformly random, let $e \in \mathbb{Z}_q$ be bounded in norm by B , and let $y = x + e \bmod q$. Let E be the event $\lfloor y \rfloor_p \neq \lfloor x \rfloor_p \bmod p$. To bound the probability of E notice that, due to the fact that $|e| < B$, e cannot change the nearest integer to x unless at least one of these coefficients lies in a ball of radius $(p/q)B$ centered at $k + \frac{1}{2}$ for some integer k . This condition is equivalent to x lying in a ball of radius B centered at $\frac{q}{p} \cdot (k + \frac{1}{2})$ for some integer k . Since $x \in \mathbb{Z}_q$ is uniformly random, the probability of this event is upper bounded by $2B/(q/p) = 2Bp/q$. \square

2.2 Gaussian Distributions and Simulatability

Definition 4. The continuous normal distribution over \mathbb{R}^m centered at $\mathbf{v} \in \mathbb{R}^m$ with standard deviation $\sigma \in \mathbb{R}$ is defined by the function

$$\rho_{\mathbf{v},\sigma}^m(\mathbf{x}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right)^m \exp \left(-\frac{\|\mathbf{x} - \mathbf{v}\|_2^2}{2\sigma^2} \right).$$

If $\mathbf{v} = \mathbf{0}$ then we just write $\rho_\sigma^m(\mathbf{x})$. For a countable set $S \subset \mathbb{R}^m$ we furthermore define $\rho_\sigma^m(S) = \sum_{\mathbf{x} \in S} \rho_\sigma^m(\mathbf{x})$.

Definition 5. The discrete normal distribution over \mathbb{Z}^m centered at $\mathbf{v} \in \mathbb{Z}^m$ with standard deviation $\sigma \in \mathbb{R}^m$ is defined as

$$\mathcal{D}_{\mathbf{v},\sigma}^m(\mathbf{x}) = \rho_{\mathbf{v},\sigma}^m(\mathbf{x}) / \rho_\sigma^m(\mathbb{Z}^m).$$

Throughout this work we apply \mathcal{D} to vectors from \mathcal{R}^k in which case we mean that $\mathcal{D}_\sigma(\mathbf{x})^k = \mathcal{D}_\sigma^{Nk}(\bar{\mathbf{x}})$ with $\bar{\mathbf{x}} \in \mathbb{Z}^{Nk}$ being the coefficient-wise embedding of \mathcal{R}^k into \mathbb{Z}^{Nk} . We similarly consider sampling \mathcal{R} -elements from \mathcal{D}_σ as sampling each coefficient independently from this distribution.

In order to use random variables sampled according to the aforementioned distribution we have to be able to estimate the size of its values. The following statement allows doing so:

Lemma 2 (See Lemma 4.4 of [37]). *For any $k > 1$,*

$$\Pr[\|\mathbf{x}\|_2 > k\sigma\sqrt{m} \mid \mathbf{x} \leftarrow \mathcal{D}_\sigma^m] < k^m \exp\left(\frac{m}{2}(1-k^2)\right).$$

Rejection Sampling for Product Distributions. We will have to perform rejection sampling on vectors that consist of discrete Gaussian distributions of multiple different standard deviations. Throughout this work we will use the following

Lemma 3 (Generalizes Theorem 4.6 of [37]). *Let $k \in \mathbb{N}^+$ and for $i \in [k]$ let $V_i \subseteq \mathbb{Z}^{m_i}$ such that all elements of V_i have norm less than T_i , $\sigma_i \in \mathbb{R}$ such that $\sigma_i = \alpha T_i$ and $h_i : V_i \rightarrow \mathbb{R}$ be a probability distribution. If $\alpha > 0$ and $M = \exp(12/\alpha + 1/(2\alpha^2))$ then the following algorithm \mathcal{A} :*

1. $\mathbf{v}_1 \leftarrow h_1, \dots, \mathbf{v}_k \leftarrow h_k$
2. $\mathbf{x}_1 \leftarrow \mathcal{D}_{\mathbf{v}_1, \sigma_1}^{m_1}, \dots, \mathbf{x}_k \leftarrow \mathcal{D}_{\mathbf{v}_k, \sigma_k}^{m_k}$
3. Output $(\mathbf{x}_1, \dots, \mathbf{x}_k, \mathbf{v}_1, \dots, \mathbf{v}_k)$ with probability $\min\left(\prod_{i \in [k]} \frac{\mathcal{D}_{\sigma_i}^{m_i}(\mathbf{x}_i)}{M \mathcal{D}_{\mathbf{v}_i, \sigma_i}^{m_i}(\mathbf{x}_i)}, 1\right)$

is within statistical distance $2^{-100+\log k}/M$ of the distribution of the following algorithm \mathcal{F} :

1. $\mathbf{v}_1 \leftarrow h_1, \dots, \mathbf{v}_k \leftarrow h_k$
2. $\mathbf{x}_1 \leftarrow \mathcal{D}_{\sigma_1}^{m_1}, \dots, \mathbf{x}_k \leftarrow \mathcal{D}_{\sigma_k}^{m_k}$
3. Output $(\mathbf{x}_1, \dots, \mathbf{x}_k, \mathbf{v}_1, \dots, \mathbf{v}_k)$ with probability $1/M^k$

where \mathcal{A} outputs something with probability at least $\left(\frac{1-2^{-100}}{M}\right)^k$.

Proof. See Section B of the Supplementary Material. □

2.3 Ring-LWE based encryption scheme

In this work we use basic ideas from RLWE-based encryption, particularly in our public-key based construction from Section 3. We describe here a simplified version of the public-key encryption scheme from [38], which we refer to as LPR. The key generation, encryption and decryption procedures are defined as follows:

Gen(a) On input a public random $a \in \mathcal{R}_q$, first sample $s \leftarrow \mathcal{D}_{\text{sk}}$ and $e \leftarrow \mathcal{D}$. Output $\text{sk} = (s)$ and $\text{pk} = (a, b)$ where $b = a \cdot s + e$.

Enc $_{p,q}(\text{pk}, x)$: On input $\text{pk} \in \mathcal{R}_q^2$ and $x \in \mathcal{R}_p$, sample $w, e_0, e_1 \leftarrow \mathcal{D}$ and output (c_0, c_1) , where $c_1 = -a \cdot w + e_1$ and $c_0 = b \cdot w + e_0 + (q/p) \cdot x$.

Dec($\text{sk}, (c_0, c_1)$): Compute $x' = c_0 + s \cdot c_1 \bmod q$, and output $x = \lfloor x' \rfloor_p \bmod p$.⁴ Notice that this works if the total noise $e = s \cdot e_1 + e \cdot w + e_0$ is bounded by $p/2q$.

On top of these standard procedures, we also use an algorithm **KDMEnc** which produces an encryption of $x \cdot s$, where s is the secret key. As observed in [18] (and implicit in [20]), this can be done using only the public key by adding the message to the second component of an encryption of zero.

KDMEnc $_{p,q}(\text{pk}, x)$: Sample $w, e_0, e_1 \leftarrow \mathcal{D}$ and output (c_0, c_1) , where $c_1 = (q/p) \cdot x - a \cdot w + e_1$ and $c_0 = b \cdot w + e_0$.

⁴ Our protocols do not directly use the decryption algorithm, but our simulator in the proof of Theorem 2 does.

2.4 Oblivious Linear Function Evaluation

The functionality we implement in this work is oblivious linear evaluation (OLE), which, in a nutshell, consists of producing an additive sharing of a multiplication. A bit more precisely, $\mathcal{P}_{\text{Alice}}$ and \mathcal{P}_{Bob} have each one secret input $v \in \mathcal{R}_m$ and $u \in \mathcal{R}_m$, respectively, and their goal is to get additive random shares of the product $u \cdot v$. The formal description of the functionality appears in Fig. 1.

Note that our OLE functionality produces several OLE instances simultaneously, and we write $\alpha \star \beta$ to denote the component-wise product of the vectors α, β .

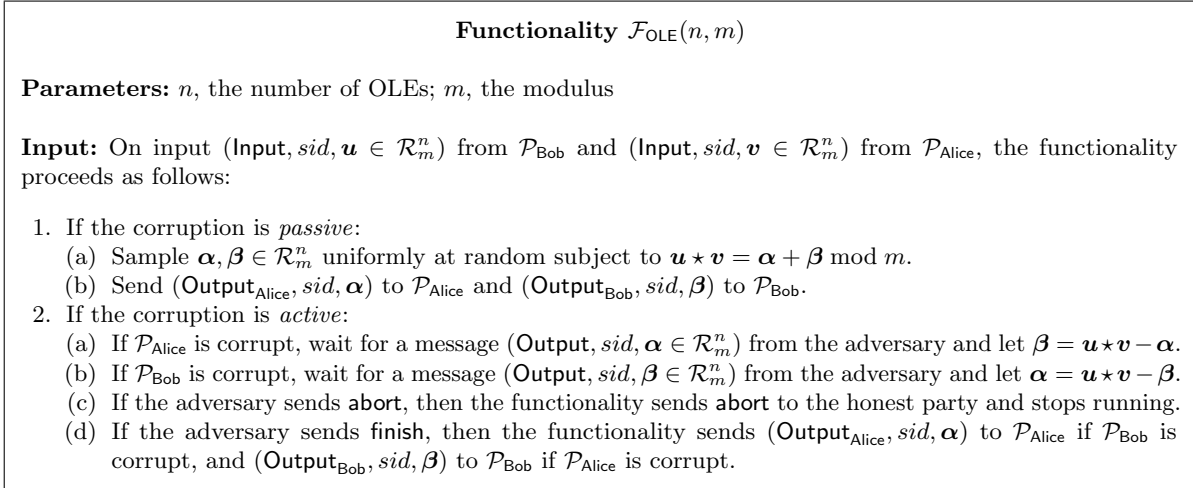


Fig. 1: Oblivious linear evaluation functionality

2.5 SIMD for Lattice-based Primitives

The OLE functionality above produces OLEs over the ring \mathcal{R}_m , however, in practice, we wish to produce OLEs over \mathbb{Z}_m . To do this, we exploit plaintext packing techniques used in homomorphic encryption [45] based on the Chinese remainder theorem. We choose $m = 1 \pmod{2N}$ such that the polynomial $X^N + 1$ splits completely into a product of linear factors modulo m . This implies that \mathcal{R}_m is isomorphic to N copies of \mathbb{Z}_m , so a single OLE over the ring \mathcal{R}_m can be directly used to obtain a batch of N OLEs over \mathbb{Z}_m . The isomorphism can be efficiently computed using fast Fourier transform techniques. Therefore, with a single call to our OLE functionality in Fig. 1, we can easily produce a batch of $N \cdot n$ OLEs over \mathbb{Z}_m .

2.6 Commitments and Zero-Knowledge Arguments

In this work, in order to achieve active security, we make extensive use of commitments schemes and zero knowledge arguments of knowledge.

Commitment Schemes. Consider the tuple $C = (\text{KG}, \text{Com}, \text{Open})$ with 1^κ as implicit input. KG is a PPT algorithm which generates a public parameter $\text{pk} \in \{0, 1\}^{\text{poly}(\kappa)}$. Com is a PPT algorithm which on input pk and a message x outputs c, r . Finally, Open is a deterministic poly-time algorithm which on input pk, x, c, r outputs a bit b .

For an algorithm \mathcal{A} let

$$\Pr \left[x \neq x' \mid \begin{array}{l} \text{pk} \leftarrow \text{KG}() \wedge (x, x', r, r', c) \leftarrow \mathcal{A}(\text{pk}) \wedge \\ \text{Open}(\text{pk}, x, c, r) = 1 \wedge \text{Open}(\text{pk}, x', c, r') = 1 \end{array} \right] \leq \text{negl}(\kappa)$$

then we say that C is computationally binding if \mathcal{A} is a PPT algorithm and C is statistically binding if \mathcal{A} is computationally unbounded.

Furthermore, for an algorithm \mathcal{A} if

$$\left| \Pr \left[i = \mathcal{A}(\text{pk}, c) \mid \begin{array}{l} \text{pk} \leftarrow \text{KG}() \wedge (x_0, x_1) \leftarrow \mathcal{A}(\text{pk}) \wedge \\ i \leftarrow \{0, 1\} \wedge (c, r) \leftarrow \text{Com}(\text{pk}, x_i) \end{array} \right] - \frac{1}{2} \right| \leq \text{negl}(\kappa)$$

then we say that the commitment scheme is statistically hiding if \mathcal{A} is computationally unbounded or computationally hiding if \mathcal{A} is a PPT algorithm.

A commitment scheme is additively homomorphic if for two commitments $(\text{com}_1, r_1) \leftarrow \text{Com}_{\text{pk}}(x_1), (\text{com}_2, r_2) \leftarrow \text{Com}_{\text{pk}}(x_2)$ as well as a constant c it holds that $\text{Open}(\text{pk}, c_1 + c_2, x_1 + x_2, r_1 + r_2) = 1$ as well as there exists a polynomial-time algorithm \mathcal{D} such that $\text{Open}(\text{pk}, \mathcal{D}(\text{pk}, c_1, x), x_1 + c, r_1) = 1$.

In this work we mainly use two different commitment schemes, namely the somewhat additively homomorphic commitment scheme of Baum et al. [8] (denoted as $C = (\text{KG}, \text{Com}, \text{Open})$) as well as a compressing statistically secure commitment scheme $C_{\text{aux}} = (\text{KG}_{\text{aux}}, \text{KG}_{\text{aux}}, \text{Open}_{\text{aux}})$. One can easily instantiate C_{aux} either using the Random Oracle or [25]. The scheme of [8] is only somewhat homomorphic, meaning that it only supports a limited number of addition of commitments due to the growth of r . More details on the used commitment scheme can be found in Section B of the Supplementary Material.

Zero-Knowledge Arguments of Knowledge (ZKA). Let \mathcal{R} be an NP relation. For $(pp, x, w) \in \mathcal{R}$ we call pp the public parameter, x the statement and w the witness. A Zero-Knowledge Proof of Knowledge for \mathcal{R} is an interactive protocol Π between a PPT prover \mathcal{P} and a PPT verifier \mathcal{V} with the following three properties:

Completeness: If \mathcal{P} with input pp, x, w and \mathcal{V} with input pp, w follow the protocol honestly, then \mathcal{V} outputs 0 only with negligible probability.

Soundness: If a PPT algorithm⁵ \mathcal{P}^* on input pp, x makes \mathcal{V} output 1 with polynomial probability p then there exists an algorithm \mathcal{E} which, given black-box access to \mathcal{P}^* outputs w' such that $(pp, x, w') \in \mathcal{R}$ in time $\text{poly}(p, \kappa)$ with at least constant probability $p' > 0$.

Honest-Verifier Zero-Knowledge: There exists a PPT algorithm \mathcal{S} called *the simulator* whose output distribution on input pp, x and interacting with a PPT algorithm \mathcal{V}^* is indistinguishable of a transcript of Π run by $\mathcal{P}, \mathcal{V}^*$.

The actual zero-knowledge arguments that are used with respect to the commitment scheme C can be found in Section B of the Supplementary Material.

Commitments and Zero-Knowledge Arguments. In this work, in order to achieve active security, we make extensive use of commitments schemes and zero knowledge arguments of knowledge. We refer the reader to Section 2.6 of the Supplementary Material for full definitions of these cryptographic notions. Here, we only introduce the basic notation.

Commitment Schemes. We consider an additively homomorphic statistically hiding commitment scheme, which we denote by a tuple $C = (\text{KG}, \text{Com}, \text{Open})$. In this work we mainly use two different commitment schemes, namely the somewhat additively homomorphic commitment scheme of Baum et al. [8] (denoted as $C = (\text{KG}, \text{Com}, \text{Open})$) as well as a compressing statistically secure commitment scheme $C_{\text{aux}} = (\text{KG}_{\text{aux}}, \text{KG}_{\text{aux}}, \text{Open}_{\text{aux}})$.

One can easily instantiate C_{aux} either using the Random Oracle or [25]. The scheme of [8] is only somewhat homomorphic, meaning that it only supports a limited number of addition of commitments due to the growth of r . More details on the used commitment scheme can be found in Section B of the Supplementary Material.

⁵ The term “argument of knowledge”, in contrast to “proof of knowledge”, relates to the setting in which soundness is only guaranteed against a polynomially bounded prover.

Zero-Knowledge Arguments of Knowledge (ZKA). Let \mathcal{R} be an NP relation. For $(pp, x, w) \in \mathcal{R}$ we call pp the public parameter, x the statement and w the witness. A Zero-Knowledge Proof of Knowledge for \mathcal{R} is an interactive protocol Π between a PPT prover \mathcal{P} and a PPT verifier \mathcal{V} satisfying completeness, soundness against bounded malicious provers and honest-verifier zero-knowledge. The actual zero-knowledge arguments that are used with respect to the commitment scheme C can be found in Section B of the Supplementary Material.

3 OLE from PKI Setup

In this section we present our first OLE construction, which is particularly simple and efficient. Furthermore, the only setup required is a correlated form of public key infrastructure for the LPR encryption scheme from Section 2.3 of the Supplementary Material in which $\mathcal{P}_{\text{Alice}}$ and \mathcal{P}_{Bob} have each a secret/public key pair for the LPR scheme, where the $a \in \mathcal{R}_q$ component of the public key is the same for both. This can be seen as a PKI setup in which the public keys are derived using some public randomness. The precise functionality \mathcal{F}_{PKI} is given in Fig. 2.

Our protocol, $\Pi_{\text{OLE-pk}}$, can be found in Fig. 3. The passively secure version $\Pi_{\text{OLE-pk}}^{\text{passive}}$ is obtained from the active one by removing the zero knowledge arguments, whose steps are framed in the description of the protocol. To provide a high level idea of our construction, we first recall the main techniques from the homomorphic secret-sharing scheme of Boyle et al. [18]. Suppose two parties have additive secret shares of a RLWE secret key $s \in \mathcal{R}_q$, and are also given secret shares modulo q of x , $x \cdot s$ and a public ciphertext $\mathbf{c}_y = (c_0, c_1) = \text{Enc}(\text{pk}, y)$, for some messages x, y . Boyle et al. observed that if each party *locally* decrypts \mathbf{c}_y using its shares, denoted $[x], [x \cdot s]$, we have:

$$[x] \cdot c_0 + [x \cdot s] \cdot c_1 = [x \cdot (c_0 + c_1 \cdot s)] \approx [(q/p) \cdot x \cdot y].$$

Applying the rounding operation from decryption on the above shares then gives *exact* additive shares of $x \cdot y$, provided the error is much smaller than q/p .

To create the initial shares of x and $x \cdot s$, it is enough to start with shares of s and ciphertexts encrypting $x, x \cdot s$, since each ciphertext can then be locally decrypted to obtain shares of these values. Boyle et al. also described a variant which removes the need for encryptions of $x \cdot s$, but at the cost of an additional setup assumption involving shares of s^2 .

Our OLE protocol from this section builds upon this blueprint, with some optimizations. First, we observe that in the two-party OLE setting, it is not necessary to give out $\text{Enc}(\text{pk}, x)$ to obtain shares of x , since one of the parties always knows x so they can simply choose these shares to be x and 0. (This is in contrast to the homomorphic secret-sharing setting, where the evaluating parties may be a set of servers who did not provide inputs.) Since we only do one multiplication, it's therefore enough to give out the two ciphertexts $\mathbf{c}_x = \text{Enc}(\text{pk}, x \cdot s)$ and $\mathbf{c}_y = \text{Enc}(\text{pk}, y)$, compared with four ciphertexts used in the HSS scheme from [18]. Since both ciphertexts can be easily generated from the public-key setup, this leads to a very simple protocol where each party (in parallel) sends a single message that is either an encryption of its input, or its input times s .

As an additional optimization, we show that the second ciphertext encrypting y can be defined at a smaller modulus p instead of q , since we only care about obtaining the result modulo $m < p$, which saves further on bandwidth.⁶

The protocol described above is passively secure, but an active adversary can break the security of this construction by sending incorrectly-formed ciphertexts. Due to our simple communication pattern this turns out to be the only potential source of attack, which we rule out by having the parties prove, in zero knowledge, that their ciphertexts are correctly formed.

⁶ This optimization is possible since we skip the ‘‘modulus lifting’’ step from [18], which is only needed when doing several repeated multiplications.

Functionality \mathcal{F}_{PKI}

The functionality runs with parties $\mathcal{P}_{\text{Alice}}$ and \mathcal{P}_{Bob} , as follows:

1. Sample $a \leftarrow \mathcal{R}_q$ and two key pairs $(s_{\text{Alice}}, (a, b_{\text{Alice}})) \leftarrow \text{LPR.Gen}(a)$ and $(s_{\text{Bob}}, (a, b_{\text{Bob}})) \leftarrow \text{LPR.Gen}(a)$.
2. Let $b = b_{\text{Alice}} + b_{\text{Bob}}$
3. Output $\text{pk} = (a, b)$ to both parties, as well as s_{Alice} to $\mathcal{P}_{\text{Alice}}$ and s_{Bob} to \mathcal{P}_{Bob} .

Fig. 2: PKI setup functionality

Protocol $\Pi_{\text{OLE-pk}}$

We use moduli $q > p > m$, where $m|p$ and $p|q$, and m is the final modulus of inputs and outputs.

