MyOPE: Malicious securitY for Oblivious Polynomial Evaluation

Malika Izabachène[1], Anca Nitulescu[2], Paola de Perthuis[1,3,4], and David Pointcheval[3,4]

[1] Cosmian
[2] Protocol Labs
[3] DIENS, École normale supérieure, CNRS, PSL University, Paris, France
[4] INRIA, Paris, France

Abstract. Oblivious Polynomial Evaluation (OPE) schemes are interactive protocols between a sender with a private polynomial and a receiver with a private evaluation point where the receiver learns the evaluation of the polynomial in their point and no additional information. They are used in Private Set Intersection (PSI) protocols. We introduce MyOPE, a "short-sighted" polynomial evaluation scheme in the presence of malicious senders. In addition to strong privacy guarantees, MyOPE enforces honest sender behavior and consistency by adding verifiability to the calculations. The main tools are Verifiable Computation (VC) of inner products between committed vectors for honest behavior enforcement and Fully Homomorphic Encryption (FHE) for input privacy. While classical techniques in pairing-based settings allow generic succinct proofs for such evaluations, they require large prime order subgroups which highly impact the computation complexity, and prevent the use of FHE with practical parameters. MyOPE builds on generic secure encoding techniques for succinct commitments, that allow real-world FHE parameters and Residue Number System (RNS) optimizations, suitable for very high-degree polynomials.

Keywords: OPE · Verifiable Computation · Malicious Adversaries

1 Introduction

1.1 Oblivious Polynomial Evaluation

Secure Two-Party Computation (2PC). Secure two-party computation enables two parties to mutually run a protocol computing the execution of a function $f(\cdot,\cdot)$ on their private inputs x, y, and allowing the parties to learn the output $f(x, y)$ but nothing else about the inputs. Seminal results from the 80s, e.g. [Yao86], have shown that with 2PC it is possible to securely evaluate any boolean circuit. Since these first feasibility results, a long line of works focused on improving the efficiency of the computational and communication costs in 2PC protocols. Two approaches were followed in these efforts: (1) improving generic protocols that compute any boolean or arithmetic circuit and (2) designing tailored protocols for practical functions. The latter approach focuses on taking advantage of these functions' particular structure to gain efficiency. Some examples of such 2PC are schemes designed for search problems [HT10,Ver11], RSA key generation [Gil99], set intersection [JL09,HN12], or polynomial evaluation [NP99].

Malicious Security of 2PC. An important aspect to consider when designing 2PC schemes is the adversarial model. In the *semi-honest* (or passive) adversarial model, a.k.a. honest-but-curious, the corrupted parties follow the protocol, but try to learn more about the private inputs of other parties, so there is no impact on the correctness of computation results. On the other hand, in the *malicious* (active) adversarial model, corrupted parties can collaborate in any way and misbehave arbitrarily, without following the protocol description.

Oblivious Polynomial Evaluation (OPE). OPE is a particular case of 2PC where the function to be evaluated is a polynomial $\mathsf{f}(X)$ of fixed degree N secretly chosen by the sender. The receiver chooses the secret evaluation point m. The receiver obtains the value $\mathsf{f}(\mathsf{m})$ without

learning anything else about the polynomial f and without giving the sender any information about the point m.

OPE is an important building block for various 2PC schemes that generally require multiple executions of an OPE protocol for the same polynomial and different evaluation points, such as for Private Set Intersection (PSI), as discussed below. However, the standard definition of receiver privacy does not preclude the sender from cheating by using a polynomial of higher degree than expected or changing the polynomial between multiple executions. Therefore, extending the security to malicious senders is essential for such applications.

Unfortunately, with generic 2PC techniques in the malicious setting, some efficiency overheads incur, with the cut-and-choose technique or an expensive preprocessing to generate correlated randomness.

Private Set Intersection (PSI). The OPE functionality can lead to various interesting applications such as data mining [LP00], private set intersection (PSI), privacy-preserving keyword search [JL09], set membership (related to PSI), and RSA key generation [Gil99].

We now focus on the PSI well-studied specialized form of 2PC that allows two parties to jointly compute the intersection of their input sets, without revealing any other information about them (other than upper bounds on their sizes).

A recent line of works reduces interactions in *unbalanced* PSI schemes by using FHE as a tool [CLR17,CHLR18]. In an unbalanced PSI protocol the sender has a set of much larger size than the receiver, and also bigger storage and computational power. The real challenge in PSI is enforcing that the sender performs the correct computation, as it could deviate from the prescribed protocol and make the receiver compute an arbitrary intersection.

We thus propose an approach to build unbalanced PSI with both integrity and privacy guarantees against a *malicious sender*.

1.2 Related Work

While [NP99] proposed a first construction for OPE, it relies on a newly introduced intractability assumption: the noisy polynomial interpolation. Naor and Pinkas conjectured that it could be reduced to a more widely studied assumption, the polynomial reconstruction problem. Nevertheless, as shown in [BN00], this conjecture seems not to hold in general.

OPE Schemes with Active Security. Among recent OPE schemes, to the best of our knowledge, only [HL09,Haz18] are secure against malicious adversaries. However, [HL09] has at least 17 rounds of interaction and the parties send each other $\mathcal{O}(\lambda N)$ Paillier encryptions, where λ is the security parameter and N the degree of the polynomial. Also, their claimed efficiency holds only for sufficiently low degree polynomials. [Haz18] shows an OPE scheme for polynomial evaluation in the exponent of a DLog group using algebraic Pseudo-Random Functions (PRF). They focus on improving the computational efficiency of [HL09] by reducing the number of modular exponentiations, and removing the trusted setup requirement, while preserving the same number of rounds of interaction and communication complexity as in [HL09], and apply their scheme to private set membership 2PC. [PRTY20] gives malicious security for PSI with symmetric set sizes, but the communication is linear in the set sizes.

Verifiable Computation (VC). Introduced by [GGP10], VC schemes are cryptographic systems that enable checking the integrity of results from delegated computations. More recent works [FNP20,BCFK20] have improved the efficiency of VC schemes to work for computations over encrypted data. These schemes, however, require proving the entire FHE circuit evaluation which is very expensive. Moreover, they neither allow using practical parameters for the FHE scheme, nor speedups through classical optimizations such as Residue Number System (RNS).

Polynomial Commitments (PC). First introduced by [KZG10], PCs are commitments for polynomials of maximal degree N and coefficients in a field $\mathbb{F}$ that support an interactive argument of knowledge to prove the correct evaluation of a committed secret polynomial in a given point. However, while this ensures verifiability of the evaluation result, it does not protect the privacy of the evaluation point.

1.3 Our Contribution

In this work, we study the OPE functionality in the malicious sender setting. When then illustrate its use to Private Set Intersection (PSI) in the unbalanced-set setting: basically, the sender, who owns the larger set $\mathcal{X} = \{x_1, \ldots, x_N\}$, defines the polynomial $\mathsf{f} = \prod_{i=1}^{N}(X - x_i)$, and the receiver asks for evaluations $\mathsf{f}(y_j)$ for each element in $\mathcal{Y} = \{y_1, \ldots, y_K\}$ to detect the common roots.

More precisely, we introduce MyOPE, a scheme for verifiable oblivious evaluation of polynomials of high degree N that achieves $\mathcal{O}(N^{2/d})$ communication costs, where d can be freely chosen and optimized with respect to the ciphertext sizes. This sublinear communication improves on the state-of-the-art.

Verifiable Computation of Inner Product. Our first tool, is a proof of correct computation of an inner product between two vectors $\vec{u}$ and $\vec{v}$, with respect to their commitments U and $\bar{V}$ respectively. Vector commitments will use polynomial evaluations in a secret point s. We define $U = [\sum_i u_i s^i] = \sum_i u_i[s^i]$ while $\bar{V}$ will be in the reverse order: $\bar{V} = \sum_i v_i[s^{n-i}]$. The notation [.] is an informal representation of a secure encoding, that leads to a computationally binding commitment. The hiding property is achieved with an additional secret component and Schnorr-like proofs. It will additionally have bilinear properties, which lead to $U \times \bar{V} = \sum_{i,j}(u_i v_j)[s^{n+i-j}] = \langle \vec{u}, \vec{v} \rangle [s^n] + \sum_{i \neq j}(u_i v_j)[s^{n+i-j}]$. By showing $U \times \bar{V} - \alpha[s^n]$ has no contribution of $[s^n]$, one proves that $\langle \vec{u}, \vec{v} \rangle = \alpha$. Whereas the analysis will be performed on polynomials evaluated in a secret point s, the Schwartz-Zippel lemma [Sch80,Zip79] will lift the relations on the polynomials, as non-zero polynomials are unlikely evaluated to 0. But this assumes good properties for the algebraic structures, which are not always satisfied. The favorable situation is considered first, then more complex structures are discussed in the Appendix C.

Verifiable Polynomial Evaluation. Now, from a polynomial $\mathsf{f} = \sum \mathsf{f}_i X^i$, which can be encoded as a vector $\vec{v} = (\mathsf{f}_i)_i$, and thus committed as $\bar{V}$ (in reverse order), and an element m, which can be encoded as a vector $\vec{u} = (\mathsf{m}^i)_i$, and thus committed as U, $\mathsf{f}(m) = \langle \vec{u}, \vec{v} \rangle$, correctness can be proven as above with respect to the binding commitments U and $\bar{V}$. Again, efficiency will depend on the actual algebraic structures. For the reader's convenience, indices for vectors start at 0, to consider the constant monomial in polynomials.

Receiver Privacy: Fully Homomorphic Encryption. When both f and m are public, which easily allows getting confidence in U and $\bar{V}$, the above approach is convincing. When the receiver privacy is expected, with a private point m, one can use fully homomorphic encryption to generate the first vector: the receiver provides M as an encryption (under their own key) of m, and FHE allows the sender to compute M_i, the encryption of m^i for each i, thanks to the homomorphic properties.

If W is a commitment of the vector $\vec{w} = (\mathsf{M}_i)$, using the linear property of the encryption scheme, $\langle \vec{w}, \vec{v} \rangle$ is the encryption of $\mathsf{f}(\mathsf{m})$, which can be proven with respect to W and $\bar{V}$, as above. When convinced, the receiver can decrypt it to get $\mathsf{f}(\mathsf{m})$. But why should the receiver trust the sender to have correctly computed the M_i's as the encryptions of the successive powers of m and the commitment W correctly? Let us assume each M_i, in $\vec{w}$ committed in W, encrypts the plaintext m_i. One can use another verifiable inner product to check $\mathsf{m}_i = \mathsf{m}^i$: from a random

common public element n, chosen after the publication of W, and the vector $\vec{z} = (\mathsf{n}^i)_i$ publicly committed into $\bar{Z}$ (in reverse order), the receiver can verifiably compute the inner product $\langle \vec{w}, \vec{z} \rangle$ with respect to W and $\bar{Z}$. Since it is proven correct, it decrypts to $\sum \mathsf{m}_i \mathsf{n}^i$, while one would like it to be $\sum \mathsf{m}^i \mathsf{n}^i$. In case of equality, this means that polynomials $\sum \mathsf{m}_i X^i$ and $\sum \mathsf{m}^i X^i$ evaluate the same way on the random point n. By applying the Schwartz-Zippel lemma in the plaintext-space, this means that $\mathsf{m}_i = \mathsf{m}^i$ with overwhelming probability. This concludes the security analysis.

Unfortunately, to draw the above conclusion, we assumed the Schwartz-Zippel lemma could be applied in both the plaintext-space and the ciphertext-space. Furthermore, for effective application of FHE, additional constraints might be added to the ciphertext-space, such as Residue Number System (RNS) representation. All these questions will be addressed below, dealing with arbitrary rings.

Sender Privacy: Noise-Flooding and Hiding Commitments OPE also expects sender privacy, with no leakage about the polynomial f. However, the commitment U might leak some information, unless it guarantees the additional *hiding* property. Furthermore, the homomorphic evaluation $\mathsf{f}(\mathsf{m})$ in the ciphertexts may leak more than just the result, and possibly the evaluation steps, as the final noise in $\langle \vec{w}, \vec{v} \rangle$ leaks them. To avoid such a leakage, the sender can add extra super-polynomial noise to $\langle \vec{w}, \vec{v} \rangle$. This is the so-called *noise-flooding* technique [Gen09]. One will of course have to prove this does not impact the decrypted result, with a norm small enough.

Applications. As already discussed, MyOPE as any OPE protocol can be used for PSI, in order to determine the intersection of the sets $\mathcal{X} = \{x_1, \ldots, x_N\}$ owned by the sender/prover and $\mathcal{Y} = \{y_1, \ldots, y_K\}$ owned by the receiver/verifier: The sender generates the polynomial $\mathsf{f} = \prod_{i=1}^{N}(X - x_i)$, commits it, and the receiver will learn whether $\mathsf{f}(y_j) = 0$ for each of their y_j's, but nothing else, by multiple evaluations of the same (or randomized) polynomial on multiple inputs.

We believe that our techniques for *verifiable computation of inner product* and *verifiable OPE* are generic and of independent interest. The former can indeed also be applied for Symmetric Private Information Retrieval (SPIR), which allows a receiver to ask for a secret index i in the sender's set $\mathcal{X} = \{x_1, \ldots, x_N\}$, and the receiver then only learns x_i. To this aim, one can revise our above notations with $\vec{v} = \vec{x} = (x_1, \ldots, x_N)$, and $\vec{u} = (\delta_{i,j})_j$, with δ the Kronecker symbol: $\delta_{i,i} = 1$ and $\delta_{i,j} = 0$ for $j \neq i$. The vector $\vec{u}$ is again encrypted with FHE by the receiver, to keep it private, whereas $\vec{v}$ is owned by the sender. More details are provided in appendix E.

Efficiency in Practice. With the Fan-Vercauteren FHE scheme [FV12], from the plaintext ring $\mathcal{R}_t = \mathbb{Z}_t[X]/\mathsf{r}(X)$, where $\mathsf{r}(X) = X^n + 1$, into the ciphertext ring $\mathcal{R}_q = \mathbb{Z}_q[X]/\mathsf{r}(X)$, the core parameters are the integers n, q, and t. For the PSI application, one needs to encode elements from $\mathcal{X}$ and $\mathcal{Y}$ into $\mathcal{R}_t$. Since $n > 128$, it will be enough to use $t = 3$: each ciphertext is $2n \log_2 q$ bits long, and the underlying plaintexts $n \log_2 t \approx 1.58n$ bits long.

With a MyOPE instantiation for polynomials of degree $N = 2^{30}$, we set $n = 2^{14}$ which leads to q over less than 512 bits to obtain appropriate semantic security for the FHE and decryption correctness, even with noise-flooding. To exploit RNS optimizations [BEHZ16], as proposed in SEAL[5], q can be the product of 10 primes on less than 60 bits each.

Then, the size of the FHE ciphertexts to be sent is less than 200 MBytes. Our proof of the sender's honest behavior consists of 100 KBytes, for a prime q and soundness of 2^{-128}. In case of use of RNS, with composite q, our commitments will have to be repeated many times for the soundness, and the proof then consists of 100MBytes (still much less than the FHE ciphertexts).

[5] https://github.com/microsoft/SEAL

2 Preliminaries

Let us recall the generic definitions of verifiable computation to illustrate our verifiable inner-product argument and oblivious polynomial evaluation, with fully homomorphic encryption. First we will need compact binding encodings with verifiable commitments.

2.1 Verifiable Commitment

A major tool for efficient verifiable computation is verifiable commitment $\mathsf{Com} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Verify}, \mathsf{Open})$ for elements in space $\mathcal{X}$:

- $\mathsf{Com.Setup}(1^\lambda) \to (\mathsf{ck}, [\mathsf{vk}])$: Given the security parameter, output the commitment key ck and possibly a secret verification key vk in case of designated-verifier proof.
- $\mathsf{Com.Commit}(\mathsf{ck}, x) \to (c_x, w)$: Given the commitment key and an element $x \in \mathcal{X}$, output a commitment c_x and an opening value w.
- $\mathsf{Com.Verify}(\mathsf{ck}, [\mathsf{vk}], c) \to acc$: Given the commitment key, optionally the verification key, and a commitment c, accept (with $acc = 1$) or reject (with $acc = 0$) a commitment c.
- $\mathsf{Com.Open}(\mathsf{ck}, x, c, w) \to acc$: Given the commitment key, an element x, a commitment c, and the opening value w, accept (with $acc = 1$) or reject (with $acc = 0$) the commitment c for x.

The *correctness* property means that the Verify and Open algorithms accept when commitments have been honestly generated on inputs in the appropriate space $\mathcal{X}$. On the other hand, the *binding* property means that no adversary can make Verify accept on committed elements that open outside the appropriate space $\mathcal{X}$ nor make Open accept on two different values. Additionally, one may expect the *hiding* property which means that c_x does not reveal any information about x (at least computationally). We stress that the target space $\mathcal{X}$ is verified: an acceptable commitment necessarily encodes an element in $\mathcal{X}$

2.2 Verifiable Computation

Here we recall the notion of *verifiable computation* from [GGP10]. We adapt the definitions to fit the setting where we might have privacy of the inputs and outputs. A VC scheme $\mathsf{VC} = (\mathsf{VC.Setup}, \mathsf{VC.KeyGen}, \mathsf{VC.QueryGen}, \mathsf{VC.Compute}, \mathsf{VC.Verify}, \mathsf{VC.Decode})$ consists of the following algorithms:

- $\mathsf{VC.Setup}(1^\lambda) \to (\mathsf{PK}, \mathsf{SK})$: Given the security parameter λ, output a pair of keys that do not depend on any function. The public key PK will be provided as input to all the subsequent algorithms.
- $\mathsf{VC.KeyGen}(\mathsf{PK}, f) \to \mathsf{pk}_f$: Given the function f, output a public key pk_f for the function f.
- $\mathsf{VC.QueryGen}(\mathsf{PK}, \mathsf{pk}_f, x) \to \sigma_x$: Given the public key pk_f and the input x, encode the query x into σ_x, and output it.
- $\mathsf{VC.Compute}(\mathsf{PK}, f, \sigma_x) \to \sigma_y$: Given the function f and the encoded input, output an encoded version σ_y of the result y.
- $\mathsf{VC.Verify}(\mathsf{PK}, [\mathsf{SK}], \mathsf{pk}_f, \sigma_x, \sigma_y) \to acc$: Given the secret key SK, in case of designated-verifier scheme, the public key pk_f for function f, and the encoding σ_x of the query, accept (with $acc = 1$) or reject (with $acc = 0$) an output encoding σ_y.
- $\mathsf{VC.Decode}(\mathsf{SK}, \sigma_y) \to y$: Given the secret key SK for function f, and an output encoding σ_y, output the result y.

For concrete use, between a sender with function f inputs and a receiver with x queries, the Setup algorithm is first run by the receiver (in case of designated-verifier) or a trusted party, and the sender executes the KeyGen algorithm with their input function f. For specific evaluations, the receiver runs $\mathsf{QueryGen}$ on their input x to allow the sender to execute the $\mathsf{Compute}$ algorithm. Eventually, the receiver can verify and decode the result with algorithms Verify and Decode. The keys $(\mathsf{SK}, \mathsf{PK})$ will at least contain the parameters for the verifiable commitments (hence

$\mathsf{Exp}^{\mathsf{SND}}_{\Pi,\mathcal{A}}(\lambda)$:
$(\mathsf{PK},\mathsf{SK}) \leftarrow \mathsf{Setup}(1^\lambda)$; $(f,x,\mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{PK})$
$\mathsf{pk}_f \leftarrow \mathsf{KeyGen}(\mathsf{PK},f)$; $\sigma_x \leftarrow \mathsf{QueryGen}(\mathsf{PK},\mathsf{pk}_f,x)$; $\sigma_y \leftarrow \mathcal{A}_2(\mathsf{PK},\mathsf{pk}_f,\sigma_x,\mathsf{st})$;
if $\mathsf{Verify}(\mathsf{PK},[\mathsf{SK}],\mathsf{pk}_f,\sigma_x,\sigma_y)=1$ and $\mathsf{Decode}(\mathsf{SK},\sigma_y)\neq f(x)$, then return 1
else return 0.

$\mathsf{Exp}^{\mathsf{R\text{-}Privacy}}_{\Pi,\mathcal{A}}(\lambda)$:
$b \xleftarrow{\$} \{0,1\}$; $(\mathsf{PK},\mathsf{SK}) \leftarrow \mathsf{Setup}(1^\lambda)$; $(f,x_0,x_1,\mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{PK})$
$\mathsf{pk}_f \leftarrow \mathsf{KeyGen}(\mathsf{PK},f)$; $\sigma_b \leftarrow \mathsf{QueryGen}(\mathsf{PK},\mathsf{pk}_f,x_b)$; $b' \leftarrow \mathcal{A}_2(\mathsf{PK},\sigma_b,\mathsf{st})$
if $f(x_0)\neq f(x_1)$, then return $\perp$ else return $(b=b')$

$\mathsf{Exp}^{\mathsf{S\text{-}Privacy}}_{\Pi,\mathcal{A}}(\lambda)$:
$b \xleftarrow{\$} \{0,1\}$; $(\mathsf{PK},\mathsf{SK}) \leftarrow \mathsf{Setup}(1^\lambda)$; $(f_0,f_1,\mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{PK})$
$\mathsf{pk}_f \leftarrow \mathsf{KeyGen}(\mathsf{PK},f_b)$; $b' \leftarrow \mathcal{A}_2^{\mathsf{CO}(.)}(\mathsf{PK},\mathsf{SK},\mathsf{pk}_f,\mathsf{st})$
if f_0 or f_1 invalid functions, if some σ_x asked to CO decodes to an x such that $f_0(x)\neq f_1(x)$, then return $\perp$ else return $(b=b')$

$\mathsf{CO}(\sigma_x)$: return $\sigma_y \leftarrow \mathsf{Compute}(\mathsf{PK},f_b,\sigma_x)$

Fig. 1. Security Games

the possible need of SK to open them, in case of designated-verifier proofs) and encryption keys when privacy properties are required.

The *correctness* of a VC scheme requires that if one runs Compute on an honestly generated query encoding of x, after honest Setup and KeyGen executions for f, then the output must verify and its decoding should be $y = f(x)$.

Note also that when there is no privacy issue, QueryGen and Decode could be the identity function and the Verify could simply consist in the computation by the receiver itself. But in such a case, we expect the QueryGen/Verify/Decode to be much more *efficient* that the computation. Verifiable commitments will be useful. For privacy, an encryption scheme will be applied.

Soundness. The primary security goal of VC is the *soundness* of the proof, which guarantees the receiver of correct computation, even in front of a malicious sender, once Setup and KeyGen have been run honestly.

Definition 1 (Soundness (SND)). *Let Π be an instance of our VC protocol and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ a two-stage adversary. Protocol Π is* SND*-secure if the advantage $\mathsf{Adv}^{\mathsf{SND}}_{\Pi,\mathcal{A}}(\lambda) = \Pr[1 \leftarrow \mathsf{Exp}^{\mathsf{SND}}_{\Pi,\mathcal{A}}(1^\lambda)]$ is negligible.*

However, one may also expect some privacy properties which are now defined.

Receiver Privacy. This notion ensures that the input x of the receiver remains hidden during the protocol execution, for an honestly generated pk_f.

Definition 2 (Receiver Privacy (R-Privacy)). *Let Π be an instance of our VC protocol and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ a two-stage adversary, where $\mathcal{A}_2$ has adaptive access to the* Compute*-oracle on legitimate queries only. Protocol Π is* R-Privacy*-secure if the advantage $\mathsf{Adv}^{\mathsf{R\text{-}Privacy}}_{\Pi,\mathcal{A}}(\lambda) = \Pr[1 \leftarrow \mathsf{Exp}^{\mathsf{R\text{-}Privacy}}_{\Pi,\mathcal{A}}(1^\lambda)]$ is negligible.*

Sender Privacy. This notion ensures that the function f of the sender remains hidden during the protocol execution, for adaptive legitimate requests by the receiver. We indeed exclude Compute-queries that trivially help to distinguish between two functions.

Definition 3 (Sender Privacy (S-Privacy)). *Let Π be an instance of our VC protocol and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ a two-stage adversary, where $\mathcal{A}_2$ has adaptive access to the* Compute*-oracle on legitimate queries only. Protocol Π is* S-Privacy*-secure if the advantage $\mathsf{Adv}^{\mathsf{S\text{-}Privacy}}_{\Pi,\mathcal{A}}(\lambda) = \Pr[1 \leftarrow \mathsf{Exp}^{\mathsf{S\text{-}Privacy}}_{\Pi,\mathcal{A}}(1^\lambda)]$ is negligible.*

2.3 Useful Functionalities

We now illustrate our VC definition with concrete functionalities that we will instantiate, with or without privacy.

Verifiable IPE: Inner Product Evaluation. We are first interested in the verifiable computation of inner products between committed vectors. As already noted, when there is no privacy issue, the commitment algorithm can simply be the identity function. But for efficiency reasons, we expect a more compact binding verifiable commitment to encode inputs. Then, from $\mathsf{pk}_x \leftarrow c_x$ with a commitment c_x of $\vec{x}$ (from $\mathsf{KeyGen}(\mathsf{PK}, \vec{x})$) and $\sigma_y \leftarrow (\vec{y}, c_y)$ with a commitment c_y of $\vec{y}$ (from $\mathsf{QueryGen}(\mathsf{PK}, \mathsf{pk}_x, \vec{y})$), the sender generates $\sigma_z = (z, \pi)$, where π is a proof of $z = \langle \vec{x}, \vec{y} \rangle$, when c_x and c_y are valid commitments of appropriate vectors $\vec{x}$ and $\vec{y}$ (in the correct vector spaces). One may additionally expect receiver and/or sender privacy, with private $\vec{y}$ and/or $\vec{x}$.

