Efficient Lattice-Based Polynomial Evaluation and Batch Zero-Knowledge Arguments

Veronika Kuchta (✉), Amin Sakzad, Ron Steinfeld, Joseph K. Liu

Faculty of Information Technology,
Monash University, Melbourne, Australia
<veronika.kuchta,amin.sakzad,ron.steinfeld,joseph.liu>@monash.edu

Abstract. In this paper we provide an efficient construction of a lattice-based polynomial argument and a polynomial batch-protocol, where the latter contains the polynomial argument as a building block. Our contribution is motivated by the discrete log based construction (EUROCRYPT'16), where in our case we employ different techniques to obtain a communication efficient lattice-based scheme. In the zero-knowledge polynomial batch-protocol, we prove the knowledge of an easy relation between two polynomials which also allows batching of several instances of the same relation. Our batch-protocol is applicable to an efficient lattice-based range proof construction which represents a useful application in cryptocurrencies. In contrast to the existing range proof (CRYPTO'19), our proof is more efficient for large number of batched instances.

1 Introduction

Lattice-based cryptography has attracted an immense interest from the cryptographic community in the last years. Lattice problems are supposed to be resistant against quantum attacks and satisfy the worst-case to average-case reductions. Zero-knowledge proofs and arguments are significant building blocks in many cryptographic protocols which are used to provide privacy. Goldwasser et al. [13] introduced the main concept of a zero-knowledge proof. We emphasize that there is a significant difference between zero-knowledge proofs and zero-knowledge arguments, where the former satisfy statistical soundness property, while the latter achieve computational soundness. One of the main advantages of zero-knowledge arguments against zero-knowledge proofs is their low communication complexity as showed in [16]. Because of the increasing interest for post-quantum cryptography, the cryptographic community has been focusing among others also on the post-quantum constructions of zero-knowledge protocols. First such protocol was introduced by Stern [21], where the security of the protocol is based on a post-quantum assumption called hardness of "syndrome decoding". Another approach for zero-knowledge proof constructions was proposed by Lyubashevsky [18,15] and further developed in [19] and is known as the "Fiat-Shamir with Aborts" technique. The main advantage of these protocols is a small soundness error with only one repetition. Using lattices for zero-knowledge proofs constructions seems to be a challenging task. While there are efficient zero-knowledge arguments [8] and polynomial evaluation arguments [4,9] in the discrete log setting, a lattice-based analogy of these protocols is a popular but not a trivial research question. Several amortization techniques are proposed and employed in such works like [5,10] to improve the efficiency of zero-knowledge proofs especially for arithmetic circuits with linear communication length [2]. Benhamouda et al. [6] introduced a new challenge space consisting of monomials such that the soundness error of the zero-knowledge proof which is defined over cyclotomic rings $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n+1)$ could be reduced from $1/2$ to $1/(2n)$. New techniques for zero-knowledge proof constructions were proposed in [11] where an efficient one-shot proof mechanism was provided. Bootle et al. [8] presented an honest-verifier zero-knowledge proof for an arithmetic circuit defined in the discrete log setting. Their protocol improves in complexity compared to [14] by achieving fewer moves. Another contribution in [8] is a subroutine to commit to a polynomial and later reveal its evaluation at any point. They achieve a square root communication complexity which motivated us to our first contribution in this paper. In a later work [9], the authors introduced a framework for simple relations between commitments and a batch protocol which proves multiple copies of the same relation in a single argument. Zero-knowledge proofs have useful applications in cryptocurrencies [23,20]. Batch polynomial proof is used to prove knowledge of multiple instances at once. For instance, range proof often requires to prove that multiple inputs are in a certain interval. In case of cryptocurrencies it proves that the sum of multiple spending inputs is equal to the multiple outputs. Batched range proofs are more efficient than running a simple range proof on different instances. Our new batch proof using the new approach of polynomial evaluation argument as a building block will provide a useful tool for blockchain and cryptocurrency applications.

1.1 Our contribution.

We provide the first construction of a lattice based polynomial evaluation argument and show how to commit to a secret polynomial and later reveal its evaluation at a certain value. Our second contribution is an extension of the first one, where the polynomial evaluation argument is used as a building block. We provide the first lattice-based version of a batch protocol with communication cost overhead of a square root in the number of batched instances.

Our Techniques. The instantiation of the polynomial evaluation protocol in [8] in the lattice setting is not a straightforward task and yields several difficulties which were not an issue in the discrete log setting. The first issue is the choice of a challenge set. In order to guarantee soundness of our polynomial evaluation protocol, which requires the extracted witnesses to be small, we use a challenge space which contains monomials of form X^{ω}. Committing to a polynomial $f(X) \in \mathcal{R}_q$ of degree $n-1 = n' \cdot m'$, we embed its coefficients into a matrix $\mathcal{A}$ of size $n' \times m'$. The polynomial $f(X)$ can be represented as the following product: $f(X) = \left(1, X, \dots, X^{n'-1}\right) \cdot \mathcal{A} \cdot \left(1, X^{n'}, \dots, X^{(m'-1)n'}\right)^{tr}$. In order to prove the knowledge of this polynomial we need to mask $\mathcal{A}$ using a procedure which we explain here briefly. Since we are using monomials as a challenge, the masking procedure is a highly challenging task. To commit to a vector $\tilde{\mathbf{f}}(X) = \big(1, X, \ldots, X^{n'-1}\big) \cdot \widetilde{\mathcal{A}}$, where $\widetilde{\mathcal{A}}$ is the masked matrix, we need to ensure that $\tilde{\mathbf{f}}(X)$ does not leak any information about the coefficients of $\mathcal{A}$. Our solution is to mask all coefficients $a_{j,i}$ as follows: We sample values $u_{j,i} \leftarrow_{\$} \mathbb{Z}_q, j \in [0, n'-1], i \in [0, m'-1]$ and compute the masked coefficients $\tilde{a}_{j,i} = a_{j,i} - u_{j,i} X^{n'-(2j+1)}$. The last of the matrix is given as $\mathbf{u} = \big(\sum_{j=0}^{n'-1} u_{n'-j-1,1} X^j, \ldots, \sum_{j=0}^{n'-1} u_{n'-j-1,m'-1} X^j, 0\big)$. This masking techniques which differs significantly from the original technique in [8], allows us to use monomials as challenges x, however the challenge space is reduced to only 3 monomials $1, X^1, X^{-1}$. The reason for that is that any other monomials of degree $\hat{n}$, $2 \leq \hat{n} \leq n-2$ do not provide the required hiding property of our commitment when we compute $\tilde{\mathbf{f}}(x)$. We explain it on a simplified example. Let $n' = m' = 2$. The equation $\tilde{\mathbf{f}}(x)$ would return the following 2 dimensional vector $\big(a_{0,0} + a_{1,0}x + x^2(u_{1,1}x^0 + u_{0,1}x), a_{0,1} - u_{0,1}x + (a_{1,1} - u_{1,1}x^{-1})x\big)$. It is obvious that the polynomial coefficients are perfectly masked when $x = 1$. The same holds, when $x = X$, since $X^2 = -1$ and $X^3 = X$, such that the first component of the vector becomes $a_{0,0} + a_{1,0}X - u_{1,1} - u_{0,1}X = (a_{0,0} - u_{1,1} + (a_{1,0} - u_{0,1})X$ such that each polynomial coefficient $a_{(\cdot,\cdot)}$ is masked by a value $u_{(\cdot,\cdot)}$. The second component is also masked perfectly. In the last case, $x = X^{-1}$ implies $X^{-2} = -1$ and $X^{-3} = X^{-1}$ and the first component is equal to $a_{0,0} + a_{1,0}X^{-1} - u_{1,1} - u_{0,1}X^{-1}$ such that each coefficient $a_{(\cdot,\cdot)}$ is fully masked. After running this experiment for higher n', m' we concluded that the only possible challenge values which fully mask the polynomial coefficients are $\{1, X, X^{-1}\}$. On the one hand the low number of challenges means that to achieve negligible soundness error of our protocol we need to repeat it $\lambda / \log(|\mathcal{CH}|)$ times, where $|\mathcal{CH}|$ is the size of the challenge space. On the other hand we deal with a generalised Vandermonde matrix which is a concatenation of simple Vandermonde matrices of dimension 3, which yields a determinant equal to 6. This determinant represents the relaxation factor of the extracted witness which is relatively small in contrast to the proof in [11]. We commit to each row of $\widetilde{\mathcal{A}}$ using a commitment scheme which is defined in [3,7] and is secure under the M-SIS assumption over $\mathcal{R}_q$. We commit to $\tilde{\mathbf{f}}(X) = \big(1, X, \ldots, X^{n'-1}\big) \cdot \widetilde{\mathcal{A}}$ and evaluate it at the challenge $x = X^{\omega}$. The verifier can check that the evaluation in the challenge is correct. While the protocol in [8] proves evaluation in an integer challenge, we needed to separate the evaluation point of the polynomial from our challenge set. For our polynomial evaluation protocol we come up with the following idea: Let $f(X) = q(X) \cdot (X - v) + w$, where $v \in \mathbb{Z}_q$ is the evaluation point and $f(v) = w \in \mathbb{Z}_q$. It is obvious that while $f(X)$ is a polynomial of degree $n-1$, the degree of $q(X)$ is $n-2$. To commit to $f(x)$ and to prove knowledge of an evaluation point v and the corresponding evaluation value w we proceed as follows. We commit to $f(X)$ and $q(X)$ using the polynomial commitment scheme from [8], while commitments to $v, w, (X - v)$ are computed using the classical technique, i.e. we encode v, w into constant coefficients and commit to them.

One of the main challenges in all lattice-based zero-knowledge proofs is the smallness property of extracted values as required by the underlying SIS problem. While in [8] the polynomial $f(X)$ has the following form: $(1, X^{n'}, \ldots, X^{(m'-1)n'}) \cdot \mathcal{A} \cdot (1, X, \ldots, X^{n'-1})^{tr}$. the same representation in the lattice setting would yield problems during the extraction procedure because a commitment to $\tilde{\mathbf{f}}(X) = (1, X^{n'}, \ldots, X^{(m'-1)n'}) \cdot \mathcal{A}$ would break the smallness property of the extracted witness. We fixed the problem by transposing the $m' \times n'$ matrix $\mathcal{A}$ to an $n' \times m'$ matrix $\mathcal{A}^{tr}$, such that the prover commits to $\hat{\mathbf{f}}(X) = (1, X, \ldots, X^{n'-1}) \cdot \mathcal{A}^{tr}$ which is a partial polynomial evaluation at a challenge point X. For the witness extraction procedure we apply a Vandermonde matrix $\mathbf{V}$ to a system of equations $\mathbf{V} \cdot \mathbf{a} = \mathbf{c}$, where $\mathbf{a}$ denotes a row of the matrix $\widetilde{\mathcal{A}}$ and $\mathbf{c}$ is a commitment of the corresponding row in the masked matrix $\widetilde{\mathcal{A}}$. The inverse of a Vandermonde matrix is small, since the determinant of $\mathbf{V}$ is a product of $(x_j - x_i)$, such that its inverse is small according to Lemma 6 in [12].

Our second contribution is motivated by the polynomial batch protocol from [9]. The main difference to their construction [9] is that our proof is defined over lattices using the modified polynomial argument from our first contribution. We compute a simple relation between two multivariate polynomials $\mathbf{P}$ and $\mathbf{Q}$, where each polynomial is represented by a vector of l_P, l_Q polynomials in multiple variables, respectively. In the construction of the batch polynomial protocol the main challenge is to find a suitable packing procedure to pack several multi-variate polynomial instances into one commitment. The inputs to the polynomials $\mathbf{P}, \mathbf{Q}$ are vectors of multivariate polynomials $\mathbf{a}_{j,i} \in \mathbb{Z}_q^{l_a}, \mathbf{b}_{j,i} \in \mathbb{Z}_q^{l_b}$ for all $j \in [0, n'-1], i \in [0, m'-1]$. Each of these polynomials is a vector of l_a, l_b monomial coefficients, respectively. It holds $\mathbf{a}_{j,i} = (a_{j,i,1}, \ldots, a_{j,i,l_a})$ and $\mathbf{b}_{j,i} = (b_{j,i,1}, \ldots, b_{j,i,l_b})$. The input vector $\mathbf{a}_j, i$ is private while the vector $\mathbf{b}_{j,i}$ is public to the protocol. Therefore following the idea of the polynomial evaluation protocol we first need to mask $\mathbf{a}_{j,i}$ using the technique from Section 3 and obtain the masked vector $\tilde{\mathbf{a}}_{j,i}$. The next challenge of this contribution is to find a suitable technique to commit to the multivariate polynomial $\mathbf{Q}(\mathbf{a}_{j,i}, \mathbf{b}_{j,i})$ over lattices. Our solution is to use a module-SIS-based commitment scheme to commit to the multivariate polynomial vector $\mathbf{Q} = (Q_1, \ldots, Q_{l_Q})$. The dimension of this vector is l_Q and corresponds to the number of batched instances that we want to prove the relation $\mathsf{R}_{\texttt{batch}}$ of. Each of the l_Q vector components $\mathbf{Q}_\iota$, for $\iota \in [l_Q]$ is again a polynomial in $l_a + l_b$ variables. The packing procedure works as follows: Let $n = n' \cdot m' - 1$ and m' be the number of instances we want to pack into one commitment $\mathtt{C}_i$ for $i \in [0, n']$. We define each component $\mathbf{Q}_\iota$ of $\mathbf{Q}$ in each of it's $l_a + l_b$ variables, by holding the remaining $l_a + l_b - 1$ variables fixed. Then, we pack polynomials which are defined in the same variable into one ring element. Finally, we obtain a matrix $\mathbf{Q}$ of size $l_Q \times (l_a + l_b)$ of packed m' polynomials. We commit to this matrix using a module-SIS commitment.

1.2 Application.

With our solution we achieve a communication-efficient lattice-based polynomial batch protocol which can be applied to an efficient range-proof protocol in post-quantum setting. The latter can be compared to the results in [11,23,22]. For the conversion of our batch polynomial protocol into a range proof we follow the technique from [9]. In the following table we provide a comparison of our range proof application with those in [11,22]. All three constructions are based on the module SIS assumption. The main difference between our construction and the proof in [11] is that a simple repetition of our range proof has a soundness error of 1/3. Thus, we require up to $\kappa = \mathcal{O}(\lambda/\log(3))$ repetitions to achieve an overall soundness error of $1/(2^{\lambda})$, while in [11] the authors provide an one-shot proof to achieve the same soundness error. Compared to [11], we achieve a square-root improvement in the number of batched instances t, while the dependence of n remains linear. We note, that for a higher number of batched instances our proof outperforms the proof from [11], while for lower batched instances it performs less than the concurrent proof in [11]. In contrast to [22], our range proof yields a significant improvement of communication costs, since the range proof in [22] is linear in the number of batched instances t and the ring dimension n and quadratic in the logarithm of the range size $N = 2^{\ell} - 1$, while our proof is only linear in ℓ. We achieve an asymptotic improvement of the proof size.

Protocols	Torres et al. [22]	Esgin et al. [11]	Our Work
1 batched instance	$\kappa\tilde{n}n(\ell+1)\log q$	$\mathcal{O}((\tilde{n}+1)n\log q)$	$\mathcal{O}(\kappa(\sqrt{n})\log q)$
t batched instances	$t\tilde{n}n(\ell+1)\log q$	$\mathcal{O}((\tilde{n}+t)n\log q + tn\ell)$	$\mathcal{O}(\kappa(\sqrt{\ell t})n\log q)$

Table 1. Comparison of communication costs, with l being the logarithm of range size $N = 2^{\ell} - 1$ in [23], t - the number of batched instances and κ is the number of repetitions and $\tilde{n}$ - module rank of M-SIS

2 Preliminaries

Definition 2.1 **(**$Module - SIS_{q,n,m,\beta}$**, [17]).** *Let $\mathcal{R}$ be some ring and $\mathcal{K}$ some distribution over $\mathcal{R}_q^{n\times m}$. Given a random matrix $\mathbf{A} \in \mathcal{R}_q^{n\times m}$ sampled from $\mathcal{K}$, find a non-zero vector $\mathbf{v} \in \mathcal{R}_q^m$ such that $\mathbf{A}\mathbf{v} = 0$ and $\|\mathbf{v}\|_2 \leq \beta$.*

Definition 2.2 **(**$Module - LWE_{q,n,m,\chi}$**, [17]).** *Let χ be a distribution over $\mathcal{R}_q$, $\mathbf{s} \leftarrow \chi^n$ be a secret key. $LWE_{q,\mathbf{s}}$ distribution is obtained by sampling $\mathbf{a} \leftarrow \mathcal{R}_q^n$ and error $e \leftarrow \chi$ and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}\rangle + e)$. The goal is to distinguish between m given samples which are either from $LWE_{q,\mathbf{s}}$ or from $\mathcal{U}(\mathcal{R}_q^n, \mathcal{R}_q)$.*

Theorem 2.3 **(Rejection Sampling[19]).** *Let V be a subset of $\mathbb{Z}^n$ in which all elements have norms less than T, and let h be a probability distribution over V. Then, for any constant M, discrete normal distribution $\mathfrak{D}_\sigma$ over $\mathbb{Z}$ with standard deviation σ, there exists a $\sigma = \tilde{\Theta}(T)$, such that the output distribution of the following algorithms A, F are statistically close:*
***Algorithm** A: (1). $\mathbf{v} \leftarrow_{\$} h$. (2). $\mathbf{z} \leftarrow_{\$} \mathfrak{D}^n_{\mathbf{v},\sigma}$. (3). Output $(\mathbf{z}, \mathbf{v})$ with probability* $\min\big(\exp\big(\frac{-2\langle \mathbf{z},\mathbf{v}\rangle + \|\mathbf{v}\|^{\mathbf{2}}}{2\sigma^2}\big), 1\big)$.
***Algorithm** F: (1). $\mathbf{v} \leftarrow_{\$} h$, (2). $\mathbf{z} \leftarrow_{\$} \mathfrak{D}^n_{\sigma}$, (3). Output $(\mathbf{z}, \mathbf{v})$ with prob. $1/M$. Moreover, the probability that A outputs something is exponentially close to that of F, i.e. $1/M$.*

Lemma 2.4 (Adapted from [6]). *Let* $\mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$ *where* $n > 1$ *is a power of* 2 *and* $0 < i, j < 2n - 1$. *Then all the coefficients of* $2(X^i - X^j)^{-1} \in \mathcal{R}$ *are in* $\{-1, 0, 1\}$. *This implies that* $\|2(X^i - X^j)^{-1}\| \leq \sqrt{n}$.

Esgin et al. [12] provided a generalization of Lemma 2.4 stating that for all monomial challenges $x_i = X^{\omega_i}$ for $0 \leq \omega_i \leq 2n - 1$ the following relation holds for the zero-coefficient of the last row of inverse of the Vandermonde matrix:

$$\|2^k a_0\| = \|\prod_{i=1}^{k} \frac{2}{x_i - x_0}\| = \|\prod_{i=1}^{k} 2\big(X^{\omega_i} - X^{\omega_0}\big)^{-1}\| \leq n^{k-1/2} \tag{1}$$

Lemma 2.5. *Let* $x_i = X^{\omega_i} \in \mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$ *for* $0 \leq \omega_i \leq 2n - 1$ *and* $0 \leq i \leq k$. *Define the Vandermonde matrix* $\mathbf{V}$ *of dimension* $k + 1$, *where* $i-th$ *row is the vector* $(1, x_i, x_i^2, \ldots, x_i^k)$. *Then* $\mathbf{V}$ *is invertible, and for any entry* α_j *in the last row of* $\mathbf{V}^{-1}$, *we have* $\|2^k \alpha_j\| \leq n^{k-0.5}$.

Lemma 2.6 (Lemma 4.4 in [19]). *For any* $\alpha > 0, \Pr[|z| > \alpha\sigma; z \leftarrow \mathfrak{D}_\sigma] \leq 2\exp\left(-\alpha^2/2\right)$. *For any* $\alpha > 1, \Pr[\|\mathbf{z}\| > \alpha\sigma\sqrt{m}, \mathbf{z} \leftarrow \mathfrak{D}_\sigma^m] \leq \alpha^m \exp\left(\frac{m(1-\alpha^2)}{2}\right)$. *In particular:* $\Pr[|z| > 12\sigma : z \leftarrow \mathfrak{D}_\sigma] < 2^{-100}$ *and* $\Pr[\|\mathbf{z}\| > 5\sigma : z \leftarrow \mathfrak{D}_\sigma^n] < 2^{-100}$, *if* $n \geq 7$.

Definition 2.7 (Commitment Scheme). *Let* $n = \nu + \nu', m, q, \mathcal{B}$ *be positive integers. Let* $S_{\mathcal{M}}$ *denote a message space. The relaxed commitment of a message* $\mathbf{x} \in S_{\mathcal{M}}$ *is defined as:*

`KeyGen`*: Create* $(\mathbf{A}_1, \mathbf{A}_2) \in \mathcal{R}_q^{\nu \times m} \times \mathcal{R}_q^{\nu' \times m}$. *Public parameters are created as follows:*

$$\mathbf{A}_1 = [\mathbf{I}_\nu \| \mathbf{A}_1'], \quad \textit{where} \quad \mathbf{A}_1' \leftarrow \mathcal{R}_q^{\nu \times (m-\nu)}$$
$$\mathbf{A}_2 = [\mathbf{0}^{\nu' \times \nu} \| \mathbf{I}_{\nu'} \| \mathbf{A}_2'], \quad \textit{where} \quad \mathbf{A}_2' \leftarrow \mathcal{R}_q^{\nu' \times (m-\nu-\nu')}$$

Set the commitment key $ck = \mathbf{A} = \begin{bmatrix} \mathbf{A_1} \\ \mathbf{A}_2 \end{bmatrix}$. *This commitment key is used to commit to messages* $\mathbf{x} \in \mathcal{R}_q^{\nu'}$.

`Com`*: To commit to a message* $\mathbf{x} \in \mathcal{R}_q^{\nu'}$, *choose a random polynomial vector* $\mathbf{r} \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^m$ *and output the commitment* $\mathtt{C} := \mathtt{Com}_{ck}(\mathbf{x}, \mathbf{r}) = \mathbf{A} \cdot \mathbf{r} + \mathbf{x} = \mathbf{A} \cdot \mathbf{r} + \mathsf{enc}(\mathbf{x})$, *where* $\mathsf{enc}(\mathbf{x}) = \begin{bmatrix} \mathbf{0}^\nu \\ \mathbf{x} \end{bmatrix}$.

`ROpen`*: A valid opening of a commitment* `C` *is a tuple consisting of* $\mathbf{x} \in \mathcal{R}_q^{\nu'}$, $\mathbf{r} \in \mathcal{R}_q^m$ *and* $d \in \Delta\mathcal{CH}$. *The verifier checks that* $d \cdot \mathtt{C} = \mathbf{A} \cdot \mathbf{r} + d \cdot \mathsf{enc}(\mathbf{x})$, *and that* $\forall\ 1 \leq i \leq k$, *we have that* $\|r_i\|_2 \leq \gamma_{bind}$. *Otherwise return* 0.

Security of this commitment scheme has been proved in [2].

Lemma 2.8. *If* $\mathsf{M}-\mathsf{LWE}_{q,m-\nu-\nu',\nu+\nu',\mathcal{U}(\{-\mathcal{B},\ldots,\mathcal{B}\}^m)}$ *problem is hard than the above commitment scheme is computationally hiding. If* $\mathsf{M}-\mathsf{SIS}_{q,\nu+\nu',m,\beta}$ *problem is hard, then our commitment scheme is computationally* $\gamma_{bind}-$*binding with respect to the relaxation factor* d.

The binding property of this commitment scheme relies on the hardness of Ring-SIS problem defined in Definition 2.1 where the number of rows of matrix $\mathbf{A}$ is set to be equal 1. If $\nu + \nu' > 1$, we obtain a commitment scheme whose binding property relies on the hardness of $Module - SIS_{q,\nu+\nu',m,\beta}$ assumption. For the full security proof of Module-SIS commitment scheme, we refer to [17].

3 Lattice-based Polynomial Zero-Knowledge Argument

We present the first lattice-based version of protocol where we commit to a polynomial and later reveal the evaluation of $f(X)$ at any point $x \in \mathbb{Z}_q$. The commitment scheme `Com` is a primitive that allows one party to commit to a chosen value while keeping it secret to other parties, then this committed value can be revealed later. We use the commitment scheme from definition 2.7.

3.1 Commitments to Polynomials (`PolyCom`)

Let $f(X) = \sum_{k=0}^{n} a_k X^k \in \mathcal{R}_q$, where $a_k \in \mathbb{Z}_q$ for $k = [0, n-1\}$ and let $n = m'n'-1$. Decomposition of this polynomial yields: $f(X) = \sum_{i=0}^{m'-1} \sum_{j=0}^{n'-1} a_{j,i} X^{in'+j}$, $a_{j,i} \in \mathbb{Z}_q$, where the coefficients form the following matrix

$$\mathcal{A} = \begin{pmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,m'-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,m'-1} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n'-1,0} & a_{n'-1,1} & \cdots & a_{n'-1,m'-1} \end{pmatrix}. \tag{2}$$

Let $S_{\mathbf{r}}(\beta) = \{\mathbf{r}_{a,j} \in \mathcal{R}_q^m : \|\mathbf{r}_{a,j}\| \leq \beta, j \in [0, m']\}$ be the randomness space. The commitment to each row $\mathbf{a}_i = (a_{i,0}, \ldots, a_{i,m'-1})$ of $\mathcal{A}$ is given as:

$$\mathbb{A}_j = \mathsf{Com}_{\mathbf{A}}(\mathbf{a}_j, \mathbf{r}_{a,j}) = \mathbf{A}\mathbf{r}_{a,j} + \mathsf{enc}(\mathbf{a}_j) \in \mathcal{R}_q^2, \tag{3}$$

where $\mathsf{enc}(\mathbf{a}_j)$ is the polynomial in $\mathcal{R}_q^2$ and $\mathbf{A} \in \mathcal{R}_q^{2\times m}$, $\mathbf{r}_{a,j} \leftarrow \mathcal{U}(S_{\mathbf{r}}(\beta))$, for all $0 \leq j \leq n'-1$ and $\mathbf{a}_j \in \mathbb{Z}_q^{m'}$ and $\mathbf{0}$ is a $(n-m')$-dimensional zero-vector. We describe the polynomial commitment via the function `PolyCom` which on input $f, \mathbf{r}_{a,j}$ follows the following computation steps:

$$\hat{\mathbf{f}}(X) = (1, X, \cdots, X^{n'-1}) \cdot \mathcal{A} = (\sum_{j=0}^{n'-1} a_{j,0} X^j, \ldots, \sum_{j=0}^{n'-1} a_{j,m'-1} X^j). \tag{4}$$

The function $\hat{\mathbf{f}}(X)$ can not be sent to the verifier, since it would leak information about the coefficients of f. To avoid any leakage of the secret values, we sample masking values, $u_{j,i} \leftarrow_{\$} \mathfrak{D}_\sigma$. We encode the masking values into monomials for all $i \in [1, m'-1]$ as follows: $\overline{\mathsf{enc}}(u_{0,i}) = u_{0,i} X^{n'-1}, \overline{\mathsf{enc}}(u_{1,i}) = u_{1,i} X^{n'-3}, \ldots, \overline{\mathsf{enc}}(u_{n'-1,i}) = u_{n'-1,i} X^{1-n'}$ and mask each entry of the coefficient matrix as follows:

$$\widetilde{\mathcal{A}} = \begin{pmatrix} a_{0,0} & a_{0,1} - u_{0,1} X^{n'-1} & \cdots & a_{0,m'-1} - u_{0,m'-1} X^{n'-1} \\ a_{1,0} & a_{1,1} - u_{1,1} X^{n'-3} & \cdots & a_{1,m'-1} - u_{1,m'-1} X^{n'-3} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n'-1,0} & a_{n'-1,1} - u_{n'-1,1} X^{1-n'} & \cdots & a_{n'-1,m'-1} - u_{n'-1,m'-1} X^{1-n'} \\ \sum_{j=0}^{n'-1} u_{n'-j-1,1} X^j & \sum_{j=0}^{n'-1} u_{n'-j-1,2} X^j & \cdots & 0 \end{pmatrix} \tag{5}$$

To show correctness we let: $f(X) = (1, X, \ldots, X^{n'}) \cdot \widetilde{\mathcal{A}} \cdot (1, X^{n'}, \ldots, X^{(m'-1)n'})^{tr}$. We compute $\tilde{\mathbf{f}}(X) = (1, X, \ldots, X^{n'}) \cdot \widetilde{\mathcal{A}}$. The result is:

$$\begin{pmatrix} a_{0,0} + a_{1,0}X + \ldots + a_{n'-1,0}X^{n'-1} + X^{n'} \sum_{j=0}^{n'-1} u_{n'-j-1,1}X^j, \\ a_{0,1} - u_{0,1}X^{n'-1} + \ldots + (a_{n'-1,1} - u_{n'-1,1}X^{1-n'})X^{n'-1} + X^{n'} \sum_{j=0}^{n'-1} u_{n'-j-1,2}X^j, \\ a_{0,2} - u_{0,2}X^{n'-1} + \ldots + (a_{n'-1,2} - u_{n'-1,2}X^{1-n'})X^{n'-1} + X^{n'} \sum_{j=0}^{n'-1} u_{n'-j-1,3}X^j, \\ \vdots \\ a_{0,m'-2} - u_{0,m'-2}X^{n'-1} + \ldots + X^{n'} \sum_{j=0}^{n'-1} u_{n'-j-1,m'-1}X^j, \\ a_{0,m'-1} - u_{0,m'-1}X^{n'-1} + \ldots + (a_{n'-1,m'-1} - u_{n'-1,m'-1}X^{1-n'})X^{n'-1} \end{pmatrix} \tag{6}$$

It is important to notice that the last component also hides all coefficients by adding the coefficients $u_{i,m'-2}$ to $a_{i,m'-1}$ for $i \in [0, n'-1]$. We reconstruct $f(X)$ by multiplying $\tilde{\mathbf{f}}(X)$ by $(1, X^{n'}, \ldots, X^{(m'-1)n'})^{tr}$ from the right. To improve the readability of the final result, we first provide the results of component-wise multiplication and add up the results in the next step:

$$(0).\ \ a_{0,0} + a_{1,0}X + \ldots + a_{n'-1,0}X^{n'-1} + X^{n'} \sum_{j=0}^{n'-1} u_{n'-j-1,1}X^j$$

$$\vdots$$

$$(m'-1).\ \ a_{0,m'-1} - u_{0,m'-1}X^{m'n'-1} + \ldots + (a_{n'-1,m'-1} - u_{n'-1,m'-1}X^{1-n'})X^{m'n'-1}$$

Finally, we add up the m' results. We need to show that the coefficients $u_i, i \in [1, m'-1]$ will be cancelled out such that the final result yields $f(X)$. This can be followed from a careful decomposition of the above system. To justify that this masking scheme does not reveal the secret coefficients of the polynomials we consider the following observation. When an adversary obtains the term $\tilde{\mathbf{f}}(X)$, she sees the vector (6). Let's have a look at the i-th coordinate of this vector, where $i \in [1, m'-1]$: $a_{0,i} - u_{0,i}X^{n'-1} + (a_{1,i} - u_{1,i}X^{n'-3})X + \ldots + X^{n'} \sum_{j=0}^{n'-1} u_{n'-j-1,i+1}X^j$. It follows that $\mathsf{enc}(u_{0,i})$ can be combined with the coefficient $a_{n'-1,i}$, $\mathsf{enc}(u_{1,i})$ can be combined with coefficient $a_{n'-2,i}$ and so on, such that all polynomial coefficients are perfectly hidden. When an adversary observes the last term $X^{n'} \sum_{j=0}^{n'-1} u_{n'-j-1,i+1}X^j$, she will get monomials $\mathtt{Coef}_0 \cdot X^{n'}, \ldots, \mathtt{Coef}_{n'-1} \cdot X^{2n-1}$, where $\mathtt{Coef}_j$ are equal to $u_{(\cdot)}$ and serve as the masking coefficients. The commitments to the rows of $\widetilde{\mathcal{A}}$ are:

$$\widetilde{\mathbb{A}}_j = \mathtt{Com}_{\mathbf{A}}(\tilde{\mathbf{a}}_j, \mathbf{r}_{a,j}) = \mathbf{A}\mathbf{r}_{a,j} + \mathsf{enc}(\tilde{\mathbf{a}}_j),\ \forall j \in [0, m'-1] \tag{7}$$

where $\tilde{\mathbf{a}}_j$ are the rows of the masked matrix $\widetilde{\mathcal{A}}$. We also commit to the vector $\mathbf{u} = \big(\sum_{j=0}^{n'-1} u_{n'-j-1,1}X^j, \ldots, \sum_{j=0}^{n'-1} u_{n'-j-1,m'-1}X^j, 0\big)$ by firstly encoding each component $\sum_{j=0}^{n'-1} u_{n'-j-1,i+1}X^j$ into the i-th coefficient c_i of the polynomial $f(X) = \sum_{i=0}^{n-1} c_iX^i$. Let $\mathbf{r}_u \leftarrow S_{\mathbf{r}}(\beta)$, then holds:

$$\mathbb{U} = \mathtt{Com}_{\mathbf{A}}(\mathbf{u}, \mathbf{r}_u) = \mathbf{A} \cdot \mathbf{r}_u + \mathsf{enc}(\mathbf{u}). \tag{8}$$

Commitment to the vector $\tilde{\mathbf{f}}(X)$ is: $\mathtt{Com}_{\mathbf{A}}\big(\sum_{j=0}^{n'-1}\tilde{\mathbf{f}}(X);\tilde{\mathbf{r}}\big)=\sum_{j=0}^{n'-1}X^j\widetilde{\mathbb{A}}_j+\mathbb{U}\cdot X^{n'}$ where $\tilde{\mathbf{r}}_{\mathbf{f}}=\sum_{j=0}^{n'-1}\mathbf{r}_{a,j}X^j+\mathbf{r}_uX^{n'}$.

Remark 3.1. Our commitments $\widetilde{\mathbb{A}}_j$ for $j\in[0,n'-1]$, $\mathbb{U}$ satisfy binding and hiding property as showed in [3], Lemma 2.6 and 2.7.

3.2 Polynomial Evaluation Protocol Π_{PEv}

Definition 3.2. *The relation and the corresponding relaxed relation of the polynomial evaluation protocol are defined as follows:*

$$\mathsf{R}_{\beta}^{\mathsf{PEv}}=\big\{(v,w,f(X),q(X),\mathbf{r}_v,\mathbf{r}_w),(\mathtt{C}_v,\mathtt{C}_w):f(X)=q(X)(X-v)+w,f(u)=w,\|\mathbf{r}_\iota\|\leq\beta,\iota\in\{v,w\}\big\}$$
$$\widehat{\mathsf{R}}_{\hat{\beta}}^{\mathsf{PEv}}=\big\{(\hat{v},\hat{w},\hat{f}(X),\hat{q}(X),\hat{\mathbf{r}}_v,\hat{\mathbf{r}}_w),(\mathtt{C}_{\hat{v}},\mathtt{C}_{\hat{w}}):\hat{f}(X)=\hat{q}(X)(X-\hat{v})+\hat{w},f(\hat{u})=\hat{w},\|\mathbf{r}_\iota\|\leq\hat{\beta},\iota\in\{\hat{v},\hat{w}\}\big\}.$$

Challenge Space. We define the challenge space $\mathcal{CH}$ being a set of monomials $X^\iota\in\mathcal{R}_q$ for $\iota\in\{-1,0,1\}$. Since the protocol does not fit into the usual security definitions of soundness, completeness and zero-knowledge property, similar to [9] we provide these definitions adapted to our protocol.

The protocol consists of four steps **Prover**, **Challenge**, **Response**, **Verification** where the prover runs `PolCom` algorithm and after receiving a challenge from the verifier, she runs the `Resp` algorithm. Finally the verifier runs the `PolVrfy` algorithm. For all $j\in[0,n'-1],j'\in[0,\nu-1]$ we set $\mathsf{pc}=\{\{\widetilde{\mathbb{A}}_j\}_j,\{\widetilde{\mathbb{T}}_{j'}\}_{j'},\mathbb{U},\mathbb{S}\},\mathsf{c}=\{\mathtt{C}_{\mathtt{m}_v},\mathtt{C}_{\mathtt{m}_w}\}$, $\mathsf{st}_1=\{\{\mathbf{r}_{a,j}\}_j,\{\mathbf{r}_{t,j'}\}_{j'},\mathbf{r}_u,\mathbf{r}_s\}$, $\mathsf{st}_2=\{\mathbf{r}_v,\mathbf{r}_w\}$ and $\mathsf{re}=\{\mathtt{C}_{\tilde{\mathbf{f}}},\mathtt{C}_{\tilde{\mathbf{q}}},\mathtt{C}_{0,\tilde{\mathbf{f}}},\mathtt{C}_{0,\tilde{\mathbf{q}}},\xi_{\mathtt{m}_v},\xi_{\mathtt{m}_w},\mathbf{r}_{\xi,\mathtt{m}_v},\mathbf{r}_{\xi,\mathtt{m}_w}\}$.

Definition 3.3 (Completeness.). *Our protocol has perfect completeness if for all non-uniform PPT adversaries $\mathfrak{A}$ and completeness error α holds:*

$$Pr\big[(\mathbf{A},m',n',f(X),q(X),p(X)\leftarrow\mathfrak{A}(1^\lambda),(\mathsf{pc},\mathsf{st}_1)\leftarrow\mathtt{PolCom}(\mathbf{A},m',n',f(X),q(X));\big(\mathsf{c},\mathsf{st}_2\big)\leftarrow$$
$$\mathtt{Com}(v,w);\mathsf{re}\leftarrow\mathtt{Resp}(f(X),q(X),x),w\leftarrow\mathtt{PolVrfy}(\mathbf{A},m',n',\mathsf{pc},\mathsf{c},\mathsf{re},u):w=f(v)\big]=1-\alpha.$$

Next, we define 3−special soundness which given 3 accepting evaluations for the challenges $x_\ell,\ell\in[1,3]$ but using the same commitments allows to either extract a witness or it breaks the binding property of our commitment scheme.

Definition 3.4 (3-Special Soundness.). *Our protocol is statistically 3-special sound if there exists a PPT algorithm χ that either extracts a valid polynomial $f(X)$ or breaks the binding property of the underlying commitment scheme. For all adversaries $\mathfrak{A}$ and all $L\geq 3$ and $\mathbf{x}_{s,n'}=(1,x_s,\ldots,x_s^{n'}),\overline{\mathbf{r}}=(\mathbf{r}_0,\ldots,\mathbf{r}_{n'-1},\mathbf{r}_u)$.*

$$Pr[\mathbf{A}\leftarrow\mathtt{KeyGen}(1^\lambda),(m',n',\mathsf{pc},\mathsf{c},\{x_i,\mathsf{re}_i\}_{i\in[1,L]})\leftarrow\mathfrak{A}(\mathbf{A});(\widetilde{\mathcal{A}},\mathsf{st}_1,\mathsf{st}_2)\leftarrow\chi(\mathbf{A},m',n',\mathsf{pc},\mathsf{c},$$
$$\{x_i,\mathsf{re}_i\}_{i\in[1,L]}),w_i\leftarrow\mathtt{PolVrfy}(\mathbf{A},m',n'\{\mathsf{pc},\mathsf{c},\mathsf{re}_i,x_i):\forall i:w_i=f(x_i)=q(v_i)p(v_i)+w_i$$
$$\vee\,\exists\ s:\ \mathtt{Com}_{\mathbf{A}}(\tilde{\mathbf{f}}(x_s),\tilde{\mathbf{r}}(x_s))=\mathbf{x}_{s,n'}\mathtt{Com}_{\mathbf{A}}(\widetilde{\mathcal{A}},\overline{\mathbf{r}})\wedge\tilde{\mathbf{f}}(x_i)\neq\mathbf{x}_{s,n'}\cdot\widetilde{\mathcal{A}}.]\approx 1$$

The next definition states that given any value v and evaluation value x it is possible to simulate the commitments and the evaluation output of `PolEv` which is distributed as in the real protocol.

Definition 3.5 (Special Honest Verifier Zero-Knowledge). *Our protocol has special honest verifier zero knowledge if there exists a PPT simulator $\mathcal{S}$ such that for all interactive non-uniform polynomial time adversaries $\mathfrak{A}$ holds:*

$$Pr\big[(\mathbf{A}, m'n', f(X), x) \leftarrow \mathfrak{A}(1^\lambda), (\mathsf{pc}, \mathsf{c}, \mathsf{st}_1, \mathsf{st}_2) \leftarrow \texttt{PolCom}(\mathbf{A}, m', n', f(X)), \mathsf{re} \leftarrow \mathsf{Resp}(\mathsf{st}_1, \mathsf{st}_2, x) :$$
$$\mathfrak{A}(\mathsf{pc}, \mathsf{c}, \mathsf{re}) = 1] \approx Pr\big[\mathbf{A}, m'n', f(X), x \leftarrow \mathfrak{A}(1^\lambda), (\mathsf{pc}, \mathsf{c}, \mathsf{re}) \leftarrow \mathcal{S}(\mathbf{A}, m', n', x, f(x)) : \mathfrak{A}(\mathsf{pc}, \mathsf{c}, \mathsf{re}) = 1\big].$$

Construction of Π_{PEv}. The common input to the protocol is given by the public parameter $\mathbf{A} \in \mathcal{R}_q^{2\times m}$ with $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n+1)$. At the beginning the prover commits to a secret value v at which the polynomial $f(X)$ should be evaluated and to the evaluated value w such that $f(v) = w$. We note that a n-degree polynomial $f(X)$ can be represented as follows: $f(X) = q(X) \cdot (X - v) + f(v) = q(X) \cdot (X - v) + w$, where $q(X)$ is a polynomial of degree $n - 1$. We apply the same decomposition technique to this polynomial as we did it for $f(X)$. Let $q(X) = \sum_{k'=0}^{n-1} t_{k'} X^{k'} = \sum_{i'=0}^{\mu-1} \sum_{j'=0}^{\nu-1} t_{j',i'} X^{i'\nu+j'}$ which can be represented by a matrix $\mathcal{T}$ with the corresponding masked matrix $\widetilde{\mathcal{T}}$. The commitments to each row of the masked matrix $\widetilde{\mathcal{T}}$ are defined as $\{\widetilde{\mathbb{T}}_{j'}\}_{j'\in[0,\nu-1]}$ and $\mathbb{S}$ is the commitment to the last row of the masked matrix $\widetilde{\mathcal{T}}$ which is given as $\mathbf{s} = \big(\sum_{j'=0}^{\nu-1} s_{\nu-j'-1,1} X^{j'}, \ldots, \sum_{j'=0}^{\nu-1} s_{\nu-j'-1,\mu-1} X^{j'}, 0\big)$.
Our protocol contains three moves, where in the first move the prover runs the `PolCom` algorithm to commit to a polynomial $f(X)$, in the second move the verifier sends a challenge $x = X^\omega$, and in the last move the prover responds. Finally, verifier runs the `PolVrfy` algorithm to verify $f(x)$. (A more formal description of the protocol is given in Figure 2, Appendix A).

Common input: Commitments to the evaluation point $v \in \mathbb{Z}_q$ and to $w = f(v) \in \mathbb{Z}_q$, i.e. $\mathtt{C}_v := \mathtt{Com}_{\mathbf{A}}(v, \mathbf{r}_v), \mathtt{C}_w := \mathtt{Com}_{\mathbf{A}}(w, \mathbf{r}_w)$ for uniformly random values $\mathbf{r}_v, \mathbf{r}_w \leftarrow_{\$} \mathcal{R}_q^m$ which are only known to the prover. These commitments to v, w are computed by encoding the integers into constant polynomials, i.e. $v = vX^0$, $w = wX^0$ and using the commitment scheme from Definition 2.7.

Prover: The prover runs $\texttt{PolCom}(n', m', \mathbf{A}, f(X))$ and $\texttt{PolCom}(\nu, \mu, \mathbf{A}, q(X))$ to commit to the polynomials $f(X) = \sum_{k=0}^{n} a_k X^k \in \mathcal{R}_q$ and $q(X) = \sum_{k'=0}^{n-1} t_{k'} X^{k'} \in \mathcal{R}_q$. Let $f(X), q(X)$ be polynomials with coefficients in $\mathbb{Z}_q$. Let $n = m'n' - 1$ and $n - 1 = \mu\nu - 1$. These polynomials can be encoded into a matrix $\mathcal{A}$ of dimension $n' \times m'$ and into a matrix $\mathcal{T}$ of dimension $\nu \times \mu$ as defined in (2). She reconstructs the polynomials $f(X)$ and $q(X)$ using the matrix representation (2) as follows:

$$f(X) = (1, X, \cdots, X^{n'-1}) \cdot \mathcal{A} \cdot (1, X^{n'}, \ldots, X^{(m'-1)n'})^{tr}, \tag{9}$$
$$q(X) = (1, X, \cdots, X^{\nu-1}) \cdot \mathcal{T} \cdot (1, X^{\nu}, \ldots, X^{(\mu-1)\nu})^{tr} \tag{10}$$

The prover commits to the polynomial and shows that it evaluates at any point from $\mathcal{R}_q$. Committing to a polynomial is done by computing commitments to each row of $\mathcal{A}$ or $\mathcal{T}$, respectively. She first needs to mask the matrices $\mathcal{A}$ and $\mathcal{T}$, by picking random values $u_{j,i} \leftarrow_{\$} \mathbb{Z}_q$, and $s_{j',i'} \leftarrow_{\$} \mathbb{Z}_q$ for $i \in [0, m'-1]$, $j \in [0, n'-1]$, $i' \in [0, \mu-1]$, $j' \in [0, \nu-1]$ where $\max |u_{j,i}| \leq \beta$.

Each entry $a_{j,i}$ of $\mathcal{A}$ is masked into $\tilde{a}_{j,i} = a_{j,i} - u_{j,i} X^{n'-(2j+1)}$ of the matrix

$\widetilde{\mathcal{A}}$. Similarly, each entry $t_{j',i'}$ of the matrix $\mathcal{T}$ is masked into $\tilde{t}_{j',i'} = t_{j',i'} - s_{j',i'}X^{\nu-(2j'+1)}$ of the new matrix $\widetilde{\mathcal{T}}$. Similarly, $\widetilde{\mathcal{T}}$ is computed as in (5) except that it's dimension is $(\nu+1) \times \mu$. We denote each entry of $\widetilde{\mathcal{A}}$ and $\widetilde{\mathcal{T}}$ by $\tilde{a}_{j,i}$ and $\tilde{t}_{j',i'}$ for $j \in [0, n'-1], i \in [0, m'-1]$ and $j' \in [0, \nu-1], i' \in [0, \mu-1]$, respectively. We add an additional row $\mathbf{u}(X)$ to $\mathcal{A}$ and $\mathbf{s}(X)$ to $\mathcal{T}$ where these rows contain the polynomially encoded values of $u_{j,i}$ and $s_{j',i'}$ as it's coordinates, respectively. The vectors $\mathbf{u}(X)$ and $\mathbf{s}(X)$ denote the last rows of $\widetilde{\mathcal{A}}, \widetilde{\mathcal{T}}$, respectively: $\mathbf{u}(X) = \big(\sum_{j=0}^{n'-1} u_{n'-j-1,1}X^j, \ldots, \sum_{j=0}^{n'-1} u_{n'-j-1,m'-1}X^j, 0\big)$ and $\mathbf{s}(X) = \big(\sum_{j'=0}^{\nu-1} s_{\nu-j'-1,1}X^{j'}, \ldots, \sum_{j'=0}^{\nu-1} s_{\nu-j'-1,\mu-1}X^{j'}, 0\big)$. To ease the notations we omit X in $\mathbf{u}(X), \mathbf{s}(X)$. The prover computes the commitments $\widetilde{\mathbb{A}}_j = \mathtt{Com}_{\mathbf{A}}(\tilde{\mathbf{a}}_j, \mathbf{r}_{a,j})$ and $\widetilde{\mathbb{T}}_{j'} = \mathtt{Com}_{\mathbf{A}}(\tilde{\mathbf{t}}_{j'}, \mathbf{r}_{t,j'})$. To commit to the last row of $\widetilde{\mathcal{A}}$ or $\widetilde{\mathcal{T}}$ respectively, the prover picks $\mathbf{r}_u, \mathbf{r}_s \leftarrow_{\$} \mathfrak{D}^{mn}_{12\mathcal{B}\sqrt{mn}}$, where $\|\mathbf{r}_u\|_\infty, \|\mathbf{r}_u\|_\infty \leq \mathcal{B}$, and computes $\mathbb{U} = \mathtt{Com}_{\mathbf{A}}(\mathbf{u}, \mathbf{r}_u), \mathbb{S} = \mathtt{Com}_{\mathbf{A}}(\mathbf{s}, \mathbf{r}_s)$ which is defined as in (8). Additionally, she picks masking elements $\mathtt{m}_v, \mathtt{m}_w \leftarrow_{\$} \mathfrak{D}^{n}_{12\beta\sqrt{n}}$ for v, w and commits to it $\mathtt{C}_{\mathtt{m}_v} := \mathtt{Com}_{\mathbf{A}}(\mathtt{m}_v; \mathbf{r}_{\mathtt{m}_v})$, $\mathtt{C}_{\mathtt{m}_w} := \mathtt{Com}_{\mathbf{A}}(\mathtt{m}_w; \mathbf{r}_{\mathtt{m}_w})$, for randomly chosen values $\mathbf{r}_{\mathtt{m}_v}, \mathbf{r}_{\mathtt{m}_w}, \in \mathfrak{D}^{mn}_{12\beta\sqrt{mn}}$. The prover outputs the commitments $\{\widetilde{\mathbb{A}}_j\}_j, \{\widetilde{\mathbb{T}}_{j'}\}_{j'}$, $\mathbb{U}, \mathbb{S}, \mathtt{C}_{\mathtt{m}_v}, \mathtt{C}_{\mathtt{m}_w}$ and $\mathsf{st} = (\mathsf{st}_1, \mathsf{st}_2) = \{f(X), q(X), \{\mathbf{r}_{a,j}\}_j, \{\mathbf{r}_{t,j'}\}_{j'}, \mathbf{r}_v, \mathbf{r}_w, \mathbf{r}_{\mathtt{m}_v}, \mathbf{r}_{\mathtt{m}_w}\}$. She sends $\mathsf{pc} = \{\{\widetilde{\mathbb{A}}_j\}_j, \{\widetilde{\mathbb{T}}_{j'}\}_{j'}, \mathbb{U}, \mathbb{S}\}$, $\mathsf{c} = \{\mathtt{C}_{\mathtt{m}_v}, \mathtt{C}_{\mathtt{m}_w}\}$ to the verifier.

Challenge: The verifier sends a challenge $x = X^\omega \in \mathcal{CH}$ to the prover.

Response: The prover computes $\mathtt{Resp}(\mathsf{st}, x)$ as follows: She first computes: $\tilde{\mathbf{f}}(x) = (1, x, \ldots, x^{n'-1}, x^{n'}) \cdot \widetilde{\mathcal{A}} = \big(\sum_{j=0}^{n'-1} \tilde{a}_{j,0}x^j, \ldots, \sum_{j=0}^{n'-1} \tilde{a}_{j,m'-1}x^j\big) + \mathbf{u}(x)$ and $\tilde{\mathbf{q}}(x) = (1, x, \ldots, x^{\nu-1}, x^\nu) \cdot \widetilde{\mathcal{T}} = \big(\sum_{j'=0}^{\nu-1} \tilde{t}_{j',0}x^{j'}, \ldots, \sum_{j'=0}^{\nu-1} \tilde{t}_{j',\mu-1}x^{j'}\big) + \mathbf{s}(x)$. She commits to $\tilde{\mathbf{f}}(x)$ and $\tilde{\mathbf{q}}(x)$ with randomness $\tilde{\mathbf{r}}_{\mathbf{f}} = \sum_{j=0}^{n'-1} \mathbf{r}_{a,j}x^j + \mathbf{r}_u x^{n'}$, $\tilde{\mathbf{r}}_{\mathbf{q}} = \sum_{j'=0}^{\nu-1} \mathbf{r}_{t,j'}x^{j'} + \mathbf{r}_s x^\nu$ as follows:

$$\widetilde{\mathtt{C}}_{\tilde{\mathbf{f}}} = \mathbf{A} \cdot \big(\sum_{j=0}^{n'-1} \mathbf{r}_{a,j}x^j\big) + \mathsf{enc}\big(\big(\sum_{j=0}^{n'-1} \tilde{a}_{j,0}x^j, \ldots, \sum_{j=0}^{n'-1} \tilde{a}_{j,m'-1}x^j\big)\big) + \mathbf{A}\mathbf{r}_u \cdot x^{n'} + \mathsf{enc}(\mathbf{u}) \cdot x^{n'}$$

$$\widetilde{\mathtt{C}}_{\tilde{\mathbf{q}}} = \mathbf{A} \cdot \big(\sum_{j'=0}^{\nu-1} \mathbf{r}_{t,j'}x^{j'}\big) + \mathsf{enc}\big(\big(\sum_{j'=0}^{\nu-1} \tilde{t}_{j',0}x^{j'}, \ldots, \sum_{j'=0}^{\nu-1} \tilde{t}_{j',\mu-1}x^{j'}\big)\big) + \mathbf{A}\mathbf{r}_s \cdot x^\nu + \mathsf{enc}(\mathbf{s}) \cdot x^\nu$$

Next, the prover masks the values v, w as follows: $\xi_{\mathtt{m}_v} = \mathtt{m}_v + xv,\ \xi_{\mathtt{m}_w} = \mathtt{m}_w + xw$, and the corresponding randomness: $\mathbf{r}_{\xi,\mathtt{m}_v} = x\mathbf{r}_v + \mathbf{r}_{\mathtt{m}_v},\ \mathbf{r}_{\xi,\mathtt{m}_w} = x\mathbf{r}_w + \mathbf{r}_{\mathtt{m}_w},\ \mathbf{r}_\gamma = \tilde{\mathbf{r}}_{\mathbf{f}} - \tilde{\mathbf{r}}_{\mathbf{q}}v - \mathbf{r}_w$. According to Theorem 2.2, the prover rejects sampled values with probability $\tilde{\rho} = \max_{i\in\{1,2,3\}}(\tilde{\rho}_i)$, where

$$\rho_1 := \frac{\mathfrak{D}^{n'mn}_{12\mathcal{B}\sqrt{n'mn}}(\tilde{\mathbf{r}}_{\mathbf{f}})}{M\mathfrak{D}^{n'mn}_{\{x^j\cdot\mathbf{r}_{a,j}\}_j, 12\mathcal{B}\sqrt{n'mn}}(\tilde{\mathbf{r}}_{\mathbf{f}})}, \qquad \rho_2 := \frac{\mathfrak{D}^{\nu mn}_{12\mathcal{B}\sqrt{\nu mn}}(\tilde{\mathbf{r}}_{\mathbf{q}})}{M\mathfrak{D}^{\nu mn}_{\{x^{j'}\cdot\mathbf{r}_{t,j'}\}_{j'}, 12\mathcal{B}\sqrt{\nu mn}}(\tilde{\mathbf{r}}_{\mathbf{q}})},$$

$$\rho_3 := \frac{\mathfrak{D}^{mn}_{12\mathcal{B}\sqrt{3mn}}(\mathbf{r}_{\xi,\mathtt{m}_v}, \mathbf{r}_{\xi,m_w})}{M\mathfrak{D}^{mn}_{x\cdot\iota, 12\mathcal{B}\sqrt{3mn}}(\mathbf{r}_{\xi,\mathtt{m}_v}, \mathbf{r}_{\xi,\mathtt{m}_w})}$$

for $\iota \in \{\mathbf{r}_v, \mathbf{r}_w\}$. It holds $\|(x^0\mathbf{r}_{a,0}, \ldots, x^{n'}\mathbf{r}_{a,n'-1})\| \leq \mathcal{B}\sqrt{n'mn}$; $\|(x^0\mathbf{r}_{t,0}, \ldots, x^\nu\mathbf{r}_{t,\nu-1})\| \leq \mathcal{B}\sqrt{\nu mn}$ and $n' = \mathcal{O}(\sqrt{n})$. She computes $\mathtt{C}_{0,\tilde{\mathbf{f}}} = \mathtt{Com}_{\mathbf{A}}(\mathbf{0}, \tilde{\mathbf{r}}_{\mathbf{f}}), \mathtt{C}_{\mathbf{0},\tilde{\mathbf{q}}} = \mathtt{Com}_{\mathbf{A}}(\mathbf{0}, \tilde{\mathbf{r}}_{\mathbf{q}})$ and sends $\mathsf{re} = \{\mathtt{C}_{\tilde{\mathbf{f}}}, \mathtt{C}_{\tilde{\mathbf{q}}}, \mathtt{C}_{\mathbf{0},\tilde{\mathbf{f}}}, \mathtt{C}_{\mathbf{0},\tilde{\mathbf{q}}}, \xi_{\mathtt{m}_v}, \xi_{\mathtt{m}_w}, \mathbf{r}_{\xi,\mathtt{m}_v}, \mathbf{r}_{\xi,\mathtt{m}_w}, \mathbf{r}_\gamma\}$ to the verifier.

Verification: the verifier runs $\texttt{PolyVerify}(\mathbf{A}, m', n', \mathsf{pc}, \mathsf{c}, \mathsf{re}, x)$ and outputs: $\texttt{Com}_{\mathbf{A}}(\xi_{\mathtt{m}_v}, \mathbf{r}_{\xi,\mathtt{m}_v}) = x\mathtt{C}_v + \mathtt{C}_{\mathtt{m}_v}, \texttt{Com}_{\mathbf{A}}(\xi_{\mathtt{m}_w}, \mathbf{r}_{\xi,\mathtt{m}_w}) = x\mathtt{C}_w + \mathtt{C}_{\mathtt{m}_w}$, $\texttt{Com}_{\mathbf{A}}(\mathbf{0}, \tilde{\mathbf{r}}_{\mathbf{f}}) = \mathbf{A}\tilde{\mathbf{r}}_{\mathbf{f}}$, $\texttt{Com}_{\mathbf{A}}(\mathbf{0}, \tilde{\mathbf{r}}_{\mathbf{q}}) = \mathbf{A}\tilde{\mathbf{r}}_{\mathbf{q}}$. She checks $\max\{\|\mathbf{r}_{\xi,\mathtt{m}_v}\|, \|\mathbf{r}_{\xi,\mathtt{m}_w}\|\} \leq 12\mathcal{B}\sqrt{m'n}$ and:

$$\sum_{j=0}^{n'-1} x^j \widetilde{\mathbb{A}}_j + x^{n'}\mathbb{U} = \widetilde{\mathtt{C}}_{\tilde{\mathbf{f}}} = \texttt{Com}_{\mathbf{A}}\big(\tilde{\mathbf{f}}(x), \tilde{\mathbf{r}}_{\mathbf{f}}\big), \quad \sum_{j'=0}^{\nu-1} x^{j'} \widetilde{\mathbb{T}}_{j'} + x^{\nu}\mathbb{S} = \widetilde{\mathtt{C}}_{\tilde{\mathbf{q}}} = \texttt{Com}_{\mathbf{A}}\big(\tilde{\mathbf{q}}(x), \tilde{\mathbf{r}}_{\mathbf{q}}\big),$$

$$\sum_{i=0}^{m'-1} \mathtt{C}_i(\tilde{\mathbf{f}}, \tilde{\mathbf{r}}_{\mathbf{f}}) \cdot x^{n'i} = \sum_{i'=0}^{\mu-1} \mathtt{C}_{i'}(\tilde{\mathbf{q}}, \tilde{\mathbf{r}}_{\mathbf{q}}) \cdot x^{\nu i'} \cdot (x - \mathtt{C}_v) + \mathtt{C}_w - \texttt{Com}_{\mathbf{A}}(0, \mathbf{r}_\gamma).$$

where $\mathtt{C}_i(\cdot)$ and $\mathtt{C}_{i'}(\cdot)$ denote the i-th or i'-th component of the corresponding commitment vector $\texttt{Com}_{\mathbf{A}}(\tilde{\mathbf{f}}, \tilde{\mathbf{r}}_{\mathbf{f}})$ and $\texttt{Com}_{\mathbf{A}}(\tilde{\mathbf{q}}, \tilde{\mathbf{r}}_{\mathbf{q}})$, respectively. Finally, she checks: $\|\tilde{\mathbf{r}}_{\mathbf{f}}\| \leq \mathcal{B}m'\sqrt{n'm}$, $\|\tilde{\mathbf{r}}_{\mathbf{q}}\| \leq \mathcal{B}\mu\sqrt{\nu m}$.

Theorem 3.6. *The polynomial commitment protocol has completeness, special honest verifier zero-knowledge and* 3*-special soundness for extracting a breach of the binding property of the commitment scheme, or extracting openings to the polynomials. The binding property relies on the underlying* $\mathtt{M-SIS}_{q,2,m,\beta}$ *assumption, while zero-knowledgeness is guaranteed by the hiding property of our commitment scheme which is given by the hardness of* $\mathtt{M-LWE}_{q,2,m}$ *assumption.*

Proof. Appendix C.1

3.3 Efficiency Analysis.

We analyze the efficiency of our protocol Π_{PEv}. The outputs are commitments to the rows $\{\mathbb{A}_j\}_j$ of the masked matrix $\mathcal{A}$ and commitments $\{\mathbb{T}_{j'}\}_{j'}$ of the masked matrix $\mathcal{T}$, commitments $\widetilde{\mathtt{C}}_{\tilde{\mathbf{f}}}, \widetilde{\mathtt{C}}_{\tilde{\mathbf{q}}}$, commitments to zero $\texttt{Com}_{\mathbf{A}}(0, \tilde{\mathbf{r}}_{\mathbf{f}}), \texttt{Com}_{\mathbf{A}}(0, \tilde{\mathbf{r}}_{\mathbf{q}})$ as well as the commitments $\mathtt{C}_{\xi,\mathtt{m}_v}, \mathtt{C}_{\xi,\mathtt{m}_v}$. The size of these commitments is given as follows: for all $j \in [0, n'-1], j' \in [0, \nu - 1]$ we have $\|\mathbb{A}_j\| \leq \sqrt{n}\|\mathbb{A}_j\|_\infty \leq \sqrt{n}\log(q)$, $\|\mathbb{T}_{j'}\| \leq \sqrt{n}\|\mathbb{T}_{j'}\|_\infty \leq \sqrt{n}\log(q)$. We get same sizes for the other sent commitments, such that the total cost yields $(2n'+9)\sqrt{n}\log(q)$. Further outputs are the 2 values $\xi_{\mathtt{m}_v}, \xi_{\mathtt{m}_w}$ of size $\|\xi_{\mathtt{m}_\iota}\| \leq 12\mathcal{B}\sqrt{n}$ for $\iota \in \{v, w\}$, i.e. total bit length $2\log(12\mathcal{B}\sqrt{n})$. Finally the 3 randomness vectors $\mathbf{r}_{\xi,\mathtt{m}_v}, \mathbf{r}_{\xi,\mathtt{m}_w}, \mathbf{r}_\gamma$ of length $\|\mathbf{r}_{\xi,\mathtt{m}_\iota}\| \leq 12\mathcal{B}\sqrt{mn}$ for $\iota \in \{v, w\}$. The total communication cost is $(2n'+9)\sqrt{n}\log q + 2 \cdot \log(12\mathcal{B}\sqrt{n}) + 3\log(12\mathcal{B}\sqrt{mn})$.

Concrete Parameters. For the instantiation of concrete parameters, we use the results from [19], where the maximum of the columns of public parameter $\mathbf{A}$ is bounded by $\sqrt{n\log q/\log\delta}$, where $\delta = 1.0035$. We balance this security level for LWE using LWE estimator [1] and get the number of integer elements $\hat{m} = \mathcal{O}(n)$ over $\mathbb{Z}$. The length of this short vector is bounded by the following condition: $\beta \geq \min\left(q, 2^{2\sqrt{n\log(q)\log\delta}}\right)$. We know that $\|\mathbf{r}_{a,j}\| \leq \beta$ and $\beta \leq 12\mathcal{B}\sqrt{mn}$. The length of the extracted witness is bounded by $\|(x^0 \cdot \mathbf{r}_0, \ldots x^{n'} \cdot \mathbf{r}_{n'})\| \leq 6 \cdot 12\mathcal{B}\sqrt{mn} = \hat{\beta}$. The length of the non-zero vector in SIS is bounded by $\hat{\beta} > \min\{q, 2^{2\sqrt{n\log q\log\delta}}\}$. The condition on m needs to satisfy $m = \sqrt{n\log q/\log\delta}$. We assume that $m' = n' = \mathcal{O}(\sqrt{n})$. Then the following estimation holds $\sigma \geq 12\mathcal{B}\sqrt{n'nm}$ is the standard deviations from rejection sampling. The probability from rejection sampling is $(1 - \tilde{\rho})$, where $\tilde{\rho} = \max_{i \in \{1,2,3\}}(\rho_i)$. It holds $\|\tilde{\mathbf{r}}_{\mathbf{f}}\|^2 = (\mathcal{B}n\sqrt{mn'})^2 = \mathcal{B}^2 n^3 m$, $\|\tilde{\mathbf{r}}_{\mathbf{q}}\|^2 = (\mathcal{B}n\sqrt{m\nu})^2 \approx \mathcal{B}^2 n^2 \nu m$. We pick the parameters so that $\mathcal{B} < \min\left(q, 2^{2\sqrt{n\log(q)\log\delta}}\right)$. In Table 2 we provide four parameter sets. To minimize the soundness error we repeat the protocol 80 times.

Parameter	Set 1	Set 2	Set 3	Set 4
Commitment modulus q	2^{18}	2^{22}	2^{25}	2^{27}
Ring dimension n	128	256	512	1024
$\hat{m}$	768	768	1024	564
$\log(\hat{\beta})$	17.19	18.19	17.67	18.99
$\mathcal{B}$	65	65	27	40
Proof size	64.64 kB	144 kB	307.14 kB	607.61 kB

Table 2. Sample parameters for Π_{PEv} for $\lambda = 128$

4 Batch Polynomial Evaluation

The protocol we present here allows to batch multiple instances into one polynomial and to commit to it. Our construction is based on the polynomial evaluation protocol we introduced in Section 3. In this section we discuss how to build such a batch polynomial evaluation zero-knowledge argument from lattices.

4.1 Preliminaries of the Protocol

The prover's witness is given by a vector $\mathbf{a}$, which satisfies some conditions and by an opening of the commitment $\mathtt{C}$ that is a commitment to the vector $\mathbf{a}$. To model these relations, we use a polynomial $\mathbf{P}$, which describes conditions on $\mathbf{a}$, and a polynomial $\mathbf{Q}$ which computes the opening of $\mathtt{Com}$. We commit to $\mathbf{Q}$ with randomness $\mathbf{r}$: $\mathbf{P}(\mathbf{a}) = 0, \quad \mathtt{C} = \mathtt{Com}(\mathbf{Q}(\mathbf{a}); \mathbf{r})$. $\mathtt{C}$ is a commitment to the polynomial $\mathbf{Q}$ as introduced in Section 3. We also introduce a public vector $\mathbf{b}$ in our polynomial $\mathbf{Q}$. We assume that $\mathbf{P}(\mathbf{a}), \mathbf{Q}(\mathbf{a}, \mathbf{b})$ are vectors of length l_P, l_Q, i.e. $\mathbf{P}(\mathbf{a}) = \big(\mathbf{P}_1(\mathbf{a}), \ldots, \mathbf{P}_{l_P}(\mathbf{a})\big), \mathbf{Q}(\mathbf{a}) = \big(\mathbf{Q}_1(\mathbf{a}, \mathbf{b}), \ldots, \mathbf{Q}_{l_Q}(\mathbf{a}, \mathbf{b})\big)$ whose components $\mathbf{P}_i(\mathbf{a}), \mathbf{Q}_j(\mathbf{a}, \mathbf{b})$, for $i' \in [1, l_P], j' \in [1, l_Q]$ are $(l_a)-$ or $(l_a + l_b)-$variate polynomials, respectively. We use bold font to indicate vectors, while each of the vector components is a multivariate polynomial and is given in plain font. Let $\mathbf{P}$ and $\mathbf{Q}$ be vectors of polynomials of degree d_P and d_Q, respectively. Let $n = m'n' - 1$ be the degree of polynomial $\mathbf{P}$. We provide an argument of knowledge of $\{\mathbf{a}_{j,i}\}_{i \in [0,m'-1], j \in [0,n'-1]} \in \mathbb{Z}_q^{l_a}$ and $\mathbf{r}_j \in \mathcal{R}_q^m$, such that $\mathbf{P}(\mathbf{a}_{j,i}) = 0$. We know that each $\mathbf{a}_{j,i}$ is a vector of l_a monomial coefficients, i.e. $\mathbf{a}_{j,i} = (a_{j,i,1}, \ldots, a_{j,i,l_a})$. We build a matrix for each monomial coefficient $a_\kappa, \kappa \in [l_a]$:

$$A_\kappa = \begin{bmatrix} a_{0,0,\kappa} & a_{0,1,\kappa} & \cdots & a_{0,m'-1,\kappa} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n'-1,0,\kappa} & a_{n'-1,1,\kappa} & \cdots & a_{n'-1,m'-1,\kappa} \end{bmatrix}.$$

The matrix is an encoding of the following polynomial of degree $m'n' - 1$ in X_κ:

$$f(X_\kappa) = \sum_{i=0}^{m'-1} \sum_{j=0}^{n'-1} a_{j,i,\kappa} X_\kappa^{in'+j}. \tag{11}$$

We mask the matrix by adding an additional row $a_{\kappa,u} = (a_{\kappa,u,1}, \ldots, a_{\kappa,u,m'-1})$ at the end of the matrix A_κ, where each masking component $a_{\kappa,u,i}, i \in [1, m'-1]$ is defined as in the previous protocol in Section 3. We obtain a masked matrix $\widetilde{A}_\kappa$ having $n'+1$ rows with indices from 0 to n'. We denote each component of the masked matrix as $\tilde{a}_{\kappa,i,j}$, where $i \in [0, m'-1]$ and $j \in [0, n'-1]$. Similarly we represent $\mathbf{b}_{j,i} \in \mathbb{Z}_q^{l_b}$, i.e. $\mathbf{b}_{j,i} = (b_{j,i,1}, \ldots, b_{j,i,l_b})$. The matrix $B_{\kappa'}$ for each monomial coefficient $b_{\kappa'}$, $\kappa' \in [l_b]$ is constructed in the same manner as above. This matrix is an encoding of the following polynomial of degree $m'n'$ in $X_{\kappa'}$:

$$f(X_{\kappa'}) = \sum_{i=0}^{m'-1} \sum_{j=0}^{n'-1} b_{\kappa',j,i} X_{\kappa'}^{in'+j}. \tag{12}$$

We take each element from row $i \in [0, m'-1]$ and column $j \in [0, n'-1]$ from matrix $\widetilde{A}_\kappa$, $\kappa \in [l_a]$ and matrix $\widetilde{B}_{\kappa'}, \kappa' \in [l_b]$ and pack them into the vectors $\tilde{\mathbf{a}}_{j,i} = (\tilde{a}_{j,i,1}, \ldots, \tilde{a}_{j,i,l_a})$ and $\mathbf{b}_{j,i} = (b_{j,i,1}, \ldots, b_{j,i,l_b})$, which are vectors of j, i-th coefficients of l_a or l_b-variate polynomials, respectively.

4.2 Detailed Protocol

In this section we provide a concrete description of our batch polynomial protocol over lattices. We first define the relations which will be proved by our protocol.

Definition 4.1. *For positive real bounds β, β' we have the following two relations to be proved:*

$$\begin{aligned} \mathsf{R}_{\texttt{batch},\beta} = \big\{ &\forall j \in [0, n'-1], i \in [0, m'-1] : \mathcal{P}\big(\{\tilde{\mathbf{a}}_{j,i}\}, \{\mathbf{r}_{a,j}\}\big), \mathcal{V}\big(\{\texttt{C}_i\}, \{\mathbf{b}_{j,i}\}, \mathbf{P}, \mathbf{Q}\big) : \\ &\mathbf{P}(\tilde{\mathbf{a}}_{j,i}) = 0 \wedge \texttt{C}_i = \texttt{Com}_{\mathbf{A}}\big(\mathbf{Q}(\tilde{\mathbf{a}}_{j,0}, \mathbf{b}_{j,0}), \ldots, \mathbf{Q}(\tilde{\mathbf{a}}_{j,m'-1}, \mathbf{b}_{j,m'-1}); \mathbf{r}_{a,j}\big) \wedge \|\mathbf{r}_{a,j}\| \leq \beta \big\} \\ \mathsf{R}'_{\texttt{batch},\beta'} = \big\{ &\forall j \in [0, n'-1], i \in [0, m'-1] : \mathcal{P}\big(\{6\tilde{\mathbf{a}}_{j,i}\}, \{\mathbf{r}'_{a,j}\}\big), \mathcal{V}\big(\{\texttt{C}_i\}, \{\mathbf{b}_{j,i}\}, \mathbf{P}, \mathbf{Q}\big) : \\ &\mathbf{P}(6\tilde{\mathbf{a}}_{j,i}) = 0 \wedge \texttt{C}_i = \texttt{Com}_{\mathbf{A}}\big(\mathbf{Q}(6\tilde{\mathbf{a}}_{j,0}, \mathbf{b}_{j,0}), \ldots, \mathbf{Q}(6\tilde{\mathbf{a}}_{j,m'-1}, \mathbf{b}_{j,m'-1}); \mathbf{r}'_{a,j}\big) \wedge \|\mathbf{r}'_{a,j}\| \leq \beta' \big\} \end{aligned}$$

Remark 4.2. The commitments in this protocol rely on the security of $Module-SIS_{\tilde{n},m,\beta}$, where $\tilde{n}$ denotes the number of rows of the public parameter $\mathbf{A} \in \mathcal{R}_q^{\tilde{n} \times m}$. Note that for the computation of commitments $\texttt{C}_i$ and $\texttt{D}_i$ we have $\tilde{n} = l_Q$ where $j \in [0, n'-1]$ and l_Q, l_a are defined in the protocol below.

Construction.

Public input: The prover computes a function on the vectors $\tilde{\mathbf{a}}_{j,i}, \mathbf{b}_{j,i}$ yielding a vector $\mathbf{Q}(\tilde{\mathbf{a}}_{j,i}, \mathbf{b}_{j,i})$ of multivariate polynomial of degree l_Q.

$$\mathbf{Q}(\tilde{\mathbf{a}}_{j,i}, \mathbf{b}_{j,i}) = \big(Q_1(\tilde{\mathbf{a}}_{j,i}, \mathbf{b}_{j,i}), \ldots, Q_{l_Q}(\tilde{\mathbf{a}}_{j,i}, \mathbf{b}_{j,i})\big)^{tr}. \tag{13}$$

The prover computes n' commitments knowing the openings to a vector of m' multivariate polynomial vectors $\{\mathbf{Q}(\tilde{\mathbf{a}}_{j,i}, \mathbf{b}_{j,i})\}_{j \in [0,n'-1], i \in [0,m'-1]}$, where each of those m' vectors are defined in (13). Note, that each of these polynomial vectors contains l_Q different $(l_a + l_b)$-variate polynomials in $l_a + l_b$ variables, i.e. for each $i \in [0, m'-1], j \in [0, n'-1]$ and for each $\iota \in [l_Q]$ we have a vector of l_Q multivariate polynomials in $l_a + l_b$ variables of the following form: $Q_\iota(\tilde{\mathbf{a}}_{j,i}, \mathbf{b}_{j,i}) = \sum_{k=0}^{d_Q} K_{\iota,k} \Big(\prod_{\substack{\{d_i\}_{i\in[l_a+l_b]} \\ \sum d_i \leq k}} X_1^{d_1} \cdots X_{l_a+l_b}^{d_{l_a+l_b}} \Big) = Q_{\iota,j,i}(X_1, \ldots, X_{l_a+l_b})$. where $K_{\iota,k}$ is the k-th coefficient in the sum which is equal to a combination of $a_{\iota,j,i}$ from (11) and $b_{\iota,j,i}$ from (12). In order to commit to a m'-dimensional vector of multivariate polynomials, we first redefine each multivariate polynomial into regular polynomials as follows: Let K_{ι,k,d_i} denote the d_ι-th coefficient of $X_k^{d_i}$, i.e. K_{ι,k,d_i} depends on the values $\{X_k\}_{k\in[l_a+l_b],k\neq d_i}$. For $k \in [l_a + l_b]$ and $\iota \in [l_Q]$ we have: $Q_{\iota,j,i}(X_k) = \sum_{d_i=0}^{d_Q} K_{\iota,k,d_i}(x_1, \ldots x_{l_a+l_b}) X_k^{d_i}$. We fix $j \in [0, n'-1]$ and $\iota \in [l_Q]$, and define a vector over the indices $i \in [0, m'-1]$ as $\mathbf{Q}_{\iota,j}(X_k) = \big(Q_{\iota,j,0}(X_k), \ldots, Q_{\iota,j,m'-1}(X_k)\big)^{tr}$ being a vector of m' polynomials in X_k for $\iota \in [l_Q]$. Each of these polynomials is of degree d_Q. The public key is $\mathbf{A} \in \mathcal{R}_q^{l_Q \times m}$ and the randomnesses $\mathbf{r}_j \in \mathcal{R}_q^{m\times(l_a+l_b)}$. Let $n = m'n'-1$. We commit to a matrix in $\mathcal{R}_q^{l_Q\times(l_a+l_b)}$ which has the following form:

$$\mathbf{Q}_j := \begin{pmatrix} \mathbf{Q}_{1,j}(X_1) & \ldots & \mathbf{Q}_{1,j}(X_{l_a+l_b}) \\ \vdots & \ddots & \vdots \\ \mathbf{Q}_{l_Q,j}(X_1) & \ldots & \mathbf{Q}_{l_Q,j}(X_{l_a+l_b}) \end{pmatrix}$$

Its commitment is defined as: $\mathtt{C}_j = \mathtt{Com}_{\mathbf{A}}\big(\mathbf{Q}(\tilde{\mathbf{a}}_{j,0}, \mathbf{b}_{j,0}), \ldots, \mathbf{Q}(\tilde{\mathbf{a}}_{j,m'-1}, \mathbf{b}_{j,m'-1}); \mathbf{r}_j\big)$ $= \mathtt{Com}_{\mathbf{A}}(\mathbf{Q}_j, \mathbf{r}_j)$. Public inputs to the protocol are $\{\mathtt{C}_j\}_j, \mathbf{Q}, \mathbf{P}$ and the vectors $\mathbf{b}_{j,i}$. For better understanding of how we commit to m' vectors of multivariate polynomials of degree d_Q, see Example in Appendix B.

Common Reference String Generation: The batch protocol embeds multiple instances of the same polynomial equality into a single polynomial by using Lagrange interpolation technique. In order to recover a single instance, we simply evaluate the polynomial in one of the interpolation points. Let $y_1, \ldots, y_{n'}$ be distinct points in $\mathcal{CH}$ (monomials) and $l_1(X), \ldots, l_{n'}(X)$ be their associated Lagrange polynomials, s.t. $l_j(y_i) = \delta_{j,i}$ and $l_0(X) = \prod_{i=1}^{n'}(X - y_i)$. For $j \in [0, n'-1]$ we have: $l_j(X) = \prod_{\substack{0\leq k\leq n-1' \\ n'-1'\neq k}} \frac{X-y_k}{y_j-y_k}$.

Public Input: $\mathbf{P}, \mathbf{Q}, \{\mathbf{b}_{j,i}\}_{i\in[0,m'-1],j\in[0,n'-1]}$, $\{\mathtt{C}_j\}_{j\in[1,n'-1]}$.

Prover's witness: $\{\tilde{\mathbf{a}}_{j,i}\}$ and $\{\mathbf{r}_j\}$ for $j \in [0, n'-1], i \in [0, m'-1]$.

Initial message: The prover picks random values $\mathbf{s}_j \in \mathcal{R}_q^m$ and computes the commitments $\mathtt{D}_j$ on $\{\tilde{\mathbf{a}}_{j,i}\}$, where $0 \leq j \leq n'-1$ and $0 \leq i \leq m'-1$ and $\tilde{\mathbf{a}}_j = (\tilde{\mathbf{a}}_{j,0}, \ldots, \tilde{\mathbf{a}}_{j,m'-1})$, with $\mathbf{A} \in \mathcal{R}_q^{l_Q\times m}$ as follows: $\mathtt{D}_j := \mathtt{Com}_{\mathbf{A}}\big(\tilde{\mathbf{a}}_{j,0}, \ldots, \tilde{\mathbf{a}}_{j,m'}; \mathbf{s}_j\big) = \mathbf{A} \cdot \mathbf{s}_j + \tilde{\mathbf{a}}_j$. For $j = n'$ we sample a vector $(\tilde{\mathbf{a}}_{n',0}, \ldots, \tilde{\mathbf{a}}_{n',m'-1}) \leftarrow_{\$} \mathfrak{D}_{12\mathcal{B}\sqrt{l_Q m' n'}}^{l_Q m' n'}$, which is a blinding vector chosen uniformly at random in the preliminary step, i.e. $\tilde{\mathbf{a}}_{n',i} = (b_{1,u,i}, \ldots, b_{l_a,u,i})$ is the additional masking row on A_κ. The prover computes $\hat{\mathbf{a}}_i = \sum_{j=0}^{n'} \tilde{\mathbf{a}}_{j,i} l_j(X)$ and $\hat{\mathbf{b}}_i = \sum_{j=0}^{n'} \mathbf{b}_{j,i} l_j(X)$. She has to show that $\tilde{\mathbf{a}}_{j,i}, \mathbf{b}_{j,i}$ satisfy polynomial relations presented above. When evaluating these vectors at a point y_i we get $\sum_{j=0}^{n'} \tilde{\mathbf{a}}_{j,i} l_j(y_i) = \tilde{\mathbf{a}}_{i,i}$ and $\sum_{j=0}^{n'} \mathbf{b}_{j,i} l_j(y_i) = \mathbf{b}_{i,i}$, respectively. This yields the single evaluation of $\mathbf{P}(\tilde{\mathbf{a}}_{j,i})$. It holds that: $\mathbf{P}_i^*(X) \cdot \prod_{j=0}^{n'}(X - y_j) = \mathbf{P}(\hat{\mathbf{a}}_i) = \mathbf{P}\big(\sum_{j=0}^{n'} \tilde{\mathbf{a}}_{j,i} l_j(X)\big)$, where evaluating this equation at y_i yields $\mathbf{P}(\tilde{\mathbf{a}}_{j,j}) = 0$. Thus, the prover computes the polynomial

$$\mathbf{P}(\hat{\mathbf{a}}_j)/l_0(X) = \frac{\mathbf{P}\big(\sum_{j=0}^{n'} \tilde{\mathbf{a}}_{j,i} l_j(X)\big)}{\prod_{j=0}^{n'}(X - y_j)} = \mathbf{P}_i^*(X)$$

of degree $(d_P - 1)n'$ and commits to its coefficients. This can be achieved using polynomial commitment scheme from Section 3 but generalized to the case of vector coefficients. We take the polynomials $\{\mathbf{P}_i^*(X)\}_i$ and encode them into matrices $\mathcal{P}_i^*$ of dimension $t_1 \times t_2$, i.e. $d_P n' - n' = t_1 t_2$ as in Sec. 3. To do so, we first write $\mathbf{P}_i^*(X) = \sum_{\mu=0}^{t_2} \sum_{\nu=0}^{t_1} p_{\nu,\mu}^{(i)} X^{\mu d_p + \nu}$ and the corresponding matrix of coefficients: $\mathcal{P}_i^* = \big(p_{\mu,\nu}^{(i)}\big)_{\mu,\nu}$, for all $i \in [0, m'-1]$ and $\mu \in [0, t_2], \nu \in [0, t_1]$. Since $\mathbf{P}_i^*$ is a vector of l_P polynomials in the variable X, i.e. $\mathbf{P}_i^* = (\mathbf{P}_{i,1}^*(X), \dots, \mathbf{P}_{i,l_P}^*(X))$, each $p_{\nu,\mu}^{(i)}$ denotes a row-vector: $p_{\nu,\mu}^{(i)} = \big(\mathsf{Coef}(\mathbf{P}_{i,1}^*(X)), \dots \mathsf{Coef}(\mathbf{P}_{i,l_P}^*(X)\big)$. Applying the same masking technique as in Section 3 and adding an additional row t_2+1 with the masking vectors $(\mathbf{u}_{\mathbf{P}^*,1}, \dots, \mathbf{u}_{\mathbf{P}^*,t_1}, 0)$, where each of these vectors is sampled from $\mathfrak{D}_\sigma^{l_p}$, we obtain a matrix $\widetilde{\mathcal{P}}_i^* = \big(\tilde{p}_{\mu,\nu}^{(i)}\big)_{\mu,\nu} \in \mathbb{Z}_q^{(t_2+2)\times(t_1+1)}$ with masked coefficients $\tilde{p}_{\mu,\nu}$. It holds:

$$\mathbf{P}_i^*(X) = \big(1, X, \cdots, X^{t_2}, X^{t_2+1}\big) \cdot \widetilde{\mathcal{P}}_i^* \cdot \big(1, X, \dots, X^{t_1}\big)^{tr}, \; i \in [0, m'-1]. \quad (14)$$

According to the polynomial commitment scheme, the prover commits to each row of the matrix $\widetilde{\mathcal{P}}_i^*$ using randomness $\mathbf{r}_{p_\mu}^{(i)} \in \mathcal{R}_q^m$, for $\mu \in [0, t_2]$ and padding with zeros until n. The commitments to the rows are $\mathbb{A}_{\mathbf{P}_i^*,\mu} = \mathtt{Com}_{\mathbf{A}}\big(\mathbf{p}_\mu^{(i)}, \mathbf{r}_{p_\mu}^{(i)}\big) = \mathbf{A} \cdot \mathbf{r}_{p_\mu}^{(i)} + \mathsf{enc}(\mathbf{p}_\mu^{(i)})$, w $\mathbf{p}_\mu^{(i)} = \big(\tilde{p}_{\mu,0}^{(i)}, \dots, \tilde{p}_{\mu,t_1}^{(i)}\big)$. The prover has also to prove that $\mathtt{C}_1, \dots, \mathtt{C}_{n'}$ are commitments to $\mathbf{Q}(\tilde{\mathbf{a}}_{j,i}, \mathbf{b}_{j,i})$. To do so she picks random vectors $\mathbf{c}_0, \dots, \mathbf{c}_{m'} \in \mathcal{R}_q^{l_Q}$ where each of the vectors is encoded into polynomials in $\mathcal{R}_q$ as follows: Let $\mathbf{c}_i = (c_{i,1}, \dots, c_{i,l_Q}) \in \mathbb{Z}_q^{l_Q}$ for all $i \in [0, m'-1]$. For $\hat{\mathbf{c}} := (\mathbf{c}_1, \dots, \mathbf{c}_{m'})$ compute $\mathtt{C}_0 = \mathtt{Com}_{\mathbf{A}}(\hat{\mathbf{c}}; \mathbf{r}_c) = \mathbf{A} \cdot \mathbf{r}_c + \mathsf{enc}(\hat{\mathbf{c}})$, while $\mathbf{A} \in \mathcal{R}_q^{l_Q \times m}, \mathbf{r}_c \in \mathcal{R}_q^{m \times (l_a + l_b)}$. We obtain the following polynomial

$$\mathbf{Q}_i^*(X) = \mathbf{c}_i + \frac{\sum_{j=0}^{n'-1} \mathbf{Q}(\tilde{\mathbf{a}}_{j,i}, \mathbf{b}_{j,i}) l_j(X) - \mathbf{Q}(\hat{\mathbf{a}}_i, \hat{\mathbf{b}}_i)}{l_0(X)},$$

which is a vector of univariate polynomials in X. The prover computes a polynomial commitment of $\mathbf{Q}_i^*$ for all $i \in [0, m'-1], \tilde{\mu} \in [0, \tau_2], \tilde{\nu} \in [0, \tau_1]$ as follows. Let $\mathbf{Q}_i^*(X) = \sum_{\tilde{\mu}=0}^{\tau_2} \sum_{\tilde{\nu}=0}^{\tau_1} q_{\tilde{\nu},\tilde{\mu}}^{(i)} X^{\tilde{\mu} d_{Q^*} + \tilde{\nu}}$ with the corresponding coefficient matrix $\mathcal{Q}_i^* = \big(q_{\tilde{\mu},\tilde{\nu}}^{(i)}\big)_{\tilde{\mu},\tilde{\nu}}$. Since $\mathbf{Q}_i^*$ is a vector of l_Q polynomials X, i.e. $\mathbf{Q}_i^* = (\mathbf{Q}_{i,1}^*(X), \dots, \mathbf{Q}_{i,l_Q}^*(X))$, and $q_{\tilde{\nu},\tilde{\mu}}^{(i)} = \big(\mathsf{Coef}(\mathbf{Q}_{i,1}^*(X)), \dots, \mathsf{Coef}(\mathbf{Q}_{i,l_Q}^*(X)\big)$. The masked matrix (computed as in Section 3) is $\widetilde{\mathcal{Q}}_i^* = \big(\tilde{q}_{\tilde{\mu},\tilde{\nu}}^{(i)}\big)_{\tilde{\mu},\tilde{\nu}}$ and has $\tau_2 + 2$ rows enumerated from 0 to $\tau_2 + 1$, where the last row is a masking vector, randomly sampled from $\mathfrak{D}^{\tau_1}_{12\beta\sqrt{\tau_1+1}}$ which is padded with 0. According to the polynomial commitment, the prover commits to each row of $\widetilde{\mathcal{Q}}^*_i$ using randomness $\mathbf{r}^{(i)}_{q_{\tilde{\mu}}} \in \mathcal{R}^m_q$, for $\tilde{\mu} \in [0, \tau_2 + 1]$ and padding with zeros until reaching the desired length n. The commitments are given as $\mathbb{A}_{\mathbf{Q}^*_i,\tilde{\mu}} = \mathtt{Com}_{\mathbf{A}}\big(\mathbf{q}_{\tilde{\mu}}, \mathbf{r}^{(i)}_{q_{\tilde{\mu}}}\big) = \mathbf{A} \cdot \mathbf{r}^{(i)}_{q_{\tilde{\mu}}} + \mathsf{enc}(\mathbf{q}^{(i)}_{\tilde{\mu}})$, where $\mathbf{q}^{(i)}_{\tilde{\mu}} = \big(q^{(i)}_{\tilde{\mu},0}, \ldots, q^{(i)}_{\tilde{\mu},\tau_1}\big)$.

The prover runs $\mathit{\Pi}_{PEv}$ from Sec. 3 on input $\mathbf{P}^*_i, \mathbf{Q}^*_i, u, \mathbf{P}^*_i(u), \mathbf{Q}^*_i(u)$. Let $\chi \in \{\mathbf{P}^*, \mathbf{Q}^*\}, \chi_i \in \{\mathbf{P}^*_i, \mathbf{Q}^*_i\}$ and $i \in [0, m' - 1]$. She outputs $\{(\mathbf{msg}_{\chi,1}, \mathbf{st}_{\chi^*})\}_\chi$ s.t. $\mathbf{msg}_{\chi,1} = \{\mathbf{msg}_{\chi_i,1}\}_i$ and $\mathbf{msg}_{\chi_i,1} = \big(\mathbb{A}_{\chi_i,\mu}, \mathbb{T}_{\chi_i,\mu}, \mathbb{U}_{\chi_i}, \mathbb{S}_{\chi_i}, \mathtt{C}_{\chi_i,v}, \mathtt{C}_{\chi_i,w}\big)$. For all $\mu \in [0, t_2 - 1], \tilde{\mu} \in [0, \tau_2 - 1]$ we set $\mathsf{st}_{\mathbf{P}^*_i} = \big(\mathbf{P}^*_i, \{\mathbf{r}^{(i)}_{p_\mu}\}_\mu\big), \mathsf{st}_{\mathbf{Q}^*_i} = \big(\mathbf{Q}^*_i, \{\mathbf{r}^{(i)}_{q_{\tilde{\mu}}}\}_{\tilde{\mu}}\big)$. Let $\mathsf{st}_\chi = \{\mathsf{st}_{\chi_i}\}_i$. The prover sends $\{\mathtt{D}_j\}_{j\in[0,n']}, \{\mathbf{msg}_{\chi,1}\}_\chi$ to the verifier.

Challenge: Verifier sends the challenge $x \in \mathcal{CH}$.

Response: The prover runs **Response** of $\mathit{\Pi}_{PEv}$ from Section 3 on input $\{\mathsf{st}_{\chi^*_i}\}_{\chi_i}$ as follows. It holds $\mathbf{P}^*_i = (x - \mathbf{y})\mathbf{H}_{\mathbf{P}^*_i} + \mathbf{z}$, where $\mathbf{P}^*_i(\mathbf{y}) = \mathbf{z}$ and $\mathbf{y}$ is a vector in which the polynomial to be evaluated and $\mathbf{z}$ being the evaluation vector of $\mathbf{P}^*_i$ in $\mathbf{y}$. According to the polynomial evaluation protocol the polynomial $\mathbf{H}_{\mathbf{P}^*_i}$ can be represented as a coefficient matrix $\mathcal{H}_{\mathbf{P}^*_i} = (h_{j',i'})_{j'\in[0,t'_2],i'\in[0,t'_1]}$. The corresponding masked matrix of $\mathbf{H}_{\mathbf{P}^*_i}$ is $\widetilde{\mathcal{H}}_{\mathbf{P}^*_i} = (\tilde{h}_{j',i'})_{j'\in[0,t'_2+1],i'\in[0,t'_1]}$ She computes $\tilde{\mathbf{f}}_{\mathbf{P}^*_i}(x) = (1, x, \ldots, x^{t_2}, x^{t_2+1}) \cdot \widetilde{\mathcal{P}}^*_i$ and $\tilde{\mathbf{h}}_{\mathbf{P}^*_i}(x) = (1, x, \ldots, x^{t'_2}, x^{t'_2+1}) \cdot \widetilde{\mathcal{H}}_{\mathbf{P}^*_i}$ with corresponding randomness $\tilde{\mathbf{r}}_{\mathbf{f},\mathbf{P}^*_i}(x) = \sum_{\mu=0}^{t_2} \mathbf{r}^{(i)}_{p_\mu} x^\mu$ and $\tilde{\mathbf{r}}_{\mathbf{h},\mathbf{P}^*_i}(x) = \sum_{\mu=0}^{t_2} \mathbf{r}^{(i)}_{h_\mu} x^\mu$ and the commitments:

$$\mathtt{F}_{\mathbf{P}^*_i} := \mathtt{Com}_{\mathbf{A}}\big(\tilde{\mathbf{f}}_{\mathbf{P}^*_i}(x), \tilde{\mathbf{r}}_{\mathbf{f},\mathbf{P}^*_i}(x)\big), \quad \mathtt{C}_{\mathbf{P}^*_i,0,\tilde{\mathbf{f}}} := \mathtt{Com}_{\mathbf{A}}\big(\mathbf{0}, \tilde{\mathbf{r}}_{\mathbf{f},\mathbf{P}^*_{\mathbf{i}}}(\mathbf{x})\big).$$

$$\mathtt{G}_{\mathbf{P}^*_i} := \mathtt{Com}_{\mathbf{A}}\big(\tilde{\mathbf{h}}_{\mathbf{P}^*_i}(x), \tilde{\mathbf{r}}_{\mathbf{h},\mathbf{P}^*_i}(x)\big), \quad \mathtt{C}_{\mathbf{P}^*_i,0,\tilde{\mathbf{h}}} := \mathtt{Com}_{\mathbf{A}}(\mathbf{0}, \tilde{\mathbf{r}}_{\mathbf{h},\mathbf{P}^*_{\mathbf{i}}}(\mathbf{x})).$$

She computes $\tilde{\mathbf{f}}_{\mathbf{Q}^*_i}(x) = (1, x, \ldots, x^{t_2}, x^{t_2+1}) \cdot \widetilde{\mathcal{P}}^*_i$ with $\tilde{\mathbf{r}}^{(i)}_{\mathbf{q},\mathbf{Q}^*}(x) = \sum_{\tilde{\mu}=0}^{\tau_2} \mathbf{r}^{(i)}_{q_{\tilde{\mu}}} x^{\tilde{\mu}}$ and the commitments $\mathtt{F}^{(i)}_{\mathbf{Q}^*}, \mathtt{G}^{(i)}_{\mathbf{Q}^*}$ computed as above. $\forall i \in [0, m'-1], \chi_i \in \{\mathbf{P}^*_i, \mathbf{Q}^*_i\}$:

$$\mathbf{msg}_{\chi_i,2} := \big(\mathtt{F}_{\chi_i}, \mathtt{G}_{\chi_i}, \mathtt{C}_{\chi_i,0,\tilde{\mathbf{f}}}, \mathtt{C}_{\chi_i,0,\tilde{\mathbf{q}}}, \xi_{\mathtt{m}_v,\chi_i}, \xi_{\mathtt{m}_w,\chi_i}, \mathbf{r}_{\xi,\mathtt{m}_v,\chi_i}, \mathbf{r}_{\xi,\mathtt{m}_w,\chi_i}\big)$$

For all $\chi \in \{\mathbf{P}^*, \mathbf{Q}^*\}$ we set $\mathbf{msg}_{\chi,2} = \{\mathbf{msg}_{\chi_i,2}\}_i$. Additionally, the prover computes $\hat{\mathbf{a}}_i(x) = \sum_{j=0}^{n'} \tilde{\mathbf{a}}_{j,i} l_j(x)$, $\hat{\mathbf{r}}(x) = \sum_{j=0}^{n'} \mathbf{r}_j l_j(x)$, $\hat{\mathbf{s}}(x) = \sum_{j=0}^{n'} \mathbf{s}_j l_j(x)$ and sends $\{\hat{\mathbf{a}}_i\}_i$, $\hat{\mathbf{r}}, \hat{\mathbf{s}}$, $\{\mathbf{msg}_{\chi,2}\}_\chi$ to the verifier. She aborts with probability $\tilde{\rho} = \max\{\rho_1, \rho_2, \rho_3\}$ which depends on the rejection sampling of $\mathit{\Pi}_{PEv}$:

$$\rho_1 := \frac{\mathfrak{D}^{t_2mn}_{12\mathcal{B}\sqrt{t_2mn}}\big(\tilde{\mathbf{r}}_{\mathbf{f},\mathbf{P}^*_i}(x)\big)}{M\mathfrak{D}^{\tau_2mn}_{\{x^\mu\cdot\mathbf{r}_\mu\}_\mu),12\mathcal{B}\sqrt{t_2mn}}\big(\tilde{\mathbf{r}}_{\mathbf{f},\mathbf{P}^*_i}(x)\big)} \cdot \frac{\mathfrak{D}^{t_2mn}_{12\mathcal{B}\sqrt{t_2mn}}\big(\tilde{\mathbf{r}}_{\mathbf{q},\mathbf{P}^*_i}(x)\big)}{M\mathfrak{D}^{t_2mn}_{\{x^\mu\cdot\mathbf{r}_{t,\mu}\}_\mu,12\mathcal{B}\sqrt{t_2mn}}\big(\tilde{\mathbf{r}}_{\mathbf{q},\mathbf{P}^*_i}(x)\big)}$$

$$\rho_2 := \frac{\mathfrak{D}^{\tau_2mn}_{12\mathcal{B}\sqrt{\tau_2mn}}\big(\tilde{\mathbf{r}}_{\mathbf{f},\mathbf{Q}^*_i}(x)\big)}{M\mathfrak{D}^{\tau_2mn}_{\{x^{\tilde{\mu}}\cdot\mathbf{r}_{\tilde{\mu}}\}_{\tilde{\mu}}),12\mathcal{B}\sqrt{\tau_2mn}}\big(\tilde{\mathbf{r}}_{\mathbf{f},\mathbf{Q}^*_i}(x)\big)} \cdot \frac{\mathfrak{D}^{\tau_2mn}_{12\mathcal{B}\sqrt{\tau_2mn}}\big(\tilde{\mathbf{r}}_{\mathbf{q},\mathbf{Q}^*_i}(x)\big)}{M\mathfrak{D}^{\tau_2mn}_{\{x^j\cdot\mathbf{r}_{t,\tilde{\mu}}\}_{\tilde{\mu}},12\mathcal{B}\sqrt{\tau_2mn}}\big(\tilde{\mathbf{r}}_{\mathbf{q},\mathbf{Q}^*_i}(x)\big)}$$

and the rejection sampling of the batch protocol:

$$\rho_3 := \frac{\mathfrak{D}^{l_an'}_{12\mathcal{B}\sqrt{l_an'}}\big(\hat{\mathbf{r}}_i(x)\big)}{M\mathfrak{D}^{l_an'}_{\{l_j(x)\cdot\tilde{\mathbf{r}}_{j,i}\}_{i,j},12\mathcal{B}\sqrt{l_an'}}\big(\hat{\mathbf{r}}_i(x)\big)}$$

Verification: The verifier runs verification of the underlying polynomial evaluation protocol from Chapter 3 on input $\{\mathtt{D}_j\}_{j\in[0,n']}, \{\mathbf{msg}_{\chi,1}\}_\chi, \{\mathbf{msg}_{\chi,2}\}_\chi$ for all $\chi \in \{\mathbf{P}^*, \mathbf{Q}^*\}$. The verifier also computes a commitment to the m' vectors $\hat{\mathbf{a}}_i$ for $i \in [0, m'-1]$ and compares it with the sum over commitments $\mathtt{D}_j$ multiplied by the corresponding Lagrange function $l_j(x)$. The verifier checks each $\hat{\mathbf{a}}_i$ against the commitment $\mathtt{D}_j$ by exploiting its homomorphic property, i.e. $\hat{\mathbf{a}} = (\hat{\mathbf{a}}_0, \ldots, \hat{\mathbf{a}}_{m'-1})$ with the randomness $\hat{\mathbf{s}}$: $\mathtt{Com}_{\mathbf{A}}(\hat{\mathbf{a}}; \hat{\mathbf{s}}_i) = \mathbf{A}\cdot\hat{\mathbf{s}}_i + \mathsf{enc}(\hat{\mathbf{a}}) = \sum_{j=0}^{n'} \mathtt{D}_{j,i} \cdot l_j(X)$, where $\hat{\mathbf{s}}_i = \sum_{j=0}^{n'} \mathbf{s}_{j,i} l_j(X)$ for all $i \in [0, m'-1]$. After accepting the commitment opening, the verifier returns $\mathbf{P}_i^*$. Then, she computes: $\mathbf{P}_i^*(X) \cdot l_0(X) = \mathbf{P}(\hat{\mathbf{a}}_i)$. Additionally she checks the following equation: $\mathtt{Com}_{\mathbf{A}}\big(\{\mathbf{Q}_i^*(x) l_0(x) + \mathbf{Q}(\hat{\mathbf{a}}_i, \hat{\mathbf{b}}_i)\}_{i\in[0,m'-1]}, \hat{\mathbf{r}}\big) = \sum_{j=0}^{m} \mathtt{C}_j l_j(x)$.

Theorem 4.3. *The batch protocol given in Figure 3 has perfect completeness, perfect special honest-verifier zero-knowledge and* 3*-special soundness.*

Proof. Appendix C.2

4.3 Efficiency Analysis

For the efficiency analysis we consider the dimensions of the matrices which are used in the polynomial commitment subprotocol to $\mathbf{P}^*$ and $\mathbf{Q}^*$, which are t_1, t_2 and τ_1, τ_2, respectively. In the underlying protocols of $\mathbf{P}^*$ and $\mathbf{Q}^*$ we communicate $2t_2 + 9$ and $2\tau_2 + 9$ commitments, 4 vectors bounded by $12\mathcal{B}\sqrt{n}$ and 6 masked randomnesses bounded by $12\mathcal{B}\sqrt{3mn}$ respectively. The communication cost is given by the $2t_2 + 2\tau_2 + 18$ commitments and 10 responses from the underlying protocol as well as the n' commitments $\{\mathtt{D}_j\}_{j\in[0,n'-1]}$ and the three elements $\hat{\mathbf{a}}, \hat{\mathbf{r}}, \hat{\mathbf{s}}$ from the batch-protocol, i.e the total costs are $2\tau_2 + 2t_2 + n' + 28$ communicated elements. Since the degree of $\mathbf{P}_i^*$ is equal to $d_P n' - n'$ and because of the definition of $\mathbf{P}_i^*$ being a vector of l_P different l_a-variate polynomials, we set: $d_P n' - n' \approx l_P m'(t_2 + 1)$. Similarly, we set $d_Q n' - n' \approx l_Q m'(\tau_2 + 1)$. We chose the parameters, so that the communication costs are proportional to the number of batched instances $\sqrt{t}$. Thus, we set $t_2 = \lceil \sqrt{(d_P n')/(l_p m')} \rceil$, which yields $t_1 \approx \sqrt{d_P l_P n'}$ and $\tau_2 = \lceil \sqrt{(d_Q n')/(l_Q m')} \rceil$, which yields $\tau_1 \approx \sqrt{d_Q l_Q n'}$. We set $n' \approx \sqrt{(l_a/8)t}$ and $m' \approx t/n' = \sqrt{(8t/l_a)}$. The total cost for the batched proof with t batch instances is $\sqrt{(l_a/8)t}n \log q + 2\sqrt{d_P l_P t}n \log q + 2\sqrt{d_Q l_Q t}n \log q + 8\sqrt{m'n'} \log q + 6\sqrt{m'n'} \log(12\mathcal{B}\sqrt{(l_a/8)m'}) + 4\sqrt{m'n'} \log(12\mathcal{B}\sqrt{3m'n'})$, where the first term describes the costs for communicated $\{\mathtt{D}_j\}_{j\in[0,n']}$. The following three terms describe the costs of the underlying polynomial evaluation protocol for $\mathbf{P}^*, \mathbf{Q}^*$, and the last two terms are costs for the sent values $\hat{\mathbf{a}}, \hat{\mathbf{r}}, \hat{\mathbf{s}}$. The number of batched instances is given for polynomials $\mathbf{P}, \mathbf{Q}$ as $t = m'n'$. We approximate each of the batched instances by t and get a total cost approximation of $\mathcal{O}((\sqrt{(l_a/8)}tn + \sqrt{l_Q}) \log q$, where l_Q denotes the module-rank of M-SIS problem.The dominant term in the communication cost is $\mathcal{O}(\sqrt{(l_a/8)t}n \log q) \log q$.

Concrete Parameters. We use the results from [19], where the maximum of public parameter $\mathbf{A}$ is bounded by $\sqrt{n \log q / \log \delta}$, where $\tilde{n}$ is the maximal number of rows of $\mathbf{A}$. From Theorem 3.1, we have $\|\mathbf{r}_x\| \leq \beta$, where

$x \in \{\mathbf{P}^*, \mathbf{Q}^*\}$ with $\beta_x \leq 12\mathcal{B}\sqrt{mn}$ and the extracted witness $\|\mathbf{r}'_x\| \leq 6\beta'$. A condition which we obtain from rejection sampling yields $\sigma_{k_x,1} \geq 6 \cdot 12\mathcal{B}\sqrt{k_x mn}$ and $\sigma_{k_x,2} \geq 6 \cdot 12\mathcal{B}\sqrt{k_x n}$, where $k_x \in \{t_2, \tau_2\}$ and $\beta' < \min\{q, 2^{2\sqrt{n \log q \log \delta}}\}$. We choose $\lambda = 128$, set $\delta = 1.0035$ being the root Hermite factor and set $q \geq \beta'$, and $n \geq 2^8$. We balance this security level for LWE using LWE estimator [1] and get the number of integer elements $\hat{m} = \mathcal{O}(n)$ over $\mathbb{Z}$ The underlying polynomial evaluation protocol needs to be repeated 63 times to achieve a negligible soundness error of 2^{-100}. We provide the results for 4 different sets in Table 3.

Parameter	Set 1	Set 2	Set 3	Set 4
Commitment modulus q	2^{20}	2^{17}	2^{24}	2^{28}
Ring dimension n	256	512	256	512
d_P	12	12	12	12
d_Q	16	16	16	16
No. of variab. in P $(= l_P)$	2	2	2	2
No. of variab. in Q $(= l_Q)$	4	4	4	4
$\tilde{m}$	768	1024	564	1536
Batches t	10	10	100	100
$\log(\beta')$	≈ 18.9	≈ 14.68	≈ 23.22	≈ 25.86
$\mathcal{B}$	30	2	128	128
Proof size	244.85 KB	332.77 KB	926.95 KB	1.89 MB

Table 3. Batch Proof Parameters

5 Application to Range Proof

In this section we apply the batch protocol to a range proof. First we define the relations to be proved.

$$\mathsf{R}_{range} = \{\mathcal{P}(\mathbf{a}, \mathbf{r}), \mathcal{V}(\mathtt{C}, \mathbf{b}, \mathbf{P}, \mathbf{Q}) : \mathbf{P}(\mathbf{a}) = 0 \wedge \mathtt{C} = \mathtt{Com}_{\mathbf{A}}(\mathbf{Q}(\mathbf{a}, \mathbf{b}), \mathbf{r})\}$$
$$\mathsf{R}'_{range} = \{\mathcal{P}(6\mathbf{a}, \mathbf{r}'), \mathcal{V}(\mathtt{C}, \mathbf{b}, \mathbf{P}, \mathbf{Q}) : \mathbf{P}(6\mathbf{a}) = 0 \wedge \mathtt{C} = \mathtt{Com}_{6\mathbf{A}}(\mathbf{Q}(6\mathbf{a}, \mathbf{b}), \mathbf{r}')\}$$

Protocol. This protocol is run between a prover and a verifier, where the former wants to convince the latter that the committed value is inside a range $[0, 2^\ell - 1]$.
Statement: Let $N = 2^\ell - 1$ be the length of an integer a and $\mathtt{C}$ a commitment to this integer.
Witness: Let $a \in \mathbb{Z}_q$ and $\mathbf{r} \in \mathcal{R}_q^m$ be the corresponding randomness used in out commitment $\mathtt{C} = \mathtt{Com}_{\mathbf{A}}(a, \mathbf{r})$. The integer a is encoded into a polynomial in $\mathcal{R}_q$ as follows: Set $a = (a_0, \ldots, a_{\ell-1})$ be the binary representation of a. We zero-pad a to reach the dimension of the ring $\mathcal{R}_q$ and encode a into a polynomial by assigning each bit a_i to a coefficient of $a(X) := \sum_{i=0}^{\ell-1} a_i X^i$.
Parameter Choice: We set $l_a = l_b = \ell$, $d_Q = 2$, $l_P = \ell$, $l_Q = 2$, $d_P = \ell + 1$. Let $\mathbf{a} = (a_0, \ldots, a_{\ell-1})$ and $\mathbf{b} = (1, 2, 2^2, \ldots, 2^{\ell-1})$. Then $\mathbf{P}(\mathbf{a}) = \mathbf{a} \circ (1 - \mathbf{a})$ and $\mathbf{Q}(\mathbf{a}, \mathbf{b}) = \sum_{i=0}^{\ell-1} a_i 2^i$.

5.1 Comparison

In this section we provide a comparison of the efficiency of our range proof with the range proof which is currently used in the lattice-based LRCT.v2 scheme

[23] and with the range proof in [11]. The total cost of that range proof is: $\tilde{n}n(l+1)\log q+\tilde{n}\kappa\log q+\tilde{n}(m-1)n\log(\sigma_{OR})$, where $\sigma_{OR}=2^{\gamma+1}\sqrt{\kappa\theta n(m-1)}$, θ is the number of repetitions of the underlying OR proof in [23] and l is the range length, i.e we prove that an integer is in the range $[0,2^{\ell}-1]$. Let n denote the dimension of $\mathcal{R}_q$, and $\tilde{n}$ denotes the module rank for M-SIS which is specified to be $\tilde{n}=2$ in [23]. The dominant term in [23] is $\tilde{n}n(\ell+1)\log q$ which considers only one batched instance. That means for t instances it grows to $tn\tilde{n}(\ell+1)\log q$, making the proof less efficient than ours.

In [11] the authors achieved asymptotic communications cost of $2(\tilde{n}+t)n\log q+\mathcal{O}(t\ell n)$, where $\tilde{n}$ is the module rank of M-SIS. It is obvious that $tn\tilde{n}(\ell+1)\log q > 2(\tilde{n}+t)n\log q$, since $\tilde{n}=2$ in [23], yielding that the result in [11] improves by reducing the result in [23] by a factor $t\ell$. As showed in the previous section, our solution achieves asymptotic costs of $\mathcal{O}(\kappa\sqrt{(l_a/8)t})n\log q$, where l_a stands for the length of integer, i.e. $l_a:=\ell$ and κ is the number of repetitions of our protocol to achieve a security level of 2^{128}. We set $\kappa=63$ and l_Q is the number of rows of public parameter $\mathbf{A}$. In this application to the range proof we assume that $l_Q=2$, i.e. module rank of M-SIS is 2. It is not obvious for which parameters our protocol outperforms the concurrent one [11]. We will compare our results with those in [11] for integer length of 32 bits with ring dimension $n=16$ and $q=32$. In order to see when our construction becomes more advantageous than [11] we set $2(\tilde{n}+t)n\log(q)+tn\ell > 63\sqrt{(\ell/8)t}n\log(q)$ and solve it for t when $n=16$, $\ell=32$, $\log(q)=38$ and $\tilde{n}\geq 92$. It yields $2(92+t)\cdot 38+t\cdot 32 > 63\cdot\sqrt{t32/8}\cdot 38$. After plotting the two functions from both sides of the last unequation we conclude that from number $t\geq 1154$ our scheme outperforms the one in [11].

We conclude that our contribution provides a significant improvement compared to [11,23]. In [11] the authors instantiated their range proof with 10 batched instances and a range width of $N=2^{64}$. In order to make a fair comparison to our work, it hold $\log N=\ell$ and $q\geq 2^{\ell}$. We adapt these sized to our batch protocol in Table 4 and obtain the following results:

Parameter	[11]	Our Work	[11]	Our Work	[11]	Our Work
Modulus q	2^{38}	2^{33}	2^{64}	2^{64}	2^{64}	2^{64}
Ring dim. n	16	16	8	8	8	8
$\log(N)$	32	32	64	64	64	64
# of batches t	1200	1200	1200	1200	10000	10000
λ	128	128	128	128	128	128
Proof size	≈ 13.85 MB	9.69 MB	≈ 22 MB	18.61 MB	≈ 160 MB	54.57

Table 4. Range Proof (table and graph)

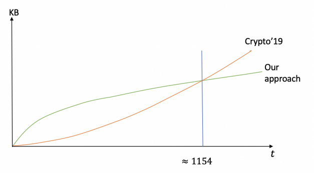

Fig. 1. Comparison graph

We conclude that our approach achieves a significant improvement if the number of batched instances is higher than 1155. Additionally we showed that for a higher number of batched instances we achieve a more significant difference to [11], namely our proof size grows approximately by a square-root factor of t, while the proof size in [11] grows almost linearly. The simplified representation of the two graphs is given in the following graph in Table (4)

References

1. M. R. Albrecht, R. Player, and S. Scott. On the concrete hardness of learning with errors. *J. Mathematical Cryptology*, 9(3):169–203, 2015.
2. C. Baum, I. Damgård, K. G. Larsen, and M. Nielsen. How to prove knowledge of small secrets. In *CRYPTO 2016*, LNCS, pages 478–498. Springer, 2016.
3. C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peikert. More efficient commitments from structured lattice assumptions. In *SCN 2018*, pages 368–385, 2018.
4. S. Bayer and J. Groth. Zero-knowledge argument for polynomial evaluation with application to blacklists. In *EUROCRYPT 2013*, LNCS, pages 646–663. Springer, 2013.
5. R. Bendlin and I. Damgård. Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems. In *TCC 2010*, LNCS, pages 201–218. Springer, 2010.
6. F. Benhamouda, J. Camenisch, S. Krenn, V. Lyubashevsky, and G. Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. In *ASIACRYPT 2014*, LNCS, pages 551–572. Springer, 2014.
7. F. Benhamouda, S. Krenn, V. Lyubashevsky, and K. Pietrzak. Efficient zero-knowledge proofs for commitments from learning with errors over rings. In *ESORICS 2015*, LNCS, pages 305–325. Springer, 2015.
8. J. Bootle, A. Cerulli, P. Chaidos, J. Groth, and C. Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In *EUROCRYPT 2016*, LNCS, pages 327–357. Springer, 2016.
9. J. Bootle and J. Groth. Efficient batch zero-knowledge arguments for low degree polynomials. In *PKC 2018*, pages 561–588, 2018.
10. I. Damgård and A. López-Alt. Zero-knowledge proofs with low amortized communication from lattice assumptions. In *SCN 2012*, LNCS, pages 38–56. Springer, 2012.
11. M. F. Esgin, R. Steinfeld, J. K. Liu, and D. Liu. Lattice-based zero-knowledge proofs: New techniques for shorter and faster constructions and applications. 2019. `https://eprint.iacr.org/2019/445`.
12. M. F. Esgin, R. Steinfeld, A. Sakzad, J. K. Liu, and D. Liu. Short lattice-based one-out-of-many proofs and applications to ring signatures. In *ACNS*, LNCS, pages 67–88. Springer, 2019.
13. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *Proceedings of the 17th Annual ACM STOC 1985*, pages 291–304. ACM, 1985.
14. J. Groth. Linear algebra with sub-linear zero-knowledge arguments. In *CRYPTO 2009*, LNCS, pages 192–208. Springer, 2009.
15. J. Groth. A verifiable secret shuffle of homomorphic encryptions. *J. Cryptology*, 23(4):546–579, 2010.
16. J. Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *ACM, Proceedings, 1992*, pages 723–732. ACM, 1992.
17. A. Langlois and D. Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptography*, pages 565–599, 2015.
18. V. Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT 2009*, LNCS, pages 598–616. Springer, 2009.
19. V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, 2012.
20. N. V. Saberhagen. Cryptonote v 2.0. 2013.

21. J. Stern. A new identification scheme based on syndrome decoding. In *CRYPTO 1993*, LNCS, pages 13–21. Springer, 1993.
22. W. A. A. Torres, V. Kuchta, R. Steinfeld, A. Sakzad, J. K. Liu, and J. Cheng. Lattice ringct v2.0 with multiple input and multiple output wallets. In *ACISP 2019*, LNCS, pages 156–175, 2019.
23. W. A. A. Torres, R. Steinfeld, A. Sakzad, J. K. Liu, V. Kuchta, N. Bhattacharjee, M. H. Au, and J. Cheng. Post-quantum one-time linkable ring signature and application to ring confidential transactions in blockchain (lattice ringct v1.0). In *ACISP 2018*, LNCS, pages 558–576, 2018.

A Table of Notations and Protocols

In this section we provide a table of notations (see Table 5) and the more formal description of the protocols Π_{PEv}, Π_{Batch}, see Fig. 2, 3, respectively.

Σ-Protocols A Σ-protocol is an interactive proof system between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ which is defined for a relation R and for a statement-witness pair $(v, w) \in \mathsf{R}$. We use the definition from [12].

Definition A.1 ([12], Definition 4). *For relations* R, R' *where* $\mathsf{R} \subseteq \mathsf{R}'$, $(\mathcal{P}, \mathcal{V}$ *is a Σ-protocol with completeness error* α*, a challenge space* $\mathcal{CH}$*, public-private inputs* (v, w) *if the following security properties are satisfied:*

Completeness: *A transcript between an honest prover and an honest verifier is accepted with probability at least* $1 - \alpha$.

$(k+1)$***-special soundness:*** *There exists an efficient PPT extractor* $\mathcal{E}$ *that computes* w' *satisfying* $(v, w') \in \mathsf{R}'$ *given* $(k+1)$ *accepting transcripts.*

Special honest-verifier zero-knowledge (SHVZK): *There exists an efficient PPT simulator* $\mathcal{S}$ *that outputs a transcript given the public input in the language of* R *and a challenge in* $\mathcal{CH}$*, such that this transcript is indistinguishable from an accepting transcript produced by a real run of the protocol.*

B Example to Batch Technique

Example: We consider the following case of 3-variate polynomials, i.e. $l_a + l_b = 3$ with variables X, Y, Z. Let $l_Q = 2$ and $m' = 2$, i.e. we have 2 elements $\mathbf{Q}(\mathbf{a}_{j,0}, \mathbf{b}_{j,0}), \mathbf{Q}(\mathbf{a}_{j,1}, \mathbf{b}_{j,1})$, where each of them is a vector which contains 2 multivariate polynomials:, i.e.:

$$\mathbf{Q}(\mathbf{a}_{j,0}, \mathbf{b}_{j,0}) = \begin{pmatrix} Q_1(\mathbf{a}_{j,0}, \mathbf{b}_{j,0}) = XYZ^2 + Y^3X^2 + YZ^2 + Y^2XZ^3 \\ Q_2(\mathbf{a}_{j,0}, \mathbf{b}_{j,0}) = XY^3 + X^3Y^2Z^2 + XY^2 + X^2Y^3 \end{pmatrix}$$

$$\mathbf{Q}(\mathbf{a}_{j,1}, \mathbf{b}_{j,1}) = \begin{pmatrix} Q_1(\mathbf{a}_{j,1}, \mathbf{b}_{j,1}) = X^3Y^2 + XZ^2 + XY^3Z + YZ^3 \\ Q_2(\mathbf{a}_{j,1}, \mathbf{b}_{j,1}) = Y^2Z + X^2Z^2 + X^3Y^2Z + XY^2Z^3 \end{pmatrix}$$

In order to construct a commitment $\mathtt{Com}_{\mathbf{A}}(\mathbf{Q}(\mathbf{a}_{j,0}, \mathbf{b}_{j,0}), \mathbf{Q}(\mathbf{a}_{j,1}, \mathbf{b}_{j,1}); \mathbf{r}_j)$ to both vectors $\mathbf{Q}(\mathbf{a}_{j,0}, \mathbf{b}_{j,0})$ and $\mathbf{Q}(\mathbf{a}_{j,1}, \mathbf{b}_{j,1})$ we first define the univariate polynomials $Q_\iota(\mathbf{a}_{j,i}, \mathbf{b}_{j,i}), \iota \in [1, 2], i \in \{0, 1\}$ in each of the variable X, Y, Z:

Notation	**Description**
$\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n+1)$	Polynomial ring
q	Prime modulus defining $\mathcal{R}_q$
λ	Security parameter
$n(= n'm'-1)$	Degree of polynomials in $\mathcal{R}_q$
$f(X) = (X-u)q(X)+w$	A polynomial of degree n in $\mathcal{R}_q$ with coefficients $a_k \in \mathbb{Z}_q$, u the evaluation value and w the evaluated value of $f(u)$.
$\mathcal{A} = (a_{j,i})_{j\in[0,n'-1],i\in[0,m'-1]}$	Coefficient matrix of $f(X)$ of dimension $n' \times m'$
$\widetilde{\mathcal{A}} = (\tilde{a}_{j,i})_{j\in[0,n'],i\in[0,m'-1]}$	Matrix $\mathcal{A}$ of dimension $(n'+1) \times m'$ after masking
$\mathcal{T} = (t_{j,i})_{j'\in[0,\nu-1],i\in[0,\mu-1]}$	Coefficient matrix of $q(X)$ of dimension $\nu \times \mu$, $\deg(q(X)) = n-1 = \nu\mu - 1$
$\widetilde{\mathcal{T}} = (\tilde{t}_{j',i'})_{j'\in[0,\nu],i'\in[0,\mu-1]}$	Masked coefficient matrix of polynomial $q(X)$
$\mathbf{a}_j, \mathbf{t}_{j'}$	The $j-$th, j'-th row of the matrix $\mathcal{A}$, $\mathcal{T}$
$\mathbf{r}_{a,j}, \mathbf{r}_{t,j'} \in \mathcal{R}_q^m$	Randomness of $\mathbf{a}_j, \mathbf{t}_{j'}$, respectively
$\mathbb{A}_j, \mathbb{T}_{j'}$	Commitment to the j-th row $\mathbf{a}_j$, j'-th row $\mathbf{t}_{j'}$, respectively
$\hat{f}(X)$	The vector-matrix product defined as: $(1, X, \ldots, X^{n'-1}) \cdot \mathcal{A}$
$\mathfrak{D}_\sigma^n$	n-dimensional discrete Normal distribution with stand. deviation σ
$u_{j,i}, s_{j',i'}$	Masking values used in $\mathcal{A}, \mathcal{T}$, respectively
$\tilde{\mathbf{a}}_j$	The $j-$th row of the matrix $\widetilde{\mathcal{A}}$
$\widetilde{\mathbb{A}}_j, \widetilde{\mathbb{T}}_{j'}$	Commitment to $\tilde{\mathbf{a}}_j, \tilde{\mathbf{t}}_{j'}$, respectively
$\mathbf{u}, \mathbf{s}$	The $(n'+1)$-th, $(\nu+1)$-th row of matrix $\widetilde{\mathcal{A}}, \widetilde{\mathcal{T}}$, respectively
$\mathbf{r}_u, \mathbf{r}_s$	Randomness to $\mathbf{u}, \mathbf{s}$, respectively
$\mathbb{U}$, $\mathbb{S}$	Commitment to $\mathbf{u}, \mathbf{s}$ with randomness $\mathbf{r}_u, \mathbf{r}_s$, respectively
$\tilde{\mathbf{r}}_{\mathbf{f}}$, $\tilde{\mathbf{r}}_{\mathbf{q}}$	Polynomials: $\sum_{j=0}^{n'-1} \mathbf{r}_{a,j} X^j + \mathbf{r}_u X^{n'}$, $\sum_{j=0}^{\nu-1} \mathbf{r}_{t,j} X^j + \mathbf{r}_s X^{n'}$, resp.
$\mathsf{R}_{(\cdot)}$	Relation to be proved
ν, μ	Decomposition of $n-1$. It holds $n-1 = \nu\mu - 1$
$\mathtt{C}_{(\cdot)}$	Commitment to a certain input $(\cdot)$
$\mathtt{m}_{(\cdot)}$	Masking value of a certain input $(\cdot)$
$\mathcal{CH}, \mathit{\Delta}\mathcal{CH} = \lvert\mathcal{CH} - \mathcal{CH}\rvert$	Set of challenges, set of challenge differences, resp.
β	Euclidean norm of a randomness $\mathbf{r}_{(\cdot)}$
$\mathcal{B}$	Infinity norm of a randomness $\mathbf{r}_{(\cdot)}$
$\mathit{\Pi}_{\mathtt{PolEv}}$	Polynomial evaluation zero-knowledge protocol
$\xi_{\mathtt{m}_{(\cdot)}}$	Masked responses
$\mathbf{r}_{\xi,\mathtt{m}_{(\cdot)}}$	Randomness to the masked response $\xi_{\mathtt{m}_{(\cdot)}}$
$\mathbf{P}$	Polynomial vector of l_P multiv. polynomials $\{P_{\iota'}\}_{\iota'\in[l_P]}$ of total degree d_P
$\mathbf{Q}$	A Polynomial vector of l_Q polynomials $\{Q_\iota\}_{\iota\in[l_Q]}$ of total degree d_Q
$n = n'm' - 1$	Degree of polynomials in $\mathcal{R}_q$
$\mathbf{a}_{j,i} = (a_{j,i,1}, \ldots, a_{j,i,l_a})$	l_a-variate polynomial input vector to $\mathbf{P}$ and $\mathbf{Q}, j \in [0, m'-1], i \in [0, n'-1]$
$\mathbf{b}_{j,i} = (b_{j,i,1}, \ldots, b_{j,i,l_b})$	l_b-variate polynomial input vector to $\mathbf{Q}, j \in [0, m'-1], i \in [0, n'-1]$
$\mathbf{Q}(\mathbf{a}_{j,i}, \mathbf{b}_{j,i})$	A vector of $l_Q(l_a + l_b)$-variate polynomials of total degree d_Q
$\mathbf{P}(\mathbf{a}_{j,i})$	A vector of $l_P l_a$-variate polynomials of total degree d_P
$A_\kappa = (a_{j,i,\kappa})_{j\in[0,n'],i\in[0,m'-1]}$	Coefficient matrix of dimension $n' \times m'$ for $\kappa \in [l_a]$
$B_{\kappa'} = (b_{j,i,\kappa})_{j\in[0,n'],i\in[0,m'-1]}$	Coefficient matrix of dimension $n' \times m'$ for $\kappa' \in [l_b]$
$\tilde{\mathbf{a}}_{j,i} = (\tilde{a}_{j,i,1}, \ldots, \tilde{a}_{j,i,l_a})$	l_a-variate polynomial input vector to $\mathbf{P}$ and $\mathbf{Q}$ after masking A_κ
$l_j(X)$	Lagrange polynomials for all $j \in [0, n']$
$\mathbf{P}_i^*(X)$	Polynomial $\mathbf{P}(\mathbf{a}_{j,i})$ evaluated at X
$\mathcal{P}_i^* = (p_{\nu,\mu})_{\nu\in[0,t_1],\mu\in[0,t_2]}$	Coefficient matrix (as in Section 3) of $\mathbf{P}_i^*(X)$ of dimension $t_1 \times t_2$
$\widetilde{\mathcal{P}}_i^* = (\tilde{p}_{\mu,\nu})_{\mu\in[0,t_2+1],\nu\in[0,t_2]}$	Masked coefficient matrix $\mathcal{P}_i^*$
$\mathcal{Q}_i^* = (q_{\tilde{\mu},\tilde{\nu}})_{\tilde{\mu}\in[0,\tau_2],\tilde{\nu}\in[0,\tau_1]}$	Coefficient matrix (as in Section 3) of $\mathbf{Q}_i^*(X)$ of dimension $\tau_1 \times \tau_2$
$\widetilde{\mathcal{Q}}_i^* = (\tilde{q}_{\tilde{\mu},\tilde{\nu}})_{\tilde{\mu}\in[0,\tau_2+1],\tilde{\nu}\in[0,\tau_1]}$	Masked coefficient matrix $\mathcal{Q}_i^*$
$\mathbb{A}_{\mathbf{P}_i^*,\mu}, \mathbb{A}_{\mathbf{Q}_i^*,\tilde{\mu}}$	Commitment to the $\mu-$th or $\tilde{\mu}-$th row of the matrix $\widetilde{\mathcal{P}}_i^*$, $\widetilde{\mathcal{Q}}_i^*$, resp.
$\mathbf{H}_{\chi_i}$, $\chi_i \in \{\mathbf{P}_i^*, \mathbf{Q}_i^*\}$	Divisor (with rest) of χ_i s.t. $\chi_i = (X - \mathbf{y}) \cdot \mathbf{H}_{\chi_i} + \mathbf{z}$, where $\chi_i(\mathbf{y}) = \mathbf{z}$
$\widetilde{\mathcal{H}}_{\chi_i}$	Masked coefficient matrix of $\mathbf{H}_{\chi_i}(X)$ for all $\chi_i \in \{\mathbf{P}_i^*, \mathbf{Q}_i^*\}$
$\mathbb{T}_{\mathbf{P}_i^*,\mu}, \mathbb{T}_{\mathbf{Q}_i^*,\tilde{\mu}}$	Commitment to the $\mu-$th or $\tilde{\mu}-$th row of the matrix $\widetilde{\mathcal{H}}_{\chi_i}$, resp.
$\mathbb{U}_{\chi_i}, \mathbb{S}_{\chi_i}$	Commitment to the last row of $\widetilde{\mathcal{P}}_i^*$, $\widetilde{\mathcal{Q}}_i^*$ for $\chi_i \in \{\mathbf{P}_i^*, \mathbf{Q}_i^*\}$

Table 5. Important Notations for Section 3 and 4(in chronological order)

For $\iota = 1, i = 0$, i.e. $Q_1(\mathbf{a}_{j,0}, \mathbf{b}_{j,0})$ we have: $Q_{j,0,1}(X) = yz^2 + (yz^2 + y^2z^3)X + y^3X^2$, $Q_{j,0,1}(Y) = (xz^2 + z^2)Y + xz^3Y^2 + x^2Y^3$, $Q_{j,0,1}(Z) = y^3x^2 + (xy + y)Z^2 + y^2xZ^3$.
For $\iota = 2, i = 0$, i.e. $Q_2(\mathbf{a}_{j,0}, \mathbf{b}_{j,0})$ we have: $Q_{j,0,2}(X) = (y^2 + y^3)X + y^3X^2 + y^2z^2X^3$, $Q_{j,0,2}(Y) = (x^3z^2 + x)Y^2 + (x + x^2)Y^3$, $Q_{j,0,2}(Z) = xy^3 + xy^2 + x^2y^3 + x^3y^2Z^2$.
For $\iota = 1, i = 1$ i.e. $Q_1(\mathbf{a}_{j,1}, \mathbf{b}_{j,1})$ we have: $Q_{j,1,1}(X) = yz^3 + (y^3z + z^2)X + y^2X^3$, $Q_{j,1,1}(Y) = xz^2 + z^3Y + x^3Y^2 + xzY^3$, $Q_{j,1,1}(Z) = x^3y^2 + xy^3Z + xZ^2 + yZ^3$.
For $\iota = 2, i = 1$, i.e. $Q_2(\mathbf{a}_{j,1}, \mathbf{b}_{j,1})$ we have: $Q_{j,1,2}(X) = y^2z + y^2z^3X + z^2X^2 + y^2zX^3$, $Q_{j,1,2}(Y) = x^2z^2 + (z + x^3z + xz^3)Y^2$, $Q_{j,1,2}(Z) = (y^2 + x^3y^2)Z + x^2Z^2 + xy^2Z^3$.
All lower-case letters x, y, z and their products and sums denote coefficients of variables $X^i, Y^i, Z^i, i \in [0, 3]$. Next, we pack polynomials $Q_{j,i,\iota}$ in a particular variable for $i \in [0, 1], \iota \in [1, 2]$ into one vector, i.e we define

$$Q_{j,1}(X) = \big(Q_{j,0,1}(X), Q_{j,1,1}(X)\big),\ \ Q_{j,1}(Y) = \big(Q_{j,0,1}(Y), Q_{j,1,1}(Y)\big),\ \ Q_{j,1}(Z) = \big(Q_{j,0,1}(Z), Q_{j,1,1}(Z)\big),$$
$$Q_{j,2}(X) = \big(Q_{j,0,2}(X), Q_{j,1,2}(X)\big),\ \ Q_{j,2}(Y) = \big(Q_{j,0,2}(Y), Q_{j,1,2}(Y)\big),\ \ Q_{j,2}(Z) = \big(Q_{j,0,2}(Z), Q_{j,1,2}(Z)\big)$$

We pack all these vectors from above into a matrix as follows:

$$\mathbf{Q}_j = \begin{pmatrix} Q_{j,1}(X) & Q_{j,1}(Y) & Q_{j,1}(Z) \\ Q_{j,1}(X) & Q_{j,1}(Y) & Q_{j,1}(Z) \end{pmatrix}$$

Now we can commit to the vectors $\mathbf{Q}(\mathbf{a}_{j,0}, \mathbf{b}_{j,0}), \mathbf{Q}(\mathbf{a}_{j,1}, \mathbf{b}_{j,1})$ which, as we have seen, are now packed into a matrix $\mathbf{Q}_j$, i.e $\mathtt{Com}_{\mathbf{A}}(\mathbf{Q}(\mathbf{a}_{j,0}, \mathbf{b}_{j,0}), \mathbf{Q}(\mathbf{a}_{j,1}, \mathbf{b}_{j,1}); \mathbf{r}_j) = \mathtt{Com}_{\mathbf{A}}(\mathbf{Q}_j, \mathbf{r}_j)$.

C Proofs of Theorem 3.6 and Theorem 4.3

C.1 Proof of Theorem 3.6

Proof. To prove the theorem we prove the three properties:

Completeness: It is easy to see that the verification equations (17) – (20) pass the test and that the verifier obtains the evaluation of the committed polynomial $f(X)$ at an evaluation point v. We verified the correctness of equation (17) in section 3.1. This implies also correctness of equation (18). Both of these equations can verify the equation (19). Further, we have to show that $\|\mathbf{u}\|$ and $\|\mathbf{s}\|$ are statistically close to $\mathfrak{D}^{m'-1}_{12\mathcal{B}\sqrt{m'}}$ and $\|x^j \cdot \mathbf{r}_{a,j}\|$, $\|x^j \cdot \mathbf{r}_{t,j'}\|$ for $j \in [0, n']$ are statistically close to $\mathfrak{D}^{m'n'}_{12\mathcal{B}\sqrt{m'n'}}, \mathfrak{D}^{\mu\nu}_{12\mathcal{B}\sqrt{\mu\nu}}$, respectively. Also $\|\mathbf{r}_{\xi,\iota}\|$ for $\iota \in \{\mathtt{m}_v, \mathtt{m}_w\}$ are statistically close to $\mathfrak{D}_{12\mathcal{B}\sqrt{mn}}$ and $\mathfrak{D}_{12\mathcal{B}}\sqrt{n'mn}$, respectively.

3**-Special Soundness:** To prove the soundness we provide a reduction to the binding property of our commitment scheme. As explained before, because of the hiding issues we consider a restricted challenge space of 3 different challenges $x \in \mathbb{Z}_q$, x_1, x_2, x_3. These vectors form a generalized Vandermonde matrix

$\mathbf{V}_{n'+1} = \begin{pmatrix} x_1^i & x_1^{i+1} & x_1^{i+2} \\ x_2^i & x_2^{i+1} & x_2^{i+2} \\ x_3^i & x_3^{i+1} & x_3^{i+2} \end{pmatrix}$. Since the columns of this matrix are independent vectors, we can reconstruct any unit vector by taking an appropriate linear combination of these columns. First we rewrite (3.2) as:

$$\mathtt{Com}_{\mathbf{A}}(\tilde{\mathbf{f}}(x); \tilde{\mathbf{r}}_{\mathbf{f}}) = \mathbf{A} \cdot \Big(\sum_{j=0}^{n'-1} \mathbf{r}_{a,j} x^j + \mathbf{r}_u x^{n'} \Big) + \mathtt{enc}(\tilde{\mathbf{f}}(x)) \tag{15}$$

We see that $\tilde{\mathbf{f}}(x) = \sum_{j=0}^{n'-1} x^j \cdot \tilde{\mathbf{a}}_j + \mathbf{u}x^{n'}$. Thus, equation (15) is equivalent to:

$$\mathtt{Com}_{\mathbf{A}}(\tilde{\mathbf{f}}(x); \tilde{\mathbf{r}}_{\mathbf{f}}) = \mathbf{A}\Big(\sum_{j=0}^{n'-1} \mathbf{r}_{a,j} x^j + \mathbf{r}_u x^{n'} \Big) + \mathtt{enc}(\sum_{j=0}^{n'-1} \tilde{\mathbf{a}}_j x^j + \mathbf{u} x^{n'}). \tag{16}$$

Since each $\mathbf{r}_{a,j}, \mathbf{r}_u, \tilde{\mathbf{a}}_j, \mathbf{u}$ for $j \in [0, n'-1]$ is a m'-dimensional vector, we can extract the openings of any commitments $\widetilde{\mathbb{A}}_j = \mathtt{Com}_{\mathbf{A}}(\tilde{\mathbf{a}}_j; \mathbf{r}_{a,j})$, by computing $2n'$ verification equations in (15) for challenges x_1, x_2, x_3. Then, (16) gives a system of polynomial equations for 3 challenges with $k \in \{1,2,3\}$: $\tilde{\mathbf{c}}_k = \mathbf{A}\big(\sum_{j=0}^{n'-1} \mathbf{r}_{a,j} x_k^j + \mathbf{r}_u x_k^{n'} \big) + \sum_{j=0}^{n'-1} \tilde{\mathbf{a}}_j x_k^j + \mathbf{u} x_k$. Next, we take $\mathtt{Com}_{\mathbf{A}}(0, \tilde{\mathbf{r}}_{\mathbf{f}})$ we compute $\mathbf{c}'_k = \tilde{\mathbf{c}}_k - \mathtt{Com}_{\mathbf{A}}(0, \tilde{\mathbf{r}}_{\mathbf{f}})$ and get: $\mathbf{c}'_k = \tilde{\mathbf{c}}_k - \mathtt{Com}_{\mathbf{A}}(0, \tilde{\mathbf{r}}_{\mathbf{f}}) = \mathbf{A}\Big(\sum_{j=0}^{n'-1} \mathbf{r}_{a,j} x_k^j + \mathbf{r}_u x_k^{n'} \Big) + \mathtt{enc}(\sum_{j=0}^{n'-1} \tilde{\mathbf{a}}_j x_k^j + \mathbf{u} x_k^{n'}) - \mathtt{Com}_{\mathbf{A}}(0, \tilde{\mathbf{r}}_{\mathbf{f}}) = \mathtt{enc}(\sum_{j=0}^{n'-1} \tilde{\mathbf{a}}_j x_k^j + \mathbf{u}(x_k)) \in \mathcal{R}_q^2$, where $\mathbf{u}(x_k) = \big(\sum_{j=0}^{n'-1} u_{n'-j-1,1} x_k^j, \ldots, \sum_{j=0}^{n'-1} u_{n'-j-1,m'-1} x_k^j, 0 \big)$. Note that $\tilde{a}_{j,i} = a_{j,i} - u_{j,i} x_k^{n'-(2j+1)}$ yielding $\sum_{j=0}^{n'-1} \tilde{a}_{j,i} x_k^j = a_{0,i} - u_{0,i} + \ldots + a_{n'-1,i} x_k^{n'-1} - u_{n'-1} x_k$.

After reshuffling the terms we get $\sum_{j=0}^{n'-1} \tilde{a}_{j,i} x_k^j = \sum_{j=0}^{n'-1} (a_{j,i} - u_{n'-j-1,i}) x_k^j$ and we set $\tilde{\tilde{a}}_{j,i} := a_{j,i} - u_{n'-j-1,i}$. For all $i \in [0, m'-1], j \in [0, n'-1]$ it yields that $(\mathbf{c}'_\iota)^{tr}$ for challenge index $\iota \in \{1,2,3\}$ is equal to:

$$\mathbf{c}'_k = \begin{bmatrix} \tilde{\tilde{a}}_{0,0} \\ \vdots \\ \tilde{\tilde{a}}_{0,m'-1} \end{bmatrix} x_k^0 + \cdots + \begin{bmatrix} \tilde{\tilde{a}}_{n'-1,0} \\ \vdots \\ \tilde{\tilde{a}}_{n'-1,m'-1} \end{bmatrix} x_k^{n'-1} + \begin{bmatrix} \sum_{j=0}^{n'-1} u_{n'-j-1,1} x_k^j \\ \vdots \\ \sum_{j=0}^{n'-1} u_{n'-j-1,m'-1} x_k^j \\ 0 \end{bmatrix} x_k^{n'}$$

For all $k \in \{1,2,3\}$, and $i \in [0, m'-1]$ we write each vector $\mathbf{c}'_{k,i}$ as a linear system of m' equations: $c_{k,i} = \tilde{\tilde{a}}_{0,i} x_k^0 + \ldots + \tilde{\tilde{a}}_{n'-1,i} x_k^{n'-1} + \sum_{j=0}^{n'-1} u_{n'-j-1,k} x_k^{n'+j}$ We obtain 3 blocks B_k, where $k \in \{1,2,3\}$ of linear systems where each such system has m' rows and $2n'$ coefficients, i.e. $B_k = [c_{k,0}, \ldots, c_{k,m'-1}]^{tr}$. We reshuffle these 3 blocks as follows: We take the $j-$th row, i.e. $c_{k,j}$ from each block B_k and collect them into a new block $B^{(j)} = [c_{1,j}, c_{2,j}, c_{3,j}]^{tr}$, such that we get in total m' blocks of linear systems with 3 rows and $2n'$ coefficients. For sake of better visualization, we show the reshuffling process on an example where we collect the 0-th row from each block $\{B_k\}_{k \in \{1,2,3\}}$ into a new block $B^{(0)}$. For all $k \in \{1,2,3\}$, we have: $c_{k,0} = \tilde{\tilde{a}}_{0,0} x_k^0 + \ldots + \tilde{\tilde{a}}_{n'-1,0} x_k^{n'-1} + \sum_{j=0}^{n'-1} u_{n'-j-1,1} x_k^{n'+j}$.

In the same manner we compute $B^{(j)}$ for all $j \in [0, n'-1]$, where $B^{(j)}$ can be represented as follows: We define $\mathbf{c}^{(j)} := (c_1^{(j)}, c_2^{(j)}, c_3^{(j)})^{tr}$ being the row from each block $\{B_k\}_k$ and $\tilde{\tilde{\mathbf{a}}}^{(j)} = (\tilde{\tilde{a}}_0^{(j)}, \dots, \tilde{\tilde{a}}_{n'-1}^{(j)}, u_{n'-1}^{(j)}, \dots u_0^{(j)})^{tr}$ being the vector of coefficients in each $c_k^{(j)}$. Then, each such system $\mathbf{c}^{(j)}$ can be represented as a product of a challenge matrix $\widetilde{\mathbf{V}}_3$ and the vector $\tilde{\tilde{\mathbf{a}}}^{(j)}$. The challenge matrix can be represented by a row-vector consisting of block Vandermonde matrices: $\widetilde{\mathbf{V}}_3 = \begin{pmatrix} 1 \; x_1^1 \; x_1^2 \cdots x_1^{(2n'-1)} \\ 1 \; x_2^1 \; x_2^2 \cdots x_2^{(2n'-1)} \\ 1 \; x_3^1 \; x_3^2 \cdots x_3^{(2n'-1)} \end{pmatrix} = \left[\begin{pmatrix} 1 \; x_1^1 \; x_1^2 \\ 1 \; x_2^1 \; x_2^2 \\ 1 \; x_3^1 \; x_3^2 \end{pmatrix} \cdots \begin{pmatrix} x_1^{2n'-3} & x_1^{2n'-2} & x_1^{2n'-1} \\ x_2^{2n'-3} & x_2^{2n'-2} & x_2^{2n'-1} \\ x_3^{2n'-3} & x_3^{2n'-2} & x_3^{2n'-1} \end{pmatrix} \right]$.

Since our challenge space is restricted to the set of $\mathcal{CH} = \{1, x^1, x^{-1}\}$, then: $\widetilde{\mathbf{V}}_3 = \left[\begin{pmatrix} 1 & x^1 & x^2 \\ 1 & x^{-1} & x^{-2} \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x^3 & x^4 & x^5 \\ x^{-3} & x^{-4} & x^{-5} \\ 1 & 1 & 1 \end{pmatrix} \cdots \begin{pmatrix} x^{2n'-3} & x^{2n'-2} & x^{2n'-1} \\ x^{3-2n'} & x^{2-2n'} & x^{1-2n'} \\ 1 & 1 & 1 \end{pmatrix} \right]$. We denote each block of $\widetilde{\mathbf{V}}_3$ as $\widetilde{\mathbf{V}}_{3,\kappa}$ for $\kappa \in [1, (2n'/3)]$. The determinant of each $\widetilde{\mathbf{V}}_{3,\kappa}$ is equal and is calculated to the following value: $2x^{-1} - 2x + x^2 - x^{-2}$ with Euclidean norm equal to 6. We define the inverses of all $\widetilde{\mathbf{V}}_{3,\kappa}$ and set it into the following $2n' \times 3$ matrix $(\widetilde{\mathbf{V}}_3^{-1})^{tr} = [(\widetilde{\mathbf{V}}_{3,1}^{-1}, \widetilde{\mathbf{V}}_{3,2}^{-1}, \widetilde{\mathbf{V}}_{3,3}^{-1})]^{tr}$. We get m' blocks which can be represented for all $i \in [0, m'-1]$ as the following equation $B^{(i)} : \; \mathbf{c}^{(i)} = \widetilde{\mathbf{V}}_3 \cdot \tilde{\tilde{\mathbf{a}}}^{(i)}$. Multiplying each block by $\widetilde{\mathbf{V}}_3^{-1}$ from the left extracts the $2n'$ vectors $\tilde{\tilde{\mathbf{a}}}^{(0)}, \dots, \tilde{\tilde{\mathbf{a}}}^{(n'-1)}$ and $\mathbf{u}_0, \dots, \mathbf{u}_{n'-1}$.

Similarly, we can extract the randomness $\{\mathbf{r}_{a,j}\}_j$ by setting $\hat{\mathbf{r}} := \sum_{j=0}^{n'-1} \mathbf{r}_{a,j} x^j + \tilde{\mathbf{r}}_u x^{n'}$ using the 3 challenges and the same extraction procedure as described above. We obtain the following system: $\widetilde{B}^{(i)} : \; \hat{\mathbf{r}}^{(i)} = \widetilde{\mathbf{V}}_3 \cdot \tilde{\mathbf{r}}^{(i)}$ for $i \in \{0, m'-1\}$ Since we use unbounded commitment scheme, we only need to compute the bound of the extracted randomness. It holds that $\|\hat{\mathbf{r}}^{(i)}\| = \|\widetilde{\mathbf{V}}_3\| \|\tilde{\mathbf{r}}^{(i)}\| \leq 6 \cdot 12\mathcal{B}\sqrt{m'n}$. Since our commitments are binding, each response is computed as it is done by an honest prover in the argument. Therefore, for a bigger number than j challenges, the extracted secrets $\mathbf{r}_{a,j}, \mathbf{r}_u, \tilde{\tilde{\mathbf{a}}}_j, \mathbf{u}$, $j \in [0, n'-1]$, yield $\tilde{\mathbf{f}}(x)$.

Similarly we extract the values $\tilde{\tilde{\mathbf{t}}}^{(0)}, \dots, \tilde{\tilde{\mathbf{t}}}^{(\nu-1)}$ and $\mathbf{s}_0, \dots, \mathbf{s}_{n'-1}$ (Due to the similarities to the extraction of $\tilde{\tilde{\mathbf{a}}}^{(0)}, \dots, \tilde{\tilde{\mathbf{a}}}^{(n'-1)}$ and $\mathbf{u}_0, \dots, \mathbf{u}_{n'-1}$ we omit the details of this extraction procedure). Since the challenge space contains only 3 elements, in order to achieve a negligible soundness error, we need to repeat the protocol $\lambda/log(3)$ times.

Special Honest Verifier Zero-Knowledgeness: To prove the zero-knowledge property of our protocol, we need to show how to simulate an evaluation of the polynomial $f(X)$ at a certain point x, where the evaluation of $f(v) = w$ is given to us and is equal to $w = \tilde{\mathbf{f}}(u) \cdot (1, u^{n'}, \dots, u^{(m'-1)n'})^t$. To run the simulation, we pick randomly the $m'-1$ values $\tilde{f}_1(x), \dots, \tilde{f}_{m'-1}(x) \leftarrow \mathfrak{D}_\sigma^n$ which are components of the vector $\tilde{\mathbf{f}}(x)$ and compute $\tilde{f}_0(x) = w - \sum_{i=1}^{m'-1} \tilde{f}_i(x) \cdot x^{in'}$. We also pick randomly the values $\widetilde{\mathbb{A}}_0, \dots, \widetilde{\mathbb{A}}_{n'-1} \in \mathcal{R}_q^m$. Then we can compute: $\widetilde{\mathbb{A}}_{n'} = \mathbb{U} = \left(\widetilde{\mathtt{C}}_{\tilde{\mathbf{f}}} - \sum_{j=0}^{n'-1} x^j \widetilde{\mathbb{A}}_j \right) \cdot x^{-n'} = \mathtt{Com}_{\mathbf{A}} \left(\tilde{\mathbf{f}}(x) \cdot x^{-n'}, \tilde{\mathbf{r}}_{\mathbf{f}} \cdot x^{-n'} \right) - \sum_{j=0}^{n'-1} x^{j-1} \widetilde{\mathbb{A}}_j$. If the protocol does not abort, then it holds that the real and the simulated values are indistinguishable under Theorem 1.

C.2 Proof of Theorem 4.3

Proof. To prove the theorem we prove the three properties:
Perfect Completeness: It follows from the underlying polynomial evaluation protocol from Section 3.

3**-Special Soundness:** For the same reason as in the $\mathit{\Pi}_{PEv}$ protocol, the challenge space is restricted to 3 challenges $\{X^{-1}, 1, X\}$ It means that we con obtain 3 accepting transcripts for the same message but different challenges x. Pick 3 challenges x_1, x_2, x_3. Because of linear independence of $l_0(X), \ldots, l_n(X)$, the columns of the following matrix are independent and therefore the matrix itself is invertible. So we can obtain any unit vector $(0, \ldots, 1, \ldots, 0)$ by taking an appropriate linear combination of the Vandermonde matrix $\mathbf{L} = \begin{pmatrix} l_0(x_1) \; l_1(x_1) \; l_2(x_1) \\ l_0(x_2) \; l_1(x_2) \; l_2(x_2) \\ l_0(x_3) \; l_1(x_3) \; l_2(x_3) \end{pmatrix}$. By multiplying the verification equation in steps 16 and 17 of $\mathit{\Pi}_{Batch}$ by $\mathbf{L}$ from both sides, we can extract the openings $\tilde{\mathbf{a}}_{j,i}, \hat{\mathbf{r}}_j$ for $j \in [0.n']$. Exploiting the special soundness of the underlying polynomial commitment protocol, we can extract the coefficients of the polynomials $\mathbf{P}_i^*$ and $\mathbf{Q}_i^*$ with their corresponding randomness as showed in the proof of soundness of Theorem 3.6. Having these results, we can apply them to the equation $\mathbf{P}_i^*(X) \cdot l_0(X) = \mathbf{P}(\hat{\mathbf{a}}_i)$. This equation holds for all 3 challenges $x_\iota, \iota \in \{1, 2, 3\}$, i.e. $\mathbf{P}_i^*(x_\iota) \cdot l_0(x_\iota) = \mathbf{P}(\hat{\mathbf{a}}_i)$. Using the same technique as in the previous proof of polynomial evaluation protocol we can extract the witness. Due to the page limitation we omit the details of this proof and only provide a sketch of it.
Evaluating at an interpolation point y_j yields $\mathbf{P}(\mathbf{a}_{j,i}) = \mathbf{P}_i^*(y_j) l_0(y_j) = 0$. In order to extract $\hat{\mathbf{a}}_i$ for all $i \in [0, m' - 1]$, we obtain 3 equations using 3 challenges $x_\iota, \iota \in \{1, 2, 3\}$: $\hat{\mathbf{a}}_i = \sum_{j=0}^{n'} \tilde{\mathbf{a}}_{j,i} l_j(x_\iota) = \tilde{\mathbf{a}}_{0,i} l_0(x_\iota) + \ldots + \tilde{\mathbf{a}}_{n',i} l_{n'}(x_\iota)$.We use the following generalised Vandermonde matrix:

$$\widetilde{\mathbf{V}}_3 = \left[\begin{pmatrix} l_0(x_1) \; l_1(x_1) \; l_2(x_1) \\ l_0(x_2) \; l_1(x_2) \; l_2(x_2) \\ l_0(x_3) \; l_1(x_3) \; l_2(x_3) \end{pmatrix} \cdots \begin{pmatrix} l_{n'-3}(x_1) \; l_{n'-2}(x_1) \; l_{n'-1}(x_1) \\ l_{n'-3}(x_2) \; l_{n'-2}(x_2) \; l_{n'-1}(x_2) \\ l_{n'-3}(x_3) \; l_{n'-2}(x_3) \; l_{n'-1}(x_3) \end{pmatrix} \right]$$

whose determinant is equal to 6, as showed in the previous proof. The, the concatenation there equations of $\hat{\mathbf{a}}_i$ are equivalent to the following product: $\hat{\mathbf{a}}_i = \widetilde{\mathbf{V}}_3 \cdot (\tilde{a}_{0,i} \ldots \tilde{a}_{n',i})^{tr}$. It holds: $6\|l_j(X)\| \leq 6\sqrt{n}^{n'}$ Let $\tilde{\mathbf{a}}_i = (\tilde{a}_{0,i}, \ldots, \tilde{a}_{n',i})$. For all $i \in \{0, \ldots, m' - 1\}$ the following equation is satisfied: $\hat{\mathbf{a}}_i = \mathbf{L} \cdot \tilde{\mathbf{a}}_i = \widetilde{\mathbf{V}}_{n'+1}^{-1} \cdot \tilde{\mathbf{a}}_i$. Multiplying each $\hat{\mathbf{a}}_i$ by $6\mathbf{L}^{-1}$, where $\mathbf{L}^{-1}$ is equivalent to the Vandermonde matrix, i.e $\mathbf{L}^{-1} = \widetilde{\mathbf{V}}_3$, we can extract the vectors $6\tilde{\mathbf{a}}_i$, for all $i \in [0, m' - 1]$. Since the entries of $\widetilde{\mathbf{V}}_3$ are monomials, it follows, that $\widetilde{\mathbf{V}}_3$ is small as required. Similarly, we can extract $\mathbf{r}_i$ from $\hat{\mathbf{s}}_i = \sum_{j=0}^{n'} \mathbf{s}_{j,i} l_j(X)$ using the 3 challenges $x_\iota, \iota \in \{1, 2, 3\}$: $\hat{\mathbf{s}}_i = \sum_{j=0}^{n'} \tilde{\mathbf{s}}_{j,i} l_j(x_\iota) = \tilde{\mathbf{s}}_{0,i} l_0(x_\iota) + \ldots + \tilde{\mathbf{s}}_{n',i} l_{n'}(x_\iota)$. For all $i \in \{0, \ldots, m' - 1\}$

the following equation holds $\hat{\mathbf{s}}_i = \mathbf{L} \cdot \tilde{\mathbf{s}}_i = \mathbf{V}_3^{-1} \cdot \tilde{\mathbf{s}}_i$. Multiplying each $\hat{\mathbf{s}}_i$ by $6\mathbf{L}^{-1}$, we can extract the vectors $6\tilde{\mathbf{s}}_i$ with the following bound: $6\|\tilde{\mathbf{s}}_i\| \leq 6\sqrt{mn}\mathcal{B}$.

Spezial Honest-Verifier-Zero-Knowledge: In order to simulate the protocol, we pick $y_1, \ldots, y_{n'}$ for the prover. We select $\hat{\mathbf{a}}_i \leftarrow_{\$} \mathfrak{D}_{\sigma}^{l_a m'}$, $\hat{\mathbf{r}} \leftarrow_{\$} \mathfrak{D}_{\sigma_r}^{mn}$, $\mathtt{D}_0, \ldots, \mathtt{D}_{n'} \leftarrow_{\$} \mathfrak{D}_{\sigma_D}^{n}$ and simulate the polynomial commitments as in the SHVZK proof of Theorem 3.6. By the perfect SHVZK property of polynomial commitment protocol, it follows that the simulation is identical to the real values.

```
Prover                                                      Verifier
Inputs: $\left(\mathtt{C}_v, \mathtt{C}_w, v, w, \mathbf{r}_v, \mathbf{r}_w, f\right)$      Inputs: $(\mathtt{C}_v, \mathtt{C}_w)$

1 :  Set $f(X) = \sum_{i=0}^{m'-1} \sum_{j=0}^{n'-1} a_{j,i} X^{in'+j}$,
       $q(X) = \sum_{i'=0}^{\mu-1} \sum_{j'=0}^{\nu-1} t_{j',i'} X^{i'\nu+j'}$
2 :  Compute $\mathcal{A}$ as in (2) and $\mathcal{T}$ correspondingly
$\forall i \in [0, m'-1], j \in [0, n'-1], i' \in [0, \mu-1], j' \in [0, \nu-1]$:
3 :  Sample $u_{j,i} \leftarrow\!\$\, \mathbb{Z}_q$, and $s_{j',i'} \leftarrow\!\$\, \mathbb{Z}_q$
4 :  Compute: $\widetilde{\mathcal{A}} = (\tilde{a}_{j,i})_{j,i}, \widetilde{\mathcal{T}} = (\tilde{t}_{j',i'})_{j',i'}$
5 :  $\tilde{\mathbf{a}}_j = (\tilde{a}_{j,0}, \ldots, \tilde{a}_{j,m'-1}), \tilde{\mathbf{t}}_{j'} = (\tilde{t}_{j',0}, \ldots, \tilde{t}_{j',\mu-1})$
6 :  Sample $\mathbf{r}_{a,j}, \mathbf{r}_{t,j'}, \mathbf{r}_u, \mathbf{r}_t \leftarrow\!\$\, \mathfrak{D}^{mn}_{12\mathcal{B}\sqrt{mn}}$
7 :  $\widetilde{\mathbb{A}}_j = \mathtt{Com}_{\mathbf{A}}(\tilde{\mathbf{a}}_j, \mathbf{r}_{a,j}),\ \widetilde{\mathbb{T}}_{j'} = \mathtt{Com}_{\mathbf{A}}(\tilde{\mathbf{t}}_{j'}, \mathbf{r}_{t,j'})$
       $\mathbb{U} = \mathtt{Com}_{\mathbf{A}}(\mathbf{u}, \mathbf{r}_u),\ \mathbb{S} = \mathtt{Com}_{\mathbf{A}}(\mathbf{s}, \mathbf{r}_s)$
8 :  Pick $\mathtt{m}_v, \mathtt{m}_w \leftarrow\!\$\, \mathfrak{D}^{n}_{12\beta\sqrt{n}},\ \mathbf{r}_{\mathtt{m}_v}, \mathbf{r}_{\mathtt{m}_w}, \in \mathfrak{D}^{mn}_{12\beta\sqrt{mn}}$
9 :  $\mathtt{C}_{\mathtt{m}_v} := \mathtt{Com}_{\mathbf{A}}(\mathtt{m}_v; \mathbf{r}_{\mathtt{m}_v}),\ \mathtt{C}_{\mathtt{m}_w} := \mathtt{Com}_{\mathbf{A}}(\mathtt{m}_w; \mathbf{r}_{\mathtt{m}_w})$
10 :  Set $\mathsf{pc} = \{\{\widetilde{\mathbb{A}}_j\}_j, \{\widetilde{\mathbb{T}}_{j'}\}_{j'}, \mathbb{U}, \mathbb{S}\}$, $\mathsf{c} = \{\mathtt{C}_{\mathtt{m}_v}, \mathtt{C}_{\mathtt{m}_w}\}$
$\mathsf{st} = \{f(X), q(X), \{\mathbf{r}_{a,j}\}_j, \{\mathbf{r}_{t,j'}\}_{j'}, \mathbf{r}_v, \mathbf{r}_w, \mathbf{r}_{\mathtt{m}_v}, \mathbf{r}_{\mathtt{m}_w}\}$
                                                $\xrightarrow{\mathsf{pc}, \mathsf{c}}$
                                                            $x \leftarrow\!\$\, \mathcal{CH}$
                                                $\xleftarrow{x}$
11 :  $\tilde{f}(x) = (1, x, \ldots, x^{n'-1}, x^{n'}) \cdot \widetilde{\mathcal{A}}$,
       $\tilde{q}(x) = (1, x, \ldots, x^{\nu-1}, x^{\nu}) \cdot \widetilde{\mathcal{T}}$
12 :  $\tilde{\mathbf{r}}_{\mathbf{f}} = \sum_{j=0}^{n'-1} \mathbf{r}_{a,j} x^j + \mathbf{r}_u x^{n'}, \tilde{\mathbf{r}}_{\mathbf{q}} = \sum_{j'=0}^{\nu-1} \mathbf{r}_{t,j'} x^{j'} + \mathbf{r}_s x^{\nu}$
13 :  $\xi_{\mathtt{m}_v} = \mathtt{m}_v + x \cdot v,\ \xi_{\mathtt{m}_w} = \mathtt{m}_w + x \cdot w$
14 :  $\mathbf{r}_{\xi,\mathtt{m}_v} = x \cdot \mathbf{r}_v + \mathbf{r}_{\mathtt{m}_v},\ \mathbf{r}_{\xi,\mathtt{m}_w} = x \cdot \mathbf{r}_w + \mathbf{r}_{\mathtt{m}_w}$,
       $\mathbf{r}_\gamma = \tilde{\mathbf{r}}_{\mathbf{f}} - \tilde{\mathbf{r}}_{\mathbf{q}} v - \mathbf{r}_w$
15 :  $\mathtt{C}_{0,\tilde{\mathbf{f}}} = \mathtt{Com}_{\mathbf{A}}(\mathbf{0}, \tilde{\mathbf{r}}_{\mathbf{f}}), \mathtt{C}_{\mathbf{0},\tilde{\mathbf{q}}} = \mathtt{Com}_{\mathbf{A}}(\mathbf{0}, \tilde{\mathbf{r}}_{\mathbf{q}})$
16 :  $\mathsf{re} := \{\mathtt{C}_{\tilde{\mathbf{f}}}, \mathtt{C}_{\tilde{\mathbf{q}}}, \mathtt{C}_{\mathbf{0},\tilde{\mathbf{f}}}, \mathtt{C}_{\mathbf{0},\tilde{\mathbf{q}}}, \xi_{\mathtt{m}_v}, \xi_{\mathtt{m}_w}, \mathbf{r}_{\xi,\mathtt{m}_v}, \mathbf{r}_{\xi,\mathtt{m}_w}, \mathbf{r}_\gamma\}$
17 :  Abort with prob. $(1-\rho)$, $\rho = \max_{i \in \{1,2,3\}}(\rho_i)$
                                                $\xrightarrow{\mathsf{re}}$
                                                            Check the following steps:
                                                            18 : $\max\{\|\mathbf{r}_{\xi,\mathtt{m}_v}\|, \|\mathbf{r}_{\xi,\mathtt{m}_w}\|\} \leq 12\mathcal{B}\sqrt{m'n}$
                                                            19 : $\|\tilde{\mathbf{r}}_{\mathbf{f}}\| \leq \mathcal{B}m'\sqrt{n'm},\ \|\tilde{\mathbf{r}}_{\mathbf{q}}\| \leq \mathcal{B}\mu\sqrt{\nu m}$
                                                            20 : $\mathtt{Com}_{\mathbf{A}}(\xi_{\mathtt{m}_v}, \mathbf{r}_{\xi,\mathtt{m}_v}) = x \cdot \mathtt{C}_v + \mathtt{C}_{\mathtt{m}_v}, \mathtt{Com}_{\mathbf{A}}(\xi_{\mathtt{m}_w}, \mathbf{r}_{\xi,\mathtt{m}_w}) = x \cdot \mathtt{C}_w + \mathtt{C}_{\mathtt{m}_w}$,
                                                                   $\mathtt{Com}_{\mathbf{A}}(\mathbf{0}, \tilde{\mathbf{r}}_{\mathbf{f}}) = \mathbf{A}\tilde{\mathbf{r}}_{\mathbf{f}},\ \mathtt{Com}_{\mathbf{A}}(\mathbf{0}, \tilde{\mathbf{r}}_{\mathbf{q}}) = \mathbf{A}\tilde{\mathbf{r}}_{\mathbf{q}}$
                                                            21 :  $\|\hat{\mathbf{r}}\| \leq \sqrt{n'}\beta,\ \|\hat{\mathbf{s}}\| \leq \sqrt{n'}\beta$
                                                            22 : $\sum_{j=0}^{n'-1} x^j \tilde{\mathbb{A}}_j + x^{n'} \mathbb{U} = \widetilde{\mathtt{C}}_{\tilde{\mathbf{f}}}; \sum_{j'=0}^{\nu-1} x^{j'} \widetilde{\mathbb{T}}_{j'} + x^{\nu} \mathbb{S} = \widetilde{\mathtt{C}}_{\tilde{\mathbf{q}}}$
                                                                   $\sum_{i=0}^{m'-1} \mathtt{C}_i(\tilde{\mathbf{f}}, \tilde{\mathbf{r}}_{\mathbf{f}}) \cdot x^{n'i} =$
                                                            $\sum_{i'=0}^{\mu-1} \mathtt{C}_{i'}(\tilde{\mathbf{q}}, \tilde{\mathbf{r}}_{\mathbf{q}}) \cdot x^{\nu i'} \cdot (x - \mathtt{C}_v) + \mathtt{C}_w - \mathtt{Com}_{\mathbf{A}}(0, \mathbf{r}_\gamma)$.
```

Fig. 2. Polynomial Evaluation Protocol $\mathit{\Pi}_{PEv}$

Prover

Inputs: $\forall i \in [0, m'-1], j \in [0, n']$

$(\mathbf{P}, \mathbf{Q}(\tilde{\mathbf{a}}_{j,i}, \mathbf{b}_{j,i}), \{\mathbf{b}_{j,i}\}_{i,j}, \{l_j(X)\}_j, \{\mathtt{C}_j\}_j, \{\tilde{\mathbf{a}}_{j,i}\}_{i,j}, \{\mathbf{r}_j\}_j)$

Verifier

Inputs: $\forall i \in [0, m'-1], j \in [0, n']$

$(\mathbf{P}, \mathbf{Q}, \{\mathbf{b}_{j,i}\}_{i,j}, \{\mathtt{C}_j, l_j(X)\}_j)$

$\forall j \in [0, n'], i \in [0, m'-1]$:

1 : $\mathbf{s}_j \leftarrow_{\$} \mathcal{U}(\mathcal{R}_q^m)$

2 : $\mathtt{D}_j := \mathtt{Com}_{\mathbf{A}}\left(\tilde{\mathbf{a}}_{j,0}, \ldots, \tilde{\mathbf{a}}_{j,m'-1}; \mathbf{s}_j\right)$

3 : $\hat{\mathbf{a}}_i = \sum_{j=0}^{n'} \tilde{\mathbf{a}}_{j,i} l_j(X), \hat{\mathbf{b}}_i = \sum_{j=0}^{n'-1} \mathbf{b}_{j,i} l_j(X)$

4 : $\frac{\mathbf{P}(\hat{\mathbf{a}}_j)}{l_0(X)} = \frac{\mathbf{P}\left(\sum_{j=0}^{n'} \tilde{\mathbf{a}}_{j,i} l_j(X)\right)}{\prod_{j=0}^{n'} (X - y_j)} = \mathbf{P}_i^*(X)$

5 : $\mathbf{c}_0, \ldots, \mathbf{c}_{m'-1} \in \mathcal{R}_q^{l_Q} \times \cdots \times \mathcal{R}_q^{l_Q}, \mathbf{r}_c \in \mathcal{R}_q^{m \times (l_a + l_b)}$

6 : $\mathtt{C}_0 = \mathtt{Com}_{\mathbf{A}}(\mathbf{c}_1, \ldots, \mathbf{c}_{m'-1}; \mathbf{r}_c)$

7 : $\mathbf{Q}_i^*(X) = \mathbf{c}_i + \frac{\sum_{j=0}^{n'} \mathbf{Q}(\tilde{\mathbf{a}}_{j,i}, \mathbf{b}_{j,i}) l_j(X) - \mathbf{Q}(\hat{\mathbf{a}}_i, \hat{\mathbf{b}}_i)}{l_0(X)}$

8 : Run $\mathtt{Prover}(\mathbf{P}_i^*)$ and $\mathtt{Prover}(\mathbf{Q}_i^*)$ of $\mathit{\Pi}_{PEv}$

9 : $\forall \chi \in \{\mathbf{P}^*, \mathbf{Q}^*\}, \chi_i \in \{\mathbf{P}_i^*, \mathbf{Q}_i^*\}$ set:

$\mathbf{msg}_{\chi,1} = \left\{\mathbb{A}_{\chi_i,\mu}, \mathbb{T}_{\chi_i,\mu'}, \mathbb{U}_{\chi_i}, \mathbb{S}_{\chi_i}, \mathtt{C}_{\chi_i,v}, \mathtt{C}_{\chi_i,w}\right\}_i$

$$\xrightarrow{\{\mathtt{D}_j\}_j, \{\mathbf{msg}_{\chi,1}\}_\chi}$$

$x \leftarrow_{\$} \mathcal{CH}$

$$\xleftarrow{x}$$

10 : From $\mathit{\Pi}_{PEv}$: $\tilde{\mathbf{f}}_{\chi_i}(x), \tilde{\mathbf{h}}_{\chi_i}(x)$,

$\mathtt{F}_{\chi_i} := \mathtt{Com}_{\mathbf{A}}\left(\tilde{\mathbf{f}}_{\chi_i}(x), \tilde{\mathbf{r}}_{\mathbf{f},\chi_i}(x)\right), \mathtt{G}_{\chi_i} := \mathtt{Com}_{\mathbf{A}}\left(\tilde{\mathbf{h}}_{\chi_i}(x), \tilde{\mathbf{r}}_{\mathbf{h},\chi_i}(x)\right)$,

$\mathtt{C}_{0,\chi_i,\tilde{\mathbf{h}}} := \mathtt{Com}_{\mathbf{A}}(0, \tilde{\mathbf{r}}_{\mathbf{h},\chi_i}(x)), \mathtt{C}_{0,\chi_i,\tilde{\mathbf{f}}} := \mathtt{Com}_{\mathbf{A}}\left(0, \tilde{\mathbf{r}}_{\mathbf{f},\chi_i}^{(i)}(x)\right)$

11 : Run **Response** of $\mathit{\Pi}_{PEv}(\chi)$ output

$\mathbf{msg}_{\chi,2} = \{F_{\chi_i}, \mathtt{C}_{0,\chi_i}, \xi_{\mathtt{m}_v,\chi_i}, \xi_{\mathtt{m}_w,\chi_i}, \mathbf{r}_{\xi,\mathtt{m}_v,\chi_i}, \mathbf{r}_{\xi,\mathtt{m}_w,\chi_i}\}_i$

12 : $\hat{\mathbf{a}}_i(x) = \sum_{j=0}^{n'} \tilde{\mathbf{a}}_{j,i} l_j(x)$,

$\hat{\mathbf{r}}(x) = \sum_{j=0}^{n'} \mathbf{r}_j l_j(x)$, $\hat{\mathbf{s}}(x) = \sum_{j=0}^{n'} \mathbf{s}_j l_j(x)$

13 : Abort with prob. $1 - \tilde{\rho}$

$$\xrightarrow{\{\hat{\mathbf{a}}_i\}_i, \hat{\mathbf{r}}, \hat{\mathbf{s}}, \{\mathbf{msg}_{\chi,2}\}_\chi}$$

14 : Run **Verification** of $\mathit{\Pi}_{PEv}$ on

$\mathbf{msg}_{\chi,1}, \mathbf{msg}_{\chi,2}$

15 : $\|\hat{\mathbf{r}}\| \leq \sqrt{n'}\beta$, $\|\hat{\mathbf{s}}\| \leq \sqrt{n'}\beta$

16 : $\mathbf{P}_i^*(X) \cdot l_0(X) = \mathbf{P}(\hat{\mathbf{a}}_i)$.

17 : $\mathtt{Com}_{\mathbf{A}}\left(\{\mathbf{Q}_i^*(x) l_0(x) + \mathbf{Q}(\hat{\mathbf{a}}_i, \hat{\mathbf{b}}_i)\}_i, \hat{\mathbf{r}}\right)$

$= \sum_{j=0}^{m'-1} \mathtt{C}_j l_j(x)$

18 : $\mathtt{Com}_{\mathbf{A}}(\hat{\mathbf{a}}_0, \ldots, \hat{\mathbf{a}}_{m'-1}, \hat{\mathbf{s}}) = \sum_{j=0}^{n'} \mathtt{D}_j l_j(x)$

Fig. 3. $\mathit{\Pi}_{Batch}$-Protocol