1. *Setup.* The parties call \mathcal{F}_{PKI} , so that both parties obtain $\text{pk} = (a, b) \in \mathcal{R}_q^2$, while $\mathcal{P}_{\text{Alice}}$ gets $s_{\text{Alice}} \in \mathcal{R}_q$ and \mathcal{P}_{Bob} gets $s_{\text{Bob}} \in \mathcal{R}_q$.
2. *First Message.* On input $\mathbf{u} \in \mathcal{R}_m^n$ from \mathcal{P}_{Bob} :
 - (a) \mathcal{P}_{Bob} sends $(\mathbf{c}_0, \mathbf{c}_1) = \text{KDMEnc}_{p,q}(\text{pk}, \mathbf{u})$ to $\mathcal{P}_{\text{Alice}}$:
 - (b) The parties engage in a zero-knowledge argument for the relation $\mathcal{R}_{\text{Bob}}^{\text{pk}}$ with $\mathcal{P}_{\text{Alice}}$ as the verifier and \mathcal{P}_{Bob} as the prover. If this fails then the parties abort.
 - (c) $\mathcal{P}_{\text{Alice}}$ computes $\boldsymbol{\rho}_{\text{Alice}} = \lfloor s_{\text{Alice}} \cdot \mathbf{c}_1 \rfloor_p$ and \mathcal{P}_{Bob} computes $\boldsymbol{\rho}_{\text{Bob}} = \lfloor \mathbf{c}_0 + s_{\text{Bob}} \cdot \mathbf{c}_1 \rfloor_p$ (it should hold that $\boldsymbol{\rho}_{\text{Alice}} + \boldsymbol{\rho}_{\text{Bob}} = s \cdot \mathbf{u} \bmod p$)
3. *Second Message.* On input $\mathbf{v} \in \mathcal{R}_m^n$ from $\mathcal{P}_{\text{Alice}}$:
 - (a) $\mathcal{P}_{\text{Alice}}$ sends $(\mathbf{d}_0, \mathbf{d}_1) = \text{Enc}_{m,p}(\text{pk}, \mathbf{v})$ to \mathcal{P}_{Bob}
 - (b) The parties engage in a zero-knowledge argument for the relation $\mathcal{R}_{\text{Alice}}^{\text{pk}}$ with \mathcal{P}_{Bob} as the verifier and $\mathcal{P}_{\text{Alice}}$ as the prover. If this fails then the parties abort.
 - (c) $\mathcal{P}_{\text{Alice}}$ outputs $\boldsymbol{\alpha} = \lfloor \mathbf{d}_1 \star \boldsymbol{\rho}_{\text{Alice}} \rfloor_m$.
 - (d) \mathcal{P}_{Bob} outputs $\boldsymbol{\beta} = \lfloor \mathbf{d}_0 \star \mathbf{u} + \mathbf{d}_1 \star \boldsymbol{\rho}_{\text{Bob}} \rfloor_m$.

We should now have $\boldsymbol{\alpha} + \boldsymbol{\beta} = \mathbf{u} \star \mathbf{v} \bmod m$.

Fig. 3: Actively secure OLE protocol from a PKI setup. The passively secure version of the protocol is obtained by removing the framed steps.

3.1 Passive Security

We now proceed to the security proof of our protocol $\Pi_{\text{OLE-pk}}^{\text{passive}}$, which consists of protocol $\Pi_{\text{OLE-pk}}$ in Fig. 3 without the zero knowledge arguments framed in the protocol description.

Our proof requires that a random element of \mathcal{R}_q is invertible with high probability. As we will see, this technicality allows the simulator to “solve equations”, matching real and ideal views. For our choice of parameters this is always the case, and for this we make use of the following lemma.

Lemma 4. *Let $q = \prod_{i=1}^k p_i$, where each p_i is an ℓ -bit prime. If the polynomial $f(x) \in \mathbb{Z}_q[x]$ of degree N used to define \mathcal{R}_q splits completely mod p_i as $f(x) = \prod_{j=1}^N f_{ij}(x) \bmod p_i$, where each $f_{ij}(x)$ is linear, then the probability that a uniformly random element from \mathcal{R}_q is not invertible is upper bounded by $\frac{N \cdot k}{2^\ell}$.*

Proof. The conditions in the lemma imply that $R_q \cong (\mathbb{F}_{p_1})^N \times \cdots \times (\mathbb{F}_{p_k})^N$, and an element in R_q is not invertible if and only if one of its components in \mathbb{F}_{p_i} from the decomposition above is zero. This happens with probability $1/2^\ell$ for each component, so by the union bound, the probability that at least one of these components is zero is bounded by $N \cdot k \cdot 2^{-\ell}$. \square

Given the above, the probability that at least one component of a vector in \mathcal{R}_q^n is not invertible is upper bounded by $n \cdot N \cdot k \cdot 2^{-\ell}$. For all our parameter sets in Section 6, this quantity is below $2^{-\lambda}$ for $\lambda \approx 36$, which is good enough for our purposes since we need it only as an artefact for the proof and it does not lead to any concrete attack or leakage.⁷ We also use

⁷ This restriction can be easily overcome by modifying the definition of security against passive adversaries, allowing the adversary to choose its output. However, we prefer to stick to more standard security definitions.

invertibility to argue correctness of the protocol, as it is required for being able to use Lemma 1 in our protocols. If this probability is not good enough for a certain application, the parties could use a PRF to rerandomize their shares so that this lemma can be applied without invertibility. However, in order to keep our exposition simple we do not discuss such extension.

Another simple but useful lemma for our construction is the following.

Lemma 5. *Assume that $p|q$. Given $y \in \mathcal{R}_p$, the set of $x \in \mathcal{R}_q$ such that $y = \lfloor x \rfloor_p$ is given by $x = \left(\frac{q}{p}\right) \cdot y + e$ for $e \in \mathbb{Z} \cap (-q/2p, q/2p]$. In particular, the mapping $\mathcal{R}_q \rightarrow \mathcal{R}_p$ given by $x \mapsto \lfloor x \rfloor_p$ is a surjective regular mapping, meaning that every element in the codomain has an equal number of preimages.*

Finally, we have the following proposition, concerning correctness of our construction. It follows as a corollary of Proposition 2 by setting the soundness slack parameter τ to be 1, so we defer the proof to that section.

Proposition 1. *Assume that $3 \cdot 2^{\kappa+1} \cdot n \cdot (mN)^2 \cdot B_{\text{err}} \cdot B_{\text{sk}} \leq p \leq \frac{q}{3 \cdot 2^{\kappa+1} \cdot n \cdot N^2 \cdot B_{\text{err}} \cdot B_{\text{sk}}}$. Let $\mathbf{u}, \mathbf{v} \in \mathcal{R}_m^n$ be the inputs to Protocol $\Pi_{\text{OLE-pk}}^{\text{passive}}$, and let $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathcal{R}_m^n$ be the outputs. Then, with probability at least $1 - 2^{-\kappa}$, $\mathbf{u} \star \mathbf{v} = \boldsymbol{\alpha} + \boldsymbol{\beta}$.*

With these tools at hand we proceed with the main result from this section.

Theorem 1. *Assume that $3 \cdot 2^{\kappa+1} \cdot n \cdot (mN)^2 \cdot B_{\text{err}} \cdot B_{\text{sk}} \leq p \leq \frac{q}{3 \cdot 2^{\kappa+1} \cdot n \cdot N^2 \cdot B_{\text{err}} \cdot B_{\text{sk}}}$. Then protocol $\Pi_{\text{OLE-pk}}^{\text{passive}}$, which consists of protocol $\Pi_{\text{OLE-pk}}$ without the underlined steps, realizes functionality \mathcal{F}_{OLE} in the \mathcal{F}_{PKI} -hybrid model under the RLWE assumption.*

The proof of this Theorem is presented in Section C of the appendix.

3.2 Active Security

As we saw in the previous section, the correctness of our construction relies on the different terms involved having a certain bound: The input \mathbf{u} must be smaller than m , the noise terms used for the encryption have to be upper bounded by B_{err} , and the randomness \mathbf{w} and \mathbf{w}' used for the encryption must be less than B_{sk} . An actively corrupted party who chooses randomness outside these bounds can easily distinguish between the real and ideal executions.

To achieve active security, each party proves in zero-knowledge that the ciphertexts they send are correctly formed. We begin by analyzing the case of a corrupt \mathcal{P}_{Bob} . Consider the message from \mathcal{P}_{Bob} , which consists of a batch of ciphertexts

$$(\mathbf{c}_0, \mathbf{c}_1) = (b \cdot \mathbf{w} + \mathbf{e}_0, (q/p) \cdot \mathbf{u} - a \cdot \mathbf{w} + \mathbf{e}_1)$$

Rewriting this, \mathcal{P}_{Bob} has to prove knowledge of vectors (over \mathcal{R}_q) $\mathbf{w}, \mathbf{u}, \mathbf{e}_0, \mathbf{e}_1$ satisfying

$$\underbrace{\begin{pmatrix} b & 1 & 0 & 0 \\ -a & 0 & 1 & q/p \end{pmatrix}}_{\mathbf{A}} \cdot \underbrace{(\mathbf{w} \ \mathbf{e}_0 \ \mathbf{e}_1 \ \mathbf{u})^\top}_{\mathbf{S}} = \underbrace{\begin{pmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \end{pmatrix}}_{\mathbf{T}} \quad (3)$$

and $\|\mathbf{w}\|_\infty \leq B_{\text{sk}}$, $\|\mathbf{u}\|_\infty \leq m$, $\|\mathbf{e}_0\|_\infty \leq B_{\text{err}}$ and $\|\mathbf{e}_1\|_\infty \leq B_{\text{err}}$. This can be written in matrix form as follows

$$\mathcal{R}_{\text{Bob}}^{\text{pk}} = \left\{ (pp, u, w) = ((\mathcal{R}, q, n, \beta, \mathbf{A}), \mathbf{T}, \mathbf{S}) \mid (\mathbf{A}, \mathbf{S}, \mathbf{T}) \in \mathcal{R}_q^{2 \times 4} \times \mathcal{R}^{4 \times n} \times \mathcal{R}_q^{2 \times n} \right. \\ \left. \wedge \mathbf{A}\mathbf{S} = \mathbf{T} \wedge \|\mathbf{s}_i\|_\infty \leq \beta_i \right\}$$

where \mathbf{s}_i is the i -th row of \mathbf{S} and the bound vector is $\beta = (B_{\text{sk}}, B_{\text{err}}, B_{\text{err}}, B_{\text{msg}})$. Such type of statements can be proven efficiently using the amortized proof from [5], as we discuss more thoroughly in Section 5.

We can similarly define a relation for the message $(\mathbf{d}_0, \mathbf{d}_1)$ that $\mathcal{P}_{\text{Alice}}$ sends, and we call this relation $\mathcal{R}_{\text{Alice}}^{\text{pk}}$. We note however that in the proof of Theorem 1 we did not actually use any bound on the message \mathbf{v} , so we may exclude the bound $\|\mathbf{v}\|_\infty \leq m$ from this relation.

For the rest of this section we assume the existence of zero knowledge arguments of knowledge for the relations $\mathcal{R}_{\text{Alice}}^{\text{pk}}$ and $\mathcal{R}_{\text{Bob}}^{\text{pk}}$. As mentioned above, we show how to construct these in Section 5.1. Note that when proving knowledge of the relation $\mathcal{R}_{\text{Bob}}^{\text{pk}}$ or $\mathcal{R}_{\text{Alice}}^{\text{pk}}$ above, if the prover is malicious then our proof actually only guarantees that $\|\mathbf{s}_i\|_2 \leq \tau \cdot \beta_i$, where τ is the soundness slack parameter of the zero knowledge argument. We therefore need to choose our parameters with respect to the larger bounds, to ensure correctness of the protocol.

We begin with the following proposition, which shows that, under an appropriate choice of parameters, our protocol guarantees correctness.

Proposition 2. *Assume that $3 \cdot 2^{\kappa+1} \cdot n \cdot \tau \cdot (mN)^2 \cdot B_{\text{err}} \cdot B_{\text{sk}} \leq p \leq \frac{q}{3 \cdot 2^{\kappa+1} \cdot n \cdot \tau \cdot N^2 \cdot B_{\text{err}} \cdot B_{\text{sk}}}$. Let $\mathbf{u}, \mathbf{v} \in \mathcal{R}_m^n$ be the inputs to Protocol $\Pi_{\text{OLE-pk}}$, and let $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathcal{R}_m^n$ be the outputs. Assume that the relations $\mathcal{R}_{\text{Alice}}^{\text{pk}}$ and $\mathcal{R}_{\text{Bob}}^{\text{pk}}$ defined in Section 3.2 hold, but that at most one of them has slack parameter τ .⁸ Then, with probability at least $1 - 2^{-\kappa}$, $\mathbf{u} \star \mathbf{v} = \boldsymbol{\alpha} + \boldsymbol{\beta}$.*

Proof. Let us begin by writing the individual public keys as $b_{\text{Bob}} = a \cdot s_{\text{Bob}} + e_{\text{Bob}}$ and $b_{\text{Alice}} = a \cdot s_{\text{Alice}} + e_{\text{Alice}}$, and let $b = b_{\text{Bob}} + b_{\text{Alice}}$ and $s = s_{\text{Bob}} + s_{\text{Alice}}$, so $b = a \cdot s + e$ where $e = e_{\text{Alice}} + e_{\text{Bob}}$. We also write $(\mathbf{c}_0, \mathbf{c}_1) = (b \cdot \mathbf{w} + \mathbf{e}_0, (q/p) \cdot \mathbf{u} - a \cdot \mathbf{w} + \mathbf{e}_1)$ and $(\mathbf{d}_0, \mathbf{d}_1) = (b \cdot \mathbf{w}' + \mathbf{e}'_0 + (p/m) \cdot \mathbf{v}, -a \cdot \mathbf{w}' + \mathbf{e}'_1)$. It follows then that

$$\begin{aligned} \mathbf{c}_0 + s \cdot \mathbf{c}_1 &= b \cdot \mathbf{w} + \mathbf{e}_0 + (q/p) \cdot s \cdot \mathbf{u} - s \cdot a \cdot \mathbf{w} + s \cdot \mathbf{e}_1 \\ &= (b - s \cdot a) \cdot \mathbf{w} + \mathbf{e}_0 + (q/p) \cdot s \cdot \mathbf{u} + s \cdot \mathbf{e}_1 \\ &= (q/p) \cdot s \cdot \mathbf{u} + \underbrace{(\mathbf{e}_0 + e \cdot \mathbf{w} + s \cdot \mathbf{e}_1)}_e \pmod{q}. \end{aligned}$$

By recalling that $s = s_{\text{Alice}} + s_{\text{Bob}}$, we can write this as $\mathbf{c}_0 + s_{\text{Bob}} \cdot \mathbf{c}_1 = (q/p) \cdot s \cdot \mathbf{u} + (-s_{\text{Alice}} \cdot \mathbf{c}_1 + e) \pmod{q}$. Rounding this equation modulo p , we see that

$$\lfloor \mathbf{c}_0 + s_{\text{Bob}} \cdot \mathbf{c}_1 \rfloor_p = s \cdot \mathbf{u} + \lfloor -s_{\text{Alice}} \cdot \mathbf{c}_1 + e \rfloor_p \pmod{p}.$$

From $\mathcal{R}_{\text{Bob}}^{\text{pk}}$ we know that $\|\mathbf{w}\|_\infty \leq \tau \cdot B_{\text{sk}}$ and that $\|\mathbf{e}_i\|_\infty, \|\mathbf{e}'_i\|_\infty$ for $i = 1, 2$ are upper bounded by $\tau \cdot B_{\text{err}}$. Hence $\|e\|_\infty \leq 3\tau N B_{\text{err}} B_{\text{sk}} \leq \frac{q}{2^{\kappa+1} \cdot n \cdot p \cdot N}$. Also, since $s_{\text{Alice}} \cdot \mathbf{c}_1$ is close to uniform, it follows from Lemma 1 that $\lfloor -s_{\text{Alice}} \cdot \mathbf{c}_1 + e \rfloor_p = \lfloor -s_{\text{Alice}} \cdot \mathbf{c}_1 \rfloor_p = -\lfloor s_{\text{Alice}} \cdot \mathbf{c}_1 \rfloor_p$ with probability at least $1 - 2^{-\kappa}$. From this we conclude that $\boldsymbol{\rho}_{\text{Alice}} + \boldsymbol{\rho}_{\text{Bob}} = s \cdot \mathbf{u} \pmod{p}$ with high probability.

Now, we can perform a similar analysis for $(\mathbf{d}_0, \mathbf{d}_1)$ by writing

$$\begin{aligned} \mathbf{d}_0 + s \cdot \mathbf{d}_1 &= b \cdot \mathbf{w}' + \mathbf{e}'_0 + (p/m) \cdot \mathbf{v} - a \cdot s \cdot \mathbf{w}' + s \cdot \mathbf{e}'_1 \\ &= (b - s \cdot a) \cdot \mathbf{w}' + \mathbf{e}'_0 + (q/p) \cdot \mathbf{v} + s \cdot \mathbf{e}'_1 \\ &= (p/m) \cdot \mathbf{v} + \underbrace{(\mathbf{e}'_0 + e \cdot \mathbf{w}' + s \cdot \mathbf{e}'_1)}_{e'} \pmod{p}. \end{aligned}$$

We can multiply both sides by \mathbf{u} to get $\mathbf{d}_0 \star \mathbf{u} + (s \cdot \mathbf{u}) \star \mathbf{d}_1 = (p/m) \cdot (\mathbf{u} \star \mathbf{v}) + \mathbf{u} \star \mathbf{e}' \pmod{p}$, and recalling that $\mathbf{u} \cdot s = \boldsymbol{\rho}_{\text{Alice}} + \boldsymbol{\rho}_{\text{Bob}} \pmod{p}$ with high probability we obtain that

$$\mathbf{d}_0 \star \mathbf{u} + \boldsymbol{\rho}_{\text{Bob}} \star \mathbf{d}_1 = \left(\frac{p}{m}\right) \cdot (\mathbf{u} \star \mathbf{v}) + (-\boldsymbol{\rho}_{\text{Alice}} \star \mathbf{d}_1 + \mathbf{u} \star \mathbf{e}') \pmod{p}.$$

If $\mathcal{R}_{\text{Bob}}^{\text{pk}}$ holds with slack τ and $\mathcal{R}_{\text{Alice}}^{\text{pk}}$ with slack 1, then we know that that $\|\mathbf{u}\|_\infty \leq \tau \cdot m$, $\|\mathbf{w}'\|_\infty \leq B_{\text{sk}}$ and $\|\mathbf{e}'_i\|_\infty \leq B_{\text{err}}$ for $i = 1, 2$, so $\|\mathbf{e}'\|_\infty \leq 3B_{\text{err}}B_{\text{sk}}$. On the other hand, if $\mathcal{R}_{\text{Bob}}^{\text{pk}}$

⁸ That is, the bounds in one of the two relations have an extra factor of τ . This corresponds to what can be guaranteed for a corrupt party via the zero knowledge argument.

holds with slack 1 and $\mathcal{R}_{\text{Alice}}^{\text{pk}}$ with slack τ , then we know that $\|\mathbf{u}\|_\infty \leq m$, $\|\mathbf{w}'\|_\infty \leq \tau \cdot B_{\text{sk}}$ and $\|e'_i\|_\infty \leq \tau \cdot B_{\text{err}}$ for $i = 1, 2$, so $\|e'\|_\infty \leq 3\tau B_{\text{err}} B_{\text{sk}}$. Either case, it holds that $\|\mathbf{u} \star e'\|_\infty \leq 3m\tau N B_{\text{err}} B_{\text{sk}} \leq \frac{p}{2^{\kappa+1} \cdot n \cdot m \cdot N}$. Hence, rounding this equation modulo m , and using Lemma 1, we conclude that $\mathbf{u} \star \mathbf{v} = \boldsymbol{\alpha} + \boldsymbol{\beta} \pmod m$ with probability at least $1 - 2^{-\kappa}$, as required. \square

With this tool at hand, the security of the actively secure version of our protocol can be proven. The proof appears in Section C of the appendix.

Theorem 2. *Assume that $3 \cdot 2^{\kappa+1} \cdot n \cdot \tau \cdot (mN)^2 \cdot B_{\text{err}} \cdot B_{\text{sk}} \leq p \leq \frac{q}{3 \cdot 2^{\kappa+1} \cdot n \cdot \tau \cdot N^2 \cdot B_{\text{err}} \cdot B_{\text{sk}}}$. Protocol $\Pi_{\text{OLE-pk}}$ realizes functionality \mathcal{F}_{OLE} under the RLWE assumption.*

4 OLE from Correlated Setup

Ciphertexts in the public key version of the LPR cryptosystem consist of two ring elements. However, in the secret key variant, we can reduce this to one element, since the first element is uniformly random so can be compressed using, for example, a PRG. Given this, a natural way of shaving off a factor of two in the communication complexity of our protocol from Section 3 would be to use secret key encryption instead of public key.

In this section we present an OLE protocol that instantiates precisely this idea. The communication pattern is very similar to the one from Protocol $\Pi_{\text{OLE-pk}}$, in which there is a setup phase, then \mathcal{P}_{Bob} sends an encryption of his input \mathbf{u} to $\mathcal{P}_{\text{Alice}}$ (and proves in zero-knowledge its correctness for the actively secure version), and then $\mathcal{P}_{\text{Alice}}$ does the same. The challenge, here, is that now, as we are using secret-key encryption to obtain his ciphertext in the first message, there is no way for Bob to encrypt \mathbf{u} multiplied by the (combined) secret key.

To make this work, we replace the PKI setup functionality from the previous section with a more specialized setup, where \mathcal{P}_{Bob} gets $\sigma_{\text{Bob}} \in \mathcal{R}_q$ and $\mathcal{P}_{\text{Alice}}$ gets $\sigma_{\text{Alice}} \in \mathcal{R}_q$ such that $s_{\text{Alice}} \cdot s_{\text{Bob}} = \sigma_{\text{Alice}} + \sigma_{\text{Bob}} \pmod q$. This can be seen as an OLE itself, where the values being multiplied are small RLWE secret keys; under this interpretation, our protocol can be seen as a form of “OLE extension” protocol. The intuition for why this setup is useful, is that Bob’s secret-key ciphertext can now be distributively “decrypted” using the shares of $s_{\text{Alice}} \cdot s_{\text{Bob}}$, which (after rounding) leads to shares of $\mathbf{u} \cdot s_{\text{Alice}}$. In the second phase, these shares are then used to “decrypt” Alice’s ciphertext, giving shares of the product $\mathbf{u} \star \mathbf{v}$.