Verifiable OPE: Oblivious Polynomial Evaluation This is another case of VC, with a polynomial $\mathsf{f}(X) \in \mathsf{R}[X^n]$ as function, and $\mathsf{m} \in \mathsf{R}$ as evaluation point. Only $\mathsf{f}(\mathsf{m})$ is learnt by the receiver, and no other information leaks. It also ensures the computations were executed as pledged in the protocol by providing verifiability.

2.4 FHE: Fully Homomorphic Encryption

We will grant privacy-preserving VCs using Fully Homomorphic Encryption (FHE). FHE has been introduced in [Gen09]. This is particular case of classical public-public encryption, with a KeyGen algorithm that generates a key-pair $(\mathsf{pk}, \mathsf{sk})$ as well as encryption and decryption algorithms Enc and Dec, but with an additional Eval algorithm to operate on ciphertexts to build a ciphertext of $f((x_i)_i)$ from the ciphertexts of the x_i's. Since the initial construction, major improvements have been made, with now practical and efficient solutions. In this work we will use the Fan-Vercauteren (FV) FHE scheme [FV12] with $\mathcal{R}_t = \mathbb{Z}_t[X]/\mathsf{r}(X)$ as the plaintext message space and $\mathcal{R}_q \times \mathcal{R}_q$ as the ciphertext space, where $\mathcal{R}_q = \mathbb{Z}_q[X]/\mathsf{r}(X)$, with $\mathsf{r}(X) = X^n + 1$ for some well-chosen integer n (usually a power of 2). To enable some optimizations as the RNS representation used in SEAL, we will allow q with small prime factors on 60 bits. Detail about this FHE scheme is given in appendix A. Essentially, the public key is $\mathsf{pk} = (\mathsf{p}, \mathsf{p}') = (-(\mathsf{a} \cdot \mathsf{s} + \mathsf{e}), \mathsf{a}) \in \mathcal{R}_q^2$, for random polynomials $\mathsf{a}, \mathsf{s} \xleftarrow{\$} \mathcal{R}_q$ and a small noise e (that follows a centered Gaussian distribution in $\mathbb{Z}$), while the secret key is $\mathsf{sk} = \mathsf{s}$. To encrypt a message $\mathsf{m} \in \mathcal{R}_t$, one computes $(\mathsf{c}, \mathsf{c}') = (\mathsf{p} \cdot \mathsf{u} + \mathsf{e}_1 + \Delta \cdot \mathsf{m} \bmod q, \mathsf{p}' \cdot \mathsf{u} + \mathsf{e}_2 \bmod q)$ with small noises $\mathsf{u}, \mathsf{e}_1, \mathsf{e}_2$, where $\Delta = \lfloor q/t \rfloor$. One can see that with

$$\mathsf{d} = \mathsf{c} + \mathsf{c}' \cdot \mathsf{s} = \Delta \cdot \mathsf{m} - \mathsf{e} \cdot \mathsf{u} + \mathsf{e}_2 \cdot \mathsf{s} + \mathsf{e}_1 = \Delta \cdot \mathsf{m} + \mathsf{v} \bmod q$$

$\mathsf{m}' = \lfloor \mathsf{d}/\Delta \rceil = \mathsf{m} + \lfloor \mathsf{v}/\Delta \rceil \bmod t$ likelily leads to m, if the error term v is small enough (infinity norm $\|\mathsf{v}\|_\infty$ less than $\Delta/2$). We do not detail Eval, but for our analysis to hold, we will need the following property, in the particular case where $L = N + 1$ (when N will be the maximal degree of our polynomial in OPE), which can be guaranteed with appropriate choice of the parameters:

Definition 4 (**(L, d)-$\mathcal{R}_t$-Linear-Homomorphism**). *For any $\mathsf{a}_i, \mathsf{m}_i \in \mathcal{R}_t$, and some ciphertexts $(\mathsf{c}_i, \mathsf{c}'_i) \in \mathcal{R}_q^2$ of m_i obtained from a circuit of multiplicative depth at most d,*

$$\mathsf{Dec}\left(\mathsf{sk}, \sum_{i=1}^{L} \mathsf{a}_i \cdot (\mathsf{c}_i, \mathsf{c}'_i)\right) = \sum_{i=1}^{L} \mathsf{a}_i \cdot \mathsf{m}_i \in \mathcal{R}_t$$

2.5 Secure Encoding Schemes

Our main ingredient will be linear encodings introduced by [BCI+13,GGPR13] for succinct non-interactive arguments of knowledge (SNARKs). We provide a general definition over rings. An encoding scheme over a ring R consists in a tuple of algorithms:

- $(\mathsf{pk}, \mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^\lambda)$, a key generation algorithm that takes as input a security parameter and outputs public information pk, a secret key sk, and a verification key vk, that can be either public or private;
- $E \leftarrow \mathsf{E}_{\mathsf{sk}}(a)$, a (probabilistic) encoding algorithm mapping a ring element $a \in \mathsf{R}$ in the encoding space $\mathcal{E}$, using the secret key sk.

It should then satisfy a few properties:

- L-Linearly homomorphic, with an algorithm $\mathsf{Eval}_{\mathsf{pk}}$ that homomorphically combine encodings into the encoding of the same linear combination of the inputs;
- L-Quadratic root verification, with an algorithm $\mathsf{QCheck}_{\mathsf{vk}}$ that can check a quadratic relation between the encoded elements, just from the encodings;
- Image verification, with an algorithm $\mathsf{Verify}_{\mathsf{vk}}$ that check the validity of the encoding. This certifies the membership of the encoded element in the appropriate space.

According to the verification key that can be either public or private, the verification processes will be either public or private. The encoding is linearly-homomorphic, but for a *secure* encoding, one expects no one to be able to derive new valid encodings except from linear combinations: any new *valid* encoding E of some $a \in \mathsf{R}$ will necessarily satisfy $a = \sum_i c_i a_i$, for extractable elements $c_i \in \mathsf{R}$. Intuitively, when an encoding E passes the verification test, one can extract the linear combination of the given initial encodings.

The above properties will be enough for a binding commitment, but additional blinding factors will be required for hiding commitments, together with zero-knowledge proofs to keep the above verifications possible, without leaking more information.

In appendix B, we provide a more formal definition, with two illustrations: encodings over $\mathbb{Z}_q$, with q a prime large enough using the Knowledge of Exponent Assumption in a pairing-friendly setting. Such encodings just consist of three group elements (2 in $\mathbb{G}_1$ and 1 in $\mathbb{G}_2$). We then discuss the situation where q is the product of smaller primes. Then, the hardness of discrete logarithm does not hold anymore, but one can use linear-only encryption schemes, such as the Paillier encryption scheme [Pai99] with large RSA modulus $\mathcal{N}$. Such encodings just consist of two Paillier ciphertexts in $\mathbb{Z}_{\mathcal{N}^2}$ each.

In the body of the paper, for the sake of clarity, hiding commitments and zero-knowledge proofs will be ignored, as their computational and communication impacts are minimal, since they only deal with few scalars.

3 Commitments

A major contribution of this paper is the construction of commitments of multivariate polynomials over rings so that succinct proofs can later be described. This is in the same vein as in [FNP20], but the latter is only defined for secure encodings based in pairings, whereas we describe here the construction from any secure encodings. With a linear-only encryption scheme, they will not be publicly verifiable anymore, but this will be useful to build compact commitments of polynomials over $\mathbb{Z}_q$ in 2PC protocols, whatever the integer q (large prime or composite).

We provide here the intuition of our approach for polynomials in $\mathcal{R}_1 = \mathbb{Z}_q[X^{n-1}]$ (polynomials of degree at most $n-1$), while more polynomial spaces will be used in the global protocol. We stress again that commitments are specific to a space $\mathcal{X}$ and when valid they ensure the committed element actually relies in $\mathcal{X}$. The verifier first generates secure encodings $E_i \leftarrow \mathsf{E}_{\mathsf{sk}}(s^i)$, for $i \in [\![0; n-1]\!]$ and a random secret element $s \xleftarrow{\$} \mathbb{Z}_q^*$. Thanks to the linear-only extractability,

when a player generates a valid encoding E, being only given $(E_0, \ldots, E_{n-1})$, one can extract (c_i) such that E is an encoding of $c_0 + c_1 s + \ldots c_{n-1} s^{n-1}$ in $\mathbb{Z}_q$, and thus of the polynomial $\mathsf{c} = \sum_i c_i X^i$ in $\mathcal{R}_1$. The encoding E is thus a commitment of $\mathsf{c} \in \mathcal{R}_1$: the list of initial encodings E_i specifies a basis of the exact space $\mathcal{X}$ we target. Here, $\mathcal{R}_1$ is spanned by $(1, X, \ldots, X^{n-1})$ in $\mathbb{Z}_q$.

In addition, thanks to the quadratic verification on the encodings, if we have four polynomials u, v, m and r such that $\mathsf{m} = \mathsf{u} \cdot \mathsf{v} \bmod \mathsf{r}$, which means that $\mathsf{m} = \mathsf{u} \cdot \mathsf{v} + \mathsf{r} \cdot \mathsf{q}$ for some polynomial q, where all the polynomials are of degree at most $n-1$, we can check such a product: from valid commitments U and V of u and v, R and Q of r and q, respectively, and M of the polynomial m, all of degree at most $n-1$, as they are all simple encodings, $\mathsf{QCheck}_{\mathsf{vk}}(X_1 X_2 + X_3 X_4 - X_5, U, V, R, Q, M) = \mathsf{true}$ implies that $\mathsf{m}(s) = \mathsf{u}(s) \cdot \mathsf{v}(s) + \mathsf{r}(s) \cdot \mathsf{q}(s)$.

Under the Schwartz-Zippel lemma, if q is a large prime, the probability to have this equality in a random point $s \in \mathbb{Z}_q$ whereas $\mathsf{m} \neq \mathsf{u} \cdot \mathsf{v} + \mathsf{r} \cdot \mathsf{q}$ in $\mathbb{Z}_q[X]$ is bounded by $2n/q$, as the total degree of the relation is at most $2n$. Hence, the probability over s to have a false positive is bounded by $2n/q$. This is negligible in the large prime case but if we want to use RNS optimizations for the FHE scheme we need to take q a product of primes. Detail about this case is given in appendix C, along with a complete description of binding and hiding polynomial commitments, for univariate and bivariate polynomials, with multiple evaluation points when necessary.

4 Verifiable Inner Product

4.1 Verifiable Computation for Inner Product

Our main tool is verifiable computation of inner products, from commitments on vectors, in various structures. To this aim, we convert vectors in polynomials to commit them as explained above.

More concretely, let us consider vectors in a field $\mathbb{Z}_q$ (with a prime q). To commit such vectors, we will consider them as coefficients of a polynomial, and then commit the corresponding polynomials, as above. Let us consider $\mathsf{A} = (a_0, \ldots, a_N)$ and $\mathsf{B} = (b_0, \ldots, b_N)$ in $\mathbb{Z}_q^{N+1}$ (equivalent to $\mathcal{R}_3 = \mathbb{Z}_q[Y^N]$, as defined in appendix C), two vectors whose inner-product is equal to $c = \langle \mathsf{A}, \mathsf{B} \rangle$ in $\mathbb{Z}_q$. As explained in Section 1.3, the commitments $\bar{A}$ of A and B of B with secure encodings are $\bar{A} = \bar{\mathsf{a}}(s)$ and $B = \mathsf{b}(s)$ for the polynomials $\bar{\mathsf{a}}(Y) = \sum_{j=0}^{N} a_j Y^{N-j}$ and $\mathsf{b}(Y) = \sum_{j=0}^{N} b_j Y^j$ in $\mathcal{R}_3$. Note that coefficients of A are set into $\bar{\mathsf{a}}$ in a reversed order:

$$\bar{\mathsf{a}}(Y) \cdot \mathsf{b}(Y) = \sum_{i,j=0}^{N} a_i b_j \cdot Y^{N+j-i} = \sum_{j=0}^{N} a_j b_j Y^N + \sum_{0 \leq i \neq j \leq N} a_i b_j Y^{N+j-i}.$$

Let us define the polynomial $\mathsf{d}(Y) = \mathsf{a}(Y) \cdot \mathsf{b}(Y) - cY^N$ of degree at most $2N$. If c is correct, d is in the subspace $\mathcal{R}_4 = \mathbb{Z}_q[Y^{2N \setminus N}]$ (the polynomials of degree at most $2N$, without monomial of degree N). By publishing a commitment D of d, that is verifiably in $\mathcal{R}_4$, one can verify the above quadratic relation, using $\bar{A}$, B, c, and D, and get convinced of the inner product value c. The proof of correct computation of $c = \langle \mathsf{A}, \mathsf{B} \rangle$ with respect to the given commitments $\bar{A}$ and B just consists of $\pi = \{D\}$ (1 commitment only), and the verification consists in checking the validity of the commitments and one quadratic relation.

Verifiable Inner-Product Algorithms. More formally, one can define the verifiable computation of inner products (IPE) on vectors A, B in $\mathbb{Z}_q^{N+1}$ for result $c \in \mathbb{Z}_q$:

$\mathsf{IPE.Setup}(1^\lambda)$ generates the secure encodings on the acceptable bases for $\mathcal{R}_3$ and $\mathcal{R}_4$ in PK to allow the generation of commitments in these vector spaces. According to the encoding, verification will need SK or not;

$\mathsf{IPE.KeyGen}(\mathsf{PK}, \mathsf{A})$, from a vector A, outputs $\bar{A}$, a commitment of A (in the reverse order);

$\mathsf{IPE.QueryGen}(\mathsf{PK}, \bar{A}, \mathsf{B})$, from a vector B, outputs (B, B), where B is a commitment of B;

$\mathsf{IPE.Compute}(\mathsf{PK}, \mathsf{A}, (\mathsf{B}, B))$, from the two vectors A and B, outputs $c = \langle \mathsf{A}, \mathsf{B} \rangle$ and $\pi = \{D\}$ (where D is a commitment of d, as defined above);

$\mathsf{IPE.Verify}(\mathsf{PK}, [\mathsf{SK}], \bar{A}, B, (c, \pi))$, with $\pi = \{D\}$, checks the relation $\mathsf{d}(Y) = \mathsf{a}(Y) \cdot \mathsf{b}(Y) - cY^N$ from $\bar{A}$, B, D and c.

Since there is no privacy in this protocol, there is no need of $\mathsf{IPE.Decode}$.

Polynomial Evaluation. It can be turned into a polynomial evaluation $y = P(x)$, with one vector A containing the coefficients of P and the other vector B built from the powers of x, and the expected inner product being y.

Infinity Norm Evaluation. It also provides a setting to compute the L_2-norm $\|\mathsf{e}\|_2$, of $\mathsf{e} \in \mathbb{Z}_q[X^{n-1}]$, as $\|\mathsf{e}\|_2^2 = \langle \mathsf{E}, \mathsf{E} \rangle$, for the vector E of the polynomial's coefficients. This leads to an approximation of the infinity norm with $\|\mathsf{e}\|_\infty \leq \|\mathsf{e}\|_2 \leq \sqrt{n} \cdot \|\mathsf{e}\|_\infty \leq \sqrt{n} \cdot \|\mathsf{e}\|_2$. One just needs $E = \mathsf{e}(s)$ and $\bar{E} = \bar{\mathsf{e}}(s)$, where

$$\mathsf{e}(X) = \sum_0^{n-1} e_i X^i \qquad\qquad \bar{\mathsf{e}}(X) = \sum_0^{n-1} e_i X^{n-1-i} = X^{n-1} \cdot \mathsf{e}(1/X)$$

which can be verified with the existence, for a random challenge $\beta \xleftarrow{\$} \mathbb{Z}_q^*$, of polynomials e' and $\bar{\mathsf{e}}'$ that satisfy, with $e = \mathsf{e}(1/\beta)$,

$$\begin{aligned} \mathsf{e}(X) - e = \mathsf{e}(X) - \mathsf{e}(1/\beta) &= (X - 1/\beta) \cdot \mathsf{e}'(X) \\ \bar{\mathsf{e}}(X) - \beta^{n-1} \cdot e = \bar{\mathsf{e}}(X) - \beta^{n-1} \cdot \mathsf{e}(1/\beta) &= \bar{\mathsf{e}}(X) - \bar{\mathsf{e}}(\beta) = (X - \beta) \cdot \bar{\mathsf{e}}'(X). \end{aligned}$$

From the commitment E of e and the result $\|\mathsf{e}\|_2$ to be proven (or a commitment of it), the proof consists of the commitments $E' = \mathsf{e}'(s)$ and $\bar{E}' = \bar{\mathsf{e}}'(s)$, plus the inner-product proof (with the commitment D as above) with the additional commitment $\bar{E}$ and the scalar e. The validity of the proof requires the verification of the validity of the commitments and three quadratic relations. For strong privacy, one can first commit the scalar $\|\mathsf{e}\|_2^2$, prove the correct computation of this hidden value with the above approach, and then perform a zero-knowledge range proof for the committed value, to show it is of appropriate size.

4.2 Verifiable Computation for Inner Product with Privacy

If we now consider vectors $\mathsf{A} = (\mathsf{a}_0, \ldots, \mathsf{a}_N)$ and $\mathsf{B} = (\mathsf{b}_0, \ldots, \mathsf{b}_N)$ in $\mathcal{R}_t^{N+1}$, where $\mathcal{R}_t = \mathbb{Z}_t[X]/\mathsf{r}(X)$, they can be committed with bivariate polynomials in $\mathbb{Z}_t[X, Y]$, using secure encodings with monomials $s^i s'^j$: $\bar{A} = \bar{\mathsf{a}}(s, s')$ and $B = \mathsf{b}(s, s')$, where

$$\bar{\mathsf{a}}(X, Y) = \sum_{i=0}^{n-1} \sum_{j=0}^{N} a_{j,i} X^i Y^{N-j} \qquad\qquad \mathsf{b}(X, Y) = \sum_{i=0}^{n-1} \sum_{j=0}^{N} b_{j,i} X^i Y^j$$

One can get $\mathsf{p} = \langle \mathsf{A}, \mathsf{B} \rangle \in \mathcal{R}_t$. If one wants to keep vector B private, the latter can be encrypted with the FV FHE scheme, in $(\mathsf{c}_i, \mathsf{c}'_i) \in \mathcal{R}_q \times \mathcal{R}_q$, for $i = 0, \ldots, N$. Thanks to the linear-homomorphism of the FHE, $\mathsf{Dec}(\langle \mathsf{A}, \mathsf{C} \rangle, \langle \mathsf{A}, \mathsf{C}' \rangle) = \langle \mathsf{A}, \mathsf{B} \rangle$, where $\mathsf{C} = (\mathsf{c}_0, \ldots, \mathsf{c}_N)$ and $\mathsf{C}' = (\mathsf{c}'_0, \ldots, \mathsf{c}'_N)$ are in $\mathcal{R}_q^{N+1}$ (equivalent to $\mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$, as denoted in appendix C). One now needs the verifiability of $\langle \mathsf{A}, \mathsf{C} \rangle$ and $\langle \mathsf{A}, \mathsf{C}' \rangle$ in $\mathcal{R}_q$: we consider $\mathsf{A} = (\mathsf{a}_0, \ldots, \mathsf{a}_N) \in \mathcal{R}_t^{N+1}$ and $\mathsf{C} = (\mathsf{c}_0, \ldots, \mathsf{c}_N) \in \mathcal{R}_q^{N+1}$, and we want to compute $\mathsf{d} = \langle \mathsf{A}, \mathsf{C} \rangle \in \mathcal{R}_q$ and prove it. One can similarly operate to compute and prove $\mathsf{d}' = \langle \mathsf{A}, \mathsf{C}' \rangle$.

We set both polynomials in $\mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$,

$$\bar{\mathsf{a}}(X, Y) = \sum_{i=0}^{n-1} \sum_{j=0}^{N} a_{j,i} X^i Y^{N-j} \qquad\qquad \mathsf{c}(X, Y) = \sum_{i=0}^{n-1} \sum_{j=0}^{N} c_{j,i} X^i Y^j.$$

They are committed as $\bar{A} = \bar{\mathsf{a}}(s, s')$ and $C = \mathsf{c}(s, s')$. The result of the inner product $\mathsf{d} \in \mathcal{R}_q$ is committed into D. However, in $\mathbb{Z}_q[X]$, the result of the inner product is equal to $\mathsf{d} + \mathsf{q} \cdot \mathsf{r}$, where r is the public quotient polynomial in rings $\mathcal{R}_t$ and $\mathcal{R}_q$, and q is the quotient, committed into Q. We want to prove $\langle \mathsf{A}, \mathsf{C} \rangle = \mathsf{d} + \mathsf{q} \cdot \mathsf{r}$ in $\mathbb{Z}_q[X]$.

Then, for a random scalar $\sigma \in \mathbb{Z}_q$, one has the following relations, with $\bar{\mathsf{a}}'(Y) = \bar{\mathsf{a}}(\sigma, Y)$ and $\mathsf{c}'(Y) = \mathsf{c}(\sigma, Y)$, committed into $\bar{A}', C'$,

$$\bar{\mathsf{a}}(X, Y) - \bar{\mathsf{a}}'(Y) = (X - \sigma) \cdot \bar{\mathsf{a}}''(X, Y) \qquad \mathsf{c}(X, Y) - \mathsf{c}'(Y) = (X - \sigma) \cdot \mathsf{c}''(X, Y)$$

for some polynomials $\bar{\mathsf{a}}''$ and c'' that can be computed from $\bar{\mathsf{a}}, \bar{\mathsf{a}}', \mathsf{c}, \mathsf{c}'$ and committed into $\bar{A}'', C''$, as well as the polynomial $X - \sigma$, so the receiver can check the above quadratic relations. Then, we also have

$$\bar{\mathsf{a}}'(Y) \cdot \mathsf{c}'(Y) = \sum_{j=0}^{N} a'_j \cdot c'_j \cdot Y^N + \sum_{0 \le i \ne j \le N} a'_i \cdot c'_j \cdot Y^{N+j-i}$$

$$\text{and } \sum_{j=0}^{N} a'_j \cdot c'_j = \sum_{j=0}^{N} \mathsf{a}_j(\sigma) \cdot \mathsf{c}_j(\sigma) = \mathsf{d}(\sigma) + \mathsf{q}(\sigma) \cdot \mathsf{r}(\sigma)$$

Setting $\delta = \mathsf{d}(\sigma)$, $\phi = \mathsf{q}(\sigma)$ and $\rho = \mathsf{r}(\sigma)$, the values can be sent and checked with respect to d, Q, and r, as $\mathsf{q}(X) - \phi = (X - \sigma) \cdot \mathsf{q}'(X)$. If we additionally set $\mathsf{e}(Y) = \bar{\mathsf{a}}'(Y) \cdot \mathsf{c}'(Y) - (\delta + \phi \cdot \rho) \cdot Y^N$, committed in E, by proving it relies in $\mathcal{R}_4 = \mathbb{Z}_q[Y^{2N \setminus N}]$, this proves the result d of the inner product in $\mathcal{R}_q$.

The proof of correct computation of $\mathsf{d} = \langle \mathsf{A}, \mathsf{C} \rangle$ in $\mathcal{R}_q$, with respect to the given commitments $\bar{A}$ and C just consists of $\pi = \{Q, Q', \bar{A}', \bar{A}'', C', C'', E, \phi\}$ (7 commitments and a scalar), for publicly generated or computed σ, δ and ρ, and the verification consists in checking the validity of the commitments and four quadratic relations. The same is needed for $\mathsf{d}' = \langle \mathsf{A}, \mathsf{C}' \rangle$. By then decrypting $(\mathsf{d}, \mathsf{d}')$ one should get $\mathsf{p} = \langle \mathsf{A}, \mathsf{B} \rangle \in \mathcal{R}_t$.

Verifiable Inner-Product with Privacy Algorithms. More formally, we can define the verifiable computation of inner products with privacy (IPEwP) on vectors A, B in $\mathcal{R}_t^{N+1}$ for result $\mathsf{p} \in \mathcal{R}_t$, while keeping B private:

- $\mathsf{IPEwP.Setup}(1^\lambda)$ generates the parameters for the FV FHE from $\mathcal{R}_t$ to $\mathcal{R}_q$. The secure encodings on the acceptable bases for $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_4$ with the FHE encryption key are put in PK to allow encryption and evaluations on ciphertexts, as well as the generation of commitments in theses vector spaces. The verification key of the secure encodings (if needed) and the FHE decryption key are put in SK;
- $\mathsf{IPEwP.KeyGen}(\mathsf{PK}, \mathsf{A})$, from a vector A, outputs $\bar{A}$, a commitment of A (in the reverse order);
- $\mathsf{IPEwP.QueryGen}(\mathsf{PK}, \bar{A}, \mathsf{B})$, from a vector A, outputs $((\mathsf{C}, \mathsf{C}'), (C, C'))$, where in $\mathsf{C} = (\mathsf{c}_i)_i$, $\mathsf{C}' = (\mathsf{c}'_i)_i$ with $(\mathsf{c}_i, \mathsf{c}'_i)$ the ciphertext of b_i, for $i = 0, \ldots, N$, and then C, C' the commitment of C and C' respectively;
- $\mathsf{IPEwP.Compute}(\mathsf{PK}, \mathsf{A}, (\mathsf{C}, \mathsf{C}', C, C'))$, from the two pairs of vectors (A, C) and $(\mathsf{A}, \mathsf{C}')$, outputs $\mathsf{d} = \langle \mathsf{A}, \mathsf{C} \rangle$ and $\mathsf{d}' = \langle \mathsf{A}, \mathsf{C}' \rangle$ and π, for proving the correct inner-product evaluations;
- $\mathsf{IPEwP.Verify}(\mathsf{PK}, [\mathsf{SK}], \bar{A}, (C, C'), (\mathsf{d}, \mathsf{d}', \pi))$, checks the proof π from the initial commitments $\bar{A}, (C, C')$ and the additional ones in π;
- $\mathsf{IPEwP.Decode}(\mathsf{SK}, (\mathsf{d}, \mathsf{d}', \pi))$, from the FHE decryption key, decrypts the pair $(\mathsf{d}, \mathsf{d}')$ to get $\mathsf{p} = \langle \mathsf{A}, \mathsf{B} \rangle$.

We used again the Schwartz-Zippel lemma to translate equalities between evaluated polynomials into equalities between polynomials, based on the unpredictability of σ. But according to q (large prime or product of smaller primes), one may reduce the bad cases by using multiple σ_κ's. In case one additionally wants the privacy of the sender's vector A, a hiding commitment can be used for $\bar{A}$, but also for Q, with additional noise in d and d', to allow receiver verifiability without leaking any information.

4.3 Verifiability of the Committed Ciphertext

As already explained in the overall description of our protocol in Section 1.3, before verifying the correct inner products $\mathsf{d} = \langle \mathsf{A}, \mathsf{C} \rangle$ and $\mathsf{d}' = \langle \mathsf{A}, \mathsf{C}' \rangle$ and decrypt the pair $(\mathsf{d}, \mathsf{d}')$ to get $\langle \mathsf{A}, \mathsf{B} \rangle$, one may want to be sure that each $(\mathsf{c}_i, \mathsf{c}'_i)$, ciphertext that would decrypt to m_i, is actually a correct encryption of m^i in $\mathcal{R}_t$. This means that B should be the vector $(\mathsf{m}^0, \ldots, \mathsf{m}^N)$ in $\mathcal{R}_t^{N+1}$. Indeed, the sender receives an encryption of m (and possibly some additional information) and generates the vectors C and C' thanks to the linearity of the FHE scheme. Why would they be honest?

To verify that, we use the above inner-product proof between each vector of ciphertexts $\mathsf{C} = (\mathsf{c}_0, \ldots, \mathsf{c}_N)$ or $\mathsf{C}' = (\mathsf{c}'_0, \ldots, \mathsf{c}'_N)$ and a vector of powers $\mathsf{N} = (\mathsf{n}^0, \mathsf{n}^1, \ldots, \mathsf{n}^N)$ derived from a public random $\mathsf{n} \in \mathcal{R}_t$ drawn by the verifier (or generated from a hash). Neither of these vectors need to be kept private as they are generated from information both parties have.

Let $\mathsf{u} = \langle \mathsf{N}, \mathsf{C} \rangle$ and $\mathsf{u}' = \langle \mathsf{N}, \mathsf{C}' \rangle$ be the results of the inner products in $\mathcal{R}_q$, proven as above (with 12 commitments and 2 scalars, and 8 quadratic relations to check). From the linear-homomorphism of the FHE, with appropriate parameters, $\mathsf{Dec}(\mathsf{u}, \mathsf{u}') = \sum_{j=0}^{N} \mathsf{n}^j \cdot \mathsf{m}_j$. The verifier checks this decryption is $\sum_{j=0}^{N} \mathsf{n}^j \cdot \mathsf{m}^j$, with appropriate error (bounded as expected). This leads to $\sum_{j=0}^{N} \mathsf{n}^j \cdot \mathsf{m}_j = \sum_{j=0}^{N} \mathsf{n}^j \cdot \mathsf{m}^j$ in $\mathcal{R}_t$. Note that $\mathcal{R}_t$ is unfortunately not a field, but possibly a ring that is the product of large fields only: $\mathsf{r} = X^{2^k} + 1$ is not irreducible in any $\mathbb{Z}_t[X]$ for a prime t, but for well-chosen prime, all the factors of $X^n + 1$ may have large degrees in $\mathbb{Z}_t[X]$: according to [BGM93], with $t + 1 = 2^\alpha(2\tau + 1)$, for any integer τ, $\alpha \geq 2$, and $n = 2^k$, then all the factors of $X^{2^k} + 1$ have degree $2^{k+1-\alpha}$. If one chooses $t = 3 \bmod 8$, $\alpha = 2$, and in particular $t = 3$ (with $\tau = 0$), there are just two irreducible factors of degree $2^{k-1} = n/2$ in $\mathbb{Z}_t[X]$: the above polynomial thus has all zero coefficients by the Schwartz-Zippel lemma, excepted with probability $2N/t^{n/2}$, as the polynomial is of degree N and n is randomly chosen among $t^{n/2}$ possible values in each of the two fields. Hence, $\mathsf{m}_i = \mathsf{m}^i$ in $\mathcal{R}_t$, excepted with probability bounded by $2N/t^{n/2}$, which is clearly negligible. Note that one cannot use $t = 2$ as $\mathsf{r}(X) = X^n + 1$ is divisible by $X + 1$ in $\mathbb{Z}_2[X]$.

By checking the noise in $(\mathsf{u}, \mathsf{u}')$, as expected with reasonable margin, as one knows the expected plaintext, one gets an upper-bound on individual errors. Even if this might might larger than initially expect, one can guarantee appropriate noise in the $(\mathsf{c}_i, \mathsf{c}'_i)$'s to satisfy the linear-homomorphism, as we will take additional margin to take care of the noise-flooding. If the sender tries to cheats with larger noise in the $(\mathsf{c}_i, \mathsf{c}'_i)$'s, they may reduce the privacy impact of the noise-flooding. But soundness remains guaranteed.

5 Verifiable OPE with Privacy

We now have the tools to allow the receiver/verifier with their private input message m to learn in a verifiable way the inner product of the vector $\mathsf{M} = (\mathsf{m}^j)_j$ with a private vector $\mathsf{F} = (\mathsf{f}_j)_j$, for indices j in $[\![0; N]\!]$, both committed by the sender/prover. More details and more applications are provided in appendix D, but we sketch here a full verifiable OPE protocol, where we assume all the global parameters set, and the sender's polynomial $\mathsf{F} = (\mathsf{f}_j)_j$ committed in a hiding way in $\bar{F}$. Once the receiver has encrypted the input $\mathsf{m} \in \mathcal{R}_t$ under their own FHE key and sent $\mathsf{Enc}(\mathsf{m}) = (\mathsf{c}, \mathsf{c}') \in \mathcal{R}_q^2$ to the sender, the latter

- computes the $(\mathsf{u}_j, \mathsf{u}'_j) = \mathsf{Enc}(\mathsf{m}^j)$, for $j \in [\![0; N]\!]$, from $(\mathsf{c}, \mathsf{c}')$ thanks to the homomorphic properties of the encryption scheme; generates the vectors $\mathsf{U} = (\mathsf{u}_j)$ and $\mathsf{U}' = (\mathsf{u}'_j)$ as well as their commitments U and U'; and provides a proof of valid computation of the inner products $\mathsf{b} = \langle \mathsf{N}, \mathsf{U} \rangle$ and $\mathsf{b}' = \langle \mathsf{N}, \mathsf{U}' \rangle$, for a common vector $\mathsf{N} = (\mathsf{n}^j)_j$ for a random $\mathsf{n} \xleftarrow{\$} \mathcal{R}_t$ (chosen with a random hash function on the previous information), with respect to the commitments U, U': ignoring scalars, the proof consists of 8 commitments (to be sent and checked) and 4 quadratic relations to be verified;

– generates a zero-ciphertext $(\mathsf{z}, \mathsf{z}')$ with a proof of small norm of the error, provides a proof of valid computation of the noisy inner products $\mathsf{d} + \mathsf{z}$ and $\mathsf{d}' + \mathsf{z}'$, where $\mathsf{d} = \langle \mathsf{F}, \mathsf{U} \rangle$ and $\mathsf{d}' = \langle \mathsf{F}, \mathsf{U}' \rangle$, with respect to the commitments $\bar{F}, U, U'$: the proof consists of 8 new commitments (and also the original commitment of the polynomial) (to be sent and checked) and 5 quadratic relations to be verified, once the noise-flooding has been proven. The latter consists of 15 commitments (to be sent and checked) and 12 quadratic relations to be verified;

By first verifying $\mathsf{Dec}(\mathsf{b}, \mathsf{b}') = \sum(\mathsf{nm})^j$ and the appropriate noise, the receiver gets convinced (U, U') commit to correct encryptions of the powers m^j. Then, with the verification of the inner products and the small noise, one gets the guarantee that $\mathsf{Dec}(\mathsf{d}+\mathsf{z}, \mathsf{d}'+\mathsf{z}') = \langle \mathsf{F}, \mathsf{M} \rangle = \mathsf{F}(\mathsf{m})$. As there are common commitments in the successive phases and one actually considers the noise components of $(\mathsf{z}^*, \mathsf{z}'^*)$ instead of these directly in practical applications, the global proof consists of 32 commitments and the verification checks them plus 21 quadratic relations (ignoring scalars and zero-knowledge proofs on scalars), as shown in figures 3 and 4. This is thus independent of the degree of the polynomials.

Security Remarks. Soundness of this protocol is guaranteed by the proofs of valid computations of inner products, first to ensure the content of the commitments U and U' (with 2 inner products), and then to convince of the correct computation of the ciphertext $(\mathsf{d} + \mathsf{z}, \mathsf{d}' + \mathsf{z}')$ (with 2 inner products). The small additional noise $(\mathsf{z}, \mathsf{z}')$ is also proven by inner products to bound the infinity norms of the 3 polynomials $\mathsf{u}, \mathsf{e}_1, \mathsf{e}_2$ involved in $(\mathsf{z}, \mathsf{z}') = (\mathsf{p} \cdot \mathsf{u} + \mathsf{e}_1 \bmod q, \mathsf{p}' \cdot \mathsf{u} + \mathsf{e}_2 \bmod q)$, and proven with quadratic linear relations:

Theorem 5. *Our MyOPE scheme is* SND*-secure against malicious adversaries (see Definition 1).*

The complete proof is provided in appendix D, together with the analysis of the privacy properties, as defined in definitions 2 and 3: basically, the receiver's privacy is ensured by the FHE encryption of m that protects its input. This is a computational security; The sender's privacy is guaranteed by the hiding commitment $\bar{F}$ and the noise $(\mathsf{z}, \mathsf{z}')$ that hides the evaluated circuit. They both provide statistical privacy to the sender.

Succinctness. As explained above, proof is succinct, and independent of the degree of the polynomial. But the size of the commitments may depend on q. Then, for low communication, one can reduce the required multiplicative depth of the computations of the $\mathsf{Enc}(\mathsf{m}^i)$, for $i = 1, \ldots, N$, from $\mathsf{Enc}(\mathsf{m})$: we can consider $d \in \mathbb{N}$, and define the basis $B = \lceil (N+1)^{1/d+1} \rceil$. Then, any $k \in [\![0; N]\!]$ can be written in basis B as $\sum_{i=0}^{d} x_i B^i$, with $x_i \in [\![0; B-1]\!]$. Hence, from the ciphertexts of all the powers $\mathsf{m}^{x_i B^i}$ for $x_i \in [\![0; b-1]\!]$ and $i \in [\![0; d]\!]$, one can compute ciphertexts of all the powers, with at most d multiplications. The receiver will thus have to provide $(d+1)B \sim dN^{1/d}$ ciphertexts, but with a smaller modulus q.

In appendix A, we study the asymptotic bound for q, to get correctness, and get $\log q = \mathcal{O}(d \log n)$, if $N \approx n^d$. Then the size of a ciphertext is in $\mathcal{O}(dn \log n)$, and dn ciphertexts have to be sent: this is globally $\mathcal{O}(d^2 n^2 \log n) = \mathcal{O}(n^2 \log n) = \tilde{\mathcal{O}}(N^{2/d})$ bits to be sent by the receiver. Then, the sender essentially sends back the result (1 ciphertext $= \mathcal{O}(dn \log n)$) and the proof which consists in a constant number of commitments in $\mathcal{O}(\log q)$, and so globally the proof is $\mathcal{O}(\log q) = \mathcal{O}(\log n) = \tilde{\mathcal{O}}(N^{1/d})$, for a constant depth d.

We estimated practical sizes using security bounds on the privacy of FHE given by the LWE estimator [APS15], for N between 2^{20} and 2^{40}, a prime q should be on 512 bits, which would lead to 100-400MBytes for the FHE ciphertexts to be sent, and less then 200KBytes for the proof. For a composite q, the size of the proof increases because of the repetitions of the commitments, as the Schwartz-Zippel lemma provides a smallest soundness. It then becomes 50-200MBytes, still much less than the FHE ciphertexts.

References

APS15. Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.

BCFK20. Alexandre Bois, Ignacio Cascudo, Dario Fiore, and Dongwoo Kim. Flexible and efficient verifiable computation on encrypted data. Cryptology ePrint Archive, Report 2020/1526, 2020. https://eprint.iacr.org/2020/1526.

BCI+13. Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. Succinct non-interactive arguments via linear interactive proofs. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 315–333. Springer, Heidelberg, March 2013.

BEHZ16. Jean-Claude Bajard, Julien Eynard, M. Anwar Hasan, and Vincent Zucca. A full RNS variant of FV like somewhat homomorphic encryption schemes. In Roberto Avanzi and Howard M. Heys, editors, *SAC 2016*, volume 10532 of *LNCS*, pages 423–442. Springer, Heidelberg, August 2016.

BGM93. Ian F. Blake, Shuhong Gao, and Ronald C. Mullin. Explicit factorization of $x^{2^k} + 1$ over $\mathbb{F}_p$ with prime $p = 3 \bmod 4$. *Appl. Algebra Eng. Commun. Comput.*, 4:89–94, 1993.

BN00. Daniel Bleichenbacher and Phong Q. Nguyen. Noisy polynomial interpolation and noisy Chinese remaindering. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 53–69. Springer, Heidelberg, May 2000.

CHLR18. Hao Chen, Zhicong Huang, Kim Laine, and Peter Rindal. Labeled PSI from fully homomorphic encryption with malicious security. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 1223–1237. ACM Press, October 2018.

CLR17. Hao Chen, Kim Laine, and Peter Rindal. Fast private set intersection from homomorphic encryption. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 1243–1255. ACM Press, October / November 2017.

Dam92. Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 445–456. Springer, Heidelberg, August 1992.

FNP04. Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. Efficient private matching and set intersection. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 1–19. Springer, Heidelberg, May 2004.

FNP20. Dario Fiore, Anca Nitulescu, and David Pointcheval. Boosting verifiable computation on encrypted data. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 124–154. Springer, Heidelberg, May 2020.

FS87. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987.

FV12. Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144, 2012. http://eprint.iacr.org/2012/144.

Gen09. Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.

GGP10. Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 465–482. Springer, Heidelberg, August 2010.

GGPR13. Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645. Springer, Heidelberg, May 2013.

Gil99. Niv Gilboa. Two party RSA key generation. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 116–129. Springer, Heidelberg, August 1999.

Haz18. Carmit Hazay. Oblivious polynomial evaluation and secure set-intersection from algebraic PRFs. *Journal of Cryptology*, 31(2):537–586, April 2018.

HL09. Carmit Hazay and Yehuda Lindell. Efficient oblivious polynomial evaluation with simulation-based security. Cryptology ePrint Archive, Report 2009/459, 2009. http://eprint.iacr.org/2009/459.

HN12. Carmit Hazay and Kobbi Nissim. Efficient set operations in the presence of malicious adversaries. *Journal of Cryptology*, 25(3):383–433, July 2012.

HT10. Carmit Hazay and Tomas Toft. Computationally secure pattern matching in the presence of malicious adversaries. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 195–212. Springer, Heidelberg, December 2010.

JL09. Stanislaw Jarecki and Xiaomin Liu. Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 577–594. Springer, Heidelberg, March 2009.

KZG10. Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 177–194. Springer, Heidelberg, December 2010.

LN14. Tancrède Lepoint and Michael Naehrig. A comparison of the homomorphic encryption schemes FV and YASHE. In David Pointcheval and Damien Vergnaud, editors, *AFRICACRYPT 14*, volume 8469 of *LNCS*, pages 318–335. Springer, Heidelberg, May 2014.

LP00. Yehuda Lindell and Benny Pinkas. Privacy preserving data mining. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 36–54. Springer, Heidelberg, August 2000.

NP99. Moni Naor and Benny Pinkas. Oblivious transfer and polynomial evaluation. In *31st ACM STOC*, pages 245–254. ACM Press, May 1999.

Pai99. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238. Springer, Heidelberg, May 1999.

PRS17. Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-LWE for any ring and modulus. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th ACM STOC*, pages 461–473. ACM Press, June 2017.

PRTY20. Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. PSI from PaXoS: Fast, malicious private set intersection. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 739–767. Springer, Heidelberg, May 2020.

Sch80. Jack T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of ACM*, 27(4):701–717, 1980.

Ver11. Damien Vergnaud. Efficient and secure generalized pattern matching via fast Fourier transform. In Abderrahmane Nitaj and David Pointcheval, editors, *AFRICACRYPT 11*, volume 6737 of *LNCS*, pages 41–58. Springer, Heidelberg, July 2011.

Yao86. Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th FOCS*, pages 162–167. IEEE Computer Society Press, October 1986.

Zip79. Richard Zippel. Probabilistic algorithms for sparse polynomials. In Edward W. Ng, editor, *Symbolic and Algebraic Computation, EUROSAM '79*, volume 72 of *LNCS*, pages 216–226. Springer, 1979.

Appendix

A Fully Homomorphic Encryption: FV Scheme

As explained in the preliminaries, we consider the Fan-Vercauteren (FV) Fully Homomorphic Encryption (FHE) scheme [FV12].

A.1 Description

Notations. Let $\mathcal{R}$ be the ring $\mathbb{Z}[X]/\mathsf{r}(X)$, where $\mathsf{r}(X) = X^n+1$ for $n = 2^k$. Given a polynomial $\mathsf{p} \in \mathcal{R}$, we denote $\|\mathsf{p}\|_\infty$ the infinity norm, *i.e.* the max of its coefficients. We also define the polynomial multiplication expansion factor δ as

$$\delta = \max_{\mathsf{c},\mathsf{d}\in\mathcal{R}} \{\|\mathsf{c}\cdot\mathsf{d}\|_\infty / (\|\mathsf{c}\|_\infty \cdot \|\mathsf{d}\|_\infty)\}.$$

By taking the cyclotomic polynomial $\mathsf{r}(X) = X^n + 1$, the worst-case bound for this expansion factor is $\delta = n$.

The Fan-Vercauteren FHE. In FV scheme, the plaintext space is $\mathcal{R}_t = \mathbb{Z}_t[X]/\mathsf{r}(X)$ while the ciphertext space is $\mathcal{R}_q \times \mathcal{R}_q$, where $\mathcal{R}_q = \mathbb{Z}_q[X]/\mathsf{r}(X)$. FV encryption scheme is thus described by parameters $\Gamma = (q, t, n, \sigma)$, with $0 < \sigma < 1$, the noise parameter. They all will depend on several constraints, namely the expected multiplicative depth for the correctness and the hardness of the Ring-LWE problem for the semantic security.

For efficiency reasons, one may want to use FHE with RNS [BEHZ16]. We then assume $q = \prod_{i=1}^{\ell} p_i$ for distinct prime factors $p_1 < \ldots < p_\ell$, assumed larger than p (that will be assumed a lower bound on the p_i's all along this paper). We additionally take $t = 3$ in our applications (though bigger primes could also be used, but still with the constraint that $t = 3 \bmod 8$ as explained in Section 4.3). We denote $\Delta = \lfloor q/t \rfloor$ and $\chi = D^n_{\mathbb{Z},\sigma}$ the centered discrete gaussian distribution over $\mathbb{Z}$ with standard deviation σ for each coordinate, for the noise. The FV encryption scheme [FV12] consists in the following algorithms:

$\mathsf{KeyGen}(1^\lambda, \Gamma) \to (\mathsf{sk}, \mathsf{pk})$**:** On input the security parameter λ and the efficiency parameters set Γ, sample $\mathsf{a}, \mathsf{s} \leftarrow \mathcal{R}_q$ and set $(\mathsf{p}, \mathsf{p}') = (-(\mathsf{a}\cdot\mathsf{s}+\mathsf{e}), \mathsf{a}) \in \mathcal{R}_q^2$. $\mathsf{sk} = \mathsf{s}$ is the secret key, where the coefficients of e are taken from χ. The public key pk contains $(\mathsf{p}, \mathsf{p}')$ with a relinearization key rlk.

$\mathsf{Enc}(\mathsf{pk}, \mathsf{m}) \to (\mathsf{c}, \mathsf{c}')$**:** Using $(\mathsf{p}, \mathsf{p}')$ from pk and a message $\mathsf{m} \in \mathcal{R}_t$, compute ciphertext $(\mathsf{c}, \mathsf{c}') = (\mathsf{p}\cdot\mathsf{u} + \mathsf{e}_1 + \Delta\cdot\mathsf{m} \bmod q, \mathsf{p}'\cdot\mathsf{u} + \mathsf{e}_2 \bmod q)$, with coefficients of $\mathsf{u}, \mathsf{e}_1, \mathsf{e}_2$ also taken from χ.

$\mathsf{Dec}(\mathsf{sk}, (\mathsf{c}, \mathsf{c}')) \to \mathsf{m}$**:** Given $\mathsf{sk} = \mathsf{s}$, compute

$$\mathsf{d} = \mathsf{c} + \mathsf{c}'\cdot\mathsf{s} = \Delta\cdot\mathsf{m} - \mathsf{e}\cdot\mathsf{u} + \mathsf{e}_2\cdot\mathsf{s} + \mathsf{e}_1 = \Delta\cdot\mathsf{m} + \mathsf{v} \bmod q$$
$$\mathsf{m}' = \lfloor \mathsf{d}/\Delta \rceil = \lfloor (\Delta\cdot\mathsf{m} + \mathsf{v})/\Delta \rceil = \mathsf{m} + \lfloor \mathsf{v}/\Delta \rceil \bmod t$$

where $\mathsf{v} = -\mathsf{e}\cdot\mathsf{u} + \mathsf{e}_2\cdot\mathsf{s} + \mathsf{e}_1$ is the error term: $\mathsf{m}' = \mathsf{m}$ if $\|\mathsf{v}\|_\infty \leq \Delta/2$.

$\mathsf{Eval}(\mathsf{pk}, f, (\mathsf{c}_i, \mathsf{c}'_i)_{i=1,\ldots,\ell}) \to (\mathsf{c}_f, \mathsf{c}'_f)$**:** Given pk, an arithmetic circuit for a function f with bounded multiplicative depth and ℓ ciphertexts $(\mathsf{c}_i, \mathsf{c}'_i)_{i=1,\ldots,\ell}$ output the ciphertext $(\mathsf{c}_f, \mathsf{c}'_f)$.

The addition of two ciphertexts is a ciphertext of the sum of the plaintexts. Multiplicative operations have also been shown possible with additional information to relinearize the ciphertext after a product using rlk (the relinearization key included in pk). For Eval to be correct, $\mathsf{Dec}(\mathsf{sk}, (\mathsf{c}_f, \mathsf{c}'_f))$ should return $f(m_1, \ldots, m_\ell)$ where $\mathsf{Dec}(\mathsf{sk}, (\mathsf{c}_i, \mathsf{c}'_i)) = m_i$ for $i = 1, \ldots, \ell$, with overwhelming probability.

A.2 Security Analysis

Semantic Security. From [PRS17]], if $0 \leq \alpha < 1$ be some real, and the modulus q is such that $\sigma = \alpha q = \omega(1)$, then there is a polynomial quantum reduction from the SIVP problem with approximation factor $\gamma(n)$ to the average-case decision Ring-LWE where $\gamma(n) \leq \max\{\omega(\sqrt{n \log n}/\alpha), \sqrt{2}n\}$. For the SIVP problem to be hard, we need $\gamma(n)$ to be polynomial in n. And the semantic security of the FV scheme relies on the decision Ring-LWE problem. We can take $\sigma = \alpha q = \sqrt{n}$, then we have $\gamma(n) \leq q\sqrt{\log n}$, which will be polynomial in n, in our case.

Correctness Assuming the coefficients of the error polynomials and the secret key polynomial are bounded by B, we just require $B \cdot (2nB + 1) \leq \Delta/2$ for correct decryption, as $\|\mathsf{v}\|_\infty \leq B \cdot (2nB + 1)$. In particular, $\Pr[\|\mathsf{e}\|_\infty > 10 \cdot \sigma, \mathsf{e} \xleftarrow{\$} \chi] \leq 2^{-128}$. Hence we can take $B = 10 \cdot \sigma$. This means that q should be taken a bit larger than $400nt\sigma^2$ for correct decryption of a fresh ciphertext.

But as already noted, we will need the (L, d)-$\mathcal{R}_t$ linear homomorphism property:

$$\mathsf{Dec}\left(\sum_{i=1}^{L} \mathsf{a}_i \cdot (\mathsf{c}_i, \mathsf{c}'_i)\right) = \sum_{i=1}^{L} \mathsf{a}_i \cdot \mathsf{m}_i$$

for any $\mathsf{a}_i \in \mathcal{R}_t$ and ciphertexts $(\mathsf{c}_i, \mathsf{c}'_i) \in \mathcal{R}_q^2$ generated with a circuit of multiplicative depth at most d, encrypting $\mathsf{m}_i \in \mathcal{R}_t$. In order to decrypt the linear combination, we compute:

$$\begin{aligned}\sum_{i=1}^{L} \mathsf{a}_i \cdot \mathsf{c}_i + \mathsf{s} \cdot \sum_{i=1}^{L} \mathsf{a}_i \cdot \mathsf{c}'_i &= \sum_{i=1}^{L} \mathsf{a}_i \left(\mathsf{c}_i + \mathsf{s} \cdot \mathsf{c}'_i\right) \\ &= \sum_{i=1}^{L} \mathsf{a}_i(\Delta \cdot \mathsf{m}_i + \mathsf{v}_i) \bmod q = \Delta \cdot \sum_{i=1}^{L} \mathsf{a}_i \cdot \mathsf{m}_i + \sum_{i=1}^{L} \mathsf{a}_i \cdot \mathsf{v}_i \bmod q\end{aligned}$$

Decryption works if

$$\left\| \sum_{i=1}^{L} \mathsf{a}_i \cdot \mathsf{v}_i \right\|_\infty \leq \sum_{i=1}^{L} n\|\mathsf{a}_i\|_\infty \cdot \|\mathsf{v}_i\|_\infty \leq nt \sum_{i=1}^{L} \|\mathsf{v}_i\|_\infty \leq ntL\|\mathsf{v}\|_\infty \leq \Delta/2,$$

for a noise bounded by $\|\mathsf{v}\|_\infty$ after the evaluation of a circuit of multiplicative depth d. Using [LN14] noise derivation formula, the error noise growth after having evaluated a sequence of d multiplications with a fresh ciphertext is bounded by $C_1^d V + dC_1^{d-1}C_2 = C_1^{d-1} \cdot (C_1 V + dC_2)$, where

$$C_1 = n^2 tB + 4nt \qquad C_2 = n^2 B(B + t^2) + nBw(\lfloor \log_w(q) \rfloor + 1) \qquad V = B(2nB + 1)$$

where w is an integer base, for relinearization, that we will take equal to 2^{32} as in [LN14], and:

$$C_1 \leq 11n^{7/2} \qquad C_2 \leq 2^{36} n^{7/2} \qquad V \leq 201n^2$$

assuming $n \geq 5$, $\sigma = \sqrt{n}$, $B = 10 \cdot \sqrt{n}$, $t^2 \leq \sqrt{n}$, and $\left\lfloor \frac{\log_2(q)}{32} \right\rfloor + 1 \leq \sqrt{n}$, from the relinearization basis. Thus the initial error in the result is bounded by

$$Lnt \cdot C_1^{d-1} \cdot \left(C_1 V + dC_2\right) \leq Lnt \cdot (11n^{7/2})^{d-1} \cdot \left(201n^2 + 2^{36} dn^{7/2}\right).$$

With $201n^2 \leq 2^{36} dn^{7/2}$, it is sufficient to have the condition: $2^{34} \cdot 11^d dLnt \cdot n^{7d/2} \leq \frac{q}{2t}$, so, with our previous assumption on t, $2^{35} \cdot 11^d dL \cdot n^{(7d+3)/2} \leq q$ guarantees the (L, d)-$\mathcal{R}_t$ linear homomorphism property.

A.3 Noise-Flooding

A linear combination of ciphertexts can leak the coefficients, from the evolution of the final noise, which can be recovered by the owner of the decryption key. To avoid this leakage, one can add super-polynomial noise to the result, this is the so-called *noise-flooding* technique: the sender will generate encryption of 0, *i.e.* polynomials $(\mathsf{z}^*, \mathsf{z}'^*)$ of the form

$$(\mathsf{z}^*, \mathsf{z}'^*) = \left(\mathsf{p} \cdot \mathsf{u} + \mathsf{e}_1 \bmod q, \mathsf{p}' \cdot \mathsf{u} + \mathsf{e}_2 \bmod q\right),$$

with coefficients of $\mathsf{u}, \mathsf{e}_1, \mathsf{e}_2$ follow the appropriate distribution for their own privacy: according to a Gaussian distribution on $\mathbb{Z}$ with standard deviation 2^λ larger than the error in the result, that we bounded by $B' = 2^{34} \cdot 11^d dLnt \cdot n^{7d/2}$. Using the verifiable inner-product for provable L_2-norm, one can prove that $\|\mathsf{u}\|_2, \|\mathsf{e}_1\|_2, \|\mathsf{e}_2\|_2$ are lower than $2^\lambda B'$, which guarantees the infinity norms are also lower than that.

A.4 Asymptotic Parameters

One now needs $q \geq 2^\lambda \times 2tB'$: With the above B', this means that

$$q \geq 2^{\lambda+35} \cdot 11^d dL \cdot n^{(7d+3)/2}$$

guarantees the (L, d)-$\mathcal{R}_t$ linear homomorphism property and noise-flooding.

In particular, we will need the (N, d)-$\mathcal{R}_t$ linear homomorphism property, so assuming $N \leq n^{d/2}$, we want:

$$q \geq 2^{\lambda+35} \cdot 11^d d \cdot n^{4d+3/2}. \tag{1}$$

Asymptotically, we will just need to take q large enough with respect to n and d. With $\lambda = 128$, typical SEAL implementation of FV values are $n \leq 2^{14}$, and $d \leq 8$, then relation (1) shows that taking q on approximately 600 bits would be more than enough to grant correctness of the decryption. When using the above exact error $Lnt \cdot C_1^{d-1} \cdot \left(C_1 V + dC_2\right)$ instead of the gross approximation, we can remark that 512 bits for q will be enough, for both semantic security and correctness, in our particular cases.

B Secure Encodings

We here give a bit more detail about secure encodings sketched in Section 2.5. Some examples are described in the Appendix B.2.

B.1 Definitions

Definition 6 (Encoding Scheme). *An encoding scheme over a ring* R *consists in a tuple of algorithms* $(\mathsf{Gen}, \mathsf{E})$.

- $(\mathsf{pk}, \mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^\lambda)$, *a key generation algorithm that takes as input a security parameter and outputs public information* pk, *a secret key* sk, *and a verification key* vk, *that can be either public or private;*
- $E \leftarrow \mathsf{E}_{\mathsf{sk}}(a)$, *a (probabilistic) encoding algorithm mapping a ring element* $a \in \mathsf{R}$ *in the encoding space* $\mathcal{E}$, *using the secret key* sk.

Properties. An encoding scheme should satisfy the following properties, with efficient and correct algorithms:

- L-Linearly homomorphic: An algorithm $\mathsf{Eval}_{\mathsf{pk}}(E_1, \ldots, E_L; c_1, \ldots, c_L)$, on input public information pk, encodings $E_1 = \mathsf{E}_{\mathsf{sk}}(a_1)$, ..., $E_L = \mathsf{E}_{\mathsf{sk}}(a_L)$, and coefficients $c_1, \ldots, c_L \in \mathsf{R}$, outputs an encoding of $\sum_{i=1}^{L} c_i \cdot a_i$;
- L-Quadratic root verification: An algorithm $\mathsf{QCheck}_{\mathsf{vk}}(Q, E_1, \ldots, E_L)$, on input the verification key vk, a quadratic polynomial $Q \in \mathsf{R}[X_1, \ldots, X_L]$ and encodings $E_1 = \mathsf{E}_{\mathsf{sk}}(a_1)$, ..., $E_L = \mathsf{E}_{\mathsf{sk}}(a_L)$, checks whether or not the relation $Q(a_1, \ldots, a_L) = 0$ is satisfied in R;
- Image verification: An algorithm $\mathsf{Verify}_{\mathsf{vk}}(E)$, on input the verification key vk and an element E, verifies E is an actual encoding of some element in R: this algorithm not only verifies that $E \in \mathcal{E}$, but also that there exists an element $a \in \mathsf{R}$ such that E can be an encoding of a.

According to the verification key that can be either public or private, the verification processes will be either public or private.

Secure Encodings. We now formally define the soundness properties for the above verification algorithms, in terms of knowledge-soundness:

Definition 7 (Linear-Only Extractability). *An encoding scheme* $(\mathsf{Gen}, \mathsf{E})$ *over* R *is extractable if for any* PPT *adversary* $\mathcal{A}$*, there exists a* PPT *extractor* $\mathsf{Ext}_{\mathcal{A}}$ *such that the following probability is negligible in the security parameter:*

$$\Pr\left[\mathsf{QCheck}_{\mathsf{vk}}\left(X - \sum_{i=1}^{n} c_i X_i, E, E_1, \ldots, E_n\right) = \mathsf{false} \;\middle|\; \mathsf{Verify}_{\mathsf{vk}}(E) = \mathsf{true}\right]$$

on the probability space $(\mathsf{pk}, \mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^\lambda)$, $a_1, \ldots, a_n \xleftarrow{\$} \mathsf{R}$, $E_i \leftarrow \mathsf{E}_{\mathsf{sk}}(a_i)$, *for* $i = 1, \ldots, n$, *and* $(E; c_1, \ldots, c_n) \leftarrow (\mathcal{A} || \mathsf{Ext}_{\mathcal{A}})(\mathsf{pk}, (E_i)_i)$.

While the encoding is linearly-homomorphic, the above extractability property requires that it be impossible to derive new valid encodings excepted by linear combinations: any new *valid* encoding E of some $a \in \mathsf{R}$ will necessarily satisfy $a = \sum_i c_i a_i$, for extractable elements $c_i \in \mathsf{R}$. Intuitively, when an encoding E passes the verification test, one can extract the linear combination of the given initial encodings.

Zero-Knowledge Proofs. The above properties will be enough for a binding commitment, but additional blinding factors in an appropriate masking set $\mathcal{M}$ will be required for hiding commitments, which will depend on the ring R (see below for the particular case of $\mathsf{R} = \mathbb{Z}_q$). Then, Zero-Knowledge proofs will be needed for Quadratic root verifications with private linear combinations: $\mathsf{ZKLQCheck}_{\mathsf{vk}}(Q, E_1, \ldots, E_L; E'_1, \ldots, E'_\mu)$, on input a quadratic polynomial $Q \in \mathsf{R}[X_1, \ldots, X_L]$ and encodings $E_1 = \mathsf{E}_{\mathsf{sk}}(a_1)$, ..., $E_L = \mathsf{E}_{\mathsf{sk}}(a_L)$, $E'_1 = \mathsf{E}_{\mathsf{sk}}(b_1)$, ..., $E'_\mu = \mathsf{E}_{\mathsf{sk}}(b_\mu)$. The verification key vk on the verifier side and the private coefficients $c_1, \ldots, c_\mu$ on the prover side prove that the relation $Q(a_1, \ldots, a_L) = \sum c_i \cdot b_i$ is satisfied. The c_i's will be the blinding factors in $\mathcal{M}$.

B.2 Examples

In the following we illustrate the definition of secure encodings with two distinct constructions. the first provides public verifiability whereas the second will be for designated-verifier only.

Encodings over $\mathsf{R} = \mathbb{Z}_q$ from Pairings. The Knowledge of Exponent Assumption, introduced by Damgard [Dam92] states that given g and g^α in a group $\mathbb{G}$, it is hard to create c and $\hat{c}$

in that group $\mathbb{G}$ so that $\hat{c} = c^\alpha$, without knowing a such that $c = g^a$. The only way to compute $\hat{c}$ being to generate $(g^\alpha)^a$.

In a pairing-friendly setting $\mathsf{pk} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, g, \mathfrak{g}, e)$, one can check the appropriate relation between c and $\hat{c}$, with $\mathfrak{g}^\alpha$. We can thus consider such a pairing-friendly setting with $q = p_1 \cdot \ldots \cdot p_\ell$ a product of ℓ prime integers $p_1 < \ldots < p_\ell$ (possibly with $\ell \neq 1$, so that $\mathbb{Z}_q$ is a ring, but not necessarily a field), we denote $G = e(g, \mathfrak{g})$. The verification key is $\mathsf{vk} = \mathfrak{g}^\alpha$ for the secret key $\alpha \xleftarrow{\$} \mathbb{Z}_q$: the encoding function E_{sk} is $\mathsf{E}_{\mathsf{sk}}(a) = (g^a, g^{\alpha \cdot a}, \mathfrak{g}^a) \in \mathbb{G}_1^2 \times \mathbb{G}_2$. It is clearly L-linearly-homomorphic for any L. The bilinear map e allows public quadratic root verification, on the elements g^a and $\mathfrak{g}^a$: for example, $Q = X_1 \cdot X_2 - X_3$ on encodings of a_1, a_2 and a_3, it can be done with $e(g^{a_1}, \mathfrak{g}^{a_2}) \cdot e(g^{-a_3}, \mathfrak{g}) = e(g, \mathfrak{g})^{Q(a_1,a_2,a_3)}$. This must be done after image verification of any individual encoding with $e(g^a, \mathfrak{g}) = e(g, \mathfrak{g}^a)$ and $e(g^a, \mathsf{vk}) = e(g^{\alpha \cdot a}, \mathfrak{g})$. They are both public, as vk is public.

To hide the content of an encoding, one just needs the encoding $E' = (g, g^\alpha, \mathfrak{g})$ of 1, multiplied by a private random factor in $\mathcal{M} = \mathbb{Z}_q$. To prove the existence of μ such private coefficients c_i such that $Q(a_1, \ldots, a_L) = \sum c_i b_i \bmod q$ is satisfied, one has to prove the knowledge of $(c_i)_i$ such that

$$V = e(g, \mathfrak{g})^{Q(a_1,\ldots,a_L)} = \prod e(g^{b_i}, \mathfrak{g})^{c_i}$$

which can be done with a classical Schnorr-proof, from the encodings $\mathsf{E}_{\mathsf{sk}}(b_i)$, as $V = e(g, \mathfrak{g})^{Q(a_1,\ldots,a_L)}$ can be computed thanks to the pairing. Using the Fiat-Shamir paradigm [FS87], this proof can be non-interactive: with random exponents $k_i \xleftarrow{\$} \mathbb{Z}_q$, the prover sets $D = \prod e(g^{b_i}, \mathfrak{g})^{k_i}$, gets a random challenge $e = \mathcal{H}(V, \{\mathsf{E}_{\mathsf{sk}}(b_i)\}, D) \in [\![0; 2^\lambda]\!]$, with $2^\lambda < p_1$, and generates the proof $\Pi = (\{s_i = k_i - e c_i \bmod q\}, e) \in \mathbb{Z}_q^\mu \times [\![0; 2^\lambda]\!]$. The verifier can compute

$$D' = \prod e(g^{b_i}, \mathfrak{g})^{s_i} \times V^e$$

and check whether $e \stackrel{?}{=} \mathcal{H}(V, \{\mathsf{E}_{\mathsf{sk}}(b_i)\}, D')$. Indeed, if the statement is true

$$\begin{aligned} \prod e(g^{b_i}, \mathfrak{g})^{s_i} \times V^e &= \prod e(g^{b_i}, \mathfrak{g})^{s_i} \times (\prod e(g^{b_i}, \mathfrak{g})^{c_i})^e \\ &= \prod e(g^{b_i}, \mathfrak{g})^{s_i + e c_i} = \prod e(g^{b_i}, \mathfrak{g})^{k_i} = D. \end{aligned}$$

The linear-only extractability relies on the above Knowledge of Exponent Assumption, that requires the hardness of the discrete logarithm in the bilinear generic group model, which additionally requires all prime factors p_k to be large enough (at least 2λ bits, for a security parameter λ). An encoding is a tuple of elements in $\mathcal{E} = \mathbb{G}_1^2 \times \mathbb{G}_2$, and a zero-knowledge proof of μ scalars contains μ elements from $\mathbb{Z}_q$ and the challenge in $[\![0; 2^\lambda]\!]$. In case of multiple proofs, one can use a unique global challenge e, and the same (k_i, s_i) can be used for the same private scalars c_i. Hence, globally, the size is thus μ' elements from $\mathbb{Z}_q$ where μ' is the total number of private scalars, independently of the number of equations, plus one challenge. The requirement of large prime factors will be prohibitive when used with Fully Homomorphic Encryption, as RNS optimizations do not apply.

Encodings over $\mathsf{R} = \mathbb{Z}_q$ from a Linear-Only Encryption Scheme. Given a linear-only encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ from $\mathbb{Z}_q$ onto $\mathcal{C}$, for any integer q, one chooses a random private element $\alpha \xleftarrow{\$} \mathbb{Z}_q^*$, to be the secret key of the encoding. Then, pk is the public key of the encryption scheme, while vk is α together with the secret decryption key of the encryption scheme. It is thus private. Then, the encoding function E_{sk} is $\mathsf{E}_{\mathsf{sk}}(a) = (\mathsf{Enc}(a), \mathsf{Enc}(\alpha \cdot a)) \in \mathcal{C} \times \mathcal{C}$. It is clearly linearly-homomorphic from an additively-homomorphic encryption scheme. The decryption algorithm allows any root verification using the decryption key in vk, while the image verification tests whether the two decryption values verify the secret ratio α. The linear-only extractability depends on the specific encryption scheme: but either the encryption scheme

is fully/somewhat homomorphic, or this property is satisfied. An encoding is a pair of elements in $\mathcal{E} = \mathcal{C} \times \mathcal{C}$.

B.3 Encodings from Paillier Encryption

More concretely, one can use Paillier encryption scheme [Pai99] with a large RSA modulus $\mathcal{N}$, the encryption of $x \in \mathbb{Z}_{\mathcal{N}}$ is $E = (1+\mathcal{N})^x \cdot r^{\mathcal{N}} \bmod \mathcal{N}^2$, for $r \xleftarrow{\$} \mathbb{Z}^*_{\mathcal{N}}$. The ciphertext space is thus $\mathcal{C} = \mathbb{Z}^*_{\mathcal{N}^2}$. For the decryption, one needs the value $\lambda = \lambda(\mathcal{N})$, where λ is Carmichael's function, which is equivalent of the knowledge of the factorisation of $\mathcal{N}$. As $\lambda(\mathcal{N}^2) = \mathcal{N}\lambda$, $E^\lambda = (1+\mathcal{N})^{x\lambda} = 1 + x\lambda \cdot \mathcal{N} \bmod \mathcal{N}^2$. If λ is invertible modulo $\mathcal{N}$, one can extract x modulo $\mathcal{N}$.

For the encoding, the public key pk is thus the modulus $\mathcal{N}$, the secret key sk is a random element $\alpha \xleftarrow{\$} \mathbb{Z}^*_{\mathcal{N}}$, and the verification key vk is the decryption key $\lambda(\mathcal{N})$ and the secret value α:

- $\mathsf{E}_{\mathsf{sk}}(a) = ((1+\mathcal{N})^a \cdot r_0^{\mathcal{N}} \bmod \mathcal{N}^2, (1+\mathcal{N})^{\alpha \cdot a} \cdot r_1^{\mathcal{N}} \bmod \mathcal{N}^2)$, for $r_0, r_1 \xleftarrow{\$} \mathbb{Z}^*_{\mathcal{N}}$;
- $\mathsf{Dec}_{\mathsf{vk}}(C_0)$ decrypts $x0$ from C_0 as $\frac{C_0^\lambda \bmod \mathcal{N}^2 - 1}{\mathcal{N}} \times \lambda^{-1} \bmod \mathcal{N}$.
- $\mathsf{Verify}_{\mathsf{vk},\mathsf{sk}}(C_0, C_1)$ first decrypts both ciphertexts using $\lambda(\mathcal{N})$ to get $x_0, x_1 \bmod \mathcal{N}$, and checks whether $x_1 = \alpha \cdot x_0 \bmod \mathcal{N}$.

Decryption and verification can be optimized using the CRT as in [Pai99]. An encoding is thus $4 \log \mathcal{N}$ bit-long.

We want an encoding on $\mathbb{Z}_q$, one can thus take $\mathcal{N} > q$ to encode elements $x \in [\![0; q-1]\!]$. Decoding first decrypts both ciphertexts modulo $\mathcal{N}$, with the elements in $[\![-\mathcal{N}/2; \mathcal{N}/2]\!]$, checks the relation with α modulo $\mathcal{N}$, and reduces it again modulo q in $[\![0; q-1]\!]$ to extract the encoded value in $\mathbb{Z}_q$. For further verifications (quadratic checks), one just considers the decryption of the first ciphertext in $[\![-\mathcal{N}/2; \mathcal{N}/2]\!]$, and relations among the plaintexts.

From L ciphertexts $E_i = (1+\mathcal{N})^{x_i} \cdot {r_i}^{\mathcal{N}} \bmod \mathcal{N}^2$, for $r_i \xleftarrow{\$} \mathbb{Z}^*_{\mathcal{N}}$, one can compute the linear combination with coefficients smaller than q, $\prod E_i^{c_i} = (1+\mathcal{N})^{\sum c_i x_i} \cdot (\prod r_i^{c_i})^{\mathcal{N}}$, which is an encryption of $\sum c_i x_i \bmod \mathcal{N}$. It decodes to $\sum c_i x_i$ if $|\sum c_i x_i| < \mathcal{N}/2$: we thus have to take $\mathcal{N} > 2L \cdot q^2$ if the encodings are fresh encodings that encrypt elements in $[\![0; q-1]\!]$. For a quadratic check, using vk, the verifier can decrypt all the encodings modulo $\mathcal{N}$ and reduce them modulo q to check the relation modulo q. There is no more constraint on $\mathcal{N}$.

To hide the content of an encoding, even with respect to the owner of the secret key, one uses a random encoding of a random private mask in $[\![0; 2^\lambda L q^2]\!]$, to act as a statistically hiding one-time pad, furthermore randomized with $\mathcal{N}$-th powers to remove any information in the initial random coins. In this case, one needs $\mathcal{N} > 2L \cdot 2^\lambda q^2$ for correct decryption modulo $\mathcal{N}$, without wrapping around modulo N before the reduction modulo q. Hence $\mathcal{M} = [\![0; 2^\lambda L q^2]\!]$.

About the zero-knowledge verification $\mathsf{ZKLQCheck}(Q, (E_i)_i; (E'_i)_i)$, to prove the existence of μ coefficients $c_i \in \mathcal{M}$ such that $Q(a_1, \ldots, a_L) = \sum c_i b_i \bmod q$ is satisfied, on the encodings $E_1 = \mathsf{E}_{\mathsf{sk}}(a_1)$, $\ldots$, $E_L = \mathsf{E}_{\mathsf{sk}}(a_L)$, $E'_1 = \mathsf{E}_{\mathsf{sk}}(b_1)$, $\ldots$, $E'_\mu = \mathsf{E}_{\mathsf{sk}}(b_\mu)$, one has to prove the knowledge of (c_i) in $V = \mathsf{Eval}(\{E'_i\}, \{c_i\})$, for $i \in [\![1; \mu]\!]$, where the verifier knows, after decryption of the encodings $(E_i)_i$ and quadratic computations modulo q, $V' = (1+\mathcal{N})^{Q(a_1,\ldots,a_L) \bmod q} \bmod \mathcal{N}^2$. One wants to prove that V and V' decrypt to the same value modulo q: $V = V' \times (1+\mathcal{N})^{qr_1} \cdot r_0^{\mathcal{N}} \bmod \mathcal{N}^2$, but for unknown values r_0, r_1. We stress that we only consider the first ciphertext in the encodings. One thus proves the knowledge of $c_i \in \mathcal{M}$ for $i \in [\![1; \mu]\!]$ such that $V = \prod {E'_i}^{c_i} = V' \times (1+\mathcal{N})^{qr_1} \cdot r_0^{\mathcal{N}} \bmod \mathcal{N}^2$. The c_i's are the masks in $[\![0; 2^\lambda L q^2]\!]$, but one can use their q-reduction. Then, as the encodings E'_i can encrypt elements in $[\![-Lq^2; Lq^2]\!]$, $|r_1 q| \leq \mu L q^3$.

For the latter zero-knowledge proof, the prover chooses random $k_i \xleftarrow{\$} \mathbb{Z}_q$, with additional noise $\nu' \xleftarrow{\$} [\![0; 2^\lambda L q^2]\!]$ and $\nu \xleftarrow{\$} \mathbb{Z}^*_N$, to hide any extra information beyond the modulo q relation, and sets $D = \prod {E'_i}^{k_i} \cdot (1+\mathcal{N})^{q\nu'} \nu^{\mathcal{N}} \bmod \mathcal{N}^2$, gets a random challenge e (possibly derived from $(Q, (E_i)_i, (E'_i)_i, D)$ in $[\![0; 2^{\lambda'} - 1]\!]$ and outputs the proof $\Pi = (D, (s_i = k_i - e c_i \bmod q)_i) \in \mathbb{Z}_{\mathcal{N}^2} \times \mathbb{Z}_q^\mu$.

For the moment, we use a different space $[\![0; 2^{\lambda'} - 1]\!]$ for the challenge, with λ' possibly smaller than λ or $-\log \varepsilon_s$, in which case $-\log \varepsilon_s/\lambda'$ parallel repetitions should be performed for correct soundness. One can check

$$e \leftarrow \mathcal{H}(Q, (E_i)_i, (E'_i)_i, D) \qquad D' \leftarrow \prod E_i'^{s_i} \cdot (V')^e \bmod \mathcal{N}^2 \qquad \mathsf{Dec}(D/D') \stackrel{?}{=} 0 \bmod q$$

Indeed,

$$\begin{aligned} D' &= \prod E_i'^{s_i} \times (V')^e = \prod E_i'^{s_i} \times V^e \cdot (1+\mathcal{N})^{-eqr_1} \cdot r_0^{-e\mathcal{N}} \\ &= \prod E_i'^{k_i - ec_i} \times \prod E_i'^{ec_i} \times (1+\mathcal{N})^{-eqr_1} \cdot r_0^{-e\mathcal{N}} \\ &= \prod E_i'^{k_i} \cdot (1+\mathcal{N})^{-eqr_1} \cdot r_0^{-e\mathcal{N}} = D \cdot (1+\mathcal{N})^{-q(er_1+\nu')} \cdot (r_0^e \nu)^{-\mathcal{N}} \bmod \mathcal{N}^2. \end{aligned}$$

One must make sure that $2eqr_1 \leq \mathcal{N}$: with $\mathcal{N} \geq 2\mu L \cdot pq^3$ (where p is the smallest prime factor of q and $2^{\lambda'} < p$), there is no reduction modulo $\mathcal{N}$ before the reduction modulo q. The zero-knowledge proof Π of μ scalars thus contains $2\log \mathcal{N} + \mu \times \log q$ bits.

In case of multiple equations involving the same secret c_i, the same challenge is used, and the same (k_i, s_i), reducing the bit-size to $2\log \mathcal{N} \times \#\text{Equations} + \log q \times \#\text{Secrets}$.

Proofs for a Hiding Commitment in $\mathcal{R}_2$. Let us illustrate on the proof of validity of a hiding commitment in $\mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$, as presented in Section C.3. $\mathsf{Commit}^*(\mathsf{u}, \mathcal{R}_2)$, outputs the tuple $C^* = (E^*_{\mathsf{u}} = (E^*_k, E^{(2*)}_k)_k, \Pi^*_{\mathsf{u}})$, where for all indices $k \in [\![1; K]\!]$ and $m \in [\![1; M]\!]$, with $\rho_k, \rho'_k \xleftarrow{\$} \mathcal{M}$ and $\sigma_k, \sigma'_k \in \mathbb{Z}^*_{\mathcal{N}}$:

$$E^*_k \leftarrow \prod_{j,i} E_{k,j,i}^{u_{j,i}} \times E_{k,0,0}^{\rho_k} \times \sigma_k^{\mathcal{N}} \bmod \mathcal{N}^2$$
$$E^{(2*)}_k \leftarrow \prod_{j,i} E^{(2)}_{k,j,i}{}^{u_{j,i}} \times E_{k,0,0}^{\rho'_k} \times \sigma_k'^{\mathcal{N}} \bmod \mathcal{N}^2$$

with $\mathsf{ZKLQCheck}(X_1 - X_2 \cdot X_3, E^{(2*)}_k, E^*_k, E^{(2)}_{k,0,0}; E_{k,0,0}, E^{(2)}_{k,0,0}) = \mathsf{true}$ for which the verifier can compute the plaintexts in $E^{(2*)}_k$ and E^*_k, and then the quadratic relation a_k, that should be $a_k = \rho'_k \times 1 - \rho_k \times r^{(2)}_k \bmod q$:

$$V'_k = (1+\mathcal{N})^{a_k} = (1+\mathcal{N})^{\rho'_k \times 1 - \rho_k \times r_k^{(2)} \bmod q} \bmod \mathcal{N}^2$$

and the encodings in the proof Π^*_{u}:

$$V^*_{k,m} \leftarrow (1+\mathcal{N})^{\mathsf{v}_m(s_k,t_k) + \rho_{k,m} \bmod q} \cdot \sigma_{k,m}^{\mathcal{N}} \bmod \mathcal{N}^2$$
$$W^*_{k,m} \leftarrow (1+\mathcal{N})^{\mathsf{w}_m(s_k) + \rho'_{k,m} \bmod q} \cdot \sigma'_{k,m}{}^{\mathcal{N}} \bmod \mathcal{N}^2$$

for random $\rho_{k,m}, \rho'_{k,m} \xleftarrow{\$} \mathcal{M}$, chosen by the prover for their privacy, and some unknown $\sigma_{k,m}, \sigma'_{k,m} \in \mathbb{Z}^*_{\mathcal{N}}$, using random $x_m, y_m \xleftarrow{\$} \mathbb{Z}_q$ obtained by a hash function, and

$$\begin{aligned} &\mathsf{ZKLQCheck}(X_1 - (X_2 - y_m) \cdot X_3 - (X_4 - x_m) \cdot X_5, \\ &\qquad E^*_k, E_{k,1,0}, V^*_{k,m}, E_{k,0,1}, W^*_{k,m}; E_{k,0,0}, E_{k,0,0}, E^{(m)}_{k,1,0}, E^{(m)}_{k,0,1}) = \mathsf{true} \end{aligned}$$

where

$$E^{(m)}_{k,1,0} = E_{k,1,0} \cdot E_{k,0,0}^{-y_m} \bmod \mathcal{N}^2 \qquad E^{(m)}_{k,0,1} = E_{k,0,1} \cdot E_{k,0,0}^{-x_m} \bmod \mathcal{N}^2$$

Again, the verifier can compute the plaintexts in the input encodings, and the plaintext $b_{k,m}$ to be proven, that should be $b_{k,m} = \rho_k + \mathsf{u}(x_m, y_m) - \rho_{k,m}(t_k - y_m) - \rho'_{k,m}(s_k - x_m) \bmod q$. They build $V'_{k,m}$:

$$V'_{k,m} = (1+\mathcal{N})^{b_{k,m}} = (1+\mathcal{N})^{\rho_k + \mathsf{u}(x_m,y_m) - \rho_{k,m}(t_k - y_m) - \rho'_{k,m}(s_k - x_m) \bmod q} \bmod \mathcal{N}^2$$

As the same ρ_k and the same $u_m = \mathsf{u}(x_m, y_m)$ are used many times, the prover first randomly chooses $T_k, T'_k, v_m, T_{k,m}, T'_{k,m} \xleftarrow{\$} [\![0; q-1]\!]$, $\nu'_k, \nu'_{k,m} \xleftarrow{\$} [\![0; 2^\lambda Lq^2 - 1]\!]$, $\nu_k, \nu_{k,m} \xleftarrow{\$} \mathbb{Z}^*_{\mathcal{N}}$ and sets

$$D_k = E_{k,0,0}^{T'_k} \cdot (E_{k,0,0}^{(2)})^{-T_k} (1+\mathcal{N})^{q\nu'_k} \nu_k^{\mathcal{N}} \bmod \mathcal{N}^2$$
$$D_{k,m} = E_{k,0,0}^{T_k} \times E_{k,0,0}^{v_m} \times E_{k,1,0}^{(m)\ -T_{k,m}} \times E_{k,0,1}^{(m)\ -T'_{k,m}} (1+\mathcal{N})^{q\nu'_{k,m}} \nu_{k,m}^{\mathcal{N}} \bmod \mathcal{N}^2$$

The huge range for $\nu'_k, \nu'_{k,m} \xleftarrow{\$} [\![0; 2^\lambda Lq^2 - 1]\!]$ is to hide the random values even if then encodings come from L-linear combinations (which is not the case in this specific proof, but will be for relation (7), with v_κ).

A challenge $e \in [\![0; 2^{\lambda'} - 1]\!]$ is provided by the verifier or drawn from a hash function evaluated on the statement to be proven and all the $(D_k)_k$ and $(D_{k,m})_{k,m}$, and the proof eventually consists of $\Pi = ((D_k)_k, (D_{k,m})_{k,m}, (S_k, S'_k)_k, (w_m)_m, (S_{k,m}, S'_{k,m})_{k,m})$, where

$$S_k = T_k - e\rho_k \bmod q \qquad S'_k = T'_k - e\rho'_k \bmod q$$
$$w_m = v_m - eu_m \bmod q$$
$$S_{k,m} = T_{k,m} - e\rho_{k,m} \bmod q \qquad S'_{k,m} = T'_{k,m} - e\rho'_{k,m} \bmod q$$

It thus contains $K(M+1)$ ciphertexts (the number of equations), of $2\log\mathcal{N}$ bits, and $2K(M+1)+M$ scalars in $[\![0; q-1]\!]$ (the number of private masks), of less than $\log q$ bits.

The verifier can first compute the challenge e, and

$$D'_k = E_{k,0,0}^{S'_k} \cdot (E_{k,0,0}^{(2)})^{-S_k} \times (V'_k)^e \bmod \mathcal{N}^2$$
$$D'_{k,m} = E_{k,0,0}^{S_k} \times E_{k,0,0}^{w_m} \times E_{k,1,0}^{(m)\ -S_{k,m}} \times E_{k,0,1}^{(m)\ -S'_{k,m}} \times (V'_{k,m})^e \bmod \mathcal{N}^2$$

They then decrypt all the D_k/D'_k and $D_{k,m}/D'_{k,m}$, that should be 0 modulo q, as there is no reduction modulo $\mathcal{N}$ before the reduction modulo q, thanks to the constraint on $\mathcal{N} > 2\mu L \cdot 2^\lambda q^3$.

Soundness. After a rewinding, with a different challenge $\tilde{e} \neq e$, such that D'_k and $\tilde{D}'_k$ decrypt to the same value modulo q, and $D'_{k,m}$ and $\tilde{D}'_{k,m}$ decrypt to the same value modulo q, where

$$D'_k = E_{k,0,0}^{S'_k} \cdot (E_{k,0,0}^{(2)})^{-S_k} \times (V'_k)^e \bmod \mathcal{N}^2$$
$$\tilde{D}'_k = E_{k,0,0}^{\tilde{S}'_k} \cdot (E_{k,0,0}^{(2)})^{-\tilde{S}_k} \times (V'_k)^{\tilde{e}} \bmod \mathcal{N}^2$$
$$D'_{k,m} = E_{k,0,0}^{S_k} \times E_{k,0,0}^{w_m} \times E_{k,1,0}^{(m)\ -S_{k,m}} \times E_{k,0,1}^{(m)\ -S'_{k,m}} \times (V'_{k,m})^e \bmod \mathcal{N}^2$$
$$\tilde{D}'_{k,m} = E_{k,0,0}^{\tilde{S}_k} \times E_{k,0,0}^{\tilde{w}_m} \times E_{k,1,0}^{(m)\ -\tilde{S}_{k,m}} \times E_{k,0,1}^{(m)\ -\tilde{S}'_{k,m}} \times (V'_{k,m})^{\tilde{e}} \bmod \mathcal{N}^2$$

we have both

$$A_k = E_{k,0,0}^{S'_k - \tilde{S}'_k} \cdot (E_{k,0,0}^{(2)})^{\tilde{S}_k - S_k} \times (V'_k)^{e-\tilde{e}} \bmod \mathcal{N}^2$$
$$A_{k,m} = E_{k,0,0}^{S_k - \tilde{S}_k} \times E_{k,0,0}^{w_m - \tilde{w}_m} \times E_{k,1,0}^{(m)\ \tilde{S}_{k,m} - S_{k,m}} \times E_{k,0,1}^{(m)\ \tilde{S}'_{k,m} - S'_{k,m}} \times (V'_{k,m})^{e-\tilde{e}} \bmod \mathcal{N}^2$$

decrypt to 0 modulo q, while the small size of the answers and the evaluation of V'_k and $V'_{k,m}$ by the verifier on small scalars guarantees no wrap-up modulo $\mathcal{N}$: $q\alpha_k, q\alpha_{k,m} < 2\mu L q^3 + 2^{\lambda'} q < \mathcal{N}$, even with encodings generated from L-linear combinations in

$$A_k = (1+\mathcal{N})^{q\alpha_k}\beta_k^{\mathcal{N}} \bmod \mathcal{N}^2 \qquad A_{k,m} = (1+\mathcal{N})^{q\alpha_{k,m}}\beta_{k,m}^{\mathcal{N}} \bmod \mathcal{N}^2$$

If we assume $\tilde{e} - e$ invertible modulo q, and set $\varepsilon = (\tilde{e}-e)^{-1} \bmod q$, we can compute

$$\rho'_k \leftarrow (S'_k - \tilde{S}'_k)\cdot\varepsilon \bmod q \qquad \rho_k \leftarrow (S_k - \tilde{S}_k)\cdot\varepsilon \bmod q$$
$$u_m \leftarrow (w_m - \tilde{w}_m)\cdot\varepsilon \bmod q$$
$$\rho_{k,m} \leftarrow (S_{k,m} - \tilde{S}_{k,m})\cdot\varepsilon \bmod q \qquad \rho'_{k,m} \leftarrow (S'_{k,m} - \tilde{S}'_{k,m})\cdot\varepsilon \bmod q$$

all smaller than q

$$E_{k,0,0}^{\rho'_k}\cdot(E_{k,0,0}^{(2)})^{-\rho_k} = V'_k\cdot(1+\mathcal{N})^{q\alpha'_k}\gamma_k^{\mathcal{N}} \bmod \mathcal{N}^2$$
$$E_{k,0,0}^{\rho_k}\times E_{k,0,0}^{u_m}\times E_{k,1,0}^{(m)\ -\rho_{k,m}}\times E_{k,0,1}^{(m)\ -\rho'_{k,m}} = V'_{k,m}\cdot(1+\mathcal{N})^{q\alpha'_{k,m}}\gamma_{k,m}^{\mathcal{N}} \bmod \mathcal{N}^2$$

Eventually, if $\mathcal{N} > 2\mu L q^3$, all the exponents to $(1+\mathcal{N})$ remain smaller than $\mathcal{N}$, which proves the soundness, until $\tilde{e} - e$ is invertible: one can take $2^{\lambda'}$ smaller than the smallest prime factor of q, so that any non-trivial difference will always be invertible, and iterate several times in parallel with multiple challenges e, to increase soundness. For $\mathcal{N}$, it is safe to take it larger than $2\mu L 2^{\lambda} q^3$. As μ will always be smaller than 4, in each individual equations, we can take $\mathcal{N} > L 2^{\lambda+3} q^3$.

Zero-Knowledge. Thanks to the random masks in $\mathcal{M}$, the plaintexts are statistically hidden. The proofs contain tuples $((D_k)_k, (D_{k,m})_{k,m}, (S_k, S'_k)_k, (w_m)_m, (S_{k,m}, S'_{k,m})_{k,m})$, where the ciphertexts statistically hide their representations modulo q, and the scalars are uniformly distributed in $[\![0; q-1]\!]$. Hence, a statistical zero-knowledge proof that guarantees the hiding property of the commitments.

C Commitments

We here develop more on the sketch provided in Section 3. In particular, when q is large and prime, as already shown, under the Schwartz-Zippel lemma, the probability to have equalities between polynomials evaluated in a random point $s \in \mathbb{Z}_q$ whereas equalities do not hold between the polynomials of total degree at most $2n$ is less than $2n/q$, which is negligible. Then, the soundness is straightforward. When q is the product of ℓ distinct prime factors greater than p, this is less than $2n\ell/p$. This probability is unfortunately non-negligible for polynomial prime factors. In particular, for SEAL implementation of FV FHE, the prime factors are at most 60-bit long.

According to the expected soundness parameter ε, to reduce the probability of false positive cases, the natural solution is to iterate K times, with multiple evaluation points s_k, for $k \in [\![1; K]\!]$, so that $(2n\ell/p)^K \le \varepsilon$. But then, we have to make sure the same polynomials are committed in each point. This is discussed below.

C.1 Binding Commitments

Because of the linear-only combinations, one can limit encodings in sub-spaces. As we can also do quadratic verifications, we will be able to check products of two polynomials. From an encoding scheme $(\mathsf{Gen}, \mathsf{E})$ over a ring R, we can define a compact binding commitment scheme over multivariate polynomials. More precisely, such commitment schemes will be defined by 4 algorithms:

- $\mathsf{Setup}(1^\lambda, \mathcal{R}, (\mathcal{R}_\pi)_\pi)$ generates the public key pk and the verification key vk, according to the polynomial space $\mathcal{R}$, and the authorized subspaces $(\mathcal{R}_\pi)_\pi$;
- $\mathsf{Commit}_{\mathsf{pk}}(\mathsf{u}, \mathcal{R}_\pi)$, for a polynomial $\mathsf{u} \in \mathcal{R}_\pi \subseteq \mathcal{R}$, outputs a commitment C of u;
- $\mathsf{Validity}_{\mathsf{vk}}(C, \mathcal{R}_\pi)$, for a commitment C, verifies whether this is a valid commitment for an (unknown) polynomial u in the appropriate subspace $\mathcal{R}_\pi$;
- $\mathsf{Quadratic}_{\mathsf{vk}}(Q, C_1, \ldots, C_\ell)$, for valid commitments C_i of (unknown) polynomials u_i and a quadratic polynomial Q in ℓ variables, verifies whether $Q(\mathsf{u}_1, \ldots, \mathsf{u}_\ell) = 0$ in $\mathcal{R}$.

For the above definition to make sense, for any adversary $\mathcal{A}$, there exists an extractor $\mathsf{Ext}_{\mathcal{A}}$ such that for any valid commitment C generated by $\mathcal{A}$, $\mathsf{Ext}_{\mathcal{A}}$ outputs the committed polynomial $\mathsf{u} \in \mathcal{R}_\pi$.

While $\mathcal{R}$ will usually be a global ring of polynomials, such as $\mathsf{R}[X]$ or $\mathsf{R}[X, Y]$, the sub-spaces $\mathcal{R}_\pi$ will only be the module generated by a limited basis, $\mathsf{R}[X^{n-1}]$, $\mathsf{R}[Y^N]$, or $\mathsf{R}[X^{n-1}, Y^N]$, where the explicit exponents limit the degrees.

A Warm-Up with the Ring of Polynomials $\mathcal{R} = \mathsf{R}[X]$. Before dealing with multivariate polynomials, we start with univariate polynomials, to illustrate some requirements and some issues:

- $\mathsf{Setup}(1^\lambda, \mathcal{R} = \mathsf{R}[X], (\mathcal{R}_1 = \mathsf{R}[X^{n-1}]))$ first runs $(\mathsf{pk}', \mathsf{sk}', \mathsf{vk}') \leftarrow \mathsf{Gen}(1^\lambda)$, chooses a random element $s \xleftarrow{\$} \mathsf{R}^*$ and, for $i \in [\![0; n-1]\!]$, sets $E_i \leftarrow \mathsf{E}_{\mathsf{sk}'}(s^i)$. Then, the public key of the commitment scheme is $\mathsf{pk} = (\mathsf{pk}', \{E_i\}_i)$, while the verification key is $\mathsf{vk} = \mathsf{vk}'$;
- $\mathsf{Commit}_{\mathsf{pk}}(\mathsf{u}, \mathcal{R}_1)$, for a polynomial $\mathsf{u} = \sum_{i=0}^{n-1} u_i X^i \in \mathcal{R}_1 \subset \mathcal{R}$ of degree at most $n-1$, outputs $E = \mathsf{Eval}(\{E_i\}_i, \{u_i\}_i) = \mathsf{E}(\sum_i u_i s^i) = \mathsf{E}(\mathsf{u}(s))$;
- $\mathsf{Validity}_{\mathsf{vk}}(E)$ is exactly the Verify of the encoding scheme, as it outputs whether this is a valid encoding or not, and thus a valid commitment or not.

Thanks to the linear-only extractability, when a player generates a valid encoding (or commitment) E, being only given $(E_0, \ldots, E_{n-1})$, one can extract (c_i) such that E is an encoding of $c_0 + c_1 s + \ldots c_{n-1} s^{n-1}$ in R, and thus of the polynomial $\mathsf{c} = \sum_i c_i X^i$ in $\mathcal{R}$. Our above commitment scheme on polynomials is thus extractable under the linear-only extractability of the secure encoding scheme.

In addition, thanks to the quadratic verification on the encodings, if we have four polynomials u, v, m and r such that $\mathsf{m} = \mathsf{u} \cdot \mathsf{v} \bmod \mathsf{r}$, which means that $\mathsf{m} = \mathsf{u} \cdot \mathsf{v} + \mathsf{r} \cdot \mathsf{q}$, where all the polynomials are of degree at most $n-1$, we can check such a product: from valid commitments U and V of u and v, R and Q of r and q, respectively, and M of the polynomial m, all of degree at most $n-1$, as they are all simple encodings, $\mathsf{QCheck}(X_1 X_2 + X_3 X_4 - X_5, U, V, R, Q, M) = \mathsf{true}$ implies that $\mathsf{m}(s) = \mathsf{u}(s) \cdot \mathsf{v}(s) + \mathsf{r}(s) \cdot \mathsf{q}(s)$. According to the ring R, this might imply the expected relation, or not. If R is a large enough field, this is true.

However, in the following, we will be interested in the particular case of $\mathsf{R} = \mathbb{Z}_q$, with $q = p_1 \cdot \ldots \cdot p_\ell$ a product of ℓ prime integers $p_1 < \ldots < p_\ell$, greater than p. Having $\mathsf{m}(s) = \mathsf{u}(s) \cdot \mathsf{v}(s) + \mathsf{r}(s) \cdot \mathsf{q}(s) \bmod q$ while $\mathsf{m} \neq \mathsf{u} \cdot \mathsf{v} + \mathsf{r} \cdot \mathsf{q}$ in $\mathbb{Z}_q[X]$ means there is an index $j \in [\![1; \ell]\!]$ such that $\mathsf{m}(s) = \mathsf{u}(s) \cdot \mathsf{v}(s) + \mathsf{r}(s) \cdot \mathsf{q}(s) \bmod p_j$ while $\mathsf{m} \neq \mathsf{u} \cdot \mathsf{v} + \mathsf{r} \cdot \mathsf{q}$ in $\mathbb{Z}_{p_j}[X]$. Under the Schwartz-Zippel lemma [Sch80,Zip79], this probability is bounded by $2n/p_j$ for each j, as the total degree of the relation is at most $2n$. Hence, the probability over s to have a false positive is bounded by $2n\ell/p$. This probability is unfortunately non-negligible for polynomial prime factors.

C.2 Commitments on Bivariate Polynomials

We now build commitments on bivariate polynomials over $\mathbb{Z}_q$ for a composite q, providing a way to deal with scalars and univariate polynomials, when they are evaluated in one-fixed coordinate or a fixed point. In addition, a bivariate polynomial is a way to encode multiple univariate polynomials: given N polynomials $\mathsf{u}_j = \sum_i u_{j,i} X^i \in \mathbb{Z}_q[X]$, of degree $n-1$, we can consider the polynomial in $\mathbb{Z}_q[X, Y]$: $\mathsf{u}(X, Y) = \sum_j Y^j \mathsf{u}_j(X) = \sum_{j,i} u_{j,i} X^i Y^j$.

As explained above, in order to reduce the probability of errors with the Schwartz-Zippel lemma, we will use encodings in K multiple points. We will additionally prove they all encode the same polynomial.

$\mathsf{Setup}(1^\lambda, \mathcal{R} = \mathbb{Z}_q[X,Y], \mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N])$ first runs $(\mathsf{pk}', \mathsf{sk}', \mathsf{vk}') \leftarrow \mathsf{Gen}(1^\lambda)$, chooses K tuples of random elements $s_k, t_k \xleftarrow{\$} \mathbb{Z}_q^*$ and, for $k \in [\![1;K]\!]$, $i \in [\![0;n-1]\!]$, and $j \in [\![0;N]\!]$, sets $E_{k,j,i} \leftarrow \mathsf{E}_{\mathsf{sk}'}(s_k^i \cdot t_k^j)$. To explicitly limit to some degrees, such as $\mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$ or even to univariate polynomials. One also chooses K tuples of random elements $r_k^{(2)} \xleftarrow{\$} \mathbb{Z}_q^*$. Then, for $k \in [\![1;K]\!]$, $i \in [\![0;n-1]\!]$, and $j \in [\![0;N]\!]$, one sets $E^{(2)}_{k,j,i} \leftarrow \mathsf{E}_{\mathsf{sk}'}(r_k^{(2)} \cdot s_k^i \cdot t_k^j)$ for bivariate polynomials in $\mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$. The public key of the commitment scheme is composed of the above encodings $\mathsf{pk} = (\mathsf{pk}', \{E_{k,j,i}, E^{(2)}_{k,j,i}\}_{k,j,i})$, and the verification key is $\mathsf{vk} = \mathsf{vk}'$. For our application, other sub-spaces will be added, but in this section, we focus on $\mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$. More spaces are detailed in Appendix C, with other random values $r_k^{(\iota)}$.

$\mathsf{Commit}(\mathsf{u}, \mathbb{Z}_q[X^{n-1}, Y^N])$, for a polynomial $\mathsf{u} = \sum_{j=0}^{N} \sum_{i=0}^{n-1} u_{j,i} X^i Y^j$ in $\mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$, outputs $C = (E_\mathsf{u} = (E_k, E_k^{(2)})_k, \Pi_\mathsf{u})$, for $k \in [\![1;K]\!]$, with Π_u detailed later, where

$$E_k \leftarrow \mathsf{Eval}(\{E_{k,j,i}\}_{j,i}, \{u_{j,i}\}_{j,i}) = \mathsf{E}(\mathsf{u}(s_k, t_k))$$
$$E_k^{(2)} \leftarrow \mathsf{Eval}(\{E^{(2)}_{k,j,i}\}_{j,i}, \{u_{j,i}\}_{j,i}) = \mathsf{E}(r_k^{(2)} \cdot \mathsf{u}(s_k, t_k))$$
$$\text{such that } \mathsf{QCheck}(X_1 - X_2 \cdot X_3, E_k^{(2)}, E_k, E^{(2)}_{k,0,0}) = \mathsf{true}$$

The idea behind the twin encodings $E_\mathsf{u} = (E_k, E_k^{(2)})_k$ is that the first element E_k is compatible between all the polynomials in $\mathcal{R} = \mathbb{Z}_q[X,Y]$, independently of the constraints on the allowed monomials, while the second element restrains the polynomial space: the limited basis in $\{E^{(2)}_{k,j,i}\}_{j,i}$ and the relation with $E^{(2)}_{k,0,0}$ limits to $\mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$. Note there is still a non-negligible probability of false positives when accepting a twin encoding, as remarked above, as the quadratic check might accept the relation on the specific points without the relation holding on the polynomials. We will say an encoding E_k is *valid* when the relation really holds on the extracted polynomials from the twin encodings $(E_k, E_k^{(2)})$. Intuitively, for each index k, the probability of false positive (accepting an invalid twin encoding) is bounded by $\ell\mathcal{D}/p$, where $\mathcal{D}$ will be the maximal total degree of the polynomials that one will be able to generate from the elements in the basis provided as input, as $E^{(2)}_{k,0,0}$ encodes a polynomial of degree 0. An additional proof Π_u (either provided from an interactive protocol or built in a non-interactive way) is thus appended to the commitment to make the validity check sound, with error-probability ε_c for each commitment: a huge fraction of the encodings are valid and encode the same polynomial.

In this section, we target the soundness of the commitments, whereas our ultimate goal will be a soundness bound ε_s for our global proof. To achieve this soundness bound, we will first require all the commitments to be correct, excepted an error probability $\varepsilon_s/3$, the relations between the commitments to also hold excepted with error probability $\varepsilon_s/3$, and the bounds on the noise-flooding to be small enough with error probability $\varepsilon_s/3$. Hence, we use a specific soundness parameter $\varepsilon_c \leq \varepsilon_s/3\nu_c$ for commitments, where ν_c will be the total number of commitments.

$\mathsf{Validity}(C)$ first verifies the twin encodings, which checks the appropriate sub-spaces for each E_k: the quadratic check with the limited bases in $\{E^{(2)}_{k,j,i}\}_{k,j,i}$ guarantees the limited list of monomials, but with an error probability bounded by $2\ell\mathcal{D}/t$. Note that in this section, $\mathcal{D} = N + n - 1$, because of the limited $\mathcal{R}_2$, but monomials with higher degrees will be required later. With K large enough, we can expect a large number of valid encodings E_k: let us assume more than $K/5$ encodings are not valid, this will remain undetected with probability at most

$(\ell\mathcal{D}/p)^{K/5}$. If we impose $p \geq 2^S \times 2\ell\mathcal{D}$ (where S will be seen as a security margin, all along this analysis), the error probability is less than $2^{-(S+1)K/5}$. Hence, the number of valid encodings E_k is greater than $4K/5$ excepted with probability upper-bounded by $2^{-(S+1)K/5}$.

But this is not enough to amplify the above Schwartz-Zippel lemma: one also has to make sure that all the valid E_k encode the same polynomial u to exploit the iterations in the quadratic check between encoded polynomials:

$$\begin{aligned}\mathsf{u}_k(X,Y) - \mathsf{u}(X',Y') &= \mathsf{u}(X,Y) - \mathsf{u}(X',Y') \\ &= \mathsf{u}(X,Y) - \mathsf{u}(X,Y') + \mathsf{u}(X,Y') - \mathsf{u}(X',Y') \\ &= (Y-Y') \cdot \mathsf{v}(X,Y,Y') + (X-X') \cdot \mathsf{w}(X,X',Y').\end{aligned}$$

This can be checked with random $x_m, y_m \xleftarrow{\$} \mathbb{Z}_q$, sent by the verifier:

$$\mathsf{u}(X,Y) - \mathsf{u}(x_m,y_m) = (Y-y_m) \cdot \mathsf{v}_m(X,Y) + (X-x_m) \cdot \mathsf{w}_m(X),$$

from the proof

$$\Pi_{\mathsf{u}} = (u_m \leftarrow \mathsf{u}(x_m,y_m), (V_{k,m} \leftarrow \mathsf{E}(\mathsf{v}_m(s_k,t_k)), W_{k,m} \leftarrow \mathsf{E}(\mathsf{w}_m(s_k)))_k)_m$$

that can be checked as

$$\begin{gathered}\mathsf{QCheck}(X_1 - u_m - (X_2 - y_m) \cdot X_3 - (X_4 - x_m) \cdot X_5, \\ E_k, E_{k,1,0}, V_{k,m}, E_{k,0,1}, W_{k,m}) = \mathsf{true}\end{gathered}$$

For each $k \in [\![1;K]\!]$, if the above relations do not hold at the polynomial level (which means for the polynomials $\mathsf{v}_{k,m}$, $\mathsf{w}_{k,m}$, encoded in $V_{k,m}$, and $W_{k,m}$ respectively, possibly in any sub-space) for more than $1/3$ of the $m \in [\![1;M]\!]$, they will remain undetected with probability at most $(\ell(\mathcal{D}+1)/p)^{M/3} \leq 2^{-(S+1)M/3}$. Otherwise

$$\mathsf{u}_k(X,Y) - u_m = (Y-y_m) \cdot \mathsf{v}_{k,m} + (X-x_m) \cdot \mathsf{w}_{k,m}$$

for more than $2M/3$ indices m. Then, for any $k' \neq k$, that correspond to valid encodings E_k and $E_{k'}$, at least $M/3$ common values (x_m, y_m) satisfy both relations in k and k', hence $\mathsf{u}_k(x_m,y_m) = \mathsf{u}_{k'}(x_m,y_m) = u_m$. As a consequence, as the polynomials u_k and $\mathsf{u}_{k'}$ were committed before seeing (x_m, y_m), there are $M/3$ random points in which the two polynomials are equal. Then $\mathsf{u}_k = \mathsf{u}_{k'}$, excepted with probability upper-bounded by $(\ell\mathcal{D}/p)^{M/3} \leq 2^{-2M}$. A false acceptance for some pair (k,k') of consecutive valid encodings is upper-bounded by $2K \cdot 2^{-(S+1)M/3}$. Globally, the probability of having a false positive among the valid encodings E_k is bounded by $2K \cdot 2^{-(S+1)M/3}$.

We thus complete the commitments in $\mathcal{R}_2$: in addition to the twin encodings $E_{\mathsf{u}} = (E_k, E_k^{(2)})_k$, after having received (or seen) $(x_m, y_m)_m$, one completes the commitment with the proofs Π_{u}, for $k \in [\![1;K]\!]$, $m \in [\![1;M]\!]$

– in $\mathsf{Commit}(\mathsf{u}, \mathbb{Z}_q[X^{n-1}, Y^N])$, $\Pi_{\mathsf{u}} = (u_m, (V_{k,m}, W_{k,m})_k)_m$;

Then, we can state the following security result, which can be extended to other subspaces:

Theorem 8 (Knowledge-Soundness of Commitments in $\mathcal{R}_2$). *For any commitment $C = (E = (E_k, E_k^{(2)})_k, \Pi = (u_m, (V_{k,m}, W_{k,m})_k)_m)$, if C successfully passes all the validity checks and quadratic root checks, on randomly chosen $(x_m, y_m) \xleftarrow{\$} \mathbb{Z}_q^2$, there exists a polynomial $\mathsf{u} \in \mathcal{R}_2$ such that at least $4K/5$ of the twin encodings actually encode u (which can be extracted from the extractability of valid encodings), excepted with probability less than $2^{-(S+1)K/5} + 2K \cdot 2^{-(S+1)M/3}$, where $q = \prod_{i=1}^{\ell} p_i$, with all $p_i \geq 2^S \times 2\ell\mathcal{D}$, and $\mathcal{D}$ is the maximal degree of the polynomials in the subspace.*

Quadratic Root Checks. According to the above analysis, if we assume $p \geq 2^S \times 2\ell\mathcal{D}$, for all the accepted commitments, we know that we have at least $4K/5$ valid encodings of the same polynomials in all twin encodings:

- if the number of invalid encodings is greater than $K/5$, they will remain undetected with probability less than $2^{-(S+1)K/5}$.
- all these valid encodings contain the same polynomial u, excepted with probability less than $2K \cdot 2^{-(S+1)M/3}$.

Therefore, when all the verifications succeed for the commitment, there are at least $4K/5$ valid encodings on the same polynomial u, excepted with small error probability. Hence, when 4 commitments are involved in a quadratic check, at least $K/5$ common indices k correspond to valid encodings on the same polynomials in the 4 commitments. Those polynomials then satisfy the relation excepted with probability less than $(2\ell\mathcal{D}/p)^{K/5} \leq 2^{-S \cdot K/5}$.

More Parameters. Commitments in $\mathbb{Z}_q[X^{n-1}]$ thus consist of $2K + KM = K(M+2)$ encodings, and M scalars in $\mathbb{Z}_q$ while commitments in $\mathbb{Z}_q[X^{n-1}, Y^N]$ consist of $2K + 2KM = 2K(M+1)$ encodings, and M scalars in $\mathbb{Z}_q$.

The above analysis was for commitments that appear in quadratic checks involving $c = 4$ commitments, hence one required $4K/5$ valid encodings. When other values of c in $\{2, 3, 4\}$, this is enough to have $cK/(c+1)$ valid encodings in the c commitments to have $K/(c+1)$ common indices, and then the soundness of the equation is $2^{-S \cdot K/(c+1)}$.

Corollary 9. *When all the tests pass in a quadratic check between at most c prover-generated commitments, that are all valid, the relation is really satisfied on the committed polynomials (which can be extracted from the extractability of valid encodings), excepted with probability less than $2^{-S \cdot K/(c+1)}$.*

Bounds on K and M are detailed in Appendix G.

C.3 Hiding Commitments

These commitments with encodings are strongly binding, because of the knowledge-soundness, but are not hiding, as sometimes expected from commitments. Because of the quadratic verification, if the verifier hesitates between two polynomials in a commitment C, they can commit them and do a simple linear verification, as they know vk.

To statistically hide the content, one has to add random blinding elements from the appropriate masking set $\mathcal{M}$ to every encodings, to get hiding encodings. We illustrate this here with with $\mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$. We later detail the case of $\mathcal{R}_3 = \mathbb{Z}_q[Y^N]$.

$\mathsf{Commit}^*(\mathsf{u}, \mathbb{Z}_q[X^{n-1}, Y^N])$, the hiding commitment for a bivariate polynomial $\mathsf{u}(X,Y) = \sum_{j=0}^{N} \sum_{i=0}^{n-1} u_{j,i} X^i Y^j \in \mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$, outputs the tuple $C^* = (E^*_{\mathsf{u}} = (E^*_k, E^{(2*)}_k, \pi_k)_k, \Pi^*_{\mathsf{u}})$, where for all indices $k \in [\![1; K]\!]$ and $m \in [\![1; M]\!]$, with $\rho_k, \rho'_k \xleftarrow{\$} \mathcal{M}$:

$$
\begin{aligned}
E^*_k &\leftarrow \mathsf{Eval}(\{E_{k,j,i}\}_{j,i}, \{E_{k,0,0}\}, \{u_{j,i}\}_{j,i}, \{\rho_k\}) = \mathsf{E}(\mathsf{u}(s_k, t_k) + \rho_k) \\
E^{(2*)}_k &\leftarrow \mathsf{Eval}(\{E^{(2)}_{k,j,i}\}_{j,i}, \{E_{k,0,0}\}, \{u_{j,i}\}_{j,i}, \{\rho'_k\}) = \mathsf{E}(r^{(2)}_k \cdot \mathsf{u}(s_k, t_k) + \rho'_k) \\
\text{with } \pi_k &= \{\mathsf{ZKLQCheck}(X_1 - X_2 \cdot X_3, E^{(2*)}_k, E^*_k, E^{(2)}_{k,0,0}; E_{k,0,0}, E^{(2)}_{k,0,0}) = \mathsf{true}\}
\end{aligned}
$$

as the quadratic relation is equal to $\rho'_k \times 1 - \rho_k \times r^{(2)}_k$, with private scalars ρ_k and ρ'_k, and a proof, using random $x_m, y_m \xleftarrow{\$} \mathbb{Z}_q$ sent by the verifier, of

$$
\begin{aligned}
\mathsf{u}(X,Y) - \mathsf{u}(x_m, y_m) &= (Y - y_m) \cdot \mathsf{v}_m(X,Y) + (X - x_m) \cdot \mathsf{w}_m(X) \\
&= (Y - y_m) \cdot \mathsf{v}_m + (X - x_m) \cdot \mathsf{w}_m
\end{aligned}
$$

which can be verified with

$$\Pi^*_{\mathsf{u}} = (V^*_{k,m} \leftarrow \mathsf{E}(\mathsf{v}_m(s_k, t_k) + \rho_{k,m}), W^*_{k,m} \leftarrow \mathsf{E}(\mathsf{w}_m(s_k) + \rho'_{k,m}), \pi_{k,m})_{k,m}$$

for random $\rho_{k,m}, \rho'_{k,m} \xleftarrow{\$} \mathcal{M}$, chosen by the prover for their privacy, and

$$\begin{aligned}\pi_{k,m} = \{&\mathsf{ZKLQCheck}(X_1 - (X_2 - y_m) \cdot X_3 - (X_4 - x_m) \cdot X_5,\\ &E^*_k, E_{k,1,0}, V^*_{k,m}, E_{k,0,1}, W^*_{k,m}; E_{k,0,0}, E_{k,0,0}, E^{(m)}_{k,1,0}, E^{(m)}_{k,0,1}) = \mathsf{true}\}\end{aligned}$$

where anyone can compute

$$\begin{aligned}E^{(m)}_{k,1,0} &= \mathsf{Eval}(\{E_{k,1,0}, E_{k,0,0}\}, \{1, -y_m\}) = \mathsf{E}(t_k - y_m)\\ E^{(m)}_{k,0,1} &= \mathsf{Eval}(\{E_{k,0,1}, E_{k,0,0}\}, \{1, -x_m\}) = \mathsf{E}(s_k - x_m)\end{aligned}$$

as the quadratic relation is equal to $\rho_k + \mathsf{u}(x_m, y_m) - \rho_{k,m}(t_k - y_m) - \rho'_{k,m}(s_k - x_m)$. One thus proves their knowledge of the 4 private scalars, $\rho_k \in \mathcal{M}$ (the same as above), $u_m = \mathsf{u}(x_m, y_m) \in \mathsf{R}$ (the same for all the k's), and $\rho_{k,m}, \rho'_{k,m} \in \mathcal{M}$.

Let us assume that a reasonable fraction of the twin encodings $(E^*_k, E^{(2*)}_k)$ are valid for a polynomial u^*_k in $\mathbb{Z}_q[X^{n-1}, Y^N]$. For each k, the quadratic check guarantees that

$$\begin{aligned}\mathsf{u}^*_k(X, Y) = \mathsf{u}_k(X, Y) + \rho_k = (Y - y_m) \cdot \mathsf{v}_{k,m} + (X - x_m) \cdot \mathsf{w}_{k,m}\\ + \rho_k + u_m - \rho_{k,m} \cdot (Y - y_m) - \rho'_{k,m} \cdot (X - x_m)\end{aligned}$$

excepted with the same error probability as in Theorem 8. On the other hand, one can state:

Theorem 10 (Hiding Property of Commitments in $\mathcal{R}_2$). *For any commitment $C^* = (E^* = (E^*_k, E^{(2*)}_k, \pi_k)_k, \Pi^* = (V^*_{k,m}, W^*_{k,m}, \pi_{k,m})_{k,m})$, thanks to the random masks, and the zero-knowledge proofs, all the encodings are statistically indistinguishable from random encodings.*

Note that for a hiding commitment, we have zero-knowledge proofs on $K(M+1)$ equations involving globally $2K(M+1)+M$ private scalars (but at most 4 in each equation). In Appendix B.3, we detail the zero-knowledge proofs when the encoding relies on the Paillier's encryption scheme: soundness is $2^{-\lambda'}$, for $\lambda' < \log_2 p$, then one needs to iterate $-\log_2(\varepsilon_s/3\nu_c)/\lambda'$ times. Each proof consists of $K(M+1)$ Paillier ciphertexts and $2K(M+1)+M$ scalars in $\mathbb{Z}_q$, iterated $\log(3\nu_c/\varepsilon_s)/\log p$ times. Soundness also implies the RSA modulus needs to be larger than $L \cdot 2^{\lambda+3} q^3$.

C.4 Quadratic Root Check

The check $\mathsf{Quadratic}(Q, C_1, \ldots, C_\ell)$, on non-hiding commitments, for a quadratic polynomial Q, that verifies whether the committed polynomials P_i's in the C_i's satisfy the relation $Q(P_1, \ldots, P_\ell) = 0$, is performed as a QCheck on the encodings for each index k. Since a huge fraction of the indices k encode the same polynomials, for the prover-generated commitments, iterations amplify the soundness.

When some hiding commitments are involved, one additionally has to prove the appropriate blinding factors. More concretely, let us consider the binding commitment D of d, and the hiding commitments C^* and H^* of c^* and h^* respectively, with blinding factors $\rho_k, \sigma_k \in \mathcal{M}$ respectively. The relation $\mathsf{c}^* = \mathsf{d} \times \mathsf{h}^*$ can be checked as:

$$\begin{aligned}&\mathsf{Quadratic}(X_1 - X_2 \cdot X_3, C^*, D, H^*)\\ &\qquad = \mathsf{ZKLQCheck}(X_1 - X_2 \cdot X_3, C^*_k, D_k, H^*_k; E_{k,0,0}, D_k) \text{ for all } k\end{aligned}$$

as the quadratic relation is equal to $\rho_k \cdot 1 - \sigma_k \cdot \mathsf{d}$, where ρ_k and σ_k are the same private values as the ones used in the validity verification of the hiding commitments C^* and H^*.

We stress that we only allow quadratic verifications where at most one polynomial is committed in a hiding way in each quadratic product. In the end, we know that the quadratic equations among polynomials are satisfied in at least $K/5$ random points. They are thus satisfied by the polynomials excepted with probability less than $2^{-S \cdot K/5}$, which is chosen to be much less than $\varepsilon_s/3\nu_e$, where ν_e is the global number of equations to be checked.

C.5 Commitments in Additional Subspaces

As already explained, in our application, we are considering $q = p_1 \cdot \ldots \cdot p_\ell$, a composite modulus q, with ℓ prime factors $p_1 < \ldots < p_\ell$ larger than p. We will work in subspaces of $\mathcal{R} = \mathbb{Z}_q[X^{n-1}, Y^{2N}]$. We will consider the subspaces $\mathcal{R}_1 = \mathbb{Z}_q[X^{n-1}]$, $\mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$, $\mathcal{R}_3 = \mathbb{Z}_q[Y^N]$, and $\mathbb{Z}_q[Y^{2N}]$ with no term in Y^N, denoted $\mathcal{R}_4 = \mathbb{Z}_q[Y^{2N\setminus N}]$, hence $\mathcal{D} = 2N$, so we will assume $p \geq 2^S \times 4N\ell$. Details on the complete construction of the commitments with system setup and detailed contents and checks are given in the following section.

In our proof of inner product, we will use polynomials in multiple subspaces of $\mathcal{R} = \mathbb{Z}_q[X, Y]$. In addition to $\mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$, we will use $\mathcal{R}_1 = \mathbb{Z}_q[X^{n-1}]$, $\mathcal{R}_3 = \mathbb{Z}_q[Y^N]$, and $\mathbb{Z}_q[Y^{2N}]$ with no term in Y^N, denoted $\mathcal{R}_4 = \mathbb{Z}_q[Y^{2N\setminus N}]$. For this, we draw additional random elements $r_k^{(1)}, r_k^{(3)}, r_k^{(4)} \xleftarrow{\$} \mathbb{Z}_q^*$ for $k \in [\![1; K]\!]$, and set:

$$\begin{aligned} E_{k,i}^{(1)} &\leftarrow \mathsf{E}(r_k^{(1)} \cdot s_k^i) && i \in [\![0; n-1]\!] \\ E_{k,j}^{(3)} &\leftarrow \mathsf{E}(r_k^{(3)} \cdot t_k^j) && j \in [\![0; N]\!] \\ E_{k,j}^{(4)} &\leftarrow \mathsf{E}(r_k^{(4)} \cdot t_k^j) && j \in [\![0; 2N]\!] \setminus \{N\} \end{aligned}$$

and we extend

$$E_{k,j,i} \leftarrow \mathsf{E}(s_k^i \cdot t_k^j) \qquad i \in [\![0; n-1]\!], j \in [\![0; 2N]\!]$$

Note that $\mathcal{D}$ becomes $\max\{N + n - 1, 2N\} = 2N$, for $N \geq n$. We then commit the polynomials $\mathsf{Commit}(\mathsf{u}, \mathbb{Z}_q[Y^N])$ or $\mathsf{Commit}(\mathsf{u}, \mathbb{Z}_q[Y^{2N\setminus N}])$ with appropriate twin encodings, that can be verified as above: for each pair $k \neq k'$ of valid encodings:

$$\mathsf{u}_k(Y) - \mathsf{u}_{k'}(Y') = \mathsf{u}(Y) - \mathsf{u}(Y') = (Y - Y') \cdot \mathsf{v}(Y, Y')$$

and so for a random $y_m \xleftarrow{\$} \mathbb{Z}_q$ chosen by the prover:

$$\mathsf{u}(Y) - \mathsf{u}(y_m) = (Y - y_m) \cdot \mathsf{v}(Y, y_m) = (Y - y_m) \cdot \mathsf{v}_m,$$

from

$$\Pi_{\mathsf{u}} = (u_m \leftarrow \mathsf{u}(y_m), (V_{k,m} \leftarrow \mathsf{E}(\mathsf{v}_m(t_k)))_k)_m$$

as

$$\mathsf{QCheck}(X_1 - u_m - (X_2 - y_m) \cdot X_3, E_k, E_{k,1,0}, V_{k,m}) = \mathsf{true}$$

and the rest of the proof follows as in Section C.2, with thus $\Pi_{\mathsf{u}} = (u_m, (V_{k,m})_k)_m$, for $k \in [\![1; K]\!]$, $m \in [\![1; M]\!]$.

C.6 Univariate Hiding Commitments

$\mathsf{Commit}^*(\mathsf{u}, \mathbb{Z}_q[Y^N])$, the hiding commitment for a univariate polynomial $\mathsf{u}(Y) = \sum_{j=0}^{N} u_j Y^j \in \mathcal{R}_3 = \mathbb{Z}_q[Y^N]$, outputs the tuple $C = (E_{\mathsf{u}}, (E_k, E_k^{(3*)})_k, \Pi_{\mathsf{u}})$, where for all indices $k \in [\![1; K]\!]$ and $m \in [\![1; M]\!]$ with $\rho_k \xleftarrow{\$} \mathcal{M}$:

$$\begin{aligned} E_k &\leftarrow \mathsf{Eval}(\{E_{k,j,0}\}_j, \{F_k\}, \{u_j\}_j, \{\rho_k\}) = \mathsf{E}(\mathsf{u}(t_k) + \rho_k \cdot r_k), \\ E_k^{(3*)} &\leftarrow \mathsf{Eval}(\{E_{k,j,0}^{(3*)}\}_j, \{F_k^{(3*)}\}, \{u_j\}_j, \{\rho_k\}) = \mathsf{E}(r_k^{(3*)}(\mathsf{u}(t_k) + \rho_k \cdot r_k)) \\ &\text{with } \mathsf{QCheck}(X_1 - X_2 \cdot X_3, E_k^{(3*)}, E_k, E_{k,0}^{(3*)}) = \mathsf{true} \end{aligned}$$

and a proof, using random $y_m \xleftarrow{\$} \mathbb{Z}_q$ sent by the verifier, of $\mathsf{u}(Y) - \mathsf{u}(y_m) = (Y - y_m) \cdot \mathsf{v}_m(Y)$ which can be checked with $\Pi_{\mathsf{u}} = (V_{k,m} \leftarrow \mathsf{E}(\mathsf{v}_m(t_k) + \rho_{k,m} \cdot r_k))_{k,m}$ for random $\rho_{k,m} \xleftarrow{\$} \mathbb{Z}_q^*$, chosen by the prover for their privacy, and

$$\mathsf{ZKLQCheck}(X_1 - (X_2 - y_m) \cdot X_3, E_k^*, E_{k,1,0}, V_{k,m}; E_{k,0,0}, F_k^{(t)}, F_k) = \mathsf{true}$$

where one proves the knowledge of the 3 scalars $u_m = \mathsf{u}(y_m)$, $\rho'_{k,m} = \rho_k - \rho_{k,m} \cdot y_m$ and $\rho_{k,m}$ so that the above relation is equal to

$$u_m \times 1 + \rho_{k,m} \times r_k \cdot t_k - \rho'_{k,m} \times r_k \bmod q.$$

The same analysis as for the $\mathcal{R}_2$ case can be done for the soundness of the proof, but with less strict bounds, as degrees are lower.

C.7 Complete Construction of the Commitment

In the following, we are considering $q = p_1 \cdot \ldots \cdot p_\ell$, a composite modulus q, with ℓ distinct prime factors $p_1 < \ldots < p_\ell$ larger than p. We will work in subspaces of $\mathcal{R} = \mathbb{Z}_q[X^{n-1}, Y^{2N}]$. We will consider the subspaces $\mathcal{R}_1 = \mathbb{Z}_q[X^{n-1}]$, $\mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$, $\mathcal{R}_3 = \mathbb{Z}_q[Y^N]$, and $\mathcal{R}_4 = \mathbb{Z}_q[Y^{2N \setminus N}]$, hence $\mathcal{D} = 2N$, so we will assume $p \geq 128N\ell$.

Setup of the System: $\mathsf{Setup}(1^\lambda, \mathcal{R}, (\mathcal{R}_i)_i)$ first runs $(\mathsf{pk}', \mathsf{vk}') \leftarrow \mathsf{Gen}(1^\lambda)$, chooses K tuples of random elements $s_k, t_k \xleftarrow{\$} \mathbb{Z}_q^*$, as well as K tuples of random elements $r_k^{(1)}, r_k^{(2)}, r_k^{(3)}, r_k^{(4)}, \xleftarrow{\$} \mathbb{Z}_q^*$, to limit the combinations of the bases. Then, for $k \in [\![1; K]\!]$, one sets

$\mathcal{R} = \mathbb{Z}_q[X^{n-1}, Y^N] + \mathbb{Z}_q[Y^{2N}]$	$E_{k,j,i} \leftarrow \mathsf{E}(s_k^i \cdot t_k^j)$	$(i,j) \in ([\![0; n-1]\!] \times [\![0; N]\!]) \cup (\{0\} \times [\![N+1; 2N]\!])$
$\mathcal{R}_1 = \mathbb{Z}_q[X^{n-1}]$	$E_{k,i}^{(1)} \leftarrow \mathsf{E}(r_k^{(1)} \cdot s_k^i)$	$i \in [\![0; n-1]\!]$
$\mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$	$E_{k,j,i}^{(2)} \leftarrow \mathsf{E}(r_k^{(2)} \cdot s_k^i \cdot t_k^j)$	$i \in [\![0; n-1]\!], j \in [\![0; N]\!]$
$\mathcal{R}_3 = \mathbb{Z}_q[Y^N]$	$E_{k,j}^{(3)} \leftarrow \mathsf{E}(r_k^{(3)} \cdot t_k^j)$	$j \in [\![0; N]\!]$
$\mathcal{R}_4 = \mathbb{Z}_q[Y^{2N \setminus N}]$	$E_{k,j}^{(4)} \leftarrow \mathsf{E}(r_k^{(4)} \cdot t_k^j)$	$j \in [\![0; 2N]\!] \setminus \{N\}$

Then, the public key of the commitment scheme is set to pk' with all these encodings, while the verification key is vk'. For $\mathcal{R}$, we can limit to $\mathbb{Z}_q[X^{n-1}, Y^N] + \mathbb{Z}_q[Y^{2N}]$, with $(n+1) \times N + n$ encodings in the public key, sent once for many proofs.

Commitment Generation: there are two commitment algorithm, with or without the hiding property.

Non-Hiding Commitment. We have defined commitments on specific subspaces: $\mathcal{R}_1 = \mathbb{Z}_q[X^{n-1}]$, $\mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$, $\mathcal{R}_3 = \mathbb{Z}_q[Y^N]$, and $\mathcal{R}_4 = \mathbb{Z}_q[Y^{2N\setminus N}]$:

$\mathsf{Commit}(\mathsf{u}, \mathbb{Z}_q[X^{n-1}])$: it outputs $C = (E_\mathsf{u} = (E_k, E_k^{(1)})_k, \Pi_\mathsf{u})$, for $k \in [\![1; K]\!]$, where

$$E_k \leftarrow \mathsf{E}(\mathsf{u}(s_k)) \qquad\qquad E_k^{(1)} \leftarrow \mathsf{E}(r_k^{(1)} \cdot \mathsf{u}(s_k)),$$

and for all $m \in [\![1; M]\!]$: $x_m \xleftarrow{\$} \mathbb{Z}_q$ chosen by the verifier (or from a hash function for a non-interactive proof), and then

$$\Pi_\mathsf{u} = (u_m \leftarrow \mathsf{u}(x_m), (W_{k,m} \leftarrow \mathsf{E}(\mathsf{w}_m(s_k)))_k)_m$$

where w_m is such that $\mathsf{u}(X) - u_m = (X - x_m) \cdot \mathsf{w}_m(X)$: in total, these are $2K$ encodings, plus M scalars and KM encodings for the proof.

$\mathsf{Commit}(\mathsf{u}, \mathbb{Z}_q[X^{n-1}, Y^N])$: it outputs $C = (E_\mathsf{u} = (E_k, E_k^{(2)})_k, \Pi_\mathsf{u})$, for $k \in [\![1; K]\!]$, where

$$E_k \leftarrow \mathsf{E}(\mathsf{u}(s_k, t_k)) \qquad\qquad E_k^{(2)} \leftarrow \mathsf{E}(r_k^{(2)} \cdot \mathsf{u}(s_k, t_k)),$$

and for all $m \in [\![1; M]\!]$: $(x_m, y_m) \xleftarrow{\$} \mathbb{Z}_q^2$ chosen by the verifier (or from a hash function for a non-interactive proof), and then

$$\Pi_\mathsf{u} = (u_m \leftarrow \mathsf{u}(x_m, y_m), (V_{k,m} \leftarrow \mathsf{E}(\mathsf{v}_m(s_k, t_k)), W_{k,m} \leftarrow \mathsf{E}(\mathsf{w}_m(s_k)))_k)_m$$

where v_m and w_m are such that

$$\mathsf{u}(X, Y) - u_m = (Y - y_m) \cdot \mathsf{v}_m(X, Y) + (X - x_m) \cdot \mathsf{w}_m(X).$$

In total, these are $2K$ encodings, plus M scalars and $2KM$ encodings for the proof.

$\mathsf{Commit}(\mathsf{u}, \mathbb{Z}_q[Y^N])$: it outputs $C = (E_\mathsf{u} = (E_k, E_k^{(3)})_k, \Pi_\mathsf{u})$, for $k \in [\![1; K]\!]$, where:

$$E_k \leftarrow \mathsf{E}(\mathsf{u}(t_k)) \qquad\qquad E_k^{(3)} \leftarrow \mathsf{E}(r_k^{(3)} \cdot \mathsf{u}(t_k)),$$

and for all $m \in [\![1; M]\!]$: $y_m \xleftarrow{\$} \mathbb{Z}_q$ chosen by the verifier (or from a hash function for a non-interactive proof), and then

$$\Pi_\mathsf{u} = (u_m \leftarrow \mathsf{u}(y_m), (V_{k,m} \leftarrow \mathsf{E}(\mathsf{v}_m(t_k)))_k)_m$$

where v_m is such that $\mathsf{u}(Y) - u_m = (Y - y_m) \cdot \mathsf{v}_m(Y)$. In total, these are $2K$ encodings, plus M scalars and KM encodings for the proof.

$\mathsf{Commit}(\mathsf{u}, \mathbb{Z}_q[Y^{2N\setminus N}])$: as the previous case but replacing $r^{(3)}$ with $r^{(4)}$ and analogously (3) indices with (4) indices.

Hiding Commitment. We only focus on $\mathcal{R}_2$ and $\mathcal{R}_3$:

$\mathsf{Commit}^*(\mathsf{u}, \mathbb{Z}_q[X^{n-1}, Y^N])$: it outputs $C^* = (E_\mathsf{u}^* = (E_k^*, E_k^{(2*)}, \pi_k)_k, \Pi_\mathsf{u}^*)$, for $k \in [\![1; K]\!]$, where, with $\rho_k, \rho'_k \xleftarrow{\$} \mathcal{M}$, where $\mathcal{M}$ is the appropriate masking set:

$$\begin{aligned} E_k^* &\leftarrow \mathsf{Eval}(\{E_{k,j,i}\}_{j,i}, \{E_{k,0,0}\}, \{u_{j,i}\}_{j,i}, \{\rho_k\}) = \mathsf{E}(\mathsf{u}(s_k, t_k) + \rho_k) \\ E_k^{(2*)} &\leftarrow \mathsf{Eval}(\{E_{k,j,i}^{(2)}\}_{j,i}, \{E_{k,0,0}\}, \{u_{j,i}\}_{j,i}, \{\rho'_k\}) = \mathsf{E}(r_k^{(2)} \cdot \mathsf{u}(s_k, t_k) + \rho'_k) \\ \text{with } \pi_k &= \{\mathsf{ZKLQCheck}(X_1 - X_2 \cdot X_3, E_k^{(2*)}, E_k^*, E_{k,0,0}^{(2)}; E_{k,0,0}, E_{k,0,0}^{(2)}) = \mathsf{true}\} \end{aligned}$$

and for all $m \in [\![1; M]\!]$: $(x_m, y_m) \xleftarrow{\$} \mathbb{Z}_q^2$ chosen by the verifier (or from a hash function for a non-interactive proof), and then for random $\rho_{k,m}, \rho'_{k,m} \xleftarrow{\$} \mathbb{Z}_q^*$ chosen by the prover for their privacy:

$$\Pi_\mathsf{u}^* = (V_{k,m}^* \leftarrow \mathsf{E}(\mathsf{v}_m(s_k, t_k) + \rho_{k,m}), W_{k,m}^* \leftarrow \mathsf{E}(\mathsf{w}_m(s_k) + \rho'_{k,m})_{k,m}$$

where v_m and w_m are such that

$$\mathsf{u}(X,Y) - \mathsf{u}(x_m,y_m) = (Y-y_m)\cdot \mathsf{v}_m + (X-x_m)\cdot \mathsf{w}_m$$

with

$$\pi_{k,m} = \{\mathsf{ZKLQCheck}(X_1 - (X_2-y_m)\cdot X_3 - (X_4-x_m)\cdot X_5,\\ E_k^*, E_{k,1,0}, V_{k,m}^*, E_{k,0,1}, W_{k,m}^*; E_{k,0,0}, E_{k,0,0}, E_{k,1,0}^{(m)}, E_{k,0,1}^{(m)}) = \mathsf{true}\}$$

KM additional zero-knowledge proofs of knowledge of 4 scalars $\rho_{k,m}$, $\rho'_{k,m}$, ρ_k, and $u_m = \mathsf{u}(x_m,y_m)$. In total, this is $2K$ encodings, plus KM encodings for the proof, and KM zero-knowledge proofs of 4 scalars.

$\mathsf{Commit}^*(\mathsf{u}, \mathbb{Z}_q[Y^N])$: it outputs $C^* = (E_{\mathsf{u}}^* = (E_k^*, E_k^{(3*)}, \pi_k)_k, \Pi_{\mathsf{u}}^*)$, for $k \in [\![1;K]\!]$, where, with $\rho_k, \rho'_k \xleftarrow{\$} \mathcal{M}$:

$$E_k \leftarrow \mathsf{E}(\mathsf{u}(t_k) + \rho_k) \qquad\qquad E_k^{(3*)} \leftarrow \mathsf{E}(r_k^{(3)}\cdot \mathsf{u}(t_k) + \rho'_k)$$

with $\pi_k = \{\mathsf{ZKLQCheck}(X_1 - X_2\cdot X_3, E_k^{(3*)}, E_k^*, E_{k,0,0}^{(3)}; E_{k,0,0}, E_{k,0,0}^{(3)}) = \mathsf{true}\}$, and for all $m \in [\![1;M]\!]$: $y_m \xleftarrow{\$} \mathbb{Z}_q$ chosen by the verifier (or from a hash function for a non-interactive proof), and then for random $\rho_{k,m} \xleftarrow{\$} \mathbb{Z}_q^*$ chosen by the prover for their privacy:

$$\Pi_{\mathsf{u}}^* = (V_{k,m}^* \leftarrow \mathsf{E}(\mathsf{v}_m(t_k) + \rho_{k,m}))_{k,m}$$

for v_m such that $\mathsf{u}(Y) - \mathsf{u}(y_m) = (Y-y_m)\cdot \mathsf{v}_m(Y)$ with KM additional zero-knowledge proofs of knowledge of 3 scalars $\rho_{k,m}$, ρ_k, and $u_m = \mathsf{u}(y_m)$. In total, this is $2K$ encodings, plus KM encodings for the proof, and KM zero-knowledge proofs of 3 scalars.

Validity Check: it also depends on hiding or non-hiding commitments and on the space $\mathcal{R}_\pi$.

Non-Hiding Commitment. $\mathsf{Validity}(C, \mathcal{R}_\pi)$ first checks the twin encodings, for $k \in [\![1;K]\!]$:

$$\begin{aligned}
&\mathcal{R}_1 && \mathsf{QCheck}(X_1 - X_2\cdot X_3, E_k^{(1)}, E_k, E_{k,0}^{(1)}) = \mathsf{true}\\
&\mathcal{R}_2 && \mathsf{QCheck}(X_1 - X_2\cdot X_3, E_k^{(2)}, E_k, E_{k,0,0}^{(2)}) = \mathsf{true}\\
&\mathcal{R}_3, \mathcal{R}_4 && \mathsf{QCheck}(X_1 - X_2\cdot X_3, E_k^{(3/4)}, E_k, E_{k,0}^{(3/4)}) = \mathsf{true}
\end{aligned}$$

and then, for $k \in [\![1;K]\!]$ and $m \in [\![1;M]\!]$, either

$$\begin{aligned}
&\mathcal{R}_1 && \mathsf{QCheck}(X_1 - u_m - (X_2-x_m)\cdot X_3, E_k, E_{k,0,1}, W_{k,m}) = \mathsf{true}\\
&\mathcal{R}_2 && \mathsf{QCheck}(X_1 - u_m - (X_2-y_m)\cdot X_3 - (X_4-x_m)\cdot X_5,\\
& && \qquad E_k, E_{k,1,0}, V_{k,m}, E_{k,0,1}, W_{k,m}) = \mathsf{true}\\
&\mathcal{R}_3, \mathcal{R}_4 && \mathsf{QCheck}(X_1 - u_m - (X_2-y_m)\cdot X_3, E_k, E_{k,1,0}, V_{k,m}) = \mathsf{true}
\end{aligned}$$

Hiding Commitment. $\mathsf{Validity}^*(C, \mathcal{R}_\pi)$ first checks the twin encodings, for $k \in [\![1;K]\!]$:

$$\begin{aligned}
&\mathcal{R}_2 && \mathsf{QCheck}(X_1 - X_2\cdot X_3, E_k^{(2*)}, E_k, E_{k,0,0}^{(2)}) = \mathsf{true}\\
&\mathcal{R}_3 && \mathsf{QCheck}(X_1 - X_2\cdot X_3, E_k^{(3*)}, E_k, E_{k,0}^{(3)}) = \mathsf{true}
\end{aligned}$$

and then, for $k \in [\![1;K]\!]$ and $m \in [\![1;M]\!]$, the zero-knowledge proofs

$$\begin{aligned}
&\mathcal{R}_2 && \mathsf{ZKLQCheck}(X_1 - (X_2-y_m)\cdot X_3 - (X_4-x_m)\cdot X_5,\\
& && \qquad E_k^*, E_{k,1,0}, V_{k,m}^*, E_{k,0,1}, W_{k,m}^*; E_{k,0,0}, E_{k,0,0}, E_{k,1,0}^{(m)}, E_{k,0,1}^{(m)}) = \mathsf{true}\\
&\mathcal{R}_3 && \mathsf{ZKLQCheck}(X_1 - (X_2-y_m)\cdot X_3, E_k, E_{k,1,0}, V_{k,m}^*; E_{k,0,0}, E_{k,1,0}^{(m)}, E_{k,0,1}^{(m)}) = \mathsf{true}
\end{aligned}$$

D More Detail on the Protocol

Here is more detail on some steps of our protocol, when explained succinctly in Section 5, with a more general description than just OPE and vector of powers.

We indeed consider the receiver/verifier with their private input message m to learn in a verifiable way the inner product of the vector $\Phi(\mathsf{m}) = (\phi(\mathsf{m}, j))_{j\in[\![0;N]\!]}$ with a private vector $\mathsf{F} = (\mathsf{f}_j)_{j\in[\![0;N]\!]}$, committed by the sender/prover, where ϕ is a function known by them both, depending on the application. If $\phi(\mathsf{m}, j) = \mathsf{m}^j$, the inner product corresponds to the Oblivious Polynomial Evaluation (OPE) of the polynomial F with coefficients $(\mathsf{f}_j)_j$ on the message m; if $\phi(\mathsf{m}, j) = \delta_{\mu,j}$, with δ the Kronecker symbol and $\mu \in [\![0; N]\!]$ such that m is the representation of μ, it provides the μ-th coefficient of f, which would coincide with a Symmetric Private Information Retrieval (SPIR) application (see Appendix E). We now show how the sender can provide this evaluation with fully homomorphic encryption in a provable way.

More precisely, we consider the above FV scheme that encrypts messages from $\mathcal{R}_t = \mathbb{Z}_t[X]/\mathsf{r}(X)$, where $\mathsf{r} = X^n + 1$, into $\mathcal{R}_q = \mathbb{Z}_q[X]/\mathsf{r}(X)$.

The receiver encrypts the input $\mathsf{m} \in \mathcal{R}_t$ under their own key, and sends $\mathsf{Enc}(\mathsf{m}) = (\mathsf{c}, \mathsf{c}') \in \mathcal{R}_q^2$, in order to get back the homomorphic inner product of $\Phi(\mathsf{m}) = (\phi(\mathsf{m}, j))_{j\in[\![0;N]\!]} \in \mathcal{R}_t^{N+1}$ with $\mathsf{F} = (\mathsf{f}_j)_{j\in[\![0;N]\!]} \in \mathcal{R}_t^{N+1}$, committed by the sender, with a proof of correct value $(\mathsf{d}, \mathsf{d}') = \mathsf{Enc}(\langle\Phi(\mathsf{m}), \mathsf{F}\rangle) \in \mathcal{R}_q^2$, that thereafter decrypts to $\langle\Phi(\mathsf{m}), \mathsf{F}\rangle \in \mathcal{R}_t$, using the receiver's decryption key.

From $(\mathsf{c}, \mathsf{c}')$, and possible additional intermediate ciphertexts, sent by the receiver, the sender is able to compute $(\mathsf{u}_j, \mathsf{u}'_j) = \mathsf{Enc}(\phi(\mathsf{m}, j))$, for $j \in [\![0; N]\!]$, in any way they want, using the homomorphic properties of the encryption scheme:

$$(\mathsf{u}_j, \mathsf{u}'_j) = \mathsf{Enc}(\phi(\mathsf{m}, j)) = \left(\sum_{i=0}^{n} u_{j,i} \cdot X^i, \sum_{i=0}^{n} u'_{j,i} \cdot X^i\right) \in \mathcal{R}_q^2.$$

Thereafter, the goal is to evaluate the inner products:

$$(\mathsf{d}, \mathsf{d}') = \mathsf{Enc}(\langle\Phi(\mathsf{m}), \mathsf{F}\rangle) = \mathsf{Enc}\left(\sum_{j=0}^{N} \mathsf{f}_j \cdot \phi(\mathsf{m}, j)\right) = \sum_{j=0}^{N} \mathsf{f}_j \cdot \mathsf{Enc}(\phi(\mathsf{m}, j))$$
$$= \sum_{j=0}^{N} \mathsf{f}_j \cdot (\mathsf{u}_j, \mathsf{u}'_j) = \left(\sum_{j=0}^{N} \mathsf{f}_j \cdot \mathsf{u}_j, \sum_{j=0}^{N} \mathsf{f}_j \cdot \mathsf{u}'_j\right).$$

As already explained, the sender can add noise to hide F in the evaluation and prove the correct evaluation of $\tilde{\mathsf{d}} = \mathsf{d} + \mathsf{z}^*$ and $\tilde{\mathsf{d}}' = \mathsf{d}' + \mathsf{z}'^*$ so that $(\tilde{\mathsf{d}}, \tilde{\mathsf{d}}')$ decrypts to $\langle\Phi(\mathsf{m}), \mathsf{F}\rangle$. The pair $(\mathsf{z}^*, \mathsf{z}'^*)$ will be committed in a hidden way, with a proof of correct noise, in the appropriate range, to ensure the correct decryption.

This verifiable evaluation will be done in two steps: first, the sender will prove the correct evaluation of all the $(\mathsf{u}_j, \mathsf{u}'_j)$, while committed in a very compact way. Then, a proof of the inner product is provided, as described in Section 4. We will later explain how everything remains succinct.

We also claim this protocol to be SND-secure, proving theorem 5, with steps provided in the following sections.

D.1 Commitment of F

One can first note that ring elements in $\mathcal{R}_t$ and $\mathcal{R}_q$ are polynomials of degree at most $n-1$, and can be encoded into $\mathcal{R}_1 = \mathbb{Z}_q[X^{n-1}]$. The sender's vector $\mathsf{F} \in \mathcal{R}_t^{N+1}$ can be committed using the polynomial $\mathsf{f} = \sum_{i=0}^{N} \mathsf{f}_j \cdot Y^j$ in $\mathcal{R}_t[Y]$, where $\mathsf{f}_j = \sum_{i=0}^{n-1} f_{j,i} \cdot X^i$, can be committed

in a hidden way in $\mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$ as $\mathsf{f}^* = \sum_{j=0}^{N}\sum_{i=0}^{n-1} f_{j,i} \cdot X^i Y^{N-j}$ (with reverse order coefficients) into F^*.

The sender can also compute and publish the noisy inner products $\tilde{\mathsf{d}}$ and $\tilde{\mathsf{d}}'$ in $\mathcal{R}_q$, the expected ciphertext of the result, for some private $\mathsf{q}^*, \mathsf{q}'^*, \mathsf{z}^*, \mathsf{z}'^* \in \mathcal{R}_q$:

$$\tilde{\mathsf{d}} = \sum_{j=0}^{N} \mathsf{f}_j \cdot \mathsf{u}_j + \mathsf{z}^* - \mathsf{q}^* \cdot \mathsf{r} \qquad\qquad \tilde{\mathsf{d}}' = \sum_{j=0}^{N} \mathsf{f}_j \cdot \mathsf{u}'_j + \mathsf{z}'^* - \mathsf{q}'^* \cdot \mathsf{r}.$$

The polynomials $\mathsf{z}^*, \mathsf{z}'^*, \mathsf{q}^*, \mathsf{q}'^*$ are committed in a hidden way in Z^*, Z'^*, Q^*, Q'^*, in $\mathcal{R}_1 = \mathbb{Z}_q[X^{n-1}]$. The inner products must be proven, which is the goal of the second part, after we have proven the correct computation of the $(\mathsf{u}_j, \mathsf{u}'_j)$'s, as both together will prove correctness of the ciphertext $(\tilde{\mathsf{d}}, \tilde{\mathsf{d}}')$.

D.2 Validity of the $(\mathsf{u}_j, \mathsf{u}'_j)$'s

From the ciphertexts $(\mathsf{u}_j, \mathsf{u}'_j)$ that the prover computed on their own, they define the polynomials:

$$\mathsf{u}(X,Y) = \sum_{j=0}^{N} \mathsf{u}_j(X) \cdot Y^j = \sum_{j=0}^{N}\sum_{i=0}^{n-1} u_{j,i} X^i Y^j \qquad\qquad \mathsf{u}'(X,Y) = \sum_{j=0}^{N} \mathsf{u}'_j(X) \cdot Y^j$$

and commit them from the sub-space $\mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$ into U and U' respectively. Our first step is to prove that the polynomials committed in U and U' by the sender indeed satisfy, for any exponents $j \in [\![0; N]\!]$, $(\mathsf{u}_j, \mathsf{u}'_j) = \mathsf{Enc}(\phi(\mathsf{m}, j))$, or more precisely that the decryptions $\mathsf{Dec}(\mathsf{u}_j, \mathsf{u}'_j)$ that we denote $\varphi_{\mathsf{m},j}$ are indeed $\phi(\mathsf{m}, j) \in \mathcal{R}_t$, for m initially encrypted by the verifier in $(\mathsf{c}, \mathsf{c}')$.

The verifier first chooses and sends a random polynomial $\mathsf{n} \xleftarrow{\$} \mathcal{R}_t$ (or it can be generated by a hash function on the previous data to remove interaction). Both parties can compute, $\mathsf{n}_j = \mathsf{n}^j = \sum_{i=0}^{n-1} n_{j,i} \cdot X^i$ in $\mathcal{R}_t$, for all $j \in [\![0; N]\!]$, and commit them in $\mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$ as $\mathsf{t} = \sum_{j=0}^{N}\sum_{i=0}^{n-1} n_{j,i} \cdot X^i Y^{N-j}$ (with reverse order coefficients) into T. Then, the prover computes and sends $(\mathsf{b}, \mathsf{b}') = \sum_{j=0}^{N} \mathsf{n}_j \cdot (\mathsf{u}_j, \mathsf{u}'_j)$ in $\mathcal{R}_q^2$.

Since this is symmetric, we now focus on the first component (without $'$), and similar analysis will have to be done on the second component (with $'$): the prover also computes and sends l so that $\sum_{j=0}^{N} \mathsf{n}_j \cdot \mathsf{u}_j = \mathsf{b} + \mathsf{l} \cdot \mathsf{r}$, that they both commit as L in $\mathcal{R}_1 = \mathbb{Z}_q[X^{n-1}]$.

The verifier chooses and sends a sequence of random scalars $\beta_\kappa \xleftarrow{\$} \mathbb{Z}_q$, for $\kappa \in [\![1; \Lambda]\!]$. They will define $v_{\kappa,j} = \mathsf{u}_j(\beta_\kappa) \in \mathbb{Z}_q$ for $j \in [\![0; N]\!]$. We thus have the polynomial:

$$\mathsf{v}_\kappa(Y) = \sum_{j=0}^{N} v_{\kappa,j} \cdot Y^j = \sum_{j=0}^{N} \mathsf{u}_j(\beta_\kappa) \cdot Y^j = \sum_{j=0}^{N}\sum_{i=0}^{n-1} u_{j,i} \beta_\kappa^i \cdot Y^j = \mathsf{u}(\beta_\kappa, Y)$$

that can be committed as V_κ in $\mathcal{R}_3 = \mathbb{Z}_q[Y^N]$, and proven correct with the quadratic check

$$\mathsf{u}(X,Y) - \mathsf{v}_\kappa(Y) = \mathsf{u}(X,Y) - \mathsf{u}(\beta_\kappa, Y) = (X - \beta_\kappa) \cdot \mathsf{w}_\kappa(X,Y) \tag{2}$$

on the commitment W_κ of $\mathsf{w}_\kappa \in \mathcal{R}_2 = \mathbb{Z}_q[X^{n-1}, Y^N]$ and the public commitment C_κ of $X - \beta_\kappa$, and thus only 3 prover-generated commitments (U, V_κ and W_κ, as C_κ is public). With $\mathsf{t}_\kappa = \mathsf{t}(\beta_\kappa, Y) = \sum_{j=0}^{N} \mathsf{n}_j(\beta_\kappa) \cdot Y^j$ publicly computed and committed in T_κ, we can note that

$$\begin{aligned}\mathsf{v}_\kappa(Y) \cdot \mathsf{t}_\kappa(Y) &= \sum_{0 \le i,j \le N} v_{\kappa,i} \cdot \mathsf{n}_j(\beta_\kappa) \cdot Y^{N+i-j} \\ &= \sum_{0 \le i \le N} v_{\kappa,i} \cdot \mathsf{n}_i(\beta_\kappa) \cdot Y^N + \sum_{0 \le i \ne j \le N} v_{\kappa,i} \cdot \mathsf{n}_j(\beta_\kappa) \cdot Y^{N+i-j}\end{aligned}$$

and

$$\sum_{0\le i\le N} v_{\kappa,i}\cdot \mathsf{n}_i(\beta_\kappa) = \sum_{0\le i\le N} \mathsf{u}_i(\beta_\kappa)\cdot \mathsf{n}_i(\beta_\kappa) = b_\kappa + l_\kappa\cdot r_\kappa$$

where $b_\kappa = \mathsf{b}(\beta_\kappa)$, $l_\kappa = \mathsf{l}_\kappa(\beta_\kappa)$, and $r_\kappa = \mathsf{r}(\beta_\kappa)$ can be publicly computed in $\mathbb{Z}_q$, so we can check:

$$\mathsf{v}_\kappa(Y)\cdot \mathsf{t}_\kappa(Y) - (b_\kappa + l_\kappa\cdot r_\kappa)\cdot Y^N = \sum_{0\le i\ne j\le N} v_{\kappa,i}\cdot \mathsf{n}_j(\beta_\kappa)\cdot Y^{N+i-j} = \mathsf{y}_\kappa(Y) \tag{3}$$

with a commitment Y_κ of y_κ in $\mathcal{R}_4 = \mathbb{Z}_q[Y^{2N\setminus N}]$ and C_N a public commitment of Y^N, which makes only 2 prover-generated commitments (V_κ and Y_κ).

We stress that because of the repetitions in each commitment (up to K, according to the kind of relations they are involved in, and the number of prover-generated commitments), we know (see Corollary 9) that when all the tests pass, the above equations (2) and (3) are all satisfied with error probability less than $2\Lambda\cdot 2^{-(S+1)K/4}$.

They altogether prove that, for each $\kappa\in[\![1;\Lambda]\!]$, $b_\kappa + l_\kappa\cdot r_\kappa$ is the coefficient of Y^N in $\mathsf{v}_\kappa(Y)\cdot\mathsf{t}_\kappa(Y)$, and so $\sum_j \mathsf{u}_j(\beta_\kappa)\cdot\mathsf{n}_j(\beta_\kappa) = \mathsf{b}(\beta_\kappa) + \mathsf{l}(\beta_\kappa)\cdot\mathsf{r}(\beta_\kappa) \bmod q$ for the random β_κ, on polynomials committed beforehand, of degree n. Hence, $\sum_j \mathsf{u}_j\cdot\mathsf{n}_j = \mathsf{b}+\mathsf{l}\cdot\mathsf{r}$, excepted with error probability bounded by $2\ell n/p \le n/2^{S+1}N$, from the Schwartz-Zippel lemma.

For Λ large enough (to be set later), after all these steps, on both b and b', one gets the proof that, in $\mathcal{R}_q$,

$$\mathsf{b} = \sum_{j=0}^{N}\mathsf{n}_j\cdot\mathsf{u}_j \quad \mathsf{b}' = \sum_{j=0}^{N}\mathsf{n}_j\cdot\mathsf{u}'_j : \qquad (\mathsf{b},\mathsf{b}') = \sum_{j=0}^{N}\mathsf{n}_j\cdot(\mathsf{u}_j,\mathsf{u}'_j).$$

These relations hold for both b and b' excepted with error probability bounded by $2\times(2\Lambda\cdot 2^{-(S+1)K/4} + (n/2^{S+1}N)^\Lambda)$.

D.3 Validity of $(\tilde{\mathsf{d}},\tilde{\mathsf{d}}')$

As above, we focus on $\tilde{\mathsf{d}}$ with respect to all $(\mathsf{u}_j)_j$ and $\mathsf{z}^*,\mathsf{q}^*$. The same should be done for $\tilde{\mathsf{d}}'$ with respect to all $(\mathsf{u}'_j)_j$ and $\mathsf{z}'^*,\mathsf{q}'^*$, but for the same $(\mathsf{f}_j)_j$ and z,r:

$$\tilde{\mathsf{d}} = \sum_{j=0}^{N}\mathsf{f}_j\cdot\mathsf{u}_j + \mathsf{z}^* - \mathsf{q}^*\cdot\mathsf{r}.$$

This following analysis is quite similar to the previous one, considering f^*, $(\mathsf{z}^*,\mathsf{q}^*)$, $\tilde{\mathsf{d}}$ instead of $\mathsf{t},\mathsf{p},\mathsf{b}$, and with hidden intermediate values, as $(\mathsf{f}_j)_j$ is private to the prover, contrarily to n which was publicly chosen by the verifier. One first proves the quadratic relation on the hidden commitment of $\mathsf{g}^*_\kappa(Y) = \mathsf{f}^*(\beta_\kappa,Y) = \sum_{j=0}^N g_{\kappa,j}\cdot Y^{N-j}$ in $\mathcal{R}_3 = \mathbb{Z}_q[Y^N]$, where $g_{\kappa,j} = \mathsf{f}_j(\beta_\kappa)$, and the hiding commitment of $\mathsf{h}^*_\kappa(X,Y)$ in $\mathcal{R}_2 = \mathbb{Z}_q[X^n,Y^N]$, so that

$$\mathsf{f}^*(X,Y) - \mathsf{g}^*_\kappa(Y) = \mathsf{f}^*(X,Y) - \mathsf{f}^*(\beta_\kappa,Y) = (X-\beta_\kappa)\cdot\mathsf{h}^*_\kappa(X,Y) \tag{4}$$

with only 3 prover-generated commitments. And similarly, one proves in a hidden way:

$$\mathsf{q}^*(X) - q^*_\kappa = \mathsf{q}^*(X) - \mathsf{q}^*(\beta_\kappa) = (X-\beta_\kappa)\cdot\mathsf{q}^*_\kappa(X) \tag{5}$$
$$\mathsf{z}^*(X) - z^*_\kappa = \mathsf{z}^*(X) - \mathsf{z}^*(\beta_\kappa) = (X-\beta_\kappa)\cdot\mathsf{z}^*_\kappa(X) \tag{6}$$

with the private $q^*_\kappa = \mathsf{q}^*_\kappa(\beta_\kappa)$ and $z^*_\kappa = \mathsf{z}^*_\kappa(\beta_\kappa)$ in $\mathbb{Z}_q$ that will be also used below. There are only 2 prover-generated commitments in each relation. We have the following relations:

$$\begin{aligned}\mathsf{v}_\kappa(Y)\cdot\mathsf{g}^*_\kappa(Y) &= \sum_{0\le i,j\le N} v_{\kappa,i}\cdot g_{\kappa,j}\cdot Y^{N+i-j}\\ &= \sum_{0\le i\le N} v_{\kappa,i}\cdot g_{\kappa,i}\cdot Y^N + \sum_{0\le i\ne j\le N} v_{\kappa,i}\cdot g_{\kappa,j}\cdot Y^{N+i-j}\end{aligned}$$

and

$$\sum_{0\le i\le N} v_{\kappa,i}\cdot g_{\kappa,i} = \sum_{0\le i\le N} \mathsf{u}_i(\beta_\kappa)\cdot \mathsf{f}_i(\beta_\kappa) \qquad = \tilde{\mathsf{d}}(\beta_\kappa) - \mathsf{z}^*(\beta_\kappa) + \mathsf{q}^*(\beta_\kappa)\cdot \mathsf{r}(\beta_\kappa)$$
$$= \tilde{d}_\kappa - z^*_\kappa + q^*_\kappa \cdot r_\kappa$$

where $\tilde{d}_\kappa = \tilde{\mathsf{d}}(\beta_\kappa)$ and $r_\kappa = \mathsf{r}(\beta_\kappa)$ can be publicly computed, and so one proves

$$\mathsf{v}_\kappa(Y)\cdot \mathsf{g}^*_\kappa(Y) - (\tilde{d}_\kappa - z^*_\kappa + q^*_\kappa\cdot r_\kappa)\cdot Y^N = \mathsf{s}_\kappa(Y) \tag{7}$$

with a commitment of s_κ in $\mathcal{R}_4 = \mathbb{Z}_q[Y^{2N\setminus N}]$ and a public commitment of Y^N.

Again, from Corollary 9, when all the tests pass, the above equations (4), (5), (6), and (7) are all satisfied with error probability less than $3\Lambda\cdot 2^{-(S+1)K/4}$.

They altogether prove that, for each $\kappa\in[\![1;\Lambda]\!]$, $d_\kappa - z^*_\kappa + q^*_\kappa\cdot r_\kappa$ is the coefficient of Y^N in $\mathsf{v}_\kappa(Y)\cdot\mathsf{g}^*_\kappa(Y)$, so

$$\tilde{\mathsf{d}}(\beta_\kappa) = \sum_j \mathsf{u}_j(\beta_\kappa)\cdot\mathsf{f}_j(\beta_\kappa) + \mathsf{z}^*(\beta_\kappa) - \mathsf{q}^*(\beta_\kappa)\cdot\mathsf{r}(\beta_\kappa) \bmod q$$

for the random β_κ, on polynomials committed beforehand. Hence,

$$\tilde{\mathsf{d}} = \sum_j \mathsf{u}_j\cdot\mathsf{f}_j + \mathsf{z}^* - \mathsf{q}^*\cdot\mathsf{r},$$

excepted with error probability bounded by $2\ell n/p \le n/2^{S+1}N$, for each κ. Again, these relations hold for both $\tilde{\mathsf{d}}$ and $\tilde{\mathsf{d}}'$ excepted with error probability bounded by $2\times(3\Lambda\cdot 2^{-(S+1)K/4} + (n/2^{S+1}N)^\Lambda)$.

Globally, the error probability is bounded by $2\times(5\Lambda\cdot 2^{-(S+1)K/4} + 2\times(n/2^{S+1}N)^\Lambda)$. We thus need to take $\Lambda = \lceil \log_2(2\nu_e/\varepsilon_s)/\log_2(2^{S+1}N/n)\rceil$ different values for β_κ so that $4\times(n/2^{S+1}N)^\Lambda \le \varepsilon_s/6$, and $K > 4\log_2(60\Lambda/\varepsilon_s)/(S+1)$ so that $2\times 5\Lambda\cdot 2^{-(S+1)K/4}\le\varepsilon_s/4$ too, so that the global error on the equations be bounded by $\varepsilon_s/2$.

On the other hand, as shown on Figure 4, the global number of commitments is bounded by $15\Lambda+17$. We thus need to take $\varepsilon_c \le \varepsilon_s/(2\times\nu_c) = \varepsilon_s/(2(15\Lambda+17))$.

D.4 SND-Security of our Scheme

For each commitment generated by the prover and sent to the receiver, Theorem 8 grants that if it passes validity and quadratic root checks, then at least $4K/5$ of the twin encodings encode the polynomial u in the appropriate subspace, that can be extracted by extractability of valid encodings, excepted with probability less than $2^{-(S+1)K/5} + 2K\cdot 2^{-(S+1)M/3}$, where $q = \prod_{i=1}^{\ell} p_i$ with all $p_i \ge 2^S\times 2\ell\times 2N$. We choose K and M such that

$$2^{-(S+1)K/5} + 2K\cdot 2^{-(S+1)M/3} \le \frac{\varepsilon_s}{2\nu_c},$$

where ε_s is the soundness security parameter and ν_c the total number of prover-generated commitments in the protocol. So, as ν_c commitments are going to be checked, this holds for all of them except with probability less than $\varepsilon_s/2$.

Then, in our protocol, a total of ν_e quadratic equations on committed polynomials need to be checked thanks to their commitments. The checks for one of these relations grants it is valid except for a probability smaller than $2^{-SK/5}$, where K is chosen so that this is less than $\frac{\varepsilon_s}{2\nu_e}$. For ν_e checked equations, they all hold with probability less than $\varepsilon_s/2$.

Then, the proof also uses the decryption of 2 FHE ciphertexts, and their soundness is ensured by the FHE security on more than 128 bits.

In the end, our soundness is granted except for a probability less than our soundness security parameter ε_s.

D.5 R-Privacy-Security of our Scheme

The scheme is R-Privacy-secure under the semantic security of the FHE FV scheme. Sending the public key containing the FHE public information and $\mathcal{O}(N^{1/d})$ FHE ciphertexts of some intermediate powers (or some other function if not performing OPE) of m, with an FHE scheme with at least 128 bits of security does not reveal any information on the cleartexts, for any malicious adversary performing polynomial time calculations.

D.6 S-Privacy-Security of our Scheme

Under the statistically hiding properties of our commitment scheme given in Theorem 10 and the statistical security provided by the noise-flooding, no sender's information leaks to a receiver that has correctly sent the initial intermediate ciphertexts (honest-but-curious receiver). See Appendix F for more discussions on malicious receivers.

E Applications

Our protocol can be deployed for several applications. Hereafter we detail the cases of PSI (using OPE), and of SPIR.

E.1 Application to Private Set Intersection

As already explained, we can apply this technique to PSI, where the receiver owns a set $\mathcal{X} = \{x_1, \ldots, x_a\}$ of cardinality $\#\mathcal{X}$, the sender a set $\mathcal{Y} = \{y_1, \ldots, y_b\}$ of cardinality $\#\mathcal{Y}$, and the receiver wants to learn the intersection. We consider the case where $\#\mathcal{Y}$ is much larger than $\#\mathcal{X}$, and hope to get a protocol essentially linear in $\#\mathcal{X}$ only. To this aim, we follow the basic approach from [FNP04], with the polynomial $P(Y) = \prod_{i=1}^{N}(Y - \mathsf{h}_i)$ committed once for all in P^*, where the values $\mathsf{h}_i \in \mathcal{R}_t$ encode the elements y_i in the sender's set, and the degree $N = \#\mathcal{Y}$. Then the receiver wants to check whether $P(x_i) = 0$ on every input x_i.

Thanks to the verifiability of our proof, and even the knowledge-soundness of our commitments, our PSI protocol is secure against malicious senders, as they cannot use distinct polynomials as P in each execution.

The protocol can be adapted to support adding elements to the sender's set, with a new check to make sure the old sender's polynomial divides the new one which only has additional roots.

E.2 Application to Symmetric Private Information Retrieval

In the Symmetric PIR setting, the sender owns a set $\mathcal{Y} = \{y_1, \ldots, y_N\}$ of cardinality N, and the receiver wants to retrieve an element of the sender's set with a private index μ. To this aim, the sender defines $\mathsf{F} = (\mathsf{f}_j)_j$ as a vector of representations of their set elements in $\mathcal{R}_t$, the receiver has a representation $\mathsf{m} = \tau(\mu)$ of their index μ in $\mathcal{R}_t$. But using our previous notations, we want to define $\mathsf{M} = (\mathsf{m}_j)_j$, with $\mathsf{m}_j = \delta_{\mu,j}$, where δ is the Kroenecker symbol ($\delta_{i,j} = 1$ if $i = j$ and 0 otherwise): $\langle \mathsf{F}, \mathsf{M} \rangle = \sum \mathsf{f}_j \mathsf{m}_j = \sum \delta_{\mu,j} \mathsf{f}_j = \mathsf{f}_\mu$, which is the μ-th element in the sender's set.

As for the successive powers, one can note that writing μ and j in a basis B with $\mu = \sum_{k=0}^{d} \mu_k B^k$ and $j = \sum_{k=0}^{d} j_k B^k$, where the $\mu_k, j_k \in [\![0; B-1]\!]$:

$$\delta_{\mu,j} = \delta_{\sum_{k=0}^{d} \mu_k b^k, \sum_{k=0}^{d} j_k b^k} = \prod_{k=0}^{d} \delta_{\mu_k, j_k}$$

So sending the encryptions of $\delta_{\mu_i,k}$ for $i \in [\![0; d]\!], k \in [\![0; B-1]\!]$ allows the sender to calculate all the required ciphertexts with circuits of multiplicative depth d. With $B^d \sim N$, the receiver sends $(d+1)B \approx dN^{1/d}$ ciphertexts. This is the same number of ciphertexts as in the OPE application.

F Towards Security against Malicious Receivers

When the receiver only sends m, they cannot learn any information other than $\mathsf{F}(\mathsf{m})$. But with encryptions of intermediate powers $\{(\mathsf{c}_{i,j}, \mathsf{c}'_{i,j}) = \mathsf{Enc}(\mathsf{m}^{iB^j}))\}_{i\in[\![0;B-1]\!],j\in[\![0;d]\!]}$ derived from their secret point $\mathsf{m} \in \mathcal{R}_t$ and a basis B to write elements in $[\![0;N]\!]$, such that N is equal to or a bit less than B^d, from which the sender will have everything necessary to calculate all the $(\mathsf{u}_j, \mathsf{u}'_j)_{j\in[\![0;N]\!]}$ in a d multiplicative depth. However, were the receiver to cheat, some information on F might leak. One would thus prevent the receiver from behaving maliciously and sending inconsistent ciphertexts. Notice we have the following relations:

$$\mathsf{m}^{iB^j} = \mathsf{m}^{(i-1)B^j} \times \mathsf{m}^{B^j} \qquad \mathsf{m}^{B^j} = \mathsf{m}^{(B-1)B^{j-1}} \times \mathsf{m}^{B^{j-1}} \qquad \mathsf{m}^1 = \mathsf{m}^0 \times \mathsf{m}^1$$

hence, the ciphertexts:

$$\mathsf{c}_{i,j} - \mathsf{c}_{i-1,j} \times \mathsf{c}_{1,j} \qquad \mathsf{c}_{1,j} - \mathsf{c}_{B-1,j-1} \times \mathsf{c}_{1,j-1} \qquad \mathsf{c}_{1,0} - \mathsf{c}_{0,0} \times \mathsf{c}_{1,0}$$

should be encodings of zero. Thus, generating random coefficients $\alpha_{i,j}, \beta_j \xleftarrow{\$} \mathbb{Z}_q^*, i \in [\![2; b-1]\!], j \in [\![0; d]\!]$, derived from a hash on the receiver communications, both parties can compute

$$\mathsf{x} = \sum_{i=2}^{b-1}\sum_{j=0}^{d} \alpha_{i,j} \cdot (\mathsf{c}_{i,j} - \mathsf{c}_{i-1,j} \times \mathsf{c}_{1,j}) + \sum_{j=1}^{d} \beta_j \cdot (\mathsf{c}_{1,j} - \mathsf{c}_{b-1,j-1} \times \mathsf{c}_{1,j-1}) + \beta_0 \cdot (\mathsf{c}_{1,0} - \mathsf{c}_{0,0} \times \mathsf{c}_{1,0})$$

which should be an encryption of zero.

A first method would be for the receiver to provide the randomness used to encrypt this ciphertext. As this global randomness just consists of $3n$ coefficients on $\mathbb{Z}_t$, this does not leak much information on the randomness used in the ciphertexts $(\mathsf{c}_{i,j}, \mathsf{c}'_{i,j})$, to still guarantee semantic security, and the sender is convinced all the intermediate ciphertexts are consistent.

A second more formal method would be to make another commitment setup, so the receiver can commit this ciphertext of zero's noise components, prove their norm is small using our inner product for norm calculation, and convince the sender of the relations verified by these secret noise components with respect with the whole public ciphertext of zero.

G Python Script for Parameters

G.1 RNS Compatible Application

In order to use the SEAL FV implementation for our scheme, we need to be able to take a modulus q that is a product of primes on less than 60 bits. This implies higher false positive probabilities given by the Schwartz-Zippel lemma, hence we need repetitions in our proofs to obtain the required soundness. This changes our parameter values, with $q = p_1 \times \ldots \times p_\ell$, $p = \min\{p_1, \ldots, p_\ell\}$, and $p \geq 2^S \cdot 4N\ell$.

The size of the proof depends a lot on Λ. The larger N/n is, the shorter the proof, but S also has a huge impact, when S makes Λ decrease, with the number of encodings. However, when S grows, so does p, which can make the FHE ciphertexts heavier.

G.2 Summary of Commitments and Checks

In Figures 2, 3, and 4, we present a summary of all the parameters according to the expected security levels.

Paola: to be dated with r ν_e

#Prover-Generated Commitments: c		2	3
$\varepsilon_c = 2^{-38}$	K	26	34
	M (and binding case #Scalars)	20	20
	#Equations (hiding case)	2184	2856
	#Secrets (hiding case)	4448	5792
	#Encodings for $1v$ Pol.	572	748
	#Encodings for $2v$ Pol.	1092	1428
$\varepsilon_c = 2^{-136}$	K	75	99
	M (and binding case #Scalars)	62	62
	#Equations (hiding case)	18900	24948
	#Secrets (hiding case)	38048	50144
	#Encodings for $1v$ Pol.	4800	6336
	#Encodings for $2v$ Pol.	9450	12474

Fig. 2. Parameters for Commitments with $p \geq 64\ell\mathcal{D}$, four repetitions in the zero-knowledge proofs, $\varepsilon_s = 2^8 \cdot \varepsilon_c$, and $\nu_e = 11$.

Equation	c
$\mathsf{u}(X,Y) - \mathsf{v}_\kappa(Y) = (X - \beta_\kappa) \cdot \mathsf{w}_\kappa(X,Y)$	3
$\mathsf{u}'(X,Y) - \mathsf{v}'_\kappa(Y) = (X - \beta_\kappa) \cdot \mathsf{w}'_\kappa(X,Y)$	3
$\mathsf{v}_\kappa(Y) \cdot \mathsf{t}_\kappa(Y) - (b_\kappa + l_\kappa \cdot r_\kappa) \cdot Y^N = \mathsf{y}_\kappa(Y)$	2
$\mathsf{v}'_\kappa(Y) \cdot \mathsf{t}_\kappa(Y) - (b'_\kappa + l'_\kappa \cdot r_\kappa) \cdot Y^N = \mathsf{y}'_\kappa(Y)$	2
$\mathsf{f}^*(X,Y) - \mathsf{g}^*_\kappa(Y) = (X - \beta_\kappa) \cdot \mathsf{h}^*_\kappa(X,Y)$	3
$\mathsf{q}^*(X) - q^*_\kappa(Y) = (X - \beta_\kappa) \cdot \mathsf{q}^*_\kappa(X)$	2
$\mathsf{q}'^*(X) - q'^*_\kappa(Y) = (X - \beta_\kappa) \cdot \mathsf{q}'^*_\kappa(X)$	2
$\mathsf{u}^*(X) - u^*_\kappa(Y) = (X - \beta_\kappa) \cdot \mathsf{u}^*_\kappa(X)$	2
$\mathsf{e}^*_1(X) - e^*_{1,\kappa}(Y) = (X - \beta_\kappa) \cdot \mathsf{e}^*_{1,\kappa}(X)$	2
$\mathsf{e}^*_2(X) - e^*_{2,\kappa}(Y) = (X - \beta_\kappa) \cdot \mathsf{e}^*_{2,\kappa}(X)$	2
$\mathsf{v}_\kappa(Y) \cdot \mathsf{g}^*_\kappa(Y) - (\tilde{\mathsf{d}}_\kappa - z^*_\kappa + q^*_\kappa \cdot r_\kappa) \cdot Y^N = \mathsf{s}_\kappa(Y)$	3
$\mathsf{v}'_\kappa(Y) \cdot \mathsf{g}^*_\kappa(Y) - (\tilde{\mathsf{d}}'_\kappa - z'^*_\kappa + q'^*_\kappa \cdot r_\kappa) \cdot Y^N = \mathsf{s}'_\kappa(Y)$	3
$\bar{\mathsf{u}}^*(X) = X^{n-1} \cdot \mathsf{u}^*(1/X)$	2
$\bar{\mathsf{e}}^*_1(X) = X^{n-1} \cdot \mathsf{e}^*_1(1/X)$	2
$\bar{\mathsf{e}}^*_2(X) = X^{n-1} \cdot \mathsf{e}^*_2(1/X)$	2
$\mathsf{u}^*(X) - u^* = (X - 1/\beta) \cdot \mathsf{u}'^*(X)$	2
$\mathsf{e}^*_1(X) - e^*_1 = (X - 1/\beta) \cdot \mathsf{e}'^*_1(X)$	2
$\mathsf{e}^*_2(X) - e^*_2 = (X - 1/\beta) \cdot \mathsf{e}'^*_2(X)$	2
$\bar{\mathsf{u}}^*(X) - \beta^{n-1} \cdot u^* = (X - \beta) \cdot \bar{\mathsf{u}}'^*(X)$	2
$\bar{\mathsf{e}}^*_1(X) - \beta^{n-1} \cdot e^*_1 = (X - \beta) \cdot \bar{\mathsf{e}}'^*_1(X)$	2
$\bar{\mathsf{e}}^*_2(X) - \beta^{n-1} \cdot e^*_2 = (X - \beta) \cdot \bar{\mathsf{e}}'^*_2(X)$	2
Total number of checked equations: $\nu_e = 12\Lambda + 9$	

Fig. 3. Equations to verify with quadratic checks in our full protocol

#variables	c	Binding or Hiding	Polynomials	Domain	#commitments
$1v$	2	B	$\mathsf{y}_\kappa, \mathsf{y}'_\kappa$	$\mathcal{R}_4$	2Λ
		H	$\mathsf{q}^*, \mathsf{q}'^*, \mathsf{q}^*_\kappa, \mathsf{q}'^*_\kappa$ $\mathsf{u}^*_\kappa, \mathsf{e}^*_{1,\kappa}, \mathsf{e}^*_{2,\kappa}$	$\mathcal{R}_1$	$5\Lambda + 2$
	3	H	$\mathsf{u}^*, \mathsf{e}^*_1, \mathsf{e}^*_2, \bar{\mathsf{u}}^*, \bar{\mathsf{e}}^*_1,$ $\bar{\mathsf{e}}^*_2, \mathsf{u}'^*, \mathsf{e}'^*_1, \mathsf{e}'^*_2,$ $\bar{\mathsf{u}}'^*, \bar{\mathsf{e}}'^*_1, \bar{\mathsf{e}}'^*_2$	$\mathcal{R}_1$	12
			g^*_κ	$\mathcal{R}_3$	Λ
		B	$\mathsf{v}_\kappa, \mathsf{v}'_\kappa$	$\mathcal{R}_3$	2Λ
			$\mathsf{s}_\kappa, \mathsf{s}'_\kappa$	$\mathcal{R}_4$	2Λ
$2v$	3	B	$\mathsf{u}, \mathsf{u}', \mathsf{w}_\kappa, \mathsf{w}'_\kappa$	$\mathcal{R}_2$	$2(\Lambda + 1)$
		H	$\mathsf{f}^*, \mathsf{h}^*_\kappa$	$\mathcal{R}_2$	$\Lambda + 1$
Total					$15\Lambda + 17$

Fig. 4. Number of Commitments in the Global Proof, and Size in Number of Encodings, with $\Lambda = \lceil (3 + \log_2(3/\varepsilon_s))/(S + 1 + \log_2(N/n)) \rceil$