Dory: Efficient, Transparent arguments for Generalised Inner Products and Polynomial Commitments

Jonathan Lee*
Microsoft Research, Nanotronics Imaging⋆

Abstract

This paper presents Dory, a transparent setup, public-coin interactive argument for proving correctness of an inner-pairing product between committed vectors of elements of the two source groups. For an inner product of length n, proofs are $6\log n$ target group elements, 1 element of each source group and 3 scalars. Verifier work is dominated by an $O(\log n)$ multi-exponentiation in the target group. Security is reduced to the symmetric external Diffie Hellman assumption in the standard model. We also show an argument reducing a batch of two such instances to one, requiring $O(n^{1/2})$ work on the Prover and $O(1)$ communication.

We apply Dory to build a multivariate polynomial commitment scheme via the Fiat-Shamir transform. For n the product of one plus the degree in each variable, Prover work to compute a commitment is dominated by a multi-exponentiation in one source group of size n. Prover work to show that a commitment to an evaluation is correct is $O(n^{log(8)/\log 25})$ in general and $O(n^{1/2})$ for univariate or multilinear polynomials, whilst communication complexity and Verifier work are both $O(\log n)$. Using batching, the Verifier can validate ℓ polynomial evaluations for polynomials of size at most n with $O(\ell + \log n)$ group operations and $O(\ell \log n)$ field operations.

1 Introduction

Zero-knowledge succinct arguments of knowledge (zkSNARKs) for the satisfiability of Rank-1 Constraint Systems are the subject of ongoing research. One general strategy for constructing these arguments uses purely information-theoretic arguments to provide a reduction to the evaluation of (possibly multi-variate) polynomials, and use some auxiliary argument between a Prover $\mathcal{P}$ and Verifier $\mathcal{V}$ with sub-linear verification time to show that these are correct. These auxiliary arguments are variously inner-product arguments, or the more restricted *polynomial commitments*, introduced by Kate [29] in the univariate context and in the multivariate context by in [33].

Spartan [34] makes the independence of the information-theoretic argument and these maxillary arguments explicit, and provides an extensive overview of the history and details of prior works, and key practical considerations relating to the uniformity of the computation to verify. Non-exhaustively, Bulletproofs [15] use inner-product arguments and Hyrax [37] utilize polynomial commitments, both based on work of Bootle et al. [12]; Spartan [34] optimizes this approach further. Virgo [38] and Fractal [19] use Interactive Oracle Proofs based on Reed-Solomon codes (RS-IOP) to prove that a polynomial is of bounded degree [6]. Supersonic [17] makes use of groups of unknown order to construct

*jlee@nanotronics.co
⋆Current Affiliation: Nanotronics Imaging; work done primarily while at MSR

Diophantine ARguments of Knowledge (DARK) proofs for polynomial evaluations over fields. Other works rely on some trusted setup, which allows the use of other commitment schemes. For example PLONK [21] makes use of Kate [29] commitments. In all cases these interactive arguments are then compiled to non-interactive arguments in the random-oracle model.

This paper introduces a new argument for generalized inner products without trusted setup, inspired by Bootle et al. [12] but applying new techniques to achieve a logarithmic Verifier complexity. This argument can be applied to give polynomial commitments for arbitrary numbers of variables, using two-tiered homomorphic commitments of Groth [24] applied to matrix commitment strategy of [37]. This approach is also followed in Bünz et al. [18] for univariate and bivariate polynomials. For transparent polynomial commitment schemes, the key operations are for $\mathcal{P}$ and $\mathcal{V}$ to generate public parameters, for $\mathcal{P}$ to commit to a polynomial and transmit that commitment to $\mathcal{V}$, and for $\mathcal{P}, \mathcal{V}$ to compute, transmit and verify a proof of evaluation of the polynomial. We present some asymptotic comparisons of transparent polynomial commitment schemes in Figure 1.

	communication complexity		time complexity			
	Commit	Eval	Gen	Commit	Eval ($\mathcal{P}$)	Eval ($\mathcal{V}$)
Supersonic [17]	$1 \lvert\mathbb{G}_U\rvert$	$\log n \lvert\mathbb{G}_U\rvert$	$n \log n\ \mathbb{G}_U$	$n\ \mathbb{G}_U$	$n \log n\ \mathbb{G}_U$	$\log n\ \mathbb{G}_U$
RS-IOP [19]	$1 \lvert\mathbb{H}\rvert$	$\log^2 n \lvert\mathbb{H}\rvert$	1	$n \log n\ \mathbb{H}$	$n \log n\ \mathbb{H}$	$\log^2 n\ \mathbb{H}$
Hyrax [37]	$n^{1/2} \lvert\mathbb{G}\rvert$	$\log n \lvert\mathbb{G}\rvert$	$n^{1/2}\mathbb{H}$	$n\ \mathbb{G}$	$n^{1/2}\ \mathbb{G}$	$n^{1/2}\ \mathbb{G}$
IPP-PC [18]	$1 \lvert\mathbb{G}_T\rvert$	$\log n \lvert\mathbb{G}_T\rvert$	$n^{1/2}\mathbb{H}$	$n\ \mathbb{G}_1$	$n^{1/2}P$	$n^{1/2}\ \mathbb{G}_2$
This work	$1 \lvert\mathbb{G}_T\rvert$	$\log n \lvert\mathbb{G}_T\rvert$	$n^{1/2}P$	$n\ \mathbb{G}_1$	$n^{1/2}P$	$\log n\ \mathbb{G}_T$

FIGURE 1—Asymptotic comparisons of transparent polynomial commitments, neglecting Pippenger-type savings in groups. $\mathbb{G}_U$ a group of unknown order. $\mathbb{H}$ denotes a hash function. $\mathbb{G}$ denotes a group. $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ denote the two source groups and the target group of a pairing P.

To allow a more concrete comparison at the 128-bit security level, we note the size and typical operation times for these systems in Figure 2. Concretely, we take Curve25519 [8] as an example group, as implemented by curve25519-dalek [31]. We use the BLS12–381 [13] curve as implemented by RELIC [3] as an example of a group with a pairing, enhanced to apply torus-based pairing compression [32] for serialization. We use an imaginary class group [28] as an example group of unknown order with trustless setup. At the 128-bit security level, a ~ 6656 bit discriminant is required [20], concretely, we fix the discriminant $\Delta = -(2^{6656} - 26745)$, as implemented by ANTIC [26]. Following Fractal [19], we measure the Blake2b hash function, hashing 64 byte messages to 32 byte digests, as implemented by rust-crypto [1].

1.1 Limitations of prior approaches

At a high level, each of the prior approaches to transparent polynomial commitments have practical problems. Schemes derived from [12] have $\mathcal{V}$'s computation to Open an evaluation being $n^{O(1)}$. Fundamentally, this is because these schemes commit to a $O(n)$ size matrix by committing to the rows and then opening a commitment to some linear combination. This necessarily bounds $\mathcal{V}$'s work below as $\Omega(n^{1/2})$. Hyrax saturates this

Setting	Implementation		Size (bytes)	Time (μs)
Group of Unknown Order	ANTIC-QFB	$\lvert\mathbb{G}_U\rvert$	832	38000
Hashing	rust-crypto	$\lvert\mathbb{H}\rvert$	32	0.180
Group	curve25519-dalek	$\lvert\mathbb{G}\rvert$	32	45
Group with Pairing	RELIC	$\lvert\mathbb{G}_1\rvert$	48	220
		$\lvert\mathbb{G}_2\rvert$	96	490
		$\lvert\mathbb{G}_T\rvert$	192	820
		P		1600

FIGURE 2—Illustrative micro-benchmarks on a single 3.2GHz core (Intel Xeon E5–1660 v4). For groups we give the serialized size in bytes of a group element, and the time taken to multiply a random point by a 256-bit scalar. For $\lvert\mathbb{H}\rvert$ denotes hashing of a 512-bit message to a 256-bit digest.

bound with small concrete constants, but for large n this can be challenging, and in applications where a commitment must be sent frequently the $O(n^{1/2})$ commitment size can be a problem.

The soundness error of Reed-Solomon based interactive proofs seems to be substantially bounded away from 0, and the *proven* soundness is lower still. So whilst asymptotically proof sizes and Verifier times are $O(\log^2 n)$ the implied constants are quite large. For example, the implementation of Fractal [19] in libiop [7] has to run the underlying proof ~ 500 times to achieve 128-bit security without heuristic assumptions.

Groups of unknown order can be constructed in a trustless way from the class groups of quadratic number fields, or essentially analogously from the Jacobians of higher genus curves [20]. Groups of unknown order have a long history [11, 28, 30], but their use for polynomial commitments is quite new [17]. Unfortunately, general sub-exponential attacks on the order are known [10], and the required security parameters for transparent setup have recently grown substantially [20]. In the particular case of Supersonic [17], even with Pippenger-type acceleration the $\mathcal{P}$ must perform $O(n\lambda)$ group operations, and generating parameters takes $O(n\lambda \log n)$ group operations. As can be seen in Figure 2, this is unlikely to be efficient in practice.

Finally, if transparent setup is given up then Kate commitments [29] and their multivariate generalization [33] are available, generally requiring $O(n)$ operations in $\mathbb{G}_1$ for $\mathcal{P}$, $O(1)$ commitment sizes and a $\mathcal{V}$ time linear in the number of variables. In addition to requiring a trusted setup phase, these systems require unprovable knowledge-of-exponent type assumptions for their security, which is undesirable.

1.2 Core techniques of Dory

Our approach builds on techniques of Bootle et al. [12] and recent work of Bünz et al. [18]. These provide arguments for inner products between vectors of scalars and group elements or generalised products between the source groups of a pairing, where the inputs are all optionally committed. In those works, the core technique is to split the inner product of two vectors u, v of length m into pieces u_L, u_R, v_L, v_R of size $m/2$, have $\mathcal{P}$ supply additional data and combine with some $\mathcal{V}$ challenge. This essentially follows the identity:

$$\forall a \neq 0 : \langle u_L, v_L\rangle + \langle u_R, v_R\rangle = \langle a u_L + u_R, a^{-1} v_L + v_R\rangle - a\langle u_L, v_R\rangle - a^{-1}\langle u_R, v_L\rangle$$

When applied to committed vectors, there is an additional challenge: the Verifier possesses a commitment to a length m vector (WLOG u), but must compute a commitment to the vector $u' = au_L + u_R$. To do this, these works use Pedersen-like commitments, which are bilinear in the committed elements and generators. Suppose u is committed with generators (g,h), and split g into two $m/2$ length pieces g_L, g_R as for u, v. Then reusing the above identity, $\mathcal{V}$ can pass from a commitment to u with generators (g,h) to a commitment to u' with generators $(a^{-1}g_L + g_R, h)$. Similar arguments apply to the commitment to v.

We say a vector equal to the Kronecker product of vectors of length 2 has *multiplicative structure*. Applying the above argument recursively, $\mathcal{V}$ and $\mathcal{P}$ compute vectors X, Y with multiplicative structure and have some claim about: $\left(\sum_i X_i u_j\right) \times \left(\sum_j Y_j v_j\right)$, where one or both of the two terms are committed. If a vector (WLOG v) was originally committed with to generators (g,h), then $\mathcal{V}$ holds a commitment to $\sum_j Y_j v_j$ with generators $(\sum_i X_i g_i, h)$. Similarly if u is committed with generators (g',h') then $\mathcal{V}$ must compute $\sum_j Y_j g'_j$. In prior works, $\mathcal{V}$ directly computes these $O(m)$-sized multi-exponentiations; a simple sigma proof shows the required product relation.

A poly-logarithmic $\mathcal{V}$: In the bilinear setting, Abe et al. [2] provide linearly homomorphic commitments to vectors of group elements. Since X has multiplicative structure, $X = (\ell X' || r X')$. So instead of computing $\sum_i X_i g_i$ directly, $\mathcal{V}$ might precompute commitments to g_L and g_R, and compute a commitment to $G = \ell g_L + r g_R$ Then $\sum_i X_i g_i = \sum_i X'_i G_i$, and so once X' is known $\mathcal{P}$ and $\mathcal{V}$ could run an $m/2$ sized inner-product argument to convince $\mathcal{V}$ of the value of $\sum_i X_i g_i$.

Naively, this only yields a constant factor improvement, as X' is an $O(m)$-length vector of scalars. However, X' has multiplicative structure, and crucially, the only computation that $\mathcal{V}$ performs with X' is to take its inner product with another vector with multiplicative structure derived from challenges in the $m/2$ sized inner product argument; the inner product of vectors with multiplicative structure can be computed in logarithmic time (i.e. without instantiating the vectors explicitly).

So using the inner-product reductions derived from Bootle et. al. [12] as a black box, we have an $O(\log m)$ reduction from a length m inner product between a vector with multiplicative structure in $\mathbb{F}$ or a committed vector in $\mathbb{G}_1$ and a committed vector in $\mathbb{G}_2$ to an $m/2$ length inner product between a public vector with multiplicative structure in $\mathbb{F}$ and a constant vector in $\mathbb{G}_2$ (and a similar reduction with $\mathbb{G}_1, \mathbb{G}_2$ swapped).

Applying this recursively gives arguments for length m inner products between committed vectors, or between committed vectors and vectors of scalars with multiplicative structure, with $O(\log^2 m)$ proof sizes and verification time. $\mathcal{P}$ and $\mathcal{V}$ would share generators for commitments of every power-of-2 length less than m in both $\mathbb{G}_1$ and $\mathbb{G}_2$, and commitments to the left and right halves of each set of generators (using the generators for half length commitments). These public parameters can be computed transparently if hashing to $\mathbb{G}_1, \mathbb{G}_2$ is possible. This use of public parameters with structure but without trusted setup can be seen as analogous to the *computational commitments* used in Spartan [34], as we perform some linear-size computation *once* during setup to accelerate the online proof generation and verification.

A logarithmic $\mathcal{V}$: To reduce further we modify the inner-product reduction. In particular, note that the scalars ℓ, r that are used to convert the generators of length m to a vector of length $m/2$ are known after the first round of inner-product reductions. So after this first round, there are two high level tasks:

- prove an $m/2$-length inner product, where any committed vectors are now committed using generators depending on $\mathcal{V}$'s challenge
- prove that these modified generators are consistent with the pre-computed commitments in the public parameters, i.e. that some product of these modified generators with a fixed vector in the other group is correct.

These tasks are both $m/2$-length inner products, and we can combine multiple inner products of the same length with a $\mathcal{V}$ supplied challenge, as:

$$\forall b : \langle bu_1 + u_2, bv_1 + v_2 \rangle = b^2 \langle u_1, v_1 \rangle + \langle u_2, v_2 \rangle + b(\langle u_1, v_2 \rangle + \langle u_2, v_1 \rangle)$$

So $\mathcal{V}$ sends a challenge b, and combine these two claims into a single $m/2$-length inner product between vectors committed with fixed generators. Applying this recursively, we obtain a $2 \log m$ round argument, and $\mathcal{V}$'s computation is dominated by an $O(\log m)$ multi-exponentiation.

Polynomial commitments, batching: We then apply ideas from Hyrax [37, §6] and BMMV [18] to construct a polynomial commitment from a two-tiered homomorphic commitment to matrices. This ultimately reduces a dense univariate or multilinear polynomial with n coefficients to two inner products of size $O(n^{1/2})$, between vectors of scalars with multiplicative structure and vectors in $\mathbb{G}_1, \mathbb{G}_2$ respectively. Conveniently, in our context we can combine the proofs of these inner products, thereby saving an additional factor 2.

We apply ideas similar of those of Bowe et al. [14] to batch these arguments to reduce verification time further. In particular, we reduce the cost of evaluating each additional polynomial commitment to $O(1)$ group operations and $O(\log n)$ additional operations in the scalar field.

2 Preliminaries

2.1 Notation

We use $\otimes$ to denote the Kronecker product, sending an $m \times n$ matrix A and $p \times q$ matrix B to an $mp \times nq$ matrix built up of appended copies of B multiplied by scalars in A. For any vector v of even length we will denote the left and right halves of the vector by v_L and v_R respectively; more formally:

$$v_L = ((1,0) \otimes I_{n/2})v, \qquad v_R = ((0,1) \otimes I_{n/2})v.$$

For any two vectors v_1, v_2 we denote their concatenation by $(v_1||v_2)$.

We write $\leftarrow_\$ S$ for a uniformly random sample of S, with the understanding that this encodes no additional structure; concretely for groups $\mathbb{G}$ we assume that samples $g_i \leftarrow_\$ \mathbb{G}$ have unrelated logarithms. In the context of curves this is known but not entirely trivial [9, 27, 35, 36].

We write all groups *additively*, and assume we are given some method to sample Type III pairings [22] at a given security level. Then we are furnished with a prime field $\mathbb{F} = \mathbb{F}_p$, three groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of order p, a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, and generators $G_1 \in \mathbb{G}_1$, $G_2 \in \mathbb{G}_2$ such that $e(G_1, G_2)$ generates $\mathbb{G}_T$. Concretely, classes of *pairing-friendly* curves (e.g. Barreto-Lynn-Scott [4] or Barreto-Naehrig [5] curves) are believed to satisfy these properties.

We generally suppress the distinction between e and multiplication of $\mathbb{F}, \mathbb{G}_1, \mathbb{G}_2$ or $\mathbb{G}_T$ by elements of $\mathbb{F}$, writing all of these bilinear maps as multiplication; we will also use $\langle , \rangle$ to denote all of the generalized inner products:

$$\mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}, \qquad \mathbb{F}^n \times \mathbb{G}^n_{\{1,2,T\}} \to \mathbb{G}_{\{1,2,T\}}, \qquad \mathbb{G}^n_1 \times \mathbb{G}^n_2 \to \mathbb{G}_T,$$

given by $\langle a, b \rangle = \sum_{i=1}^n a_i b_i$ in each case.

We will initially present our arguments as requiring some precomputation depending on public, independent and uniformly random samples of $\mathbb{G}_1, \mathbb{G}_2$. These can be computed from a hash function as in [9, 27, 35, 36]. Ultimately these precomputed values are computed form part of the public parameters.

2.2 Computationally hard problems in Type III pairings

Since the pairing is of Type III, there are no efficiently computable morphisms between $\mathbb{G}_1, \mathbb{G}_2$, and the standard security assumption in this case is Symmetric eXternal Diffie-Hellman (SXDH) [2]:

Definition 2.1 (SXDH)**.** For $(\mathbb{F}_p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$ as above, the Decisional Diffie-Hellman (DDH) assumption holds for $(\mathbb{F}_p, \mathbb{G}_1, G_1)$ and $(\mathbb{F}_p, \mathbb{G}_2, G_2)$

Note that a DDH instance in $\mathbb{G}_1$ can be mapped to a DDH instance in $\mathbb{G}_T$ by the map $g \to e(g, G_2)$, so SXDH implies that DDH holds in $\mathbb{G}_T$. In any group, DDH implies DLOG, and so:

Lemma 2.1. *For $(\mathbb{F}_p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$ satisfying SXDH, for $\mathbb{G} \in \{\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T\}$ and polynomial n, given $B \leftarrow_\$ \mathbb{G}^n$ no non-uniform polynomial-time adversary can compute a non-trivial $A \in \mathbb{F}^n$ such that: $\langle A, B \rangle = 0$.*

SXDH also implies the more useful Double Pairing assumption and reverse Double Pairing assumption, which we combine:

Definition 2.2 (DBP)**.** For $(\mathbb{F}_p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$ as above, given $A_1, A_2 \leftarrow_\$ \mathbb{G}_1$ no non-uniform polynomial-time adversary can compute non-trivial $B_1, B_2 \in \mathbb{G}_2$ such that: $A_1 B_1 + A_2 B_2 = 0$. Similarly, given $A_1, A_2 \leftarrow_\$ \mathbb{G}_2$ no adversary can compute non-trivial $B_1, B_2 \in \mathbb{G}_1$ such that $B_1 A_1 + B_2 A_2 = 0$.