The setup functionality is described in Figure 4, where we present both the passive and active versions of the functionality, with the main difference being that in the active setting we must ensure that the corrupt party uses the same secret key for encrypting its input as the secret key distributed in the setup phase. Thus, in this case, when the corrupted party proves in zero knowledge the correctness of its encryption, it also proves that the secret key is the same as in the setup phase. This requires the setup functionality in the active case to output some extra information that allows us to “bind” the key from the setup with the key from the encryption sent, for which we use commitments. We discuss this in more detail when we look at active security in Section 4.2.

Our protocol is described in full detail in Fig. 5. As in Section 3, we present the full, actively secure version, but outline in a box those steps that are only necessary for active security.

4.1 Passive Security

The following proposition states that our construction satisfies correctness when the parties are honest, and follows from Proposition 4 in Section 4.2, which analyzes the case where the bounds satisfied by the values from one of the parties may not be sharp.

Proposition 3. *Assume that $2^{\kappa+1} \cdot n \cdot (mN)^2 \cdot B_{\text{err}} \leq p \leq \frac{q}{2^{\kappa+1} \cdot n \cdot N^2 \cdot B_{\text{err}} \cdot B_{\text{sk}}}$. Let $\mathbf{u}, \mathbf{v} \in \mathcal{R}_m^n$ be the inputs to Protocol $\Pi_{\text{OLE-sk}}$, and let $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathcal{R}_m^n$ be the outputs. Then, with probability at least $1 - 2^{-\kappa}$, $\mathbf{u} \star \mathbf{v} = \boldsymbol{\alpha} + \boldsymbol{\beta}$.*

Functionality $\mathcal{F}_{\text{setup}}$

This functionality interacts with two parties $\mathcal{P}_{\text{Alice}}, \mathcal{P}_{\text{Bob}}$ as well as an ideal adversary \mathcal{S} . Upon initialization, \mathcal{S} is allowed to passively corrupt either $\mathcal{P}_{\text{Alice}}$ or \mathcal{P}_{Bob} or none of them. The functionality is parameterized by the ring \mathcal{R} , the moduli q, p, m and the bound $0 < B_{\text{sk}} < q$. For the active case it is also parameterized by the homomorphic commitment scheme $C = (\text{KG}, \text{Com}, \text{Open})$ together with the parameters pk of C .

Sample: On input (Sample, sid) by both $\mathcal{P}_{\text{Alice}}, \mathcal{P}_{\text{Bob}}$ and if no such message has been sent before to the functionality, the functionality proceeds as follows:

- If the corruption is *passive*:
 1. Sample $s_{\text{Alice}}, s_{\text{Bob}} \leftarrow \mathcal{R}_q$ uniformly at random such that $\|s_{\text{Alice}}\|_{\infty}, \|s_{\text{Bob}}\|_{\infty} \leq B_s$.
 2. Sample $\sigma_{\text{Alice}}, \sigma_{\text{Bob}} \in \mathcal{R}_q$ subject to $s_{\text{Alice}} \cdot s_{\text{Bob}} = \sigma_{\text{Alice}} + \sigma_{\text{Bob}} \pmod q$.
 3. Send $(\text{Output}_{\text{Alice}}, sid, s_{\text{Alice}}, \sigma_{\text{Alice}})$ to $\mathcal{P}_{\text{Alice}}$ and $(\text{Output}_{\text{Bob}}, sid, s_{\text{Bob}}, \sigma_{\text{Bob}})$ to \mathcal{P}_{Bob} .
- If the corruption is *active*:
 1. Sample $s_{\text{Alice}}, s_{\mathcal{P}_{\text{Bob}}} \leftarrow \mathcal{R}_q$ uniformly at random such that $\|s_{\text{Alice}}\|_{\infty}, \|s_{\mathcal{P}_{\text{Bob}}}\|_{\infty} \leq B_s$.
 2. Compute $(\text{com}_{\text{Alice}}, r_{\text{Alice}}) \leftarrow \text{Com}_{pk}(s_{\text{Alice}})$ and $(\text{com}_{\text{Bob}}, r_{\text{Bob}}) \leftarrow \text{Com}_{pk}(s_{\text{Bob}})$.
 3. Sample $\sigma_{\text{Alice}}, \sigma_{\text{Bob}} \in \mathcal{R}_q$ subject to $s_{\text{Alice}} \cdot s_{\text{Bob}} = \sigma_{\text{Alice}} + \sigma_{\text{Bob}} \pmod q$.
 4. Send $(\text{Output}_{\text{Alice}}, sid, s_{\text{Alice}}, \sigma_{\text{Alice}}, r_{\text{Alice}}, \text{com}_{\text{Alice}}, \text{com}_{\text{Bob}})$ to $\mathcal{P}_{\text{Alice}}$ and $(\text{Output}_{\text{Bob}}, sid, s_{\text{Bob}}, \sigma_{\text{Bob}}, r_{\text{Bob}}, \text{com}_{\text{Alice}}, \text{com}_{\text{Bob}})$ to \mathcal{P}_{Bob} .

Fig. 4: Preprocessing for OLE with passive security

With this proposition, we proceed to the proof of security of our passively secure protocol.

Theorem 3. *Assume that $m^2 \cdot B_{\text{err}} \cdot 2^{\kappa+1} \cdot n \cdot N^2 \leq p \leq \frac{q}{2^{\kappa+1} \cdot n \cdot N^2 \cdot B_{\text{sk}} \cdot B_{\text{err}}}$. Then protocol $\Pi_{\text{OLE-sk}}^{\text{passive}}$, which consists of protocol $\Pi_{\text{OLE-sk}}$ without the underlined steps, realizes functionality \mathcal{F}_{OLE} in the $\mathcal{F}_{\text{setup}}$ -hybrid model under the RLWE assumption.*

The proof bears similarity with the proof of Theorem 1, and we defer it to Section C of the appendix.

4.2 Active Security

An active adversary in the protocol $\Pi_{\text{OLE-sk}}$ can cheat by sending incorrect messages. For example, a corrupt \mathcal{P}_{Bob} may send an incorrectly formed \mathbf{c} , and one can show that, in fact, by choosing \mathbf{c} appropriately a corrupt \mathcal{P}_{Bob} may learn some information about $\mathcal{P}_{\text{Alice}}$'s input \mathbf{v} . A similar attack can be carried out by a corrupt $\mathcal{P}_{\text{Alice}}$. Hence, to achieve active security, we must ensure that the message \mathbf{c} sent by \mathcal{P}_{Bob} and the message \mathbf{d} sent by $\mathcal{P}_{\text{Alice}}$ are computed honestly.

We implement zero knowledge arguments to show precisely these statements. \mathcal{P}_{Bob} proves that he knows \mathbf{u}, \mathbf{e} and s_{Bob} of the appropriate sizes such that $\mathbf{c} = \left(\frac{q}{p}\right) \cdot \mathbf{u} + (\mathbf{a} \cdot s_{\text{Bob}} + \mathbf{e}_{\text{Bob}}) \pmod q$, and $\mathcal{P}_{\text{Alice}}$ proceeds similarly.

An additional technicality, however, is that s_{Bob} (and respectively s_{Alice}) has to be exactly the same value that was distributed during the setup phase. To enforce this, we consider a modified setup functionality for the actively secure setting that, on top of distributing $s_{\text{Bob}} \cdot s_{\text{Alice}} = \sigma_{\text{Bob}} + \sigma_{\text{Alice}}$, also distributes commitments to s_{Bob} and s_{Alice} that can be used in the relation of the zero knowledge argument (Fig. 4).

Given that the protocol is essentially symmetric with respect to the roles of $\mathcal{P}_{\text{Alice}}$ and \mathcal{P}_{Bob} , from now on we focus on discussing the case of a corrupt \mathcal{P}_{Bob} . A similar argument applies for the case of corrupt $\mathcal{P}_{\text{Alice}}$. The message \mathbf{c} that \mathcal{P}_{Bob} sends is formed by adding n RLWE samples to $\left(\frac{q}{p}\right) \cdot \mathbf{u}$, which is a scaled version of its input \mathbf{u} . Furthermore, the RLWE samples must be generated using the secret s_{Bob} distributed in the setup phase. As a result, the relation that \mathcal{P}_{Bob} will prove is

$$\mathcal{R}_{\text{Bob}}^{\text{sk}}(\tau) = \left\{ \begin{array}{l} (pp, u, w) = \\ \left((\mathcal{R}, q, p, m, \beta, \text{pk}, \mathbf{a}, \text{com}_{\text{Bob}}), \right. \\ \left. \mathbf{c}, (\mathbf{u}, \mathbf{e}, s, r) \right) \end{array} \middle| \begin{array}{l} \mathbf{c} = \left(\frac{q}{p}\right) \cdot \mathbf{u} + \mathbf{a} \cdot s + \mathbf{e} \pmod{q} \wedge \\ \|\mathbf{u}\|_{\infty} \leq \tau \cdot \beta_1 \wedge \|\mathbf{e}\|_{\infty} \leq \tau \cdot \beta_2 \wedge \\ \text{Open}_{\text{pk}}(\text{com}_{\text{Bob}}, s, r) = 1 \end{array} \right\}$$

Protocol $\Pi_{\text{OLE-sk}}$

We use moduli $q > p > m$, where m is the final modulus of inputs and outputs. We assume that m divides p and that p divides q .

1. *Setup phase.*

- (a) *Passive case.* $\mathcal{P}_{\text{Alice}}, \mathcal{P}_{\text{Bob}}$ each send **(Sample, sid)** to $\mathcal{F}_{\text{setup}}$. $\mathcal{P}_{\text{Alice}}$ obtains $s_{\text{Alice}}, \sigma_{\text{Alice}}$ while \mathcal{P}_{Bob} obtains $s_{\text{Bob}}, \sigma_{\text{Bob}}$.

Active case. $\mathcal{P}_{\text{Alice}}, \mathcal{P}_{\text{Bob}}$ each send **(Sample, sid)** to $\mathcal{F}_{\text{setup}}$. $\mathcal{P}_{\text{Alice}}$ obtains $s_{\text{Alice}}, \sigma_{\text{Alice}}, r_{\text{Alice}}, c_{\text{Alice}}, c_{\text{Bob}}$ and \mathcal{P}_{Bob} obtains $s_{\text{Bob}}, \sigma_{\text{Bob}}, r_{\text{Bob}}, c_{\text{Alice}}, c_{\text{Bob}}$.

- (b) The parties sample two public random values $\mathbf{a}, \mathbf{a}' \in \mathcal{R}_q^n$ ^a.

2. *First Message.* On input $\mathbf{u} \in \mathcal{R}_m^n$ from \mathcal{P}_{Bob} :

- (a) \mathcal{P}_{Bob} samples a noise vector $\mathbf{e}_{\text{Bob}} \leftarrow \mathcal{D}^n$ and sends $\mathbf{c} = \left(\frac{q}{p}\right) \cdot \mathbf{u} + (\mathbf{a} \cdot s_{\text{Bob}} + \mathbf{e}_{\text{Bob}}) \pmod q$ to $\mathcal{P}_{\text{Alice}}$.

- (b) The parties engage in a zero-knowledge argument for the relation $\mathcal{R}_{\text{Bob}}^{\text{sk}}$ with $\mathcal{P}_{\text{Alice}}$ as the verifier and \mathcal{P}_{Bob} as the prover with witness $(\mathbf{u}, \mathbf{e}_{\text{Bob}}, s_{\text{Bob}}, r_{\text{Bob}})$. If this fails then the parties abort.

- (c) $\mathcal{P}_{\text{Alice}}$ computes $\boldsymbol{\rho}_{\text{Alice}} = \lfloor s_{\text{Alice}} \cdot \mathbf{c} - \mathbf{a} \cdot \sigma_{\text{Alice}} \rfloor_p$.

- (d) \mathcal{P}_{Bob} computes $\boldsymbol{\rho}_{\text{Bob}} = -\lfloor \mathbf{a} \cdot \sigma_{\text{Bob}} \rfloor_p$. It should now hold that $\mathbf{u} \cdot s_{\text{Alice}} = \boldsymbol{\rho}_{\text{Alice}} + \boldsymbol{\rho}_{\text{Bob}} \pmod p$.

3. *Second Message.* On input $\mathbf{v} \in \mathcal{R}_m^n$ from $\mathcal{P}_{\text{Alice}}$:

- (a) $\mathcal{P}_{\text{Alice}}$ samples a noise vector $\mathbf{e}_{\text{Alice}} \leftarrow \mathcal{D}^n$ and sends $\mathbf{d} = \left(\frac{p}{m}\right) \cdot \mathbf{v} + (\mathbf{a}' \cdot s_{\text{Alice}} + \mathbf{e}_{\text{Alice}}) \pmod p$ to \mathcal{P}_{Bob} .

- (b) The parties engage in a zero-knowledge argument for the relation $\mathcal{R}_{\text{Alice}}^{\text{sk}}$ with \mathcal{P}_{Bob} as the verifier and $\mathcal{P}_{\text{Alice}}$ as the prover, with witness $(\mathbf{v}, \mathbf{e}_{\text{Alice}}, s_{\text{Alice}}, r_{\text{Alice}})$. If this fails then the parties abort.

- (c) \mathcal{P}_{Bob} outputs, $\boldsymbol{\beta} = \lfloor \mathbf{u} \star \mathbf{d} - \mathbf{a}' \star \boldsymbol{\rho}_{\text{Bob}} \rfloor_m \pmod m$.

- (d) $\mathcal{P}_{\text{Alice}}$ outputs, $\boldsymbol{\alpha} = -\lfloor \mathbf{a}' \star \boldsymbol{\rho}_{\text{Alice}} \rfloor_m \pmod m$. It should hold that $\mathbf{u} \star \mathbf{v} = \boldsymbol{\alpha} + \boldsymbol{\beta} \pmod m$.

^a In practice this can be done by using a PRF with some pre-shared key. In our proofs we use a random oracle that can be programmed by the simulator.

Fig. 5: Actively secure OLE protocol based on RLWE. The passively secure version of the protocol is obtained by removing the framed steps.

and $\mathcal{R}_{\text{Alice}}^{\text{sk}}$ can be defined similarly.⁹ Here in the honest case \mathcal{P}_{Bob} starts with $\mathcal{R}_{\text{Bob}}^{\text{sk}}$, but the guarantee given by the zero-knowledge argument will be for a substantially larger factor τ (see Section 5). The relation essentially shows that the message that \mathcal{P}_{Bob} sends is well formed, and furthermore, that the s_{Bob} used for constructing this message is exactly the same as the one provided in the setup phase.

For the purpose of this section we assume the existence of zero knowledge arguments for the relations $\mathcal{R}_{\text{Alice}}^{\text{sk}}$ and $\mathcal{R}_{\text{Bob}}^{\text{sk}}$. We develop such results in Section 5.

Now, to proceed with the security proof of our protocol, we first present the following proposition, which states that our construction satisfies correctness even when the bound on the parameters may have some slack.

Proposition 4. *Assume that $2^{\kappa+1} \cdot n \cdot \tau \cdot (mN)^2 \cdot B_{\text{err}} \leq p \leq \frac{q}{2^{\kappa+1} \cdot n \cdot \tau \cdot N^2 \cdot B_{\text{err}} \cdot B_{\text{sk}}}$. Let $\mathbf{u}, \mathbf{v} \in \mathcal{R}_m^n$ be the inputs to Protocol $\Pi_{\text{OLE-sk}}$, and let $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathcal{R}_m^n$ be the outputs. Assume that the relations $\mathcal{R}_{\text{Alice}}^{\text{sk}}$ and $\mathcal{R}_{\text{Bob}}^{\text{sk}}$ hold, but that at most one of them has slack parameter τ . Then, with probability at least $1 - 2^{-\kappa}$, $\mathbf{u} \star \mathbf{v} = \boldsymbol{\alpha} + \boldsymbol{\beta}$.*

Proof. We begin by noticing that $s_{\text{Alice}} \cdot \mathbf{c} = \left(\frac{q}{p}\right) \cdot s_{\text{Alice}} \cdot \mathbf{u} + (\mathbf{a} \cdot s_{\text{Alice}} \cdot s_{\text{Bob}} + s_{\text{Alice}} \cdot \mathbf{e}_{\text{Bob}}) \pmod q$, so, taking into account that in the real world $\mathcal{F}_{\text{setup}}$ guarantees that $s_{\text{Alice}} \cdot s_{\text{Bob}} = \sigma_{\text{Alice}} + \sigma_{\text{Bob}}$, we have that $\mathbf{a} \cdot \sigma_{\text{Bob}} + s_{\text{Alice}} \cdot \mathbf{e}_{\text{Bob}} = (s_{\text{Alice}} \cdot \mathbf{c} - \mathbf{a} \cdot \sigma_{\text{Alice}}) - \left(\frac{q}{p}\right) \cdot s_{\text{Alice}} \cdot \mathbf{u} \pmod q$. Rounding this equation mod p , and noticing that $\left(\frac{q}{p}\right) \cdot s_{\text{Alice}} \cdot \mathbf{u}$ rounds exactly to the integer $s_{\text{Alice}} \cdot \mathbf{u}$, we obtain $\lfloor \mathbf{a} \cdot \sigma_{\text{Bob}} + s_{\text{Alice}} \cdot \mathbf{e}_{\text{Bob}} \rfloor_p = \lfloor s_{\text{Alice}} \cdot \mathbf{c} - \mathbf{a} \cdot \sigma_{\text{Alice}} \rfloor_p - s_{\text{Alice}} \cdot \mathbf{u} \pmod p$.

Now, if $\mathcal{R}_{\text{Alice}}^{\text{sk}}$ has slack τ and $\mathcal{R}_{\text{Bob}}^{\text{sk}}$ has slack 1, it follows that $\|s_{\text{Alice}}\|_{\infty} \leq \tau \cdot B_{\text{sk}}$ and $\|\mathbf{e}_{\text{Bob}}\|_{\infty} \leq B_{\text{err}}$, which implies that $\|s_{\text{Alice}} \cdot \mathbf{e}_{\text{Bob}}\|_{\infty} \leq N\tau B_{\text{err}} B_{\text{sk}}$. On the other hand, if $\mathcal{R}_{\text{Alice}}^{\text{sk}}$

⁹ As in public-key protocol from Section 3.2, $\mathcal{P}_{\text{Alice}}$ does not need to prove the bound on her input \mathbf{v} .

has slack 1 and $\mathcal{R}_{\text{Bob}}^{\text{sk}}$ has slack τ , it follows that $\|s_{\text{Alice}}\|_{\infty} \leq B_{\text{sk}}$ and $\|e_{\text{Bob}}\|_{\infty} \leq \tau \cdot B_{\text{err}}$, which also implies that $\|s_{\text{Alice}} \cdot e_{\text{Bob}}\|_{\infty} \leq N\tau B_{\text{err}} B_{\text{sk}}$. Since $\mathbf{a} \cdot \sigma_{\text{Bob}}$ is uniformly random, and given that $\|s_{\text{Alice}} \cdot e_{\text{Bob}}\|_{\infty} \leq \tau N B_{\text{err}} B_{\text{sk}} \leq \frac{q}{2^{\kappa+1} n p N}$, Lemma 1 implies that $[\mathbf{a} \cdot \sigma_{\text{Bob}} + s_{\text{Alice}} \cdot e_{\text{Bob}}]_p = [\mathbf{a} \cdot \sigma_{\text{Bob}}]_p$ with probability at least $1 - 2^{-\kappa}$.

For the second message, we see that in the real world $\mathcal{P}_{\text{Alice}}$ sends $\mathbf{d} = \left(\frac{p}{m}\right) \cdot \mathbf{v} + (\mathbf{a}' \cdot s_{\text{Alice}} + e_{\text{Alice}}) \bmod m$ to \mathcal{P}_{Bob} , where \mathbf{v} is $\mathcal{P}_{\text{Alice}}$'s input. Now, if $\mathcal{R}_{\text{Alice}}^{\text{sk}}$ has slack τ and $\mathcal{R}_{\text{Bob}}^{\text{sk}}$ has slack 1, it follows that $\|e_{\text{Alice}}\|_{\infty} \leq \tau \cdot B_{\text{err}}$ and $\|\mathbf{u}\|_{\infty} \leq m$, which implies that $\|\mathbf{u} \star e_{\text{Alice}}\|_{\infty} \leq \tau m N B_{\text{err}}$. On the other hand, if $\mathcal{R}_{\text{Alice}}^{\text{sk}}$ has slack 1 and $\mathcal{R}_{\text{Bob}}^{\text{sk}}$ has slack τ , it follows that $\|e_{\text{Alice}}\|_{\infty} \leq B_{\text{err}}$ and $\|\mathbf{u}\|_{\infty} \leq \tau \cdot m$, which implies too that $\|\mathbf{u} \star e_{\text{Alice}}\|_{\infty} \leq \tau m N B_{\text{err}}$. Using Lemma 1, we can conclude that $\mathbf{u} \star \mathbf{v} = \boldsymbol{\alpha} + \boldsymbol{\beta}$, except with probability at most $2^{-\kappa}$. \square

Given this, we can prove the security of our actively secure OLE protocol, as stated in the following theorem. The proof appears in Section C of the appendix.

Theorem 4. *Assume that $2^{\kappa+1} \cdot n \cdot \tau \cdot (mN)^2 \cdot B_{\text{err}} \leq p \leq \frac{q}{2^{\kappa+1} \cdot n \cdot \tau \cdot N^2 \cdot B_{\text{err}} \cdot B_{\text{sk}}}$. Then protocol $\Pi_{\text{OLE-sk}}$ realizes functionality \mathcal{F}_{OLE} in the $\mathcal{F}_{\text{setup}}$ -hybrid model under the RLWE assumption.*

5 Zero-Knowledge Arguments

In this section we describe in detail the zero-knowledge arguments that are necessary to implement the actively secure versions of our OLE protocols. Both of the arguments are amortized, meaning that i) they prove n statements in parallel and ii) for large enough n the communication from the prover \mathcal{P} to the verifier \mathcal{V} becomes strictly sublinear in n . While the first argument follows directly from [5], the second one is a non-trivial modification of this approach. It implies an amortized argument for proving well-formedness of *secret-key* (R)LWE ciphertexts, which to the best of our knowledge, has not been done previously and may be of independent interest.

5.1 Argument for Public OLE

Recall the relations $\mathcal{R}_{\text{Alice}}^{\text{pk}}$ and $\mathcal{R}_{\text{Bob}}^{\text{pk}}$ considered in Section 3.2. We present in this section a zero-knowledge argument of knowledge for these relations. We begin by noticing that these relations can be expressed in matrix-form as $\mathbf{T} = \mathbf{A}\mathbf{S}$ where \mathbf{A} is derived from the public parameters, \mathbf{T} is the target and \mathbf{S} is the secret we want to prove knowledge of. More concretely

$$\underbrace{\begin{pmatrix} b & 1 & 0 & 0 \\ -a & 0 & 1 & q/p \end{pmatrix}}_{\mathbf{A}} \cdot \underbrace{(\mathbf{f} \ e_0 \ e_1 \ \mathbf{x})^{\top}}_{\mathbf{S}} = \underbrace{(\mathbf{c}_0, \mathbf{c}_1)}_{\mathbf{T}}$$

where we show that \mathbf{S} has a bound of $B_{\text{sk}}, B_{\text{err}}, B_{\text{err}}, B_{\text{msg}}$ for each row, respectively.

We consider one single relation \mathcal{R}_{pk} , defined as in $\mathcal{R}_{\text{Bob}}^{\text{pk}}$ (or $\mathcal{R}_{\text{Alice}}^{\text{pk}}$), but using the 2-norm instead of the ∞ -norm.¹⁰ To prove this relation, we use the interactive argument between \mathcal{P} and \mathcal{V} that is outlined in Fig. 6. One can easily show the following theorem, whose proof we defer to Section B.5 of the Supplementary Material.

Theorem 5. *Assume that $\forall i \in [n] : \|\mathbf{f}[i]\|_2 \leq T_{\text{sk}}, \|e_0[i]\|_2, \|e_1[i]\|_2 \leq T_{\text{err}}, \|\mathbf{x}[i]\|_2 \leq T_{\text{msg}}$. Let $\ell \geq \kappa + 2$, $M > 1$ and $\sigma_x \geq 12/(\ln M)\sqrt{n\ell T_x}$. Furthermore, let C_{aux} be a statistically hiding and computationally binding commitment scheme. Then the aforementioned protocol is a zero-knowledge argument of knowledge for \mathcal{R}_{pk} with $(B_{\text{sk}}, B_{\text{err}}, B_{\text{msg}}) = (\sqrt{8N}\sigma_{\text{sk}}, \sqrt{8N}\sigma_{\text{err}}, \sqrt{8N}\sigma_{\text{msg}})$ that is complete with probability $1/M^4$, computationally sound and statistically zero-knowledge.*

¹⁰ The choice of norm is irrelevant for the purpose of the existence of a proof, since bounds in one norm can be translated to the other.

Zero Knowledge Argument for \mathcal{R}_{pk}

The following is a zero knowledge argument for the relation \mathcal{R}_{pk} . The public instance is $pp = (\mathcal{R}, q, n, \beta, \mathbf{A})$ and furthermore let $C_{\text{aux}} = (\text{KG}_{\text{aux}}, \text{Com}_{\text{aux}}, \text{Open}_{\text{aux}})$ be a statistically hiding auxiliary commitment scheme with public key pk_{aux} known to both \mathcal{P}, \mathcal{V} .

1. \mathcal{P} samples $\mathbf{y}_1 \leftarrow \mathcal{D}_{\sigma_{\text{sk}}}^\ell, \mathbf{y}_2, \mathbf{y}_3 \leftarrow \mathcal{D}_{\sigma_{\text{err}}}^\ell, \mathbf{y}_4 \leftarrow \mathcal{D}_{\sigma_{\text{msg}}}^\ell$, defines $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_4)$ as well as $\mathbf{W} \leftarrow \mathbf{A}\mathbf{Y}^\top$, generates $(\text{com}_{\mathbf{W}}, r_{\mathbf{W}}) \leftarrow \text{Com}_{\text{aux}}(\text{pk}_{\text{aux}}, \mathbf{W})$ and sends $\text{com}_{\mathbf{W}}$ to \mathcal{V} .
2. \mathcal{V} samples $\mathbf{C} \leftarrow \{0, 1\}^{\ell \times n}$ uniformly at random and sends it \mathcal{P} .
3. \mathcal{P} computes $(\mathbf{z}_1, \dots, \mathbf{z}_4) = \mathbf{Z} \leftarrow \mathbf{C}(\mathbf{f}, \mathbf{e}_0, \mathbf{e}_1, \mathbf{x}) + \mathbf{Y}$. Continue with probability

$$\min \left(1, \frac{1}{M^4} \frac{\mathcal{D}_{\sigma_{\text{sk}}}^\ell(\mathbf{z}_1)}{\mathcal{D}_{\mathbf{C}\mathbf{f}, \sigma_{\text{sk}}}^\ell(\mathbf{z}_1)} \frac{\mathcal{D}_{\sigma_{\text{err}}}^\ell(\mathbf{z}_2)}{\mathcal{D}_{\mathbf{C}\mathbf{e}_0, \sigma_{\text{err}}}^\ell(\mathbf{z}_2)} \frac{\mathcal{D}_{\sigma_{\text{err}}}^\ell(\mathbf{z}_3)}{\mathcal{D}_{\mathbf{C}\mathbf{e}_1, \sigma_{\text{err}}}^\ell(\mathbf{z}_3)} \frac{\mathcal{D}_{\sigma_{\text{msg}}}^\ell(\mathbf{z}_4)}{\mathcal{D}_{\mathbf{C}\mathbf{x}, \sigma_{\text{msg}}}^\ell(\mathbf{z}_4)} \right)$$

and send \mathbf{Z} as well as $r_{\mathbf{W}}$ to \mathcal{V} , otherwise abort.

4. \mathcal{V} accepts iff
 - (a) $\text{Open}_{\text{aux}}(\text{pk}_{\text{aux}}, \text{com}_{\mathbf{W}}, \mathbf{A}\mathbf{Z}^\top - \mathbf{T}\mathbf{C}^\top, r_{\mathbf{W}}) = 1$
 - (b) $\|\mathbf{z}_1\|_2 \leq B_{\text{sk}}/2, \|\mathbf{z}_2\|_2, \|\mathbf{z}_3\|_2 \leq B_{\text{err}}/2, \|\mathbf{z}_4\|_2 \leq B_{\text{msg}}/2$

Fig. 6: Zero knowledge argument for \mathcal{R}_{pk}

5.2 Argument for Private OLE

For the OLE in Fig. 5 with preprocessing the relation which each party has to prove looks different from the aforementioned \mathcal{R}_{pk} and we cannot apply the same argument as above. In more detail, one of the two relations (the other follows along the same lines) that we want to prove is

$$\mathcal{R}_{\text{Bob}}^{\text{sk}} = \left\{ \left(\begin{array}{l} (pp, u, w) = \\ (\mathcal{R}, q, p, m, \beta, \text{pk}, \mathbf{a}, \text{com}), \\ \mathbf{c}, (\mathbf{u}, \mathbf{e}, s, \mathbf{r}) \end{array} \right) \left| \begin{array}{l} (\mathbf{a}, \mathbf{c}, \mathbf{u}, \mathbf{e}) \in \mathcal{R}_q^n \times \mathcal{R}_q^n \times \mathcal{R}^n \times \mathcal{R}^n \wedge \\ \mathbf{c} = \begin{pmatrix} q \\ p \end{pmatrix} \cdot \mathbf{u} + \mathbf{a} \cdot s + \mathbf{e} \text{ mod } q \wedge \\ \|\mathbf{u}\|_2 \leq B_{\mathbf{u}} \wedge \|\mathbf{e}\|_2 \leq B_{\text{err}} \wedge \\ \text{Open}_{\text{pk}}(\text{com}, s, \mathbf{r}) = 1 \end{array} \right. \right\}$$

where $\beta = (B_{\mathbf{u}}, B_{\text{err}})$.¹¹

Towards constructing an interactive argument, we first observe that the equation to be proven can be alternatively written as

$$-\mathbf{a} \cdot s = (q/p) \cdot \mathbf{u} + \mathbf{e} - \mathbf{c}$$

which means that all such statements are in fact linear in s . At first glance it might seem plausible to apply a standard amortization technique such as in Fig. 6 but that is not possible: amortization techniques require that the instances are linear in a *public* value, whereas in our case we need to exploit linearity in the *secret*. In Fig. 7 we present a zero-knowledge argument for the specific relation whose overhead is $O(\kappa)$, i.e. the number of additional vectors in \mathcal{R}_q^n only depends on the statistical security parameter κ .

Theorem 6. *Assume that $\forall i \in [n] : \|\mathbf{v}[i]\|_2 \leq T_{\mathbf{u}}, \|\mathbf{e}[i]\|_2 \leq T_{\mathbf{e}}$. Let $\ell \geq \kappa + 3$, $M > 1$ and $\sigma_x \geq 12/(\ln M)\sqrt{n\ell}T_x$. Furthermore, let C be a statistically hiding and computationally binding commitment scheme. Then the aforementioned protocol is a zero-knowledge argument of knowledge for $\mathcal{R}_{\text{Bob}}^{\text{sk}}$ with $(B_{\mathbf{u}}, B_{\text{err}}) = (\sqrt{8N}\sigma_{\mathbf{u}}, \sqrt{8N}\sigma_{\mathbf{e}})$ that is complete with probability $1/M^4$, computationally sound and statistically zero-knowledge.*

A full proof of this theorem is presented in Section B.5 of the Supplementary Material. In terms of complexity of the aforementioned protocol, we start by sending $\ell = O(\kappa)$ commitments com_j of $O(1)$ \mathcal{R}_q elements, then sample $\ell \cdot n$ bits and then send two \mathcal{R}_q -vectors $\boldsymbol{\epsilon}, \boldsymbol{\mu}$ which are of

¹¹ Observe that, again, we have switched to the 2-norm.

Zero Knowledge Argument for \mathcal{R}_{Bob}

The following is a zero knowledge argument for the relation \mathcal{R}_{Bob} . The public instance is $pp = (\mathcal{R}, q, p, m, \beta, \text{pk}, \mathbf{a}, \text{com})$ where we use the same commitment scheme $C = (\text{KG}, \text{Com}, \text{Open})$ together with the parameters pk from the setup phase.

1. \mathcal{P} runs an interactive proof for \mathcal{R}_{PoK} from Appendix B.3 with \mathcal{V} to show knowledge of an opening of com using the opening (s, \mathbf{r}) . Here we use the same ℓ for \mathcal{R}_{PoK} .
2. \mathcal{P} samples $\mathbf{f} \leftarrow \mathcal{D}_{\sigma_e}^\ell, \mathbf{v} \leftarrow \mathcal{D}_{\sigma_u}^\ell$. Then for $j \in [\ell]$ \mathcal{P} generates commitments $(\text{c}\tilde{\text{om}}_j, \tilde{\mathbf{r}}_j) \leftarrow \text{Com}_{\text{pk}}((q/p) \cdot \mathbf{v}[j] + \mathbf{f}[j])$ and sends $\{\text{c}\tilde{\text{om}}_j\}_{j \in [\ell]}$ to \mathcal{V} .
3. \mathcal{V} samples $\mathbf{C} \leftarrow \{0, 1\}^{\ell \times n}$ uniformly at random and sends it to \mathcal{P} . Both \mathcal{V}, \mathcal{P} set $\boldsymbol{\alpha} \leftarrow \mathbf{C}\mathbf{a}$ and $\boldsymbol{\gamma} \leftarrow \mathbf{C}\mathbf{c}$.
4. \mathcal{P} locally computes $\boldsymbol{\epsilon} \leftarrow \mathbf{f} + \mathbf{C}\mathbf{e}$ and $\boldsymbol{\mu} \leftarrow \mathbf{v} + \mathbf{C}\mathbf{u}$. With probability

$$\min \left(1, \frac{1}{M^2} \frac{\mathcal{D}_{\sigma_e}^\ell(\boldsymbol{\epsilon})}{\mathcal{D}_{\mathbf{C}\mathbf{e}, \sigma_e}^\ell(\boldsymbol{\epsilon})} \frac{\mathcal{D}_{\sigma_u}^\ell(\boldsymbol{\mu})}{\mathcal{D}_{\mathbf{C}\mathbf{u}, \sigma_u}^\ell(\boldsymbol{\mu})} \right)$$

\mathcal{P} outputs $(\boldsymbol{\epsilon}, \boldsymbol{\mu})$ to \mathcal{V} , otherwise aborts.

5. \mathcal{V} checks that $\|\boldsymbol{\epsilon}\|_2 \leq B_{\text{err}}/2$ and $\|\boldsymbol{\mu}\|_2 \leq B_u/2$ and otherwise aborts.
6. For each $j \in [\ell]$ \mathcal{V}, \mathcal{P} locally set $\text{c}\hat{\text{om}}_j \leftarrow \text{c}\tilde{\text{om}}_j + \boldsymbol{\gamma}[j] - (q/p) \cdot \boldsymbol{\mu}[j] - \boldsymbol{\epsilon}[j]$.
7. \mathcal{P} runs an interactive argument for \mathcal{R}_{Lin} from Appendix B.4 using

$$(pp, u, w) = ((\mathcal{R}, N, q, \text{pk}, \sigma_{\text{com}}, \ell, \boldsymbol{\alpha}), (\{\text{com}, \text{c}\hat{\text{om}}_j\}_{j \in [\ell]}), (\{s, \mathbf{r}, \tilde{\mathbf{r}}_j, 1\}_{j \in [\ell]}))$$

Observe that the first step is only necessary for the generality of Theorem 6. In our application \mathcal{P} will show in the preprocessing that it knows an opening of com so that this step must not be run.

Fig. 7: Zero knowledge argument for \mathcal{R}_{Bob}

length ℓ each. Furthermore, we need to run a ZK argument of both relations $\mathcal{R}_{\text{PoK}}, \mathcal{R}_{\text{Lin}}$ which each require additional $O(\kappa)$ \mathcal{R}_q -elements of communication (see Section B). Therefore we obtain $O(\kappa)$ total communication.

Finally, we discuss in Section E of the Supplementary Material why we chose the specific approach taken in this work instead of other zero-knowledge argument techniques.

6 Evaluation

In this section, we evaluate the efficiency of our OLE protocols, and compare this with protocols based on previous techniques. Firstly, we look at the communication complexity and compare this with other protocols. Then, in Section 6.2, we present implementation results for our passively secure secret-key protocol to demonstrate its practicality.

Choosing Parameters. We estimate parameters for our OLE protocols according to the correctness requirement in Proposition 2. For RLWE we use a ternary secret distribution (so, $B_{\text{sk}} = 1$) a Gaussian error distribution with $\sigma = 3.19$ and $B_{\text{err}} = 6\sigma$; the soundness slack parameter is $\tau = 1$ for passive protocols and $\tau \approx 24\sqrt{8Nn\kappa}$ otherwise. The statistical security parameter is $\kappa = 40$.

6.1 Comparison to Previous Protocols

Protocol	Security		Rounds*	Total comm. (bits)			
	passive	active		$\log m \approx 128$		$\log m \approx 64$	
				passive	active	passive	active
PK-OLE	RLWE	+ FS [†]	1	1516	1630	1004	1120
SK-OLE	RLWE	+ FS	1	758	815	502	560
AHE	RLWE	+ FS + LOE [‡]	2	1320	1476	800	956
SHE	RLWE	+ FS	2	3682	3682	2310	2310
RS	noisy encodings	–	8	4096	4096	2048	2048

[†] FS is Fiat-Shamir

[‡] LOE is linear-only encryption [13]

* 1 round means that each party sends one message simultaneously. 2 rounds either means that each party sequentially sends one message (for AHE), or one simultaneous message, twice in succession (for SHE).

Table 1: Comparison of the complexity of our OLE protocols with previous works based on homomorphic encryption

Table 1 presents the communication complexity, measured from the protocol specifications, of our two public-key and secret-key OLE protocols, and compares this with two other protocols based on RLWE-based homomorphic encryption, either additively homomorphic (AHE) or somewhat homomorphic (SHE), as well as a protocol based on noisy Reed-Solomon encodings (RS). As can be seen from the table, ours is the only protocol with just a single round of communication, where each party simultaneously sends just one message (as in a non-interactive key exchange), whereas both other protocols require two rounds. Our secret-key protocol, which requires some special preprocessing, has the lowest communication cost of all the protocols, with both passive and active security. Furthermore, compared with the previously most efficient protocol based on AHE with active security, our active protocols avoid the need for assuming linear-only encryption, which is a relatively strong and un-studied assumption, compared with standard RLWE.

A full description of these protocols can be found in Section D of the Supplementary Material.

6.2 Experimental Results

We have implemented the passive version of the secret-key protocol (see in Fig. 5) in Go language, making use of the ring package provided by the lattigo library [1]. Our implementation features a full-RNS (Residue Number System) realization of all the protocol operations, using a moduli of 60-bit limbs. For comparison purposes, we have also implemented the AHE-based OLE protocol described in Section A of the Supplementary Material.

The execution times of the protocol steps were tested on a laptop with an Intel Core i7-8550U processor with 16GB RAM, running Arch Linux with kernel 5.6.4 and Go 1.14.2. The latency is not simulated, as it is highly dependent on the particular deployment; we include instead the communication complexity of the involved messages, from which the latency can be derived.

Parameter	Par. set 1	Par. set 2	Bob	Par. set 1	Par. set 2	Alice	Par. set 1	Par. set 2
q	360 bits (6 limbs)	480 bits (8 limbs)	Step 2.(a)	462 ms	601 ms	Step 2.(c)	564 ms	817 ms
p	240 bits (4 limbs)	360 bits (6 limbs)	Step 2.(d)	533 ms	772 ms	Step 3.(a)	350 ms	479 ms
m	60 bits (1 limb)	120 bits (2 limbs)	Step 3.(c)	263 ms	438 ms	Step 3.(d)	242 ms	412 ms
bit security	≈ 159	≈ 116	1st msg.	995 ms	1373 ms	1st msg.	564 ms	817 ms
# OLEs	2097152	2097152	2nd msg.	263 ms	438 ms	2nd msg.	591 ms	890 ms

(a) Example parameter sets ($n = 128$ and $N = 16384$) and global run times for the passive case of Fig. 5 (uniformly random ternary secret keys $\{-1, 0, 1\}$ and Gaussian noise with $\sigma = 3.19$).

(b) Run times in the passive case of Fig. 5 for the example parameter sets of Table 2a ($n = 128$ and $N = 16384$, uniformly random ternary secret keys $\{-1, 0, 1\}$ and Gaussian noise with $\sigma = 3.19$).

Table 2: Parameter sets and run times in the passive case of Fig. 5

We have chosen two practical parameter sets for both protocols (see Tables 2a and 4a), both featuring more than 110 bits of security,¹² and achieving more than 2 million scalar OLEs per protocol run. Table 2b includes the run times corresponding to each party (Alice and Bob) and Table 4b (see Section A of the Supplementary Material) shows the communication costs.

It is worth noting that the public key version from Fig. 3 is not explicitly tested, but it incurs in a similar computational complexity as the one from Fig. 5; it presents, though, an increase on the communication complexity, as the interchanged messages are composed of two polynomials instead of one.

As the latency is not simulated, in order to compare with other protocols, we must consider that the total run time of ours would be $\max(T_{\text{Bob}}, T_{\text{Alice}})$, being T_{Bob} (resp. T_{Alice}) the corresponding run time for Bob (resp. Alice). Tables 3a and 3b include the corresponding expressions and also extrapolate total protocol run times for some specific values of network bandwidth $\{600\text{Mbit/s}, 1\text{Gbit/s}, 10\text{Gbit/s}\}$. T_{step} corresponds to the time of each step included in Table 2b, and $T_{\mathbf{d}}$ (resp. $T_{\mathbf{c}}$) is the time needed to transmit ciphertext \mathbf{d} (resp. \mathbf{c}).

Extrapolated runtimes are approximately equal or lower than those obtained with the protocol based on AHE from Section A of the Supplementary Material; note that for the last one we are not taking into account transmission runtimes. Consequently, we can see that the proposed protocols in this paper achieve both a better efficiency and lower communication cost than the one based on AHE from Section A of the Supplementary Material.

Acknowledgements. We thank the anonymous reviewers for comments which helped to improve the paper. This work has been supported by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme under grant agreement No 669255 (MPCPRO), the Danish Independent Research Council under Grant-ID DFF-6108-00169 (FoCC), an Aarhus University Research Foundation starting grant, the Xunta de Galicia & ERDF under projects ED431G2019/08 and Grupo de Referencia ED431C2017/53, and by the grant #2017-201 (DPPH) of the Strategic Focal Area “Personalized Health and Related Technologies (PHRT)” of the ETH Domain.

¹² We have used the LWE security estimator of Albrecht et al. [2] (available online in <https://bitbucket.org/malb/lwe-estimator>.) to give bit-security estimates.

Run time expressions for Bob and Alice
$T_{\text{Bob}} = \max(T_{2.a} + T_{2.d}, T_{3.a} + T_d) + T_{3.c}$
$T_{\text{Alice}} = \max(T_{3.a}, T_{2.a} + T_c) + T_{2.c} + T_{3.d}$

Total time	600Mbit/s	1Gbit/s	10Gbit/s
Par. Set 1	2526 ms	2023 ms	1344 ms
Par. Set 2	3508 ms	2837 ms	1931 ms

- (a) Total run time expressions for Bob (T_{Bob}) and Alice (T_{Alice}). (b) Extrapolated run times ($\max(T_{\text{Bob}}, T_{\text{Alice}})$) in the passive case of Fig. 5.

Table 3: Total run times expressions and extrapolated run times in the passive case of Fig. 5

Parameter	Par. set 1	Par. set 2
$\{n, N\}$	{256, 8192}	{128, 16384}
p	240 bits (4 limbs)	360 bits (6 limbs)
m	60 bits (1 limb)	120 bits (2 limbs)
bit security	≈ 115	≈ 159
# OLEs	2097152	2097152
Alice time	1441 ms	2129 ms
Bob time	1024 ms	1375 ms
Total time	2465 ms	3504 ms

Proposed protocol of Fig. 5		
{Bob Alice}	Par. set 1	Par. set 2
1st msg. {2.(a) —}	{94.37 —} MB	{125.83 —} MB
2nd msg. {— 3.(a)}	{— 62.91} MB	{— 94.37} MB
AHE-based protocol from Section A		
1st round	{— 62.91} MB	{— 94.37} MB
2nd round	{125.83 —} MB	{188.74 —} MB

- (a) Example parameter sets and global run times for the passively secure OLE based on AHE from Section A of the Supplementary Material (uniformly random ternary secret keys $\{-1, 0, 1\}$ and Gaussian noise with $\sigma = 3.19$). (b) Communication cost in the passive case of Fig. 5 and the passively secure OLE based on AHE from Section A of the Supplementary Material.

Table 4: Parameter sets and communication costs for the passive case of Fig. 5 and Section A of the Supplementary Material

References

- Lattigo 1.3.1. Online: <http://github.com/ldsec/lattigo>, Feb. 2020. EPFL-LDS.
- M. R. Albrecht, R. Player, and S. Scott. On the concrete hardness of learning with errors. *J. Mathematical Cryptology*, 9(3):169–203, 2015.
- S. Ames, C. Hazay, Y. Ishai, and M. Venkatasubramanian. Liger: Lightweight sublinear arguments without a trusted setup. In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *ACM CCS 2017*, pages 2087–2104. ACM Press, Oct. / Nov. 2017.
- B. Applebaum, I. Damgård, Y. Ishai, M. Nielsen, and L. Zichron. Secure arithmetic computation with constant computational overhead. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 223–254. Springer, Heidelberg, Aug. 2017.
- C. Baum, J. Bootle, A. Cerulli, R. del Pino, J. Groth, and V. Lyubashevsky. Sub-linear lattice-based zero-knowledge arguments for arithmetic circuits. In H. Shacham and A. Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 669–699. Springer, Heidelberg, Aug. 2018.
- C. Baum, D. Cozzo, and N. P. Smart. Using TopGear in overdrive: A more efficient ZKPoK for SPDZ. In K. G. Paterson and D. Stebila, editors, *SAC 2019*, volume 11959 of *LNCS*, pages 274–302. Springer, Heidelberg, Aug. 2019.
- C. Baum, I. Damgård, K. G. Larsen, and M. Nielsen. How to prove knowledge of small secrets. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 478–498. Springer, Heidelberg, Aug. 2016.
- C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peikert. More efficient commitments from structured lattice assumptions. In D. Catalano and R. De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 368–385. Springer, Heidelberg, Sept. 2018.
- C. Baum and A. Nof. Concretely-efficient zero-knowledge arguments for arithmetic circuits and their application to lattice-based cryptography. In *Public-Key Cryptography - PKC 2020*. Springer, 2020.
- D. Beaver. Efficient multiparty protocols using circuit randomization. In J. Feigenbaum, editor, *CRYPTO’91*, volume 576 of *LNCS*, pages 420–432. Springer, Heidelberg, Aug. 1992.
- E. Ben-Sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward. Aurora: Transparent succinct arguments for R1CS. In Y. Ishai and V. Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 103–128. Springer, Heidelberg, May 2019.
- R. Bendlin, I. Damgård, C. Orlandi, and S. Zakarias. Semi-homomorphic encryption and multiparty computation. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 169–188. Springer, Heidelberg, May 2011.

13. D. Boneh, Y. Ishai, A. Sahai, and D. J. Wu. Lattice-based SNARGs and their application to more efficient obfuscation. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 247–277. Springer, Heidelberg, Apr. / May 2017.
14. E. Boyle, G. Couteau, N. Gilboa, and Y. Ishai. Compressing vector OLE. In D. Lie, M. Mannan, M. Backes, and X. Wang, editors, *ACM CCS 2018*, pages 896–912. ACM Press, Oct. 2018.
15. E. Boyle, G. Couteau, N. Gilboa, Y. Ishai, L. Kohl, P. Rindal, and P. Scholl. Efficient two-round OT extension and silent non-interactive secure computation. In L. Cavallaro, J. Kinder, X. Wang, and J. Katz, editors, *ACM CCS 2019*, pages 291–308. ACM Press, Nov. 2019.
16. E. Boyle, G. Couteau, N. Gilboa, Y. Ishai, L. Kohl, and P. Scholl. Efficient pseudorandom correlation generators: Silent OT extension and more. In A. Boldyreva and D. Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 489–518. Springer, Heidelberg, Aug. 2019.
17. E. Boyle, N. Gilboa, and Y. Ishai. Breaking the circuit size barrier for secure computation under DDH. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 509–539. Springer, Heidelberg, Aug. 2016.
18. E. Boyle, L. Kohl, and P. Scholl. Homomorphic secret sharing from lattices without FHE. In Y. Ishai and V. Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 3–33. Springer, Heidelberg, May 2019.
19. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In S. Goldwasser, editor, *ITCS 2012*, pages 309–325. ACM, Jan. 2012.
20. Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 505–524. Springer, Heidelberg, Aug. 2011.
21. M. Chase, Y. Dodis, Y. Ishai, D. Kraschewski, T. Liu, R. Ostrovsky, and V. Vaikuntanathan. Reusable non-interactive secure computation. In A. Boldyreva and D. Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 462–488. Springer, Heidelberg, Aug. 2019.
22. R. Cramer, I. Damgård, C. Xing, and C. Yuan. Amortized complexity of zero-knowledge proofs revisited: Achieving linear soundness slack. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 479–500. Springer, Heidelberg, Apr. / May 2017.
23. I. Damgård, M. Keller, E. Larraia, V. Pastro, P. Scholl, and N. P. Smart. Practical covertly secure MPC for dishonest majority - or: Breaking the SPDZ limits. In J. Crampton, S. Jajodia, and K. Mayes, editors, *ESORICS 2013*, volume 8134 of *LNCS*, pages 1–18. Springer, Heidelberg, Sept. 2013.
24. I. Damgård, V. Pastro, N. P. Smart, and S. Zakarias. Multiparty computation from somewhat homomorphic encryption. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 643–662. Springer, Heidelberg, Aug. 2012.
25. I. Damgård, T. P. Pedersen, and B. Pfitzmann. On the existence of statistically hiding bit commitment schemes and fail-stop signatures. In D. R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 250–265. Springer, Heidelberg, Aug. 1994.
26. L. de Castro, C. Juvekar, and V. Vaikuntanathan. Fast vector oblivious linear evaluation from ring learning with errors. Cryptology ePrint Archive, Report 2020/685, 2020. <https://eprint.iacr.org/2020/685>.
27. Y. Dodis, S. Halevi, R. D. Rothblum, and D. Wichs. Spooky encryption and its applications. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 93–122. Springer, Heidelberg, Aug. 2016.
28. N. Döttling, S. Ghosh, J. B. Nielsen, T. Nilges, and R. Trifiletti. TinyOLE: Efficient actively secure two-party computation from oblivious linear function evaluation. In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *ACM CCS 2017*, pages 2263–2276. ACM Press, Oct. / Nov. 2017.
29. D. Genkin, Y. Ishai, M. Prabhakaran, A. Sahai, and E. Tromer. Circuits resilient to additive attacks with applications to secure computation. In D. B. Shmoys, editor, *46th ACM STOC*, pages 495–504. ACM Press, May / June 2014.
30. S. Ghosh, J. B. Nielsen, and T. Nilges. Maliciously secure oblivious linear function evaluation with constant overhead. In T. Takagi and T. Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 629–659. Springer, Heidelberg, Dec. 2017.
31. S. Ghosh and T. Nilges. An algebraic approach to maliciously secure private set intersection. In Y. Ishai and V. Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 154–185. Springer, Heidelberg, May 2019.
32. N. Gilboa. Two party RSA key generation. In M. J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 116–129. Springer, Heidelberg, Aug. 1999.
33. Y. Ishai, J. Kilian, K. Nissim, and E. Petrank. Extending oblivious transfers efficiently. In D. Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 145–161. Springer, Heidelberg, Aug. 2003.
34. Y. Ishai, M. Prabhakaran, and A. Sahai. Secure arithmetic computation with no honest majority. In O. Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 294–314. Springer, Heidelberg, Mar. 2009.
35. M. Keller, E. Orsini, and P. Scholl. MASCOT: Faster malicious arithmetic secure computation with oblivious transfer. In E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, editors, *ACM CCS 2016*, pages 830–842. ACM Press, Oct. 2016.

36. M. Keller, V. Pastro, and D. Rotaru. Overdrive: Making SPDZ great again. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 158–189. Springer, Heidelberg, Apr. / May 2018.
37. V. Lyubashevsky. Lattice signatures without trapdoors. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, Heidelberg, Apr. 2012.
38. V. Lyubashevsky, C. Peikert, and O. Regev. A toolkit for ring-LWE cryptography. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 35–54. Springer, Heidelberg, May 2013.
39. V. Lyubashevsky and G. Seiler. Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 204–224. Springer, Heidelberg, Apr. / May 2018.
40. P. Mohassel and Y. Zhang. SecureML: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy*, pages 19–38. IEEE Computer Society Press, May 2017.
41. M. Naor and B. Pinkas. Oblivious transfer and polynomial evaluation. In *31st ACM STOC*, pages 245–254. ACM Press, May 1999.
42. D. Rathee, T. Schneider, and K. K. Shukla. Improved multiplication triple generation over rings via RLWE-based AHE. In *CANS 2019*, 2019. <https://eprint.iacr.org/2019/577>.
43. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In H. N. Gabow and R. Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.
44. P. Schoppmann, A. Gascón, L. Reichert, and M. Raykova. Distributed vector-OLE: Improved constructions and implementation. In L. Cavallaro, J. Kinder, X. Wang, and J. Katz, editors, *ACM CCS 2019*, pages 1055–1072. ACM Press, Nov. 2019.
45. N. P. Smart and F. Vercauteren. Fully homomorphic SIMD operations. *Des. Codes Cryptography*, 71(1):57–81, Apr. 2014.

A Passively Secure OLE From Linearly Homomorphic Encryption

We start out with two parties having $\mathbf{u}, \mathbf{v} \in \mathcal{R}_m^n$ as inputs where $\mathcal{P}_{\text{Alice}}$ has \mathbf{u} and \mathcal{P}_{Bob} \mathbf{v} . Furthermore, assume that \mathcal{P}_{Bob} has $\beta \in \mathcal{R}_m^n$ as well. Our goal is, again, to compute $\alpha \in \mathcal{R}_m^n$ such that $\alpha + \beta = \mathbf{u} \star \mathbf{v}$. Here \cdot is the product of ring elements or of a scalar multiplication with a vector, while \star denotes the element-wise multiplication of two vectors.

1. First, $\mathcal{P}_{\text{Alice}}$ samples $s \in \mathcal{R}_m$ according to the ternary distribution, $c \in \mathcal{R}_p$ uniformly at random as well as $r \in \mathcal{R}^n$ as a discrete Gaussian with the same standard deviation $\sigma = 3.19$ (approximated using a Binomial distribution). It sends $pk = (c, d = c \cdot s + r)$ to \mathcal{P}_{Bob} .
2. In the first round of the OLE, $\mathcal{P}_{\text{Alice}}$ samples $\mathbf{e}_1 \in \mathcal{R}^n$ as a discrete Gaussian with the same standard deviation $\sigma = 3.19$ (approximated using a Binomial distribution). $\mathcal{P}_{\text{Alice}}$ then samples $\mathbf{a} \in \mathcal{R}_p^n$ uniformly at random, computes $\mathbf{b} = \mathbf{a} \cdot s - (p/m) \cdot \mathbf{u} + \mathbf{e}_1$ and sends \mathbf{a}, \mathbf{b} to \mathcal{P}_{Bob} .
3. In the second round, \mathcal{P}_{Bob} samples $\mathbf{t} \in \mathcal{R}_m^n$ according to the ternary distribution, $\mathbf{e}_2 \in \mathcal{R}^n$ according to a discrete Gaussian with $\sigma = 3.19$ but moreover $\mathbf{e}_3 \in \mathcal{R}^n$ uniform with infinity norm smaller than $3mN^2\sigma \cdot 2^\kappa < p/(2m)$. It sets $\mathbf{x} = \mathbf{a} \star \mathbf{v} + c \cdot \mathbf{t} + \mathbf{e}_2$ and $\mathbf{y} = \mathbf{b} \star \mathbf{v} + d \cdot \mathbf{t} + (p/m) \cdot \beta + \mathbf{e}_3$ and sends \mathbf{x}, \mathbf{y} to $\mathcal{P}_{\text{Alice}}$.
4. Finally, $\mathcal{P}_{\text{Alice}}$ computes $\alpha = \lfloor (\mathbf{x} \cdot s - \mathbf{y} \bmod p) / (p/m) \rfloor$ and outputs this.

We see that

$$\begin{aligned} \mathbf{x} \cdot s - \mathbf{y} &= (\mathbf{a} \star \mathbf{v} + c \cdot \mathbf{t} + \mathbf{e}_2) \cdot s - \mathbf{b} \star \mathbf{v} - d \cdot \mathbf{t} - \mathbf{e}_3 - (p/m) \cdot \beta \\ &= (\mathbf{a} \star \mathbf{v} + c \cdot \mathbf{t}) \cdot s + \mathbf{e}_2 \cdot s - (\mathbf{a} \cdot s) \star \mathbf{v} - \mathbf{e}_1 \star \mathbf{v} - (c \cdot s) \star \mathbf{t} - r \cdot \mathbf{t} \\ &\quad - \mathbf{e}_3 + (p/m)(\mathbf{u} \star \mathbf{v} - \beta) \end{aligned}$$

and the desired relation holds. The noise terms add up to $r \cdot \mathbf{t} + \mathbf{e}_1 \star \mathbf{v} + \mathbf{e}_2 \cdot s + \mathbf{e}_3$ where \mathbf{e}_3 due to its size statistically drowns the other three terms, which makes simulation possible. This is because $\|\mathbf{e}_1\|_\infty \leq 6\sigma$ with overwhelming probability and $\|\mathbf{v}\|_\infty \leq m/2$, so $\|\mathbf{e}_1 \star \mathbf{v}\|_\infty \leq 3mN^2$ with overwhelming probability, while $r \cdot \mathbf{t}$ and $\mathbf{e}_2 \cdot s$ are a lot smaller.

Furthermore, since we also require that $3mN^2\sigma \cdot 2^\kappa < p/(2m)$ this term will not lead to a wrap-around mod p during decryption and the result will be correct with overwhelming probability. Observe that $\mathcal{P}_{\text{Alice}}$ will not actually have to send \mathbf{a} because it can be sampled from a PRG-seed by both $\mathcal{P}_{\text{Alice}}, \mathcal{P}_{\text{Bob}}$, meaning that the communication only requires $3 \mathcal{R}_p$ -elements for one \mathcal{R}_m -OLE once pk is set up.

B Commitments & Zero-Knowledge Arguments, continued

B.1 Rejection Sampling

Lyubashevsky [37] proved the following Theorem, which is at the core of many lattice-based constructions:

Theorem 7 (See Theorem 4.6 of [37]). *Let $V \subseteq \mathbb{Z}^m$ such that all elements of V have norm less than T , $\sigma \in \mathbb{R}$ such that $\sigma = \omega(T\sqrt{\log m})$ and $h : V \rightarrow \mathbb{R}$ be a probability distribution. Then there exists a constant $M = O(1)$ such that the distributions of the following algorithm \mathcal{A} :*

1. $\mathbf{v} \leftarrow h$
2. $\mathbf{x} \leftarrow \mathcal{D}_{\mathbf{v}, \sigma}^m$
3. Output (\mathbf{x}, \mathbf{v}) with probability $\min\left(\frac{\mathcal{D}_\sigma^m(\mathbf{x})}{M\mathcal{D}_{\mathbf{v}, \sigma}^m(\mathbf{x})}, 1\right)$

is within statistical distance $2^{-\omega(\log m)}/M$ of the distribution of the following algorithm \mathcal{F} :

1. $\mathbf{v} \leftarrow h$

2. $\mathbf{x} \leftarrow \mathcal{D}_\sigma^m$
3. Output (\mathbf{x}, \mathbf{v}) with probability $1/M$

where \mathcal{A} outputs something with probability at least $\frac{1-2^{-\omega(\log m)}}{M}$.

For concreteness, if $\sigma = \alpha T$ for $\alpha > 0$ then $M = \exp(12/\alpha + 1/(2\alpha^2))$, meaning that the output of \mathcal{A} is within statistical distance $2^{-100}/M$ of \mathcal{F} and that \mathcal{A} outputs something with probability at least $\frac{1-2^{-100}}{M}$.

We can now use the aforementioned theorem to prove Lemma 3:

Proof. We perform the experiment from Theorem 7 for each of the k components using discrete Gaussian distributions $\mathcal{D}_{\sigma_i}^{m_i}$ in the process. Since α is fixed among all components, then in particular the constant M will be the same across all k individual experiments. By a union bound, we obtain the statistical distance $2^{-100+\log k}/M$ by adding up all k identical terms. The fact that \mathcal{F} outputs the vector with probability at least $((1-2^{-100})/M)^k$ follows by the independence of the k experiments. \square

B.2 Instantiating (Somewhat Homomorphic) Commitments

After having introduced the formal definitions for commitment schemes and zero-knowledge proofs in Appendix 2.6 we will now recap an implementation which we use in our construction, namely the somewhat homomorphic commitment scheme of Baum et al. [8] and its accompanying zero-knowledge proofs.

A specific property of the scheme of [8] is that it allows “relaxed openings”, meaning that an opening of a commitment \mathbf{com} does not just consist of the message x and randomness r but also some additional factor f . The guarantee is then that the scheme is binding as long as it is hard to come up with two (x, r, f) and (x, r', f') that open the same commitment \mathbf{com} . In order to define f properly let $\mathcal{C} = \{c \in \mathcal{R}_q \mid \|c\|_\infty = 1 \wedge \|c\|_1 = \kappa\}$ as well as $\bar{\mathcal{C}} = \{c - c' \mid c \neq c' \in \mathcal{C}\}$. [39] showed under which conditions all elements of $\mathcal{C}, \bar{\mathcal{C}}$ are invertible.

Somewhat Homomorphic Commitments. The commitment scheme of [8] consists of the following algorithms:

KG: Given $\mathcal{R}_q, N, k, n, \beta, \sigma_{\text{Com}}$ sample $\mathbf{A}'_1 \leftarrow \mathcal{R}_q^{n \times (k-n)}$ as well as $\mathbf{a}'_2 \leftarrow \mathcal{R}_q^{k-n-1}$. Set $\mathbf{A}_1 = [\mathbf{I}_n \ \mathbf{A}'_1]$ and $\mathbf{a}_2 = [\mathbf{0}^n \ 1 \ \mathbf{a}'_2]$ and output $\mathbf{pk} = (\mathcal{R}_q, N, k, n, \beta, \mathbf{A}_1, \mathbf{a}_2)$.

Com: On input a valid public key \mathbf{pk} and a value $x \in \mathcal{R}_q$ sample $\mathbf{r} \leftarrow S_\beta^k$, compute

$$\mathbf{com} = \begin{bmatrix} \mathbf{com}_1 \\ \mathbf{com}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{a}_2 \end{bmatrix} \times \mathbf{r} + \begin{bmatrix} \mathbf{0}^n \\ x \end{bmatrix}$$

and output $(\mathbf{com}, \mathbf{r})$.

Open: On input a valid public key \mathbf{pk} and $\mathbf{com} = \begin{bmatrix} \mathbf{com}_1 \\ \mathbf{com}_2 \end{bmatrix} \in \mathcal{R}_q^{n+1}$, $x \in \mathcal{R}_q$, $\mathbf{r} \in \mathcal{R}_q^k$ as well as $f \in \bar{\mathcal{C}}$ output 1 iff

$$f \cdot \begin{bmatrix} \mathbf{com}_1 \\ \mathbf{com}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{a}_2 \end{bmatrix} \times \mathbf{r} + f \cdot \begin{bmatrix} \mathbf{0}^n \\ x \end{bmatrix} \quad \text{and} \quad \forall i \in [k] : \|\mathbf{r}[i]\|_2 \leq 4\sigma_{\text{Com}}\sqrt{N}$$

and 0 otherwise.

[8] showed that their construction is binding under the $\text{MSIS}_{n,k,16\sigma_{\text{Com}}\kappa N}$ assumption and hiding under the $\text{MLWE}_{n+1,k,\beta}$ assumption.

It can directly be seen that the commitment scheme is *somewhat homomorphic*:

Zero Knowledge Argument for Opening \mathcal{R}_{PoK}

The following is a zero knowledge argument for proving knowledge of an opening of a commitment, i.e. for the relation \mathcal{R}_{PoK} . Let $C_{\text{aux}} = (\text{KG}_{\text{aux}}, \text{Com}_{\text{aux}}, \text{Open}_{\text{aux}})$ be a statistically hiding auxiliary commitment scheme with public key pk_{aux} known to both \mathcal{P}, \mathcal{V} .

1. For $i \in [\ell]$ \mathcal{P} samples $\mathbf{y}_i \leftarrow \mathcal{D}_{\sigma_{\text{Com}}}^k$ and computes $\mathbf{t}_i \leftarrow \mathbf{A}_1 \mathbf{y}_i$. \mathcal{P} then generates the auxiliary commitments $(\text{com}_{\mathbf{t},i}, r_{\mathbf{t},i}) \leftarrow \text{Com}_{\text{aux}}(\mathbf{t}_i)$ and sends these to \mathcal{V} .
2. \mathcal{P}, \mathcal{V} run a coin-flipping protocol based on C_{aux} to sample $\mathbf{d} \leftarrow \{0, 1\}^\ell$.
3. For each $i \in [\ell]$ \mathcal{P} sets $\mathbf{z}_i \leftarrow \mathbf{y}_i + \mathbf{d}[i] \cdot \mathbf{r}$. Let \mathbf{z} be the concatenation of all \mathbf{z}_i and ϵ of $\mathbf{d}[i] \cdot \mathbf{r}$ respectively. Then with probability

$$\min \left(1, \frac{\mathcal{D}_{\sigma_{\text{Com}}}^{k\ell}(\mathbf{z})}{M \cdot \mathcal{D}_{\epsilon, \sigma_{\text{Com}}}^{k\ell}(\mathbf{z})} \right)$$

\mathcal{P} sends \mathbf{z} to \mathcal{V} , otherwise he aborts. \mathcal{P} furthermore sends the openings for the commitments generated with C_{aux} .

4. If \mathcal{V} receives \mathbf{z} then he accepts if for all $i \in [\ell]$:
 - (a) $\text{Open}_{\text{aux}}(\text{com}_{\mathbf{t},i}, \mathbf{A}_1 \mathbf{z}_i - \mathbf{d}[i] \text{com}[1], r_{\mathbf{t},i}) = 1$
 - (b) $\forall j \in [k] : \|\mathbf{z}_i[j]\|_2 \leq \sigma_{\text{Com}} \sqrt{8N}$

Fig. 8: Zero knowledge argument for Opening \mathcal{R}_{PoK}

- If $\text{com} = \begin{bmatrix} \text{com}_1 \\ \text{com}_2 \end{bmatrix}$ is a commitment that can be opened as (x, \mathbf{r}, f) then for a public $x' \in \mathcal{R}_q$ the commitment $\text{com}' = \begin{bmatrix} \text{com}_1 \\ \text{com}_2 + x' \end{bmatrix}$ can be opened as $(x + x', \mathbf{r}, f)$.
- For h commitments $\text{com}^i = \begin{bmatrix} \text{com}_1^i \\ \text{com}_2^i \end{bmatrix}$ with openings (x^i, \mathbf{r}^i, f) we have that $\text{com} = \begin{bmatrix} \sum_{i \in [h]} \text{com}_1^i \\ \sum_{i \in [h]} \text{com}_2^i \end{bmatrix}$ has opening $(\sum_{i \in [h]} x^i, \sum_{i \in [h]} \mathbf{r}^i, f)$ if $\sum_{i \in [h]} \mathbf{r}^i$ still fulfills the bound of Open .

In comparison to fully linearly homomorphic commitments it is not possible to generate a commitment com' from com with opening x such that com' opens to $\alpha \cdot x$ for a public α without losing the binding property. Instead, one has to use a zero-knowledge proof to show such a relation. Below we will describe both the standard proof of knowledge for the commitment scheme as well as the proof of linear relation.

B.3 Zero-Knowledge Proof of Opening

The commitment scheme of [8] comes with a highly efficient proof of knowledge and other efficient proofs. The disadvantage of this class of proofs is that soundness does not reduce to a standard opening $(x, \mathbf{r}, 1)$ but instead will extract (x, \mathbf{r}, f) with $f \in \mathcal{C}$. These “extended openings” are never generated by an honest party but possibly by an adversary and are not sufficient in our application.

$$\mathcal{R}_{\text{PoK}} = \left\{ \left(\begin{array}{l} (pp, u, w) = \\ (\mathcal{R}, N, q, \text{pk}, \sigma_{\text{Com}}), \\ \text{com}, (x, \mathbf{r}) \end{array} \right) \middle| \begin{array}{l} (x, \mathbf{r}) \in \mathcal{R}_q \times \mathcal{R}_q^k \wedge \\ \text{Open}_{\text{pk}}(\text{com}, x, \mathbf{r}, 1) = 1 \end{array} \right\}$$

Instead, we use a “standard” proof of knowledge for a commitment which follows the standard Fiat-Shamir with Aborts signature. The algorithm can be found in Fig. 8.

Lemma 6. *Let $M > 1$ and $\sigma_{\text{Com}} \geq 12/\ln M \sqrt{\beta N k}$ as well as $\ell > \kappa$. The algorithm from Fig. 8 is a zero-knowledge argument of knowledge that is complete with probability $1/M$, computationally sound and statistically honest-verifier zero knowledge.*

Zero Knowledge Argument for Linear Relation \mathcal{R}_{Lin}

The following is a zero knowledge argument for proving knowledge of openings for the relation \mathcal{R}_{Lin} . Let $C_{\text{aux}} = (\text{KG}_{\text{aux}}, \text{Com}_{\text{aux}}, \text{Open}_{\text{aux}})$ be a statistically hiding auxiliary commitment scheme with public key pk_{aux} known to both \mathcal{P}, \mathcal{V} .

1. For $i \in [n]$ \mathcal{P} samples $\mathbf{y}_i, \mathbf{y}'_i \leftarrow \mathcal{D}_{\sigma_{\text{Com}}}^k$, computes $\mathbf{t}_i \leftarrow \mathbf{A}_1 \mathbf{y}_i, \mathbf{t}'_i \leftarrow \mathbf{A}_1 \mathbf{y}'_i$ as well as $u_i \leftarrow \alpha[i] \cdot \langle \mathbf{a}_2, \mathbf{y}_i \rangle - \langle \mathbf{a}_2, \mathbf{y}'_i \rangle$. \mathcal{P} generates auxiliary commitments

$$\begin{aligned} (\text{com}_{\mathbf{t},i}, r_{\mathbf{t},i}) &\leftarrow \text{Com}_{\text{aux}}(\mathbf{t}_i), \\ (\text{com}_{\mathbf{t}',i}, r_{\mathbf{t}',i}) &\leftarrow \text{Com}_{\text{aux}}(\mathbf{t}'_i), \\ (\text{com}_{u,i}, r_{u,i}) &\leftarrow \text{Com}_{\text{aux}}(u_i) \end{aligned}$$

and sends these to \mathcal{V} .

2. \mathcal{P}, \mathcal{V} run a coin-flipping protocol based on C_{aux} to sample $\mathbf{d} \leftarrow \mathcal{C}^n$.
3. For each $i \in [n]$ \mathcal{P} sets $\mathbf{z}_i \leftarrow \mathbf{y}_i + \mathbf{d}[i] \cdot \mathbf{r}_i, \mathbf{z}'_i \leftarrow \mathbf{y}'_i + \mathbf{d}[i] \cdot \mathbf{r}'_i$. Let \mathbf{z}, \mathbf{z}' be the concatenation of all $\mathbf{z}_i, \mathbf{z}'_i$ respectively and ϵ, ϵ' of $\mathbf{d}[i] \cdot \mathbf{r}_i$ and $\mathbf{d}[i] \cdot \mathbf{r}'_i$. Then with probability

$$\min \left(1, \frac{\mathcal{D}_{\sigma_{\text{Com}}}^{kn}(\mathbf{z}) \cdot \mathcal{D}_{\sigma_{\text{Com}}}^{kn}(\mathbf{z}')}{M^2 \cdot \mathcal{D}_{\epsilon, \sigma_{\text{Com}}}^{kn}(\mathbf{z}) \cdot \mathcal{D}_{\epsilon', \sigma_{\text{Com}}}^{kn}(\mathbf{z}')} \right)$$

\mathcal{P} sends \mathbf{z}, \mathbf{z}' to \mathcal{V} , otherwise he aborts. \mathcal{P} furthermore sends the openings for the commitments generated with C_{aux} .

4. If \mathcal{V} receives \mathbf{z}, \mathbf{z}' then he accepts if for all $i \in [n]$:
 - (a) $\text{Open}_{\text{aux}}(\text{com}_{\mathbf{t},i}, \mathbf{A}_1 \mathbf{z}_i - \mathbf{d}[i] \text{com}_{\mathbf{t},i}[1], r_{\mathbf{t},i}) = 1$
 - (b) $\text{Open}_{\text{aux}}(\text{com}_{\mathbf{t}',i}, \mathbf{A}_1 \mathbf{z}'_i - \mathbf{d}[i] \text{com}_{\mathbf{t}',i}[1], r_{\mathbf{t}',i}) = 1$
 - (c) $\text{Open}_{\text{aux}}(\text{com}_{u,i}, h_i, r_{u,i}) = 1$
 - (d) $\forall j \in [k] : \|\mathbf{z}_i[j]\|_2 \leq \sigma_{\text{Com}} \sqrt{8N}$ and $\|\mathbf{z}'_i[j]\|_2 \leq \sigma_{\text{Com}} \sqrt{8N}$
 where $h_i \leftarrow \alpha[i] \cdot \langle \mathbf{a}_2, \mathbf{z}_i \rangle - \langle \mathbf{a}_2, \mathbf{z}'_i \rangle - (\alpha[i] \text{com}_{\mathbf{t},i}[2] - \text{com}_{\mathbf{t}',i}[2]) \cdot \mathbf{d}[i]$.

Fig. 9: Zero knowledge argument for Linear Relation \mathcal{R}_{Lin}

Observe that this argument has a much higher communication complexity (by a factor κ) than the optimized AoK of [8] for this commitment scheme, but it has the aforementioned advantage of being more “exact”. Furthermore, in the overall work we will only use it once.

B.4 Zero-Knowledge Proof of Linear Relation

Based on [8] one can easily prove a linear dependency $\alpha \in \mathcal{R}_q$ between the openings of two commitments. Consider the relation

$$\mathcal{R}_{\text{Lin}} = \left\{ \left((\mathcal{R}, N, q, \text{pk}, \sigma_{\text{Com}}, n, \alpha), \begin{array}{l} \{\text{com}_i, \text{com}'_i\}_{i \in [n]}, \\ \{x_i, \mathbf{r}_i, \mathbf{r}'_i, f_i\}_{i \in [n]} \end{array} \right) \left| \begin{array}{l} \forall i \in [n] : \\ (x_i, \mathbf{r}_i, \mathbf{r}'_i, f_i) \in \mathcal{R}_q \times \mathcal{R}_q^k \times \mathcal{R}^k \times \overline{\mathcal{D}} \wedge \\ \text{Open}_{\text{pk}}(\text{com}_i, x_i, \mathbf{r}_i, f_i) = 1 \wedge \\ \text{Open}_{\text{pk}}(\text{com}'_i, \alpha[i]x_i, \mathbf{r}'_i, f_i) = 1 \end{array} \right. \right\}$$

One can use the zero-knowledge proof as outlined in Fig. 9 to prove \mathcal{R}_{Lin} , meaning that each of the n commitments com'_i has an opening that is $\alpha[i]$ away from the opening of com_i .

Lemma 7. *Let $M > 1$ and $\sigma_{\text{Com}} \geq 12/\ln M \sqrt{\kappa \beta N k n}$. The algorithm from Fig. 9 is a zero-knowledge argument of knowledge that is complete with probability $1/M^2$, computationally sound and statistically honest-verifier zero knowledge.*

Proof. The proof follows as a generalization of the linearity-proof for two commitments of [8]. The only difference is that we adjusted the size of σ_{Com} to allow a rejection-sampling for the whole vector $\mathbf{z}_i, \mathbf{z}'_i$ at once, which leads to a slightly worse bound on the extracted openings. \square

One important property of the protocol is that the extracted opening for both $\text{com}_i, \text{com}'_i$ will have the same value f for both commitments.

B.5 Some Missing Proofs

Theorem 8 (Theorem 5 restated). *Assume that $\forall i \in [n] : \|\mathbf{f}[i]\|_2 \leq T_{\text{sk}}, \|\mathbf{e}_0[i]\|_2, \|\mathbf{e}_1[i]\|_2 \leq T_{\text{err}}, \|\mathbf{x}[i]\|_2 \leq T_{\text{msg}}$. Let $\ell \geq \kappa + 2$, $M > 1$ and $\sigma_x \geq 12/(\ln M)\sqrt{n\ell}T_x$. Furthermore, let C_{aux} be a statistically hiding and computationally binding commitment scheme. Then the aforementioned protocol is a zero-knowledge argument of knowledge for \mathcal{R}_{pk} with $(B_{\text{sk}}, B_{\text{err}}, B_{\text{msg}}) = (\sqrt{8N}\sigma_{\text{sk}}, \sqrt{8N}\sigma_{\text{err}}, \sqrt{8N}\sigma_{\text{msg}})$ that is complete with probability $1/M^4$, computationally sound and statistically zero-knowledge.*

Proof. The proof follows directly from Theorem 1 of [5] where the zero-knowledge argument is done individually for each row of \mathbf{S} using Lemma 3. We will therefore only sketch the most important parts.

Completeness. We will focus on the contribution of \mathbf{f} throughout the argument but it generalizes to $\mathbf{e}_0, \mathbf{e}_1, \mathbf{x}$ accordingly. We know that for $i \in [n]$ it holds that $\|\mathbf{f}[i]\|_2 \leq T_{\text{sk}}$. Thus $\|(\mathbf{C}\mathbf{f})[i]\|_2 \leq \sqrt{n} \cdot T_{\text{sk}}$ for all elements of the vector and hence $\|\mathbf{C}\mathbf{f}\|_2 \leq \sqrt{\ell n}T_{\text{sk}}$. Using σ_{sk} from the statement as well as Lemma 3 we can see that \mathbf{Z} will be sent with the required probability. As therefore particularly $\mathbf{z}_1 \sim \mathcal{D}_{\sigma_{\text{sk}}}^\ell$ we can upper-bound its norm using Lemma 2 setting $k = 2$ and using $N \gg \kappa$. Therefore Step 4b succeeds with overwhelming probability, while Step 4a holds by linearity.

Honest-Verifier Zero-Knowledge. In the simulation \mathcal{S} will simply sample \mathbf{C} as in the protocol, generate

$$(\mathbf{z}_1, \dots, \mathbf{z}_4) \leftarrow \mathcal{D}_{\sigma_{\text{sk}}}^\ell \times \mathcal{D}_{\sigma_{\text{err}}}^\ell \times \mathcal{D}_{\sigma_{\text{err}}}^\ell \times \mathcal{D}_{\sigma_{\text{msg}}}^\ell,$$

set $\mathbf{W} \leftarrow \mathbf{A}\mathbf{Z}^\top - \mathbf{T}\mathbf{C}^\top$ and commit to the respective value \mathbf{W} . \mathcal{S} then outputs the transcript $(c_{\mathbf{W}}, \mathbf{C}, (r_{\mathbf{W}}, \mathbf{Z}))$ with probability $1/M^4$. Observe that indistinguishability follows by Lemma 3 as well as the statistical hiding property of C_{aux} .

Soundness. By a standard heavy-column argument we can rewind a successful prover \mathcal{P}^* on different challenges for the same first message $c_{\mathbf{W}}$ and are able to obtain multiple transcripts for different (\mathbf{C}, \mathbf{Z}) , but fixed $\text{com}_{\mathbf{W}}$. Observe that by the binding property of C_{aux} we must have that $\text{com}_{\mathbf{W}}$ always opens to the same value. This allows us to obtain multiple equations of the form $\mathbf{T}(\mathbf{C} - \mathbf{C}')^\top = \mathbf{A}(\mathbf{Z} - \mathbf{Z}')^\top$. We will extract $\mathbf{f}[i], \mathbf{e}_0[i], \mathbf{e}_1[i], \mathbf{x}[i]$ by rewinding with all columns of \mathbf{C}, \mathbf{C}' being fixed during rewinding except for the i -th column, which means that all except for the i -th part of \mathbf{T} will disappear when being multiplied by $(\mathbf{C} - \mathbf{C}')^\top$. Then by the aforementioned equation and the bounds on the \mathbf{z}_i from successfully produced transcripts the result follows. The lower bound of $\ell \geq \kappa + 2$ is a consequence of the heavy-column argument of [5]. \square

Theorem 9 (Theorem 6 restated). *Assume that $\forall i \in [n] : \|\mathbf{v}[i]\|_2 \leq T_{\mathbf{u}}, \|\mathbf{e}[i]\|_2 \leq T_{\mathbf{e}}$. Let $\ell \geq \kappa + 3$, $M > 1$ and $\sigma_x \geq 12/(\ln M)\sqrt{n\ell}T_x$. Furthermore, let C be a statistically hiding and computationally binding commitment scheme. Then the aforementioned protocol is a zero-knowledge argument of knowledge for $\mathcal{R}_{\text{Bob}}^{\text{sk}}$ with $(B_{\mathbf{u}}, B_{\text{err}}) = (\sqrt{8N}\sigma_{\mathbf{u}}, \sqrt{8N}\sigma_{\mathbf{e}})$ that is complete with probability $1/M^4$, computationally sound and statistically zero-knowledge.*

Proof. The proof uses elements of the proof of Theorem 5, though some modifications are necessary to obtain the full statement.

Completeness. The proof of completeness follows along the exact same lines as the proof of Theorem 5, except that we now only have 2 vectors to perform rejection sampling on and not 4. All of the bounds can be determined the exact same way, and the correctness of the opening follows by homomorphism of the commitment scheme and the completeness of the arguments for \mathcal{R}_{PoK} and \mathcal{R}_{Lin} . We assume that both arguments succeed with probability $1/M$ which yields the claim.

Honest-Verifier Zero-Knowledge. The simulator will start by a simulation of the argument for \mathcal{R}_{PoK} . If this sub-simulator outputs τ_1 then we will continue simulating, otherwise abort if the sub-simulator for \mathcal{R}_{PoK} aborts and output its aborting transcript.

As in the proof of Theorem 5, the simulator then generates $\mathbf{C} \leftarrow \{0, 1\}^{\ell \times n}$ honestly and samples $\boldsymbol{\epsilon} \leftarrow \mathcal{D}_{\sigma_e}^\ell, \boldsymbol{\mu} \leftarrow \mathcal{D}_{\sigma_u}^\ell$. Given this, we can compute $\boldsymbol{\alpha}, \boldsymbol{\gamma}$ as in the protocol.

Next, sample the commitments $\text{c}\tilde{\mathbf{m}}_j$ as uniformly random commitments and generate the $\text{c}\tilde{\mathbf{m}}_j$ such that the equation from 6 holds. With probability $1/M^2$ we output $(\tau_1, \{\text{c}\tilde{\mathbf{m}}_j\}_{j \in [\ell]}, \mathbf{C})$ and abort, otherwise we generate τ_2 as the output of the statistical zero-knowledge simulator of the argument for \mathcal{R}_{Lin} . Here, we abort and only output $(\tau_1, \{\text{c}\tilde{\mathbf{m}}_j\}_{j \in [\ell]}, \mathbf{C}, \boldsymbol{\epsilon}, \boldsymbol{\mu})$ if this sub-simulator aborts or otherwise output $(\tau_1, \{\text{c}\tilde{\mathbf{m}}_j\}_{j \in [\ell]}, \mathbf{C}, \boldsymbol{\epsilon}, \boldsymbol{\mu}, \tau_2)$ if the sub-simulator succeeds.

Observe that the overall probability of outputting a full simulated transcript is $1/M^4$, a transcript without τ_2 $1/M^3$ and a transcript only containing τ_1 with probability $1/M$ which is the same as in the protocol. The values $\boldsymbol{\epsilon}, \boldsymbol{\mu}$ are distributed as in the protocol by Lemma 3 whereas $\{\text{c}\tilde{\mathbf{m}}_j\}_{j \in [\ell]}$ are statistically indistinguishable from those of the protocol due to the statistical hiding of the commitment scheme. Then, since τ_1, τ_2 are statistically indistinguishable from the real transcript of the arguments for $\mathcal{R}_{\text{PoK}}, \mathcal{R}_{\text{Lin}}$ the statement follows.

Soundness. Assume that there exists a prover \mathcal{P}^* that can convince a verifier with probability $\epsilon > 2^{-\kappa+3}$. Observe that \mathcal{P}^* 's randomness tape is fixed, except for the inputs coming from the interaction. We first rewind the prover on the subprotocol for \mathcal{R}_{PoK} where we extract the opening $(s, \mathbf{r}, 1)$. Observe that by the binding property of C \mathcal{P}^* is not able to open c to any (s', \mathbf{r}', f') with $s' \neq s$ throughout the rest of this argument.

Next, we fix some choice for \mathcal{R}_{PoK} and continue with the rest of the protocol with \mathcal{P}^* , calling this new hybrid algorithm \mathcal{P}_1 . This fixes the commitments $\text{c}\tilde{\mathbf{m}}_j$ for the remaining protocol. By the standard heavy-column Lemma, with probability at least $1/2$ we have that the fraction of choices of \mathbf{C} and randomness in \mathcal{R}_{Lin} which make \mathcal{P}_1 output a valid transcript is at least $2^{-\kappa+2}$.

Using the same fixing of \mathbf{C} we now arrive at a prover \mathcal{P}_2 which with probability $1/4$ outputs valid transcripts with probability at least $> 2^{-\kappa+1}$. Using the soundness of \mathcal{R}_{Lin} in Step 7 we have that any $\text{c}\tilde{\mathbf{m}}_j$ must contain a value of the form $-\boldsymbol{\alpha}[j] \cdot s$ for some $\boldsymbol{\alpha}[j]$ as com in \mathcal{R}_{Lin} can only be opened to s due to the binding property of the scheme. Observe that these openings of $\text{c}\tilde{\mathbf{m}}_j$ can be transformed into openings of $\text{c}\tilde{\mathbf{m}}_j$ by linearity of the scheme, i.e. they do also work for \mathcal{P}_1 .

Next, using the same heavy-column argument as in the proof of Theorem 5 we can then for each $i \in [n]$ extract two accepting transcripts that have \mathbf{C}, \mathbf{C}' where all columns of \mathbf{C} are identical to \mathbf{C}' except for column i , which means that there is a j such that

$$-(\boldsymbol{\alpha}'[j] - \boldsymbol{\alpha}[j])s = \boldsymbol{\gamma}[j] - \boldsymbol{\gamma}'[j] - (q/p)(\boldsymbol{\mu}[j] - \boldsymbol{\mu}'[j]) - (\boldsymbol{\epsilon}[j] - \boldsymbol{\epsilon}'[j])$$

where by the definition of $\boldsymbol{\alpha}, \boldsymbol{\gamma}$ it must now hold that

$$\mathbf{a}[i] \cdot s = \mathbf{c}[i] - (q/p)(\boldsymbol{\mu}[j] - \boldsymbol{\mu}'[j]) - (\boldsymbol{\epsilon}[j] - \boldsymbol{\epsilon}'[j]).$$

Setting $\mathbf{u}[i] = \boldsymbol{\mu}[j] - \boldsymbol{\mu}'[j]$ and $\mathbf{e}[i] = \boldsymbol{\epsilon}[j] - \boldsymbol{\epsilon}'[j]$ then yields the necessary values of the appropriate bounds. Observe that the loss in success probability is only constant (i.e. we can amplify it back by repeating the experiment) and that the extractor runs in time $\text{poly}(\epsilon, \kappa)$. \square

C Missing Proofs in OLE Sections

Theorem 10 (Theorem 1 restated). *Assume that $3 \cdot 2^{\kappa+1} \cdot n \cdot (mN)^2 \cdot B_{\text{err}} \cdot B_{\text{sk}} \leq p \leq \frac{q}{3 \cdot 2^{\kappa+1} \cdot n \cdot N^2 \cdot B_{\text{err}} \cdot B_{\text{sk}}}$. Then protocol $\Pi_{\text{OLE-pk}}^{\text{passive}}$, which consists of protocol $\Pi_{\text{OLE-pk}}$ without the underlined steps, realizes functionality \mathcal{F}_{OLE} in the \mathcal{F}_{PK1} -hybrid model under the RLWE assumption.*

Proof. Let us consider first the case in which \mathcal{P}_{Bob} is passively corrupt. The transcript of the computation for an environment \mathcal{Z} that corrupts \mathcal{P}_{Bob} consists of the inputs (\mathbf{u}, \mathbf{v}) , the values from the setup phase $(s_{\text{Bob}}, b_{\text{Bob}}, a)$, the intermediate messages $(\mathbf{d}_0, \mathbf{d}_1)$ and the outputs $(\boldsymbol{\alpha}, \boldsymbol{\beta})$. Our goal is to show the existence of a simulator \mathcal{S} that, on input $(\mathbf{u}, \boldsymbol{\beta})$, can simulate the transcript above so that it is indistinguishable from a real execution.

The simulator proceeds as follows. It emulates the setup phase by sampling $a, s_{\text{Alice}}, s_{\text{Bob}}, b_{\text{Alice}}, b_{\text{Bob}}$ as in the real execution, and it sets $(\mathbf{c}_0, \mathbf{c}_1) = \text{KDMEnc}(\text{pk}, \mathbf{u})$. Then it lets $\boldsymbol{\rho}_{\text{Bob}} = \lfloor \mathbf{c}_0 + s_{\text{Bob}} \cdot \mathbf{c}_1 \rfloor_p$, samples $\mathbf{d}_0 \leftarrow \mathcal{R}_p^n$ and samples a uniformly random $\mathbf{d}_1 \in \mathcal{R}_p^n$ subject to $\boldsymbol{\beta} = \lfloor \mathbf{d}_0 \star \mathbf{u} + \mathbf{d}_1 \star \boldsymbol{\rho}_{\text{Bob}} \rfloor_m$. This is possible since, by applying Lemma 5, sampling such \mathbf{d}_1 boils down to finding one single \mathbf{x} such that $\mathbf{y} = \mathbf{d}_0 \star \mathbf{u} + \mathbf{x}_1 \star \boldsymbol{\rho}_{\text{Bob}} \pmod q$ for some fixed $\mathbf{y} \in \mathcal{R}_q^n$, which can be done since all entries in $\boldsymbol{\rho}_{\text{Bob}}$ are invertible with probability at least $2^{-\lambda}$.

To argue that the distribution of the values outputted by \mathcal{S} are computationally indistinguishable from those in the real execution, we consider a hybrid distribution as follows:

Hybrid H₁. The execution is as in the ideal world, but the functionality \mathcal{F}_{OLE} is modified so that \mathcal{S} can choose the output $\boldsymbol{\beta}$. This way, instead of getting $\boldsymbol{\beta}$ and *then* sampling $(\mathbf{d}_0, \mathbf{d}_1)$ so that $\boldsymbol{\beta} = \lfloor \mathbf{d}_0 \star \mathbf{u} + \mathbf{d}_1 \star \boldsymbol{\rho}_{\text{Bob}} \rfloor_m$, the simulator samples $(\mathbf{d}_0, \mathbf{d}_1)$ completely at random and then defines $\boldsymbol{\beta} := \lfloor \mathbf{d}_0 \star \mathbf{u} + \mathbf{d}_1 \star \boldsymbol{\rho}_{\text{Bob}} \rfloor_m$, and sends this to the functionality \mathcal{F}_{OLE} .

Hybrid H₂. The execution is as in the hybrid H₁, but instead of sampling $(\mathbf{d}_0, \mathbf{d}_1)$ uniformly at random, these are obtained as encryptions of the input \mathbf{v} from \mathcal{Z} for the real-world $\mathcal{P}_{\text{Alice}}$.

Claim. The ideal execution and the hybrid H₁ are statistically indistinguishable.

This follows from Lemma 5 and from the invertibility of all the entries in $\boldsymbol{\rho}_{\text{Bob}}$, since these imply that the function $\lfloor \cdot \rfloor_m$ is a regular function and as a result one can either sample its output $\boldsymbol{\beta}$ and then sample a uniform input $\mathbf{d}_0 \star \mathbf{u} + \mathbf{d}_1 \star \boldsymbol{\rho}_{\text{Bob}}$ that maps to $\boldsymbol{\beta}$, which corresponds to the ideal world, or one can sample the input uniformly first and then compute the output, which corresponds to the hybrid H₁.

Claim. The hybrids H₁ and H₂ are computationally indistinguishable.

This is based on the security of the RLWE problem. We define an adversary \mathcal{A}' playing the CPA game for the LPR encryption scheme, which is defined as follows: \mathcal{A}' receives a public key pk from the challenger and it then replies with some message \mathbf{v} , the challenger then tosses an internal coin and returns $(\mathbf{d}_0, \mathbf{d}_1)$ to \mathcal{A}' where the pair is either an encryption of \mathbf{v} or a uniformly random pair. The goal of \mathcal{A}' is to guess the internal bit from the challenger.

Our adversary \mathcal{A}' proceeds by playing an honest $\mathcal{P}_{\text{Alice}}$ and the simulator \mathcal{S} above, using the public key received from the challenger. Upon receiving $\mathcal{P}_{\text{Alice}}$'s input \mathbf{v} from \mathcal{Z} , \mathcal{A}' sends \mathbf{v} to the challenger and receives $(\mathbf{d}_0^*, \mathbf{d}_1^*)$. \mathcal{A}' plays the protocol, but uses $(\mathbf{d}_0^*, \mathbf{d}_1^*)$ in place of $(\mathbf{d}_0, \mathbf{d}_1)$.

We notice the following. If $(\mathbf{d}_0^*, \mathbf{d}_1^*)$ is an encryption of \mathbf{v} , then the execution corresponds to the hybrid H₂. On the other hand, if $(\mathbf{d}_0^*, \mathbf{d}_1^*)$ is uniform, then this is precisely the hybrid H₁. Therefore, the advantage of \mathcal{Z} is upper bounded by the advantage of \mathcal{A}' , which is negligible from the security of the LPR encryption scheme.

Claim. The real execution and the hybrid H₂ are statistically indistinguishable.

Our first observation is that the intermediate values from the setup phase $(s_{\text{Bob}}, b_{\text{Bob}}, a)$ and the intermediate messages $(\mathbf{d}_0, \mathbf{d}_1)$ follow the exact same distribution in both executions, so the only potential distinguishing point is the output pair $(\boldsymbol{\alpha}, \boldsymbol{\beta})$: In the hybrid H₂ it holds that $\boldsymbol{\beta} = \lfloor \mathbf{d}_0 \star \mathbf{u} + \mathbf{d}_1 \star \boldsymbol{\rho}_{\text{Bob}} \rfloor_m$ and $\boldsymbol{\alpha} = \mathbf{u} \star \mathbf{v} - \boldsymbol{\beta}$, whereas in the real world $\boldsymbol{\alpha}$ is defined as $\boldsymbol{\alpha} = \lfloor \mathbf{d}_1 \cdot \boldsymbol{\rho}_{\text{Alice}} \rfloor_m$. It suffices to show then that $\mathbf{u} \star \mathbf{v} = \boldsymbol{\alpha} + \boldsymbol{\beta}$ holds in the real world as well. This holds with probability at least $1 - 2^{-\kappa}$, as shown in Proposition 1.

This concludes the proof of the claim, and with it, the proof of the theorem. \square

Theorem 11 (Theorem 2 restated). *Assume that $3 \cdot 2^{\kappa+1} \cdot n \cdot \tau \cdot (mN)^2 \cdot B_{\text{err}} \cdot B_{\text{sk}} \leq p \leq \frac{q}{3 \cdot 2^{\kappa+1} \cdot n \cdot \tau \cdot N^2 \cdot B_{\text{err}} \cdot B_{\text{sk}}}$. Protocol $\Pi_{\text{OLE-pk}}$ realizes functionality \mathcal{F}_{OLE} under the RLWE assumption.*

Proof. We define a simulator \mathcal{S} that interacts with the environment \mathcal{Z} as follows.

Corrupt Bob: The simulator emulates an honest party $\mathcal{P}_{\text{Alice}}$, and it also emulates the PKI resource by sampling $a \leftarrow \mathcal{R}_q$ and two key pairs $(s_{\text{Alice}}, (a, b_{\text{Alice}})) \leftarrow \text{Gen}(a)$ and $(s_{\text{Bob}}, (a, b_{\text{Bob}})) \leftarrow \text{Gen}(a)$, and sending the public key $\text{pk} = (a, b)$ to both $\mathcal{P}_{\text{Alice}}$ and \mathcal{P}_{Bob} , s_{Alice} to $\mathcal{P}_{\text{Alice}}$ and s_{Bob} to \mathcal{P}_{Bob} . After this setup phase the emulated $\mathcal{P}_{\text{Alice}}$ receives $(\mathbf{c}_0, \mathbf{c}_1)$ from \mathcal{P}_{Bob} , and she runs the zero knowledge argument for $\mathcal{R}_{\text{Bob}}^{\text{pk}}$ honestly and, if the proof succeeds, \mathcal{S} uses the knowledge on the secret key $s = s_{\text{Alice}} + s_{\text{Bob}}$ to decrypt a message \mathbf{u} .

The emulated $\mathcal{P}_{\text{Alice}}$ computes $(\mathbf{d}_0, \mathbf{d}_1) = \text{Enc}(\text{pk}, 0)$ and sends the input \mathbf{u} and output β to the functionality \mathcal{F}_{OLE} , where $\beta = \lfloor \mathbf{d}_0 \star \mathbf{u} + \mathbf{d}_1 \star \rho_{\text{Bob}} \rfloor_m \bmod m$. Then it sends $(\mathbf{d}_0, \mathbf{d}_1)$ to \mathcal{P}_{Bob} and then runs the zero knowledge argument for $\mathcal{R}_{\text{Alice}}^{\text{pk}}$ honestly, which she can do since she knows the witness.

To argue that the real and the ideal world are indistinguishable, it is useful to consider the following hybrid:

Hybrid H. The execution is as in the ideal world, except that the actual input \mathbf{v} from real-world $\mathcal{P}_{\text{Alice}}$ is used to define $(\mathbf{d}_0, \mathbf{d}_1)$, and the zero knowledge argument for $\mathcal{R}_{\text{Alice}}^{\text{pk}}$ uses this input.

Claim. The real execution is statistically indistinguishable from the hybrid H.

Begin by noticing that there is no difference between the intermediate messages in these two executions. The only potential difference originates in the input/output relation: in the hybrid it is the case that $\alpha + \beta = \mathbf{u} \star \mathbf{v}$, so we only need to show that this is the case too in the real execution.

To see this, first observe that the extractor \mathcal{E} from the ZKA for $\mathcal{R}_{\text{Bob}}^{\text{pk}}$ can interact with \mathcal{Z} to obtain a witness for $\mathcal{R}_{\text{Bob}}^{\text{pk}}$, i.e. $w = (\mathbf{w}, \mathbf{e}_0, \mathbf{e}_1, \mathbf{u})$ such that $(\mathbf{c}_0, \mathbf{c}_1) = (b \cdot \mathbf{w} + \mathbf{e}_0, (q/p) \cdot \mathbf{u} - a \cdot \mathbf{w} + \mathbf{e}_1)$ with $\|\mathbf{u}\|_\infty \leq \tau \cdot m$, $\|\mathbf{w}\|_\infty \leq \tau \cdot B_{\text{sk}}$, $\|\mathbf{e}_0\|_\infty \leq \tau \cdot B_{\text{err}}$ and $\|\mathbf{e}_1\|_\infty \leq \tau \cdot B_{\text{err}}$. What the simulator computes, on the other hand, is $\mathbf{c}_0 + s \cdot \mathbf{c}_1 = (q/p) \cdot s \cdot \mathbf{u} + \mathbf{e}$, where $\mathbf{e} = \mathbf{e}_0 + e \cdot \mathbf{w} + s \cdot \mathbf{e}_1$, and then rounds to p . The bounds above imply that $\|\mathbf{e}\|_\infty \leq 3\tau N B_{\text{err}} B_{\text{sk}}$, which just needs to be below $q/2p$ to guarantee correct decryption. This is clearly implied by the bound in the theorem statement.

As a result, the value that the simulator decrypts is exactly the \mathbf{u} from the witness. At this point we can apply Proposition 2 with τ being the slack parameter from the ZKA to conclude that $\mathbf{u} \star \mathbf{v} = \alpha + \beta$ holds in the real world with probability at least $1 - 2^{-\kappa}$.

Claim. The hybrid H is computationally indistinguishable from the ideal world.

Here we notice that the only difference between these two scenarios is the message $(\mathbf{d}_0, \mathbf{d}_1)$ (and its corresponding ZKA). It suffices then to argue that these two are indistinguishable, which intuitively hold because of the security of the encryption scheme and the zero knowledge property of the ZKA for $\mathcal{R}_{\text{Alice}}^{\text{pk}}$.

In a bit more detail, consider an adversary \mathcal{A}' for the following game: \mathcal{A}' gets a public key pk and sends $\mathbf{v} \in \mathcal{R}_m^n$ to a challenger who samples a bit internally and returns $(\mathbf{d}_0, \mathbf{d}_1) = \text{Enc}(\text{pk}, 0)$ if the bit is 0, and $(\mathbf{d}_0, \mathbf{d}_1) = \text{Enc}(\text{pk}, \mathbf{v})$ if the bit is 1. This is essentially the semantic security of the LPR encryption scheme and its security can be proved based on RLWE [38]. Our adversary \mathcal{A}' is defined as follows: it runs a copy of \mathcal{Z} internally and interacts with by playing the honest $\mathcal{P}_{\text{Alice}}$ and the simulator \mathcal{S} . \mathcal{A}' gets pk from the challenger and uses this for the PKI setup. Notice that \mathcal{A}' does not know the secret key s that the challenger used, but it still can give \mathcal{P}_{Bob} a uniformly random secret key s_{Bob} , which implicitly defines a secret key s_{Alice} for $\mathcal{P}_{\text{Alice}}$ (that \mathcal{A}' does not know!).

The simulator then receives \mathbf{v} from \mathcal{Z} and \mathcal{A}' passes this value to the challenger, getting back the challenge $(\mathbf{d}_0, \mathbf{d}_1)$. \mathcal{S} uses this as the second message from $\mathcal{P}_{\text{Alice}}$ to \mathcal{P}_{Bob} . Now, the main issue with the interaction above is that $\mathcal{P}_{\text{Alice}}$ cannot prove $\mathcal{R}_{\text{Alice}}^{\text{sk}}$ now, since she does not know the witness. To this end, consider the simulator \mathcal{M} for the ZKA for $\mathcal{R}_{\text{Alice}}^{\text{sk}}$. \mathcal{A}' has the public parameters and the instance information from the interaction with \mathcal{Z} , so it can use \mathcal{M} to interact with \mathcal{Z} and pass the ZKA. If \mathcal{Z} claims the resulting interaction is from the ideal world, then \mathcal{A}' outputs 0, and if \mathcal{Z} claims the interaction is in the real world then \mathcal{A}' outputs 1.

We claim now that the adversary \mathcal{A}' above wins the game with an advantage that is negligibly close to the advantage with which \mathcal{Z} distinguishes H from the ideal world, which implies that the latter must be negligible, as the original claim stated. To see this, first notice that, for a fixed $(\mathbf{d}_0, \mathbf{d}_1)$, \mathcal{Z} cannot distinguish between interacting with \mathcal{M} and interacting with a prover who knows the witness. This holds by the zero knowledge property of the ZKA. As a result, the advantage of \mathcal{A}' is negligibly close to the difference between the probability of \mathcal{Z} outputting “ideal world” and the probability of \mathcal{Z} outputting “ H ” when the ZKA is computed correctly. Finally, given that the interaction of \mathcal{Z} with \mathcal{A}' corresponds to either the ideal world (for challenge 0) or the hybrid H (for challenge 1) when the ZKA is computed correctly, we conclude that this difference is precisely the distinguishing advantage of \mathcal{Z} , which concludes the proof of the claim and with it the proof of indistinguishability for a corrupt \mathcal{P}_{Bob} .

Corrupt Alice (sketch): The proof in this setting is similar to the one considered above and therefore we only provide a sketch. The simulator uses the same idea of encrypting a dummy input $\mathbf{u} = 0$ on behalf of the emulated honest \mathcal{P}_{Bob} , and running the ZKA for $\mathcal{R}_{\text{Bob}}^{\text{pk}}$ honestly. To prove indistinguishability of the ideal and real worlds, a similar hybrid as above is considered where this message uses the real \mathbf{u} .

Proving that the hybrid is indistinguishable from the real world follows along the same lines as above. However, proving that the hybrid is indistinguishable from the ideal world requires a small change, given that now \mathbf{u} is not encrypted, but KDM-encrypted. This, fortunately, is not a problem since KDM-encryptions are indistinguishable from proper encryptions, as shown in [18]. \square

Theorem 12 (Theorem 3 restated). *Assume that $m^2 \cdot B_{\text{err}} \cdot 2^{\kappa+1} \cdot n \cdot N^2 \leq p \leq \frac{q}{2^{\kappa+1} \cdot n \cdot N^2 \cdot B_{\text{sk}} \cdot B_{\text{err}}}$. Then protocol $\Pi_{\text{OLE-sk}}^{\text{passive}}$, which consists of protocol $\Pi_{\text{OLE-sk}}$ without the underlined steps, realizes functionality \mathcal{F}_{OLE} in the $\mathcal{F}_{\text{setup}}$ -hybrid model under the RLWE assumption.*

Proof. Let \mathcal{S} be a simulator interacting with the real-world adversary \mathcal{A} and the functionality \mathcal{F}_{OLE} . We begin by considering the case in which \mathcal{P}_{Bob} is passively corrupt. It turns out that the case in which $\mathcal{P}_{\text{Alice}}$ is corrupt is completely symmetric.

The simulator receives as input the corrupt party’s input $\mathbf{u} \in \mathcal{R}_m^n$, and it emulates an honest $\mathcal{P}_{\text{Alice}}$ with a dummy input. It also emulates the resource $\mathcal{F}_{\text{setup}}$ by sending random s_{Bob} and σ_{Bob} to \mathcal{P}_{Bob} from the proper distributions. \mathcal{S} begins by invoking the ideal functionality \mathcal{F}_{OLE} on input \mathbf{u} to get back $\beta_i \in \mathcal{R}_m$ for $i = 1, \dots, n$. Then \mathcal{S} samples $\mathbf{a}' \in \mathcal{R}_q^n$ and $\mathbf{d} \in \mathcal{R}_m^n$ uniformly at random, and chooses a uniformly random $\mathbf{a} \in \mathcal{R}_q^n$ such that $\beta = \left[\mathbf{u} \star \mathbf{d} - \mathbf{a}' \star \left(- \lfloor \mathbf{a} \cdot \sigma_{\text{Bob}} \rfloor_p \right) \right]_m$. This is possible since, by applying Lemma 5 twice, sampling such \mathbf{a} boils down to finding \mathbf{x} such that $\mathbf{y} = \mathbf{x} \cdot \sigma_{\text{Bob}} \bmod q$ for some fixed $\mathbf{y} \in \mathcal{R}_q^n$, which can be found since σ_{Bob} is invertible with probability at least $2^{-\lambda}$ from Lemma 4. Finally, \mathcal{S} emulates the protocol interaction by setting the initial public value \mathbf{a} , and it waits for \mathbf{c}_{Bob} from \mathcal{P}_{Bob} . At this point \mathcal{S} simulates the second message by sending \mathbf{d} to \mathcal{P}_{Bob} on behalf of the emulated honest party $\mathcal{P}_{\text{Alice}}$.

Now we argue that the simulation above is indistinguishable from a real-world execution, which amounts to showing that the view of the environment, defined as $(\mathbf{u}, \mathbf{v}, s_{\text{Bob}}, \sigma_{\text{Bob}}, \mathbf{a}, \mathbf{a}', \mathbf{c}, \mathbf{d}, \alpha, \beta)$, is indistinguishable in both executions. To this end, we consider an intermediate hybrid H which is defined as follows:

Hybrid H. \mathbf{d} , instead of being uniformly random, it is sampled as in the real execution:

$$\mathbf{d} = \left(\frac{p}{m}\right) \cdot \mathbf{v} + (\mathbf{a}' \cdot s_{\text{Alice}} + e_{\text{Alice}}) \pmod{m}.$$

Claim. The hybrid H and the real world executions are statistically indistinguishable.

To see this, first notice that in the hybrid above α and β are uniformly random subject to $\mathbf{u} \star \mathbf{v} = \alpha + \beta$. Also, \mathbf{a}' , s_{Bob} and σ_{Bob} are uniformly random, \mathbf{c} is defined as $\left(\frac{q}{p}\right) \cdot \mathbf{u} + (\mathbf{a} \cdot s_{\text{Bob}} + e_{\text{Bob}}) \pmod{q}$, \mathbf{d} equals $\left(\frac{p}{m}\right) \cdot \mathbf{v} + (\mathbf{a}' \cdot s_{\text{Alice}} + e_{\text{Alice}}) \pmod{m}$ (for secrets s_{Alice} and e_{Alice} that \mathcal{Z} does not know) and \mathbf{a} is uniformly random subject to $\beta = \left[\mathbf{u} \star \mathbf{d} - \mathbf{a}' \cdot \left(- \lfloor \mathbf{a} \cdot \sigma_{\text{Bob}} \rfloor_p \right) \right]_m$.

Our goal is to show that the distribution above is the same distribution as in the real execution. To this end, first notice that in the real execution s_{Bob} , σ_{Bob} and \mathbf{a}' are also uniformly random. Also, \mathbf{c} and \mathbf{d} are computed as in the hybrid. Furthermore, \mathbf{a} is uniformly random and β is defined as $\beta = \left[\mathbf{u} \star \mathbf{d} - \mathbf{a}' \cdot \left(- \lfloor \mathbf{a} \cdot \sigma_{\text{Bob}} \rfloor \right) \right]_m$, which implies that β is uniformly random from Lemma 5. This is the same as choosing first β uniformly at random and then sampling \mathbf{a} conditioned on the equation above, which is what happens in the hybrid. Given the above, it remains to argue that $\mathbf{u} \star \mathbf{v} = \alpha + \beta$ in the real world with overwhelming probability, which is precisely what Proposition 3 shows.

We conclude then from the above that the distributions in the real world and in the hybrid are identical. Hence, to finish with the proof of the theorem, it suffices to show that the hybrid and the ideal world are indistinguishable.

Claim. The hybrid H and the ideal world executions are computationally indistinguishable.

Begin by noticing that the only difference between these distributions is the choice of \mathbf{d} : In the ideal execution it is completely uniform, but in the hybrid it is sampled as $\left(\frac{p}{m}\right) \cdot \mathbf{v} + (\mathbf{a}' \cdot s_{\text{Alice}} + e_{\text{Alice}}) \pmod{m}$. As a result, \mathcal{Z} distinguishes H from the ideal world if and only if it distinguishes \mathbf{d} . Such \mathcal{Z} would imply an adversary \mathcal{A}' for the RLWE game, defined as follows: This adversary plays an honest $\mathcal{P}_{\text{Alice}}$ and the simulator \mathcal{S} . On input a sample $(\mathbf{a}^*, \mathbf{b}^*) \in (\mathcal{R}_q^n)^2$ for the RLWE game, \mathcal{S} above is invoked with \mathbf{a}^* and $\mathbf{d}^* = \left(\frac{p}{m}\right) \cdot \mathbf{v} + \mathbf{b}^*$ in place of \mathbf{a}' and \mathbf{d} , respectively, where \mathbf{v} is the input that $\mathcal{P}_{\text{Alice}}$ received from \mathcal{Z} .

We see that if the sample is an RLWE sample, i.e. $\mathbf{b}^* = \mathbf{a}^* \cdot s + e$ for some s , then the distribution generated by \mathcal{A}' is identical to the distribution in the hybrid H, where $\mathcal{P}_{\text{Alice}}$ gets s as s_{Alice} . On the other hand, if \mathbf{b}^* is uniformly random, then the distribution generated by \mathcal{A}' is identical to the distribution in the ideal world. Therefore, if \mathcal{Z} claims the execution is in the ideal world, \mathcal{A}' concludes that $(\mathbf{a}^*, \mathbf{b}^*)$ is a uniform sample, and otherwise it concludes it is an RLWE sample. \mathcal{A}' would break the $\text{RLWE}_{n, \mathcal{D}}$ game with essentially the same distinguishing advantage as \mathcal{Z} 's, which contradicts the security of $\text{RLWE}_{n, \mathcal{D}}$. \square

Theorem 13 (Theorem 4 restated). *Assume that $2^{\kappa+1} \cdot n \cdot \tau \cdot (mN)^2 \cdot B_{\text{err}} \leq p \leq \frac{q}{2^{\kappa+1} \cdot n \cdot \tau \cdot N^2 \cdot B_{\text{err}} \cdot B_{\text{sk}}}$. Then protocol $\Pi_{\text{OLE-sk}}$ realizes functionality \mathcal{F}_{OLE} in the $\mathcal{F}_{\text{setup}}$ -hybrid model under the RLWE assumption.*

Proof. As usual, we define a simulator \mathcal{S} that interacts with the real-world adversary \mathcal{A} and the functionality \mathcal{F}_{OLE} . As stated before, we only consider the case in which \mathcal{P}_{Bob} is actively corrupt, since the analysis for $\mathcal{P}_{\text{Alice}}$ is similar.

The simulator emulates an honest $\mathcal{P}_{\text{Alice}}$ with dummy input, and it also emulates the resource $\mathcal{F}_{\text{setup}}$ honestly which distributes $(s_{\text{Bob}}, \sigma_{\text{Bob}}, r_{\text{Bob}}, c_{\text{Alice}}, c_{\text{Bob}})$ to \mathcal{P}_{Bob} . The emulated $\mathcal{P}_{\text{Alice}}$ also interacts with \mathcal{P}_{Bob} to sample \mathbf{a} and \mathbf{a}' .

Upon receiving \mathbf{c} from \mathcal{P}_{Bob} , the emulated $\mathcal{P}_{\text{Alice}}$ engages with \mathcal{P}_{Bob} in the zero knowledge argument for $\mathcal{R}_{\text{Bob}}^{\text{sk}}$, and if the emulated $\mathcal{P}_{\text{Alice}}$ accepts, \mathcal{S} uses defines \mathbf{u} as the quotient of the (componentwise) division between $\mathbf{c} - \mathbf{a} \cdot s_{\text{Bob}}$ and q/p . Then \mathcal{S} defines \mathbf{d} as an honest $\mathcal{P}_{\text{Alice}}$ would do, with dummy input $\mathbf{v} = 0$, and sends (\mathbf{u}, β) to the functionality \mathcal{F}_{OLE} , where β is

defined as $\left[\mathbf{u} \star \mathbf{d} + \mathbf{a}' \star [\mathbf{a} \cdot \sigma_{\text{Bob}}]_p \right]_m \bmod m$. $\mathcal{P}_{\text{Alice}}$ then sends \mathbf{d} to \mathcal{P}_{Bob} , and at this point the emulated $\mathcal{P}_{\text{Alice}}$ can play the zero knowledge argument for $\mathcal{R}_{\text{Alice}}^{\text{sk}}$ honestly since it knows the witness for the relation (recall that \mathcal{S} used a dummy input $\mathbf{v} = 0$ for $\mathcal{P}_{\text{Alice}}$).

We claim that the ideal and real world executions are indistinguishable to \mathcal{Z} . To this end, we consider the following hybrid:

Hybrid H. The execution is as in the ideal world, except that the actual input \mathbf{v} from real-world $\mathcal{P}_{\text{Alice}}$ is used to define \mathbf{d} , and the zero knowledge argument for $\mathcal{R}_{\text{Alice}}^{\text{sk}}$ uses this input.

Claim. The real execution is computationally indistinguishable from the hybrid H.

This argument is similar to the one applied in the first claim in the proof of Theorem 3. Intuitively, the fact that the actual input \mathbf{v} is used to define \mathbf{d} implies that from this message onwards the hybrid H and the real world look identical. Furthermore, if the first zero knowledge argument succeeds, then we know that \mathbf{c} is well formed, and at this point the proof becomes essentially identical to the one in the passive case. We now proceed with the details.

Let us begin by considering the extractor \mathcal{E} for the zero knowledge argument for $\mathcal{R}_{\text{Bob}}^{\text{sk}}$. Notice that \mathcal{P}_{Bob} convinces $\mathcal{P}_{\text{Alice}}$ with the same probability in both worlds since, up to that point, the two interactions are indistinguishable. If this probability is negligible then clearly H and the real execution would be indistinguishable, as the interaction would terminate here. Otherwise, the extractor \mathcal{E} would extract from \mathcal{P}_{Bob} a witness $w = (\mathbf{u}, \mathbf{e}_{\text{Bob}}, s'_{\text{Bob}}, r_{\text{Bob}})$ for the relation $\mathcal{R}_{\text{Bob}}^{\text{sk}}$, which means that $\mathbf{c} = \left(\frac{q}{p}\right) \cdot \mathbf{u} + (\mathbf{a} \cdot s'_{\text{Bob}} + \mathbf{e}_{\text{Bob}}) \bmod q$, where $\|\mathbf{u}\|_{\infty} \leq \tau \cdot m \in \mathcal{R}_m^n$, $\|\mathbf{e}_{\text{Bob}}\|_{\infty} \leq \tau \cdot B_{\text{err}}$, and $\text{Open}_{pk}(c_{\text{Bob}}, s'_{\text{Bob}}, r_{\text{Bob}}) = 1$, where c_{Bob} was the commitment produced in the setup phase. Furthermore, from the computationally binding property of the commitment scheme we see that, with overwhelming probability, $s_{\text{Bob}} = s'_{\text{Bob}}$, where s_{Bob} was the value distributed in the setup phase.

Now, notice that the real world and the hybrid H coincide in the second zero knowledge argument for the relation $\mathcal{R}_{\text{Alice}}^{\text{sk}}$, so \mathcal{Z} cannot distinguish up to this point either. Hence, to conclude the argument about the two worlds being indistinguishable, it remains to be shown then that the input/output relation is the same in both worlds. To see this, first observe that the \mathbf{u} extracted from the first zero knowledge argument is the same as the one extracted by \mathcal{S} in the world H, which follows directly from the uniqueness of the quotient and the fact that $\|\mathbf{e}_{\text{Bob}}\|_{\infty} \leq \tau \cdot B_{\text{err}} < q/p$. As a result, we can apply Proposition 4 to obtain that $\mathbf{u} \star \mathbf{v} = \alpha + \beta$ with overwhelming probability in the real world, which is precisely the relation in the hybrid H. With this we conclude then that H is indistinguishable from the real world.

Claim. The hybrid H is computationally indistinguishable from the ideal world.

Intuitively, this holds since the only difference between the hybrid H and the ideal world is the message \mathbf{d} and its zero knowledge argument: In H the actual input \mathbf{v} is used to construct \mathbf{d} , but in the real world a dummy input of 0 is used instead. These cannot be distinguished due to the security of RLWE and the zero knowledge property of the ZKA.

In a bit more detail, consider an adversary \mathcal{A}' for the following game: \mathcal{A}' gets \mathbf{d} from a challenger, where \mathbf{d} is either $\mathbf{a} \cdot s + \mathbf{e}$ (challenge 0) or $\left(\frac{q}{p}\right) \cdot \mathbf{v} + (\mathbf{a} \cdot s + \mathbf{e})$ (challenge 1) for some \mathbf{v} chosen by \mathcal{A}' and $s \leftarrow \mathcal{R}_q, \mathbf{e} \leftarrow \mathcal{D}$ that are kept secret. This is essentially the semantic security of the LPR encryption scheme and its security can be proved based on RLWE [38]. Our adversary \mathcal{A}' is defined as follows: it runs a copy of \mathcal{Z} internally, getting \mathbf{v} from \mathcal{Z} which is passed to the challenger. Furthermore, \mathcal{A}' interacts with \mathcal{Z} as the simulator \mathcal{S} , using the message \mathbf{d} received by the challenger as the second message from $\mathcal{P}_{\text{Alice}}$ to \mathcal{P}_{Bob} .

The main issue with the interaction above is that \mathcal{A}' cannot prove $\mathcal{R}_{\text{Alice}}^{\text{sk}}$ now, since it does not know the witness. To this end, consider the simulator \mathcal{M} for the ZKA for $\mathcal{R}_{\text{Alice}}^{\text{sk}}$. \mathcal{A}' has the public information $x = (\mathbf{c}, \mathbf{a}, B_{\text{err}}, B_{\text{sk}}, m, \tau, c_{\text{Bob}})$ from the interaction with \mathcal{Z} , so it can use \mathcal{M}

on input x to interact with \mathcal{Z} and pass the ZKA. If \mathcal{Z} claims the resulting interaction is from the ideal world, then \mathcal{A}' outputs 0, and if \mathcal{Z} claims the interaction is in the real world then \mathcal{A}' outputs 1.

We claim now that the adversary \mathcal{A}' above wins the game with an advantage that is negligibly close to the advantage with which \mathcal{Z} distinguishes \mathbf{H} from the ideal world. To see this, first notice that, for a fixed \mathbf{d} , \mathcal{Z} cannot distinguish between interacting with \mathcal{M} and interacting with a prover who knows the witness. This holds by the zero knowledge property of the ZKA. As a result, the advantage of \mathcal{A}' is negligibly close to the difference between the probability of \mathcal{Z} outputting “ideal world” and the probability of \mathcal{Z} outputting “ \mathbf{H} ” when the ZKA is computed correctly. Finally, given that the interaction of \mathcal{Z} with \mathcal{A}' corresponds to either the ideal world (for challenge 0) or the hybrid \mathbf{H} (for challenge 1) when the ZKA is computed correctly, we conclude that this difference is precisely the distinguishing advantage of \mathcal{Z} .

The above implies that the distinguishing advantage of \mathcal{Z} has to be negligible, which concludes the proof of the claim and with it the proof of the theorem. \square

Remark 1. For our protocol we used a proof of knowledge, but a careful analysis of the proof above shows that we only need the existence of a witness, and not its extractability. As a result, we may relax the conditions on the zero knowledge argument by not requiring a proof of knowledge, as the mere soundness from the ZKA would suffice. This in fact also applies to the proof of Theorem 2.

D Previous Works

OLE from additively homomorphic encryption (AHE): There is a standard approach to building OLE using linearly homomorphic encryption, where $\mathcal{P}_{\text{Alice}}$ sends $\text{Enc}(u)$ to \mathcal{P}_{Bob} , who multiplies this with his input v and adds it to $\text{Enc}(\beta)$ for a random $\beta \in \mathcal{R}_m$. For example, this is the method that is implicitly used in the BDOZ [12] and LowGear [36] protocols for actively secure multi-party computation. One drawback of this method, seen in [12], is that to obtain active security, \mathcal{P}_{Bob} needs to prove that he multiplied v correctly into the ciphertext sent by $\mathcal{P}_{\text{Alice}}$. This proof of correct multiplication is prohibitively more expensive than proofs of plaintext knowledge, since we do not know of any efficient way to amortize a large batch of them. The LowGear protocol [36] avoids the proof of correct multiplication (while still using a proof of plaintext knowledge from $\mathcal{P}_{\text{Alice}}$) by assuming an additional property of the RLWE encryption scheme called “enhanced CPA” security. This is implied by linear-only encryption, a non-falsifiable assumption used in some zero-knowledge constructions [13].

We evaluate this approach in the “AHE” row of Table 1, using parameters based on [36].

OLE from RLWE: Concurrently and independently to our work, an efficient instantiation of the AHE-based template described above is presented in [26], essentially using the LPR encryption scheme as described here. To achieve circuit privacy, the authors in [26] do not rely on traditional “noise drowning” techniques. Instead, the authors devise a more efficient method involving “quotient-and-rounding”. This method reduces the sizes of the ciphertexts and allows for cheaper arithmetic.

OLE from somewhat homomorphic encryption (SHE): Another approach is to use a somewhat (or partially) homomorphic encryption scheme that supports one multiplication, as well as addition, such as BGV [19]. This method has been used to create multiplication triples in many protocols in the SPDZ family [24,23]. To use this to create OLE, each party first sends an encryption of its input u or v , and then multiply the ciphertexts homomorphically. Next, one party sends an encryption of a random value, which is added to this before being decrypted

towards the other party with a distributed decryption protocol. This requires 3 ciphertexts to be sent in all, plus one additional \mathcal{R}_q element for the distributed decryption, where one party sends its “partial decryption” to the other party, who decrypts the result.¹³

We evaluate this approach in row “SHE” of Table 1, using SHE parameters from the Overdrive variant of SPDZ [36, Table 1].

OLE from noisy Reed-Solomon encodings (RS): Another approach is based on oblivious transfer and noisy encodings via Reed-Solomon codes [41,34,30]. Here, the protocol of Ghosh et al. [30] has the best concrete efficiency, and achieves active security almost for free on top of previous passive protocols using simple consistency check. The protocol has not been implemented, but according to estimates from [28], an optimistic choice of parameters for the underlying security assumption leads to a communication cost of 32 field elements per OLE. This still leads to a higher communication cost than our protocols. Note that although the other protocols, being based on RLWE encryption, will likely have similar computational costs, we cannot easily compare the computational efficiency of the RS protocol, since it has not been implemented.

Other OLE Protocols. There are several other ways of constructing OLE which are not presented in Table 1, which we now briefly discuss here. Recently, Rathee et al. implemented passively secure protocols for Beaver triple generation over rings using RLWE [42]. These are based on the AHE approach described above, except they also use a CRT optimization where the plaintext space is reduced by using several ciphertexts with different plaintext moduli. This optimization is better suited to their setting with multiplication over general rings mod 2^k , and does not seem to give a benefit for our setting of a large prime plaintext space.

As mentioned earlier, we can also use Paillier encryption to build OLE from linearly homomorphic encryption, and add active security using either zero-knowledge proofs [12] or a common reference string [21]. However, Paillier ciphertexts are very large (at least 4096 bits) and have a high computational overhead, since exponentiations are relatively much more costly than polynomial operations in RLWE. It may be advantageous to use Paillier when only a few OLEs are desired, but in the amortized setting it seems unlikely to be competitive. OLE can also be constructed from string oblivious transfer, with Gilboa’s method [32]. Using OT extension [33] this can be quite cheap computationally [35], but has a much higher communication cost that is quadratic in the field bit length, instead of linear for all the protocols in Table 1, and around an order of magnitude higher than our protocol.

Finally, Boyle et al. [16] combined homomorphic secret sharing with a PRG based on the hardness of solving multivariate quadratic equations, to produce a large batch of n Beaver triples or OLEs with $o(n)$ communication. This interesting approach clearly has much lower communication than our methods, but it only achieves such low communication when producing a very large number of triples (more than 2^{30}), so will not be suitable for many applications. Furthermore, its computational efficiency is much worse than our protocol.

E On utilizing other Zero-Knowledge Arguments

Both \mathcal{R}_{pk} and R_{sk} can also be proven using other zero-knowledge arguments, and we will now explain why we chose this specific approach.

On individual arguments. An alternative to the use of amortized arguments such as those used in our protocol is to use arguments for each linear relation separately, e.g. using [9]. For n such

¹³ This has slightly lower costs than the SPDZ protocol, since we have simplified the distributed decryption for the two-party OLE setting.

instances the overall communication will then at least be $O(nN)$ and also computation must now be performed for each of the n instances. Amortization allows to instead reduce computation and communication to net κ instances, thus reducing the overhead.

On other amortized proofs. The recent work of [6] extended previous MPC preprocessing to use the more efficient challenge spaces of [5]. This can most likely be applied in our setting too and would lead to an earlier point at which amortization outperforms individual arguments. We leave this as interesting future work.

On generic proofs. Generic argument systems such as [3] or [11] outperform our proofs in terms of communication for large enough instances due to their sublinear (in the proven circuit) communication. This low communication comes at the expense of higher computation on the prover side, thus potentially decreasing the throughput of our protocol. [3,11] can use amortization over multiple instances, but so far it has not been studied how these perform in the lattice setting in comparison to specialized amortization techniques.