This in turn implies that:

Lemma 2.2. *For $(\mathbb{F}_p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$ as above and polynomial n, given $A \leftarrow_\$ \mathbb{G}^n_1$ no non-uniform polynomial-time adversary can compute a non-trivial $B \in \mathbb{G}^n_2$ such that: $\langle A, B \rangle = 0$. Similarly, given $A \leftarrow_\$ \mathbb{G}^n_2$, no adversary can compute non-trivial $B \in \mathbb{G}^n_1$ such that $\langle B, A \rangle = 0$.*

2.3 Succinct interactive arguments of knowledge

These definitions largely follow follow the presentation in [34]. Let $\mathcal{P}, \mathcal{V}$ be a pair of interactive PPT algorithms, and fix public parameters *pp* output by some algorithm Gen given λ a security parameter, such that $O(2^{-\lambda}) = \text{negl}(\lambda)$ is negligibly small.

Definition 2.3. A public-coin succinct interactive argument of knowledge for a language $\mathcal{L}$ is a protocol between $\mathcal{P}, \mathcal{V}$ with the following properties:

- **Completeness.** If $\mathbb{x} \in \mathcal{L}$, there exists a *witness* w such that for all $r \in \{0,1\}^*$, $\Pr\{\langle \mathcal{P}(pp, w), \mathcal{V}(pp, r)\rangle(\mathbb{x}) = 1\} \geq 1 - \text{negl}(\lambda)$.
- **Soundness.** For any non-satisfiable problem instance $\mathbb{x}$, any PPT Prover $\mathcal{P}^*$, and for all $w, r \in \{0,1\}^*$, $\Pr\{\langle \mathcal{P}^*(pp, w), \mathcal{V}(pp, r)\rangle(\mathbb{x}) = 1\} \leq \text{negl}(\lambda)$.
- **Knowledge soundness.** For any PPT adversary $\mathcal{A}$, there exists a PPT *extractor* $\mathcal{E}$ such that $\forall \mathbb{x} \in \mathcal{L}, \forall w, r \in \{0,1\}^*$, if $\Pr\{\langle \mathcal{A}(pp, w), \mathcal{V}(pp, r)\rangle(\mathbb{x}) = 1\} \geq \text{negl}(\lambda)$, then $\Pr\{\mathtt{Sat}_{\mathcal{L}}(\mathbb{x}, \mathcal{E}^{\mathcal{A}}(pp, \mathbb{x})) = 1\} \geq \text{negl}(\lambda)$.
- **Succinctness.** The total communication between $\mathcal{P}$ and $\mathcal{V}$ is sub-linear in the size of the NP statement $\mathbb{x} \in \mathcal{L}$.
- **Public coin.** $\mathcal{V}$'s messages are chosen uniformly at random.

We denote the *transcript* of the interaction of two PPTs $\mathcal{P}, \mathcal{V}$ with random tapes $z_{\mathcal{P}}, z_{\mathcal{V}} \in \{0,1\}^*$ on $\mathbb{x}$ by $tr\langle \mathcal{P}(z_{\mathcal{P}}), \mathcal{V}(z_{\mathcal{V}})\rangle(\mathbb{x})$

Definition 2.4. A succinct interactive argument of knowledge is *publicly verifiable* if there is a polynomial time algorithm Accept of the transcript t such that:
$\Pr\left(\mathsf{Accept}(tr\langle \mathcal{P}(z_{\mathcal{P}}), \mathcal{V}(z_{\mathcal{V}})\rangle(\mathbb{x}), \mathbb{x}) \neq \langle \mathcal{P}(z_{\mathcal{P}}), \mathcal{V}(z_{\mathcal{V}})\rangle(\mathbb{x})\right) = \text{negl}(\lambda)$.

Note that any public-coin succinct interactive argument of knowledge is necessarily publicly verifiable. Then from [37]:

Definition 2.5 (Witness-extended emulation [25])**.** An publicly verifiable interactive argument $(\mathsf{Gen}, \mathcal{P}, \mathcal{V})$ for $\mathcal{L}$ has witness-extended emulation if for all deterministic polynomial time programs $\mathcal{P}^*$ there exists an expected polynomial time emulator E such that for all non-uniform polynomial time adversaries A and all $z_{\mathcal{V}} \in \{0,1\}^*$, the following probabilities differ by at most $\text{negl}(\lambda)$: $\Pr\{pp \leftarrow \mathsf{Gen}(1^\lambda)\ ;\ (\mathbb{x}, z_{\mathcal{P}}) \leftarrow A(pp)\ ;\ t \leftarrow tr\langle \mathcal{P}^*(z_{\mathcal{P}}), \mathcal{V}(z_{\mathcal{V}})\rangle(\mathbb{x}) : \mathcal{A}(t, \mathbb{x}) = 1\}$ and $\Pr\{pp \leftarrow \mathsf{Gen}(1^\lambda)\ ;\ (\mathbb{x}, z_{\mathcal{P}}) \leftarrow A(pp)\ ;\ (t, w) \leftarrow E^{\mathcal{P}^*(z_{\mathcal{P}})}(\mathbb{x}) : \mathcal{A}(t, \mathbb{x}) = 1 \wedge (\mathsf{Accept}(t) = 1 \Rightarrow \mathtt{Sat}_{\mathcal{L}}(\mathbb{x}, w) = 1)\}$.

Note that witness-extended emulation implies *soundness* and *knowledge soundness*. In particular, we will use [37, Lemma 13] heavily:

Lemma 2.3. *Let $(\mathcal{P}, \mathcal{V})$ be a $(2\mu + 1)$-move interactive protocol. Let χ be a witness extraction algorithm extracting a witness from an $(w_1, \ldots, w_\mu)$-tree of accepting transcripts in probabilistic polynomial time with negligible failure probability, and that $\prod_i w_i$ is bounded above by some polynomial in λ the security parameter. Then $(\mathcal{P}, \mathcal{V})$ has witness-extended emulation.*

Remark 2.1. *The main advantage of Lemma 2.3 is that composing reductions between* $\mathbb{x} \in \mathcal{L}$ *and* $\mathbb{x}' \in \mathcal{L}'$ *is straightforward. In the reduction, one may assume that the argument showing* $\mathbb{x}' \in \mathcal{L}'$ *is already witness-extracted, so one has to extract a witness for* $\mathbb{x} \in \mathcal{L}$ *from a tree of accepting transcripts of the reduction where* $\mathcal{P}$ *additionally provides witnesses for* $\mathbb{x}' \in \mathcal{L}'$*. We will have logarithmic numbers of* $O(1)$ *round interactive protocols reducing from membership in one language to another, so the composed witness extraction algorithm has* $\mu = O(\log n)$*, all* $w_i = O(1)$ *and* $n = \lambda^{O(1)}$.

Definition 2.6. An interactive argument $(\mathsf{Gen}, \mathcal{P}, \mathcal{V})$ for $\mathcal{L}$ is Honest-Verifier Statistical Zero-Knowledge (HVSZK) if there exists a PPT algorithm S called the simulator, running in time polynomial in the length of its first input such that for every problem instance $\mathbb{x} \in \mathcal{L}$, $w \in \mathcal{R}_{\mathbb{x}}$, and $z \in \{0,1\}^*$, the following holds when the distinguishing gap is considered as a function of $|\mathbb{x}|$:

$$\mathit{View}(\langle \mathcal{P}(w), \mathcal{V}(z) \rangle(\mathbb{x})) \approx S(\mathbb{x}, z),$$

where $\mathit{View}(\langle \mathcal{P}(w), \mathcal{V}(z) \rangle(\mathbb{x}))$ denotes the distribution of the transcript of interaction between $\mathcal{P}$ and $\mathcal{V}$, and $\approx$ denotes that the statistical distance between the two distributions is negligible.

Remark 2.2. *As is standard, when compiled with the Fiat-Shamir transform, honest verifier zero knowledge, public-coin succinct interactive arguments are transformed into zkSNARKs. Standard techniques [23] can also be used to remove the honest-verifier constraint.*

2.4 Commitments

Our aim is to define polynomial commitments. As in [34], we work with the definitions from Bünz et al. [17] rather than directly with Kate et al. [29], as this permits interactive proofs for evaluations.

A *commitment scheme* for some space of messages $\mathcal{X}$ is a tuple of three protocols $(\mathsf{Gen}, \mathsf{Commit}, \mathsf{Open})$:

- $pp \leftarrow \mathsf{Gen}(1^\lambda)$: produces public parameters pp.
- $(\mathcal{C}; \mathcal{S}) \leftarrow \mathsf{Commit}(pp; x)$: takes as input some $x \in \mathcal{X}$; produces a public commitment $\mathcal{C}$ and a secret opening hint $\mathcal{S}$.
- $b \leftarrow \mathsf{Open}(pp, \mathcal{C}, x, \mathcal{S})$: verifies the opening of commitment $\mathcal{C}$ to $x \in \mathcal{X}$ with the opening hint $\mathcal{S}$; outputs $b \in \{0,1\}$.

In the sequel, our commitment schemes will generally have $\mathcal{S}$ sampled uniformly from some space. In this case it is often convenient to pass in $\mathcal{S}$, and we will write this $\mathsf{Commit}(pp; x, \mathcal{S})$.

Definition 2.7. A tuple of three protocols $(\mathsf{Gen}, \mathsf{Commit}, \mathsf{Open})$ is a binding commitment scheme for $\mathcal{X}$ if:

Binding. For any PPT adversary $\mathcal{A}$,

$$\Pr\left\{ \begin{array}{c} pp \leftarrow \mathsf{Gen}(1^\lambda);\ (\mathcal{C}, \mathbb{G}_0, \mathbb{G}_1, \mathcal{S}_0, \mathcal{S}_1) = \mathcal{A}(pp); \\ b_0 \leftarrow \mathsf{Open}(pp, \mathcal{C}, \mathbb{G}_0, \mathcal{S}_0);\ b_1 \leftarrow \mathsf{Open}(pp, \mathcal{C}, \mathbb{G}_1, \mathcal{S}_1): \\ b_0 = b_1 \neq 0 \wedge \mathbb{G}_0 \neq \mathbb{G}_1 \end{array} \right\} \leq \mathrm{negl}(\lambda)$$

Definition 2.8. A commitment scheme $(\mathsf{Gen}, \mathsf{Commit}, \mathsf{Open})$ provides hiding commitments if for all PPT adversaries $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$:

$$\left| 1 - 2 \cdot \Pr \left\{ \begin{array}{c} b = \bar{b} : \\ pp \leftarrow \mathsf{Gen}(1^\lambda); \\ (\mathbb{G}_0, \mathbb{G}_1, st) = \mathcal{A}_0(pp); \\ b \leftarrow_R \{0,1\}; \\ (\mathcal{C}, \mathcal{S}) \leftarrow \mathsf{Commit}(pp; \mathbb{G}_b); \\ \bar{b} \leftarrow \mathcal{A}_1(st, \mathcal{C}) \end{array} \right\} \right| \leq \mathrm{negl}(\lambda)$$

If the above holds for all algorithms, then the commitment is statistically hiding.

Pedersen and AFGHO Commitments For messages $\mathcal{X} = \mathbb{F}^n$ and any $i \in \{1, 2, T\}$, the Pedersen commitment scheme is defined by:

$$\begin{aligned} \mathsf{Gen}(1^\lambda) &= \{g \leftarrow_\$ G_i^n, h \leftarrow G_i\} \\ (\mathcal{C}, \mathcal{S}) \leftarrow \mathsf{Commit}(pp; x) &= \{r \leftarrow_\$ \mathbb{F} \; ; \; (\langle x, g \rangle + rh \; ; \; r)\} \\ \mathsf{Open}(pp, \mathcal{C}, x, \mathcal{S}) &= (\langle x, g \rangle + r(h) \stackrel{?}{=} \mathcal{C}) \end{aligned}$$

As is standard, if DLOG in $\mathbb{G}_i$ is hard, then this is a hiding, binding commitment scheme. Similarly, Abe et. al. [2] define a structure preserving commitment to group elements. In this case we have $\mathcal{X} = \mathbb{G}_i^n$ for $i \in \{1, 2\}$ and:

$$\begin{aligned} \mathsf{Gen}(1^\lambda) &= \{g \leftarrow_\$ G_{3-i}^n, H_1 \leftarrow_\$ G_1, H_2 \leftarrow_\$ G_2\} \\ (\mathcal{C}, \mathcal{S}) \leftarrow \mathsf{Commit}(pp; x) &= \{r \leftarrow_\$ \mathbb{F} \; ; \; (\langle x, g \rangle + re(H_1, H_2) \; ; \; r)\} \\ \mathsf{Open}(pp, \mathcal{C}, x, \mathcal{S}) &= (\langle x, g \rangle + \mathcal{S}e(H_1, H_2) \stackrel{?}{=} \mathcal{C}) \end{aligned}$$

This is a hiding commitment as $re(H_1, H_2)$ is uniformly random in $\mathbb{G}_T$. The commitment is binding conditional on the DBP assumption (since providing two distinct openings would violate Lemma 2.2). Note that this commitment reduces to that of [2]; in that work an opening for a commitment to a vector $x \in \mathbb{G}_1^n$ would supply some $R \in G_1$ such that:

$$\mathcal{C} = \langle x, g \rangle + e(R, H_2).$$

Here, an opening provides $r \in \mathbb{F}$ such that $R = rH_1$, which is strictly stronger. Both the Pedersen and AFGHO commitments are additively homomorphic.

Commitments to matrices Composing the Pedersen and AFGHO commitments yields a two-tiered homomorphic commitment [24] to matrices. We recommend Bünz et al. [18] for a longer exposition of these ideas. Formally, we take $\mathcal{X} = \mathbb{F}^{n \times m}$, and for $M_{ij} \in \mathcal{X}$ we have:

$$\begin{aligned} \mathsf{Gen}(1^\lambda) &= \{\Gamma_1 \leftarrow_\$ \mathbb{G}_1^m, H_1 \leftarrow_\$ \mathbb{G}_1, \Gamma_2 \leftarrow_\$ \mathbb{G}_2^n, H_2 \leftarrow_\$ \mathbb{G}_2\} \\ (\mathcal{C}, \mathcal{S}) \leftarrow \mathsf{Commit}(pp; M_{ij}) &= \left\{ \begin{array}{c} r_{rows} \leftarrow_\$ \mathbb{F}^n \; ; \; r_{fin} \leftarrow_\$ \mathbb{F} \; ; \; H_T \leftarrow e(H_1, H_2) \; ; \\ V_i \leftarrow \mathsf{Commit}_{Pedersen}((\Gamma_1, H_1) \; ; \; M_{ij}, r_{rows,i}) \; ; \\ (\mathsf{Commit}_{AFGHO}((\Gamma_2, H_T) \; ; \; V, r_{fin}), (r_{rows}, r_{fin}, V)) \end{array} \right\} \\ \mathsf{Open}(pp, \mathcal{C}, M, \mathcal{S}) &= \left(\mathcal{C} \stackrel{?}{=} \sum_i \Gamma_{2i} \left(\sum_j M_{ij} \Gamma_{1j} + r_{rows,i} H_1 \right) + r_{fin} e(H_1, H_2) \right) \end{aligned}$$

Remark 2.3. *$V \in \mathbb{G}_1^n$ is a vector of hiding, binding commitments to the rows of M. So if $L \in \mathbb{F}^n$ then $\sum L_i V_i \in \mathbb{G}_1$ is a hiding, binding commitment to $L^T M \in \mathbb{F}^m$.*

2.5 Polynomial commitments and evaluation from vector-matrix-vector products

We define a polynomial commitment scheme for multilinear polynomials, following [17, 34], which permit interactive evaluation proofs contra Kate [29]. Let $(\mathsf{Gen}_{\mathbb{F}}, \mathsf{Commit}_{\mathbb{F}}, \mathsf{Open}_{\mathbb{F}})$ be a commitment scheme for $\mathcal{X} = \mathbb{F}$ with public parameters $pp_{\mathbb{F}}$.

Definition 2.9. A tuple of four protocols $(\mathsf{Gen}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Eval})$ is an honest-verifier, zero-knowledge, extractable polynomial commitment scheme for ℓ-variable multilinear polynomials over $\mathbb{F}$ if $(\mathsf{Gen}, \mathsf{Commit}, \mathsf{Open})$ is a commitment scheme for ℓ-variable multilinear polynomials over $\mathbb{F}$, and:

- $pp \leftarrow \mathsf{Gen}(1^\lambda)$, $pp_{\mathbb{F}} \leftarrow \mathsf{Gen}_{\mathbb{F}}(1^\lambda)$. Both $\mathcal{V}$ and $\mathcal{P}$ hold a commitment $\mathcal{C}_G$ to G.
- $\mathcal{V}$ selects a public-coin $r \in \mathbb{F}^\ell$; $\mathcal{P}$ then supplies a commitment $\mathcal{C}_v$ to a scalar $v \in \mathbb{F}$.
- $b \leftarrow \mathsf{Eval}(pp, pp_{\mathbb{F}}, \mathcal{C}_G, r, \mathcal{C}_v; G, \mathcal{S}_G, \mathcal{S}_v)$ is an interactive public-coin protocol between a PPT Prover $\mathcal{P}$ and Verifier $\mathcal{V}$. $\mathcal{P}$ additionally knows a ℓ-variate multilinear polynomial $G \in \mathbb{F}[X_1, \ldots, X_\ell]$ and its secret opening hint $\mathcal{S}_G$, and the scalar $v \in \mathbb{F}$ and its secret opening hint $\mathcal{S}_v$. $\mathcal{P}$ attempts to convince $\mathcal{V}$ that $G(r) = v$. At the end of the protocol, $\mathcal{V}$ outputs $b \in \{0, 1\}$.
- **Completeness.** For any ℓ-variate multilinear polynomial $G \in \mathbb{F}[X_1, \ldots, X_\ell]$,

$$\Pr\left\{ \begin{array}{c} pp \leftarrow \mathsf{Gen}(1^\lambda); \\ (\mathcal{C}_G, \mathcal{S}_G) \leftarrow \mathsf{Commit}(pp; G); (\mathcal{C}_v, \mathcal{S}_v) \leftarrow \mathsf{Commit}_{\mathbb{F}}(pp_{\mathbb{F}}; v): \\ \mathsf{Eval}(pp, pp_{\mathbb{F}}, \mathcal{C}_G, r, \mathcal{C}_v, ; G, \mathcal{S}_G, \mathcal{S}_v) = 1 \wedge v = G(r) \end{array} \right\} \geq 1 - \mathrm{negl}(\lambda)$$

- **Knowledge soundness.** Eval is a public-coin succinct interactive argument of knowledge with witness-extended emulation (Definition 2.5) for the following NP relation given $pp \leftarrow \mathsf{Gen}(1^\lambda)$, $pp_{\mathbb{F}} \leftarrow \mathsf{Gen}_{\mathbb{F}}(1^\lambda)$, and $r \in \mathbb{F}^\ell$ chosen after $\mathcal{C}_{\mathbb{G}}$ is fixed:

$$\mathcal{R}_{\mathsf{Eval}}(pp, pp_{\mathbb{F}}) = \left\{ \begin{array}{c} \langle (\mathcal{C}_G, \mathcal{C}_v), (G, \mathcal{S}_G, \mathcal{S}_v) \rangle : \\ G \in \mathbb{F}[X_1, \ldots, X_\ell] \text{ is multilinear } \wedge v \in \mathbb{F} \wedge G(r) = v \\ \wedge \mathsf{Open}(pp; \mathcal{C}_G, G, \mathcal{S}_G) = 1 \wedge \mathsf{Open}_{\mathbb{F}}(pp_{\mathbb{F}}; \mathcal{C}_v, v, \mathcal{S}_v) = 1 \end{array} \right\}$$

- **Zero-knowledge.** Eval is a public-coin succinct interactive argument of knowledge with witness-extended emulation (Definition 2.5) and honest-verifer zero-knowledge (Definition 2.6) for the following NP relation given $pp \leftarrow \mathsf{Gen}(1^\lambda)$, $pp_{\mathbb{F}} \leftarrow \mathsf{Gen}_{\mathbb{F}}(1^\lambda)$, and $r \in \mathbb{F}^\ell$ chosen after $\mathcal{C}_{\mathbb{G}}$ is fixed:

$$\mathcal{R}_{\mathsf{Eval}}(pp, pp_{\mathbb{F}}) = \left\{ \begin{array}{c} \langle (\mathcal{C}_G, \mathcal{C}_v), (G, \mathcal{S}_G, v, \mathcal{S}_v) \rangle : \\ G \in \mathbb{F}[X_1, \ldots, X_\ell] \text{ is multilinear } \wedge G(r) = v \wedge \\ \mathsf{Open}(pp; \mathcal{C}_G, G, \mathcal{S}_G) = 1 \wedge \mathsf{Open}_{\mathbb{F}}(pp_{\mathbb{F}}; \mathcal{C}_v, v, \mathcal{S}_v) = 1 \end{array} \right\}$$

Remark 2.4. *We modify the definition from [17] to have the evaluation be committed, and weaken it to explicitly state that the point of evaluation is not chosen adversarially. This weakening can be avoided (see Remark 5.2)), but is not relevant to the typical use cases for polynomial commitment schemes.*

Bünz et al. [18] use the matrix commitment of Section 2.4 to commit to univariate and bivariate polynomials. Using ideas from Hyrax [37], we present a fundamentally similar commitment for general polynomials.

Any polynomial f in variables $x_1, \ldots, x_\ell$ of degree $d_1, \ldots, d_\ell$ can be reformulated as a multilinear polynomial in the variables $\{x_i, x_i^2, \ldots x_i^{2^{\lceil \log(d_i+1) \rceil}} : i \in [\ell]\}$. The evaluation of a multilinear polynomial in r variables at some point $x \in \mathbb{F}^r$ is the contraction of an order r tensor T with r vectors $(1, x_i)$. Furthermore, for any $n_1 \times \ldots \times n_r$ tensor T and vectors $v_i \in \mathbb{F}^{n_i}$:

$$\sum_{i_1,\ldots,i_r} T^{i_1 \ldots i_r} (v_j)_{i_j} = (\otimes_{i<k} v_i)^T M (\otimes_{i \geq k} v_i)$$
$$\text{where } M_{ij} = T^{i_1 \ldots i_r} \Leftrightarrow i \prod_{i \geq k} n_i + j = i_r + n_r(\ldots(i_2 + n_2(i_1))\ldots)$$

So the evaluation of f at some point r can be replaced with the evaluation of a multilinear polynomial in $r = \sum_i \lceil \log(d_i + 1) \rceil$, variables, which can in turn be replaced by a vector-matrix-vector product with vectors of length at most $2^m = 2^{\lceil r/2 \rceil} = O((\prod_i d_i)^{1/2} 2^{\ell/2})$. Note also that the vectors in this product have multiplicative structure, being formed as Kronecker products of vectors $(1, x_i^{2^j})$ for $i, j \in \mathbb{N}$. For univariate polynomials of degree d, $m \leq (3 + \log d)/2$, and for multilinear polynomials in ℓ variables $m \leq (\ell + 1)/2$.

In the particular case of multilinear polynomials in ℓ variables, it is often convenient to specify the polynomial on some cube, e.g. $\{0,1\}^\ell$. In this case, analogous formulae exist, as reading off the evaluations on the cube as a tensor:

$$f(x) = \sum_{i_1,\ldots,i_r \in \{0,1\}} T^{i_1 \ldots i_r} \prod_{j \in 1,\ldots,r} (v'_j)_{i_j},$$

where now the vectors have the slightly different form $v'_j = (1 - x_j, x_j)$. Similar formulae exist for interpolating from the evaluations on any product $\{a_1, b_1\} \times \ldots \times \{a_\ell, b_\ell\}$.

3 An inner-product argument with logarithmic Verifier costs

We begin by showing the simplest form of Dory: an argument for inner products between two vectors in $v_1 \in \mathbb{G}_1^n$, $v_2 \in \mathbb{G}_2^n$, AFGHO committed with generators $(\Gamma_2, e(H_1, H_2)) \in \mathbb{G}_2^n \times \mathbb{G}_T$ and $(\Gamma_1, e(H_1, H_2)) \in \mathbb{G}_1^n \times \mathbb{G}_T$ respectively. Formally, we define a language:

$$\begin{aligned}(C, D_1, D_2) &\in \mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2} \subset \mathbb{G}_T^3 \\ &\Leftrightarrow \exists (v_1 \in \mathbb{G}_1^n, v_2 \in \mathbb{G}_2^n, r_C \in \mathbb{F}, r_{D_1} \in \mathbb{F}, r_{D_2} \in \mathbb{F}) : \\ &\quad D_1 = \langle v_1, \Gamma_2 \rangle + r_{D_1} e(H_1, H_2), \quad D_2 = \langle \Gamma_1, v_2 \rangle + r_{D_2} e(H_1, H_2), \\ &\quad C = \langle v_1, v_2 \rangle + r_C e(H_1, H_2)\end{aligned}$$

For n even, and $\Gamma'_{1,2} \in \mathbb{G}_{1,2}^{2^{n/2}}$, we will show an interactive protocol reducing membership in $\mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}$ to membership in $\mathcal{L}_{n/2,\Gamma'_1,\Gamma'_2,H_1,H_2}$. For $\Gamma_{1,2} \in \mathbb{G}_{1,2}$, we will also show an interactive argument of knowledge for $\mathcal{L}_{1,\Gamma_1,\Gamma_2,H_1,H_2}$. We will also show a generic argument reducing two claims of membership of $\mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}$ to one. We will briefly discuss concrete efficiency of these protocols and optimizations for $\mathcal{V}$.

3.1 Scalar-Product

We give a interactive argument of knowledge for $\mathcal{L}_{1,\Gamma_1,\Gamma_2,H_1,H_2}$. This requires showing the product of two elements $v_1 \in \mathbb{G}_1$ and $v_2 \in \mathbb{G}_2$ under AFGHO; for Pedersen commitments this is folklore. Since pairings are more expensive than multiplications in $\mathbb{G}_1$ or $\mathbb{G}_2$, we combine the usual final three checks into a single pairing with a Verifier challenge.

$\mathsf{Scalar\text{-}Product}_{\Gamma_1,\Gamma_2,H_1,H_2}(C, D_1, D_2)$

Precompute: $H_T = e(H_1, H_2)$, $\chi = e(\Gamma_1, \Gamma_2)$

$\mathcal{P}$ **witness:** $(v_1, v_2, r_C, r_{D_1}, r_{D_2})$ for $(C, D_1, D_2) \in \mathcal{L}_{1,\Gamma_1,\Gamma_2,H_1,H_2}$

$\mathcal{P}$**:** $r_{P_1}, r_{P_2}, r_Q, r_R \leftarrow_\$ \mathbb{F}$, $d_1 \leftarrow_\$ \mathbb{G}_1$, $d_2 \leftarrow_\$ \mathbb{G}_2$

$\mathcal{P} \to \mathcal{V}$**:** $P_1 = e(d_1, \Gamma_2) + r_{P_1} H_T$, $\quad P_2 = e(\Gamma_1, d_2) + r_{P_2} H_T$,
$Q = e(d_1, v_2) + e(v_1, d_2) + r_Q H_T$, $\quad R = e(d_1, d_2) + r_R H_T$,

$\mathcal{V} \to \mathcal{P}$**:** $c \leftarrow_\$ \mathbb{F}$

$\mathcal{P} \to \mathcal{V}$**:** $E_1 \leftarrow d_1 + c v_1$, $\quad E_2 \leftarrow d_2 + c v_2$,
$r_1 \leftarrow r_{P_1} + c r_{D_1}$, $\quad r_2 \leftarrow r_{P_2} + c r_{D_2}$,
$r_3 \leftarrow r_R + c r_Q + c^2 r_C$

$\mathcal{V}$**:** $d \leftarrow_\$ \mathbb{F}$, accept if:

$$e(E_1 + d\Gamma_1, E_2 + d^{-1}\Gamma_2) = \chi + R + cQ + c^2 C + dP_2 + dcD_2 + d^{-1}P_1 + d^{-1}cD_1 - (r_3 + dr_2 + d^{-1}r_1)H_T$$

Theorem 3.1. *For $\Gamma_2, H \leftarrow_\$ \mathbb{G}_2$, $\Gamma_1, H_1 \leftarrow_\$ \mathbb{G}_1$, Scalar-Product is an HVSZK, succinct interactive argument of knowledge for $\mathcal{L}_{1,\Gamma_1,\Gamma_2,H_1,H_2}$ with witness extended emulation under SXDH.*

Proof. Succinctness and the Public Coin property are immediate. Completeness holds as for an honest Prover:

$$\begin{aligned} e(E_1 + d\Gamma_1, E_2 + d^{-1}\Gamma_2) &= e(d_1 + cv_1, d_2 + cv_2) + de(\Gamma_1, d_2 + cv_2) \\ &\quad + d^{-1}e(d_1 + cv_1, \Gamma_2) + e(\Gamma_1, \Gamma_2) \\ &= \chi + c^2 e(v_1, v_2) + c[e(d_1, v_2) + e(v_1, d_2)] + e(d_1, d_2) \\ &\quad + de(\Gamma_1, d_2) + dce(\Gamma_1, v_2) + d^{-1}e(d_2, \Gamma_2) + d^{-1}ce(v_2, \Gamma_2) \\ &= \chi + R + cQ + c^2 C + dP_2 + dcD_2 + d^{-1}P_1 + d^{-1}cD_1 - (r_3 + dr_2 + d^{-1}r_1)H_T \end{aligned}$$

so $\mathcal{V}$ accepts.

Witness extended emulation: Apply Lemma 2.3, taking $\mu = 2$ and $w_1 = w_2 = 3$. We are given tree of accepting transcripts for 3 values c, and for each c there are 3 accepting values of d. Across all transcripts, P_1, P_2, Q, R, C, D_1, D_2 are constant, and E_1, E_2, r_1, r_2, r_3 can be interpolated as quadratics in c.

The final randomized check contains terms in d only of form $d, 1, d^{-1}$. Since there are three accepting choices of d, we can eliminate any two of $d, 1, d^{-1}$ leaving a non-trivial constraint. So for each of the three challenge c:

$$\begin{aligned} e(E_1(c), E_2(c)) + r_3(c)H_T &= R + cQ + c^2C \\ e(E_1(c), \Gamma_2) + r_1(c)e(H_1, H_2) &= P_1 + cD_1 \\ e(\Gamma_1, E_2(c)) + r_2(c)e(H_1, H_2) &= P_2 + cD_2 \end{aligned}$$

Then if $E_1(c), r_1(c)$ are not linear functions, we can eliminate P_1, D_1 and recover a non-trivial linear relationship between elements of Γ_2 and H_2, contradicting Lemma 2.2. Similarly $E_2(c), r_2(c)$ must be linear. So for $i = 1, 2$, $E_i(c) = d_i + cv_i$ and $r_i = r_{P_i} + cr_{D_i}$. We interpolate $r_3(c) = r_R + cr_Q + c^2 r_C$, and substitute into the last verification equation:

$$\begin{aligned} R + cQ + c^2C = e(d_1, d_2) + r_R H_T + c(e(d_1, v_2) + e(v_1, d_2) + r_Q H_T) \\ + c^2(e(v_1, v_2) + r_C H_T) \end{aligned}$$

Since this holds for 3 distinct c, we can eliminate any two of $1, c, c^2$ leaving a non-trivial constraint. So comparing c^2 terms we find $C = e(v_1, v_2) + r_C H_T$, and so we have extracted a witness.

HVSZK: Note that for $\mathcal{P}$, $E_1, E_2, Q \leftarrow_\$ \mathbb{G}_T^3$ and $r_1, r_2, r_3 \leftarrow_\$ \mathbb{F}$. We split the final check into terms that are proportional to $d^{-1}, d, 1$:

$$\begin{aligned} P_1 &= e(E_1, \Gamma_2) + r_1 H_T - cD_1, & P_2 &= e(\Gamma_1, E_2) + r_2 H_T - cD_2, \\ R &= e(E_1, E_2) + r_3 H_T - cQ - c^2 C \end{aligned}$$

We construct a simulator as follows: Sample $Q, E_1, E_2 \leftarrow_\$ \mathbb{G}_T^3$ and compute the challenge c from $\mathcal{V}$'s coins. Then sample $r_1, r_2, r_3 \leftarrow_\$ \mathbb{F}$ and compute P_1, P_2, R as above. □

3.2 Dory-Reduce

We now show an interactive argument reducing membership of $\mathcal{L}_{2^m, \Gamma_1, \Gamma_2, H_1, H_2}$ to membership of $\mathcal{L}_{2^{m-1}, \Gamma_1', \Gamma_2', H_1, H_2}$:

Dory-Reduce$_{m, \Gamma_1, \Gamma_2, \Gamma_1', \Gamma_2', H_1, H_2}(C, D_1, D_2)$

Precompute: $H_T = e(H_1, H_2)$, $\chi = \langle \Gamma_1, \Gamma_2 \rangle$, and:

$$\begin{aligned} \Delta_{1L} &= \langle \Gamma_{1L}, \Gamma_2' \rangle, & \Delta_{1R} &= \langle \Gamma_{1R}, \Gamma_2' \rangle, \\ \Delta_{2L} &= \langle \Gamma_1', \Gamma_{2L} \rangle, & \Delta_{2R} &= \langle \Gamma_1', \Gamma_{2R} \rangle, \\ \chi_i &= \langle \Gamma_1, \Gamma_2 \rangle \end{aligned}$$

$\mathcal{P}$ **witness:** $(v_1, v_2, r_c, r_{D_1}, r_{D_2})$ for $(C, D_1, D_2) \in \mathcal{L}_{2^m, \Gamma_1, \Gamma_2, H_1, H_2}$

$\mathcal{P}$**:** $r_{D_{1L}}, r_{D_{1R}}, r_{D_{2L}}, r_{D_{2R}} \leftarrow_\$ \mathbb{F}$

$\mathcal{P} \rightarrow \mathcal{V}$**:** $\begin{aligned} D_{1L} &= \langle v_{1L}, \Gamma_2' \rangle + r_{D_{1L}} H_T, & D_{1R} &= \langle v_{1R}, \Gamma_2' \rangle + r_{D_{1R}} H_T \\ D_{2L} &= \langle \Gamma_1', v_{2L} \rangle + r_{D_{2L}} H_T, & D_{2R} &= \langle \Gamma_1', v_{2R} \rangle + r_{D_{2R}} H_T \end{aligned}$

$\mathcal{V} \to \mathcal{P}$: $\beta \leftarrow_{\$} \mathbb{F}$

$\mathcal{P}(*)$: $v_1 \leftarrow v_1 + \beta\Gamma_1, \quad v_2 \leftarrow v_2 + \beta^{-1}\Gamma_2, \quad r_C \leftarrow r_C + \beta r_{D_2} + \beta^{-1} r_{D_1}$

$\mathcal{P}$: $r_{C_+}, r_{C_-} \leftarrow_{\$} \mathbb{F}$

$\mathcal{P} \to \mathcal{V}$: $C_+ = \langle v_{1L}, v_{2R} \rangle + r_{C_+} H_T$,
$C_- = \langle v_{1R}, v_{2L} \rangle + r_{C_-} H_T$

$\mathcal{V} \to \mathcal{P}$: $\alpha \leftarrow_{\$} \mathbb{F}$

$\mathcal{P}(**)$: $v'_1 \leftarrow \alpha v_{1L} + v_{1R}, \qquad v'_2 \leftarrow \alpha^{-1} v_{2L} + v_{2R}$
$r'_{D_1} \leftarrow \alpha r_{D_{1L}} + r_{D_{1R}}, \qquad r'_{D_2} \leftarrow \alpha^{-1} r_{D_{2L}} + r_{D_{2R}},$
$r'_C \leftarrow r_C + \alpha r_{C_+} + \alpha^{-1} r_{C_-}$

$\mathcal{V}(**)$: $C' \leftarrow C + \chi + \beta D_2 + \beta^{-1} D_1 + \alpha C_+ + \alpha^{-1} C_-$
$D'_1 \leftarrow \alpha D_{1L} + D_{1R} + \alpha\beta\Delta_{1L} + \beta\Delta_{1R}$
$D'_2 \leftarrow \alpha^{-1} D_{2L} + D_{2R} + \alpha^{-1}\beta^{-1}\Delta_{2L} + \beta^{-1}\Delta_{2R}$

$\mathcal{V}$: Accept if $(C', D'_1, D'_2) \in \mathcal{L}_{2^{m-1}, \Gamma'_1, \Gamma'_2, H_1, H_2}$

$\mathcal{P}$ **witness:** $(v'_1, v'_2, r'_C, r'_{D_1}, r'_{D_2})$

Theorem 3.2. *For $\Gamma'_2 \leftarrow_{\$} \mathbb{G}_2^{m-1}$, $H_2 \leftarrow_{\$} \mathbb{G}_2$, $\Gamma_1 \leftarrow_{\$} \mathbb{G}_2^{m-1}$, $H_1 \leftarrow_{\$} \mathbb{G}_1$, Dory-Reduce is an HVSZK, succinct interactive argument of knowledge for $\mathcal{L}_{2^m, \Gamma_1, \Gamma_2, H_1, H_2}$ with witness extended emulation under SXDH.*

Proof. Succinctness and the Public Coin properties are immediate. Completeness holds if $(v'_1, v'_2, r'_C, r'_{D_1}, r'_{D_2})$ witnesses $(C', D'_1, D'_2) \in \mathcal{L}_{2^{m-1}, \Gamma'_1, \Gamma'_2, H_1, H_2}$. To see this, observe that:

$$v'_1 = \alpha v_{1L} + v_{1R} + \beta\alpha\Gamma_{1L} + \beta\Gamma_{1R}, \quad v'_2 = \alpha^{-1} v_{2L} + v_{2R} + \beta^{-1}\alpha^{-1}\Gamma_{2L} + \beta^{-1}\Gamma_{2R},$$

and so:

$$\begin{aligned}
C' &= \langle v_1, v_2 \rangle + \langle \Gamma_1, \Gamma_2 \rangle + r_C H_T + \beta(\langle \Gamma_1, v_2 \rangle + r_{D_2} H_T) + \beta^{-1}(\langle v_1, \Gamma_2 \rangle + r_{D_1} H_T) \\
&\quad + \alpha(\langle v_{1L}, v_{2R} \rangle + r_{C_+} H_T) + \alpha^{-1}(\langle v_{1R}, v_{2L} \rangle + r_{C_-} H_T) = \langle v'_1, v'_2 \rangle + r'_C H_T, \\
D'_1 &= \alpha(\langle v_{1L}, \Gamma'_2 \rangle + r_{D_{1L}} H_T) + (\langle v_{1R}, \Gamma'_2 \rangle + r_{D_{1R}} H_T) + \alpha\beta\langle \Gamma_{1L}, \Gamma'_2 \rangle + \beta\langle \Gamma_{1R}, \Gamma'_2 \rangle \\
&= \langle v'_1, \Gamma'_2 \rangle + r'_{D_1} H_T, \\
D'_2 &= \alpha^{-1}(\langle \Gamma'_1, v_{2L} \rangle + r_{D_{2L}} H_T) + (\langle \Gamma'_1, v_{2R} \rangle + r_{D_{2R}} H_T) \\
&\quad + \alpha^{-1}\beta^{-1}\langle \Gamma'_1, \Gamma_{2L} \rangle + \beta^{-1}\langle \Gamma'_1, \Gamma_{2R} \rangle = \langle v'_1, \Gamma'_2 \rangle + r'_{D_1} H_T,
\end{aligned}$$

Witness extended emulation Apply Lemma 2.3, taking $\mu = 2$ and $w_1 = w_2 = 3$. We are given a tree of accepting transcripts for 3 values β, and for each β 3 values of α. For each leaf, the Prover reveals the witness $(v'_1, v'_2, r'_C, r'_{D_1}, r'_{D_2})$. Our witness extraction is analogous to witness extraction of SIPP in [18]["Proof of Theorem 4.1"] or of the improved inner product argument of [16][Appendix B].

Across all transcripts, D_{1L}, D_{1R} are constant. Furthermore, C_+, C_- can be interpolated as a Laurent polynomials with coefficients in $\mathbb{G}_T$ of degree 1 and order -1 in β, and $v'_1, v'_2, r'_{D_1}, r'_{D_2}, r'_C$ can be interpolated as bivariate Laurent polynomials with coefficients in $\mathbb{G}_T$ of degree 1 and order -1 in α, β. From membership of $\mathcal{L}_{2^{m-1},\Gamma'_1,\Gamma'_2,H_1,H_2}$:

$$\begin{aligned} D'_1 &= \alpha D_{1L} + D_{1R} + \alpha\beta\langle\Gamma_{1L}, \Gamma'_2\rangle + \beta\langle\Gamma_{1R}, \Gamma'_2\rangle \\ &= \langle v'_1(\alpha,\beta), \Gamma'_2\rangle + r'_{D_1}(\alpha,\beta)e(H_1, H_2) \end{aligned}$$

for all 9 (β, α) pairs. So if v'_1, r'_{D_1} are not multilinear in α, β, we can eliminate the $1, \alpha, \beta, \alpha\beta$ terms to obtain a non-trivial relationship between Γ'_2, H_2, contradicting Lemma 2.2. Similarly if the $\alpha\beta, \beta$ terms of v_1 are not Γ_{1L}, Γ_{1R} respectively and the same terms in r'_{D_1} are not 0, we find a non-trivial linear relationship between Γ'_2, H_2, contradicting Lemma 2.2. Applying the same considerations to v'_2, r'_{D_2} we find:

$$\begin{aligned} v'_1(\alpha,\beta) &= \alpha v_{1L} + v_{1R} + \beta(\alpha\Gamma_{1L} + \Gamma_{1R}) \\ v'_2(\alpha,\beta) &= \alpha^{-1} v_{2L} + v_{2R} + \beta^{-1}(\alpha^{-1}\Gamma_{2L} + \Gamma_{2R}) \end{aligned}$$

Let us say $r'_C(\alpha,\beta) = r_C + \beta r_{D_2} + \beta^{-1} r_{D_1} + \alpha(\ldots) + \alpha^{-1}(\ldots)$. Then substituting into the constraint of $\mathcal{L}_{2^{m-1},\Gamma'_1,\Gamma'_2,H_1,H_2}$ on C':

$$\begin{aligned} C' &= C + \chi + \beta D_2 + \beta^{-1} D_1 + \alpha C_+(\beta) + \alpha^{-1} C_-(\beta) \\ &= \langle v'_1(\alpha,\beta), v'_2(\alpha,\beta)\rangle + r'_C(\alpha,\beta) H_T \\ &= \langle v_{1L}, v_{2L}\rangle + \langle v_{1R}, v_{2R}\rangle + \chi \\ &+ \beta(\langle\Gamma_{1L}, v_{2L}\rangle + \langle\Gamma_{1R}, v_{2R}\rangle) + \beta^{-1}(\langle v_{1L}, \Gamma_{2L}\rangle + \langle v_{1L}, \Gamma_{2L}\rangle) \\ &+ \alpha(\langle v_{1L}, v_{2R}\rangle + \langle\Gamma_{1L}, \Gamma_{2R}\rangle + \beta\langle\Gamma_{1L}, v_{2R}\rangle + \beta^{-1}\langle v_{1L}, \Gamma_{2R}\rangle) \\ &+ \alpha^{-1}(\langle v_{1R}, v_{2L}\rangle + \langle\Gamma_{1R}, \Gamma_{2L}\rangle + \beta\langle\Gamma_{1R}, v_{2L}\rangle + \beta^{-1}\langle v_{1R}, \Gamma_{2L}\rangle) \end{aligned}$$

Since these are two bivariate Laurent series of degree 1 and order -1, equal at 3 values of α for each of 3 values of β, we conclude they are equal. In particular comparing the $1, \beta, \beta^{-1}$ coefficients:

$$\begin{aligned} C &= \langle v_{1L}, v_{2L}\rangle + \langle v_{1R}, v_{2R}\rangle + r_C H_T \\ D_1 &= \langle v_{1L}, \Gamma_{2L}\rangle + \langle v_{1R}, \Gamma_{2R}\rangle + r_{D_1} H_T \\ D_2 &= \langle \Gamma_{1L}, v_{2L}\rangle + \langle \Gamma_{1R}, v_{2R}\rangle + r_{D_2} H_T \end{aligned}$$

and so the vectors $v_1 = (v_{1L}||v_{2L})$, $v_2 = (v_{2L}||v_{2R})$ and the values r_C, r_{D_1}, r_{D_2} are the desired witness.

HVSZK: All messages from $\mathcal{P}$ to $\mathcal{V}$ are uniformly random elements of $\mathbb{G}_T$, and so are trivially simulated.

□

3.3 Dory-Innerproduct

We now discuss the full inner product argument explicitly, which requires public parameters: $\forall j \in 0 \ldots m, \Gamma_{1,j} \in \mathbb{G}_1^{2^{m-j}}, \Gamma_{2,j} \in \mathbb{G}_2^{2^{m-j}}$.

$\mathsf{Dory\text{-}Innerproduct}_{\Gamma_{ij},H_i}(C, D_1, D_2)$

Precompute: $H_T = e(H_1, H_2)$, for all $i \in 0 \dots m$ compute $\chi_i = \langle \Gamma_{1i}, \Gamma_{2i} \rangle$, and for all $i \in 0 \dots m-1$ compute:

$$\Delta_{1L,i} = \langle (\Gamma_{1,i})_L, \Gamma_{2,i+1} \rangle, \qquad \Delta_{1R,i} = \langle (\Gamma_{1,i})_R, \Gamma_{2,i+1} \rangle,$$
$$\Delta_{2L,i} = \langle \Gamma_{1,i+1}, (\Gamma_{2,i})_L \rangle, \qquad \Delta_{2R,i} = \langle \Gamma_{1,i+1}, (\Gamma_{2,i})_R \rangle,$$

$\mathcal{P}$ **witness:** $(v_1, v_2, r_C, r_{D_1}, r_{D_2})$ for $(C, D_1, D_2) \in \mathcal{L}_{2^m,\Gamma_{1,0},\Gamma_{2,0},H_1,H_2}$

For $j = 0 \dots m-1$

$\mathcal{P}, \mathcal{V}$: $(C, D_1, D_2) \leftarrow \mathsf{Dory\text{-}Reduce}_{m-j,\Gamma_{1,j}\Gamma_{2,j},\Gamma_{1,j+1},\Gamma_{2,j+1},H_1,H_2}(C, D_1, D_2)$

$\mathcal{P}, \mathcal{V}$: $\mathsf{Scalar\text{-}Product}_{\Gamma_{1,m},\Gamma_{2,m},H_1,H_2}(C, D_1, D_2)$

Theorem 3.3. *If each Γ_{ij} is uniformly random in $\mathbb{G}_i^{2^{m-j}}$ and $H_i \leftarrow_\$ \mathbb{G}_i$, $\Gamma_1, H_1 \leftarrow_\$ \mathbb{G}_1$, then Dory-Innerproduct is an HVSZK, succinct interactive argument of knowledge for $\mathcal{L}_{2^m,\Gamma_1,\Gamma_2,H_1,H_2}$ with witness extended emulation under SXDH.*

Proof. Succinctness, the Public Coin property, Completeness and HVSZK follow from the same properties of the two sub-arguments.

Soundness holds as for each i, all entries in $\Gamma_{1,i}$ and H_1 are uniformly random and independent and all entries in $\Gamma_{2,i}$ and H_2 are uniformly random and independent, so each sub-argument is computationally sound.

Witness extended emulation follows from composition for Lemma 2.3 as in Remark 2.1, giving $\mu = 2m+2$ and all $w_i = 3$. Then the tree has size $9^{1+\log n} = O(n^{\log 9})$ transcripts, which is polynomial in λ. □

Remark 3.1. *For $i \neq j$, $\Gamma_{1,i}$ and $\Gamma 1, j$ can be dependent (similarly $\Gamma_{2,i}$ and $\Gamma 2, j$). So in the sequel we set $\Gamma_{1,i+1} = (\Gamma_{1,i})_L$ and $\Gamma_{2,i+1} = (\Gamma_{2,i})_L$, which implies $(\Delta_{1,i})_L = (\Delta_{2,i})_L$.*

3.3.1 Concrete costs of Dory-Innerproduct

$\mathcal{P}$**:** In each call to Dory-Reduce, $\mathcal{P}$ sends 6 elements of $\mathbb{G}_T$ to $\mathcal{V}$. The j-th call requires $\mathcal{P}$ to perform 6 multi-pairings of size 2^{m-j-1}, $O(2^{m-j})$ operations in $\mathbb{F}$ to update their witness, and $O(1)$ additional operations in $\mathbb{G}_T$ and $\mathbb{F}$. The call to Scalar-Product requires $\mathcal{P}$ compute $O(1)$ pairings and exponentiations in $\mathbb{G}_T$. So the overall cost to $\mathcal{P}$ is dominated by multi-pairings of total size 6×2^m, $O(m)$ group operations, and $O(2^m)$ field arithmetic.

$\mathcal{V}$**:** Naively, in each invocation of Dory-Reduce $\mathcal{V}$ computes 10 exponentiations in $\mathbb{G}_T$, 2 inversions and 2 multiplications in $\mathbb{F}$, and $O(1)$ additional operations in $\mathbb{G}_T$ and additions in $\mathbb{F}$. In the invocation of Scalar-Product $\mathcal{V}$ must compute 1 pairing, 7 exponentiations in $\mathbb{G}_T$, 1 inversion and 5 multiplications in $\mathbb{F}$, and $O(1)$ additional operations in $\mathbb{G}_T$ and additions in $\mathbb{F}$.

Deferring $\mathcal{V}$ Computation: $\mathcal{V}$'s computation depends only on the messages from $\mathcal{P}$ and the $5m+1$ precomputed values. For each call to Dory-Reduce, $\mathcal{V}$ uses some

values $\Delta_{1L} = \Delta_{2L}, \Delta_{1R}, \Delta_{2R}, \chi$, and in the final check $\mathcal{V}$ uses $e(\Gamma_{1m}, \Gamma_{2m})$. We will use superscripts on group elements and subscripts on the challenge scalars to denote which call they came from. We assume that we precompute $\Delta^j_{\{1,2\}\{L,R\}}$ as before, but instead of computing χ_i for $i \in 0 \ldots m$, we compute: $\chi = \sum_{j=0}^{m-1} \langle \Gamma_{1j}, \Gamma_{2j} \rangle$ and $\chi_{fin} = \langle \Gamma_{1m}, \Gamma_{2m} \rangle$. Collapsing the Dory-Reduce rounds, we obtain the arguments for Scalar-Product:

$$
\begin{aligned}
C \leftarrow C + \chi + \beta_0 D_2^0 + \beta_0^{-1} D_1^0 + \sum_{j=0}^{m-1} (\alpha_j C_+^j + \alpha_j^{-1} C_-^j) + \\
+ \sum_{j=1}^{m-1} \beta_j (\alpha_{j-1}^{-1} D_{2L}^{j-1} + D_{2R}^{j-1} + \alpha_{j-1}^{-1} \beta_{j-1}^{-1} \Delta_{2L}^{j-1} + \beta_{j-1}^{-1} \Delta_{2R}^{j-1}) \\
+ \sum_{j=1}^{m-1} \beta_j^{-1} (\alpha_{j-1} D_{1L}^{j-1} + D_{1R}^{j-1} + \alpha_{j-1} \beta_{j-1} \Delta_{1L}^{j-1} + \beta_{j-1} \Delta_{1R}^{j-1})
\end{aligned}
$$

$$
\begin{aligned}
D_1 &\leftarrow \alpha_{j-1} D_{1L}^{m-1} + D_{1R}^{m-1} + \alpha_{m-1} \beta_{m-1} \Delta_{1L}^{m-1} + \beta_{m-1} \Delta_{1R}^{m-1} \\
D_2 &\leftarrow \alpha_{j-1}^{-1} D_{2L}^{m-1} + D_{2R}^{m-1} + \alpha_{m-1}^{-1} \beta_{m-1}^{-1} \Delta_{2L}^{m-1} + \beta_{m-1}^{-1} \Delta_{2R}^{m-1}
\end{aligned}
$$

which are substituted into the check in Scalar-Product. This reduces $\mathcal{V}$'s group operations to a multi-exponentiation in $\mathbb{G}_T$ of size $9m + 9$, two exponentiations in $\mathbb{G}_T$, and one pairing. Computation of the required coefficients with Montgomery's trick for batch inversions requires one inversion and $O(m)$ multiplications and additions in $\mathbb{F}$.

3.4 Batching inner products

Suppose we have $(C, D_1, D_2), (C', D_1', D_2') \in \mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}$, and $\mathcal{P}$ possesses witnesses $(v_1, v_2, r_C, r_{D_1}, r_{D_2})$ and $(v_1', v_2', r_C', r_{D_1}', r_{D_2}')$ respectively. Then we have the following two-to-one interactive argument:

Batch-Innerproduct$_{\Gamma_1,\Gamma_2}(C, D_1, D_2, C', D_1', D_2')$

Precompute: $H_T = e(H_1, H_2) \in G_T$

$\mathcal{P}$ **witness:** $(v_1, v_2, r_c, r_{D_1}, r_{D_2})$ for $(C, D_1, D_2) \in \mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}$, and $(v_1', v_2', r_c', r_{D_1}', r_{D_2}')$ for $(C', D_1', D_2') \in \mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}$

$\mathcal{P}$**:** $r_X \leftarrow_\$ \mathbb{F}$

$\mathcal{P} \rightarrow \mathcal{V}$**:** $X = \langle v_1, v_2' \rangle + \langle v_1', v_2 \rangle + r_X H_T$

$\mathcal{V} \rightarrow \mathcal{P}$**:** $\gamma \leftarrow_\$ \mathbb{F}$

$\mathcal{P}$**:** $v_1'' \leftarrow \gamma v_1 + v_1'$, $\quad v_2'' \leftarrow \gamma v_2 + v_2'$,
$r_{D_1}'' \leftarrow \gamma r_{D_1} + r_{D_1}'$, $\quad r_{D_2}'' \leftarrow \gamma r_{D_2} + r_{D_2}'$,
$r_C'' \leftarrow \gamma^2 r_C + \gamma r_X + r_C'$

$\mathcal{V}$**:** $C'' \leftarrow \gamma^2 C + \gamma X + C'$, $\quad D_1'' \leftarrow \gamma D_1 + D_1'$, $\quad D_2'' \leftarrow \gamma D_2 + D_2'$,

$\mathcal{V}$**:** Accept if $(C'', D_1'', D_2'') \in \mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}$

$\mathcal{P}$ **witness:** $(v_1'', v_2'', r_C'', r_{D_1}'', r_{D_2}'')$

Theorem 3.4. *For $\Gamma_i \leftarrow_\$ \mathbb{G}_i^n$, $H_i \leftarrow_\$ G_i$, Batch-Innerproduct is an HVSZK, succinct interactive argument of knowledge for $\mathcal{L}^2_{n,\Gamma_1,\Gamma_2,H_1,H_2}$ with witness extended emulation under SXDH.*

Proof. Succinctness, the Public Coin property, Completeness, Soundness and HVSZK of this protocol are immediate.

To show witness extended emulation, apply Lemma 2.3, with $\mu = 1$ and $w_1 = 3$. We are given witnesses for 3 distinct challenges γ. For $i \in \{1, 2\}$, from the 3 values D_i' we either find a non trivial relationship between Γ_i, H_i for some i, contradicting Lemma 2.2, or: $v_i''(\gamma) = \gamma v_i + v_i'$. We interpolate $r_C''(\gamma) = r_C' + \gamma r_X + \gamma^2 r_C$, and get:

$$\begin{aligned} C''(\gamma) &= \gamma^2 C + \gamma X + C' = \langle v_1''(\gamma), v_2''(\gamma)\rangle + r_C''(\gamma) H_T \\ &= \gamma^2(\langle v_1, v_2\rangle + r_C H_T) + \gamma(\dots) + (\langle v_1', v_2'\rangle + r_C' H_T) \end{aligned}$$

Since this holds for 3 values of γ, the $1, \gamma, \gamma^{-1}$ coefficients must be equal, which immediately implies we have extracted the required witness. □

Concretely, in Batch-Innerproduct messages from $\mathcal{P}$ to $\mathcal{V}$ have size $|G_T|$; $\mathcal{P}$'s computation is clearly dominated by an $2n$-sized multi-pairing and $\mathcal{V}$'s computation is clearly $O(1)$ exponentiations in $\mathbb{G}_T$.

4 Extending to inner products with public vectors of scalars

In the previous section, we constructed Dory-Innerproduct, a succinct argument of knowledge for generalised inner products between committed vectors in $\mathbb{G}_1^n$ and $\mathbb{G}_2^n$. For a polynomial commitment scheme we also require the ability to prove products of committed vectors with vectors of scalars with multiplicative structure. However, this structure is not preserved when instances are batched, so we will extend our arguments to allow for general vectors in $\mathbb{F}^n$.

So we define a family of languages, parameterised by an additional pair of vectors $s_1, s_2 \in \mathbb{F}^n$:

$$\begin{aligned} (C, D_1, D_2, E_1, E_2) \in \mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}(s_1, s_2) \subset \mathbb{G}_T^3 \times \mathbb{G}_1 \times \mathbb{G}_2 \\ \Leftrightarrow \exists (v_1 \in \mathbb{G}_1^n, v_2 \in \mathbb{G}_2^n, r_C, r_{D_1}, r_{D_2}, r_{E_1}, r_{E_2} \in \mathbb{F}) : \\ D_1 = \langle v_1, \Gamma_2\rangle + r_{D_1} e(H_1, H_2), \quad D_2 = \langle \Gamma_1, v_2\rangle + r_{D_2} e(H_1, H_2), \\ E_1 = \langle v_1, s_2\rangle + r_{E_1} H_1, \quad E_2 = \langle s_1, v_2\rangle + r_{E_2} H_2, \\ C = \langle v_1, v_2\rangle + r_C e(H_1, H_2), \end{aligned}$$

We will show how the arguments of the previous section naturally extend to proving membership in this language. Note that $(C, D_1, D_2, E_1, E_2) \in \mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}(s_1, s_2)$ implies that $(C, D_1, D_2) \in \mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}$.

4.1 General reduction with $O(n)$ cost

There is a reduction from $\mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}(s_1,s_2)$ to $\mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}$, with $O(n)$ cost to $\mathcal{P},\mathcal{V}$:

$\mathsf{Fold\text{-}Scalars}_{n,\Gamma_1,\Gamma_2,H_1,H_2}(C,D_1,D_2,E_1,E_2,s_1,s_2)$

Precompute: $H_T = e(H_1,H_2)$

$\mathcal{P}$ **witness:** $(v_1,v_2,r_C,r_{D_1}r_{D_2},r_{E_1},r_{E_2})$ for
$(C,D_1,D_2,E_1,E_2) \in \mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}(s_1,s_2)$

$\mathcal{V} \to \mathcal{P}$**:** $\gamma \leftarrow_\$ \mathbb{F}$

$\mathcal{P}(**)$**:** $v_1' \leftarrow v_1 + \gamma s_1 H_1$, $\quad v_2' \leftarrow v_2 + \gamma^{-1} s_2 H_2$,
$r_{D_1}' \leftarrow r_{D_1}$, $\quad r_{D_2}' \leftarrow r_{D_2}$,
$r_C' \leftarrow r_C + \gamma r_{E_2} + \gamma^{-1} r_{E_1}$

$\mathcal{V}(**)$**:** $C' \leftarrow C + \langle s_1,s_2\rangle H_T + \gamma e(H_1,E_2) + \gamma^{-1} e(E_1,H_2)$,
$D_1' \leftarrow D_1 + e(H_1,\langle s_1,\gamma\Gamma_2\rangle)$, $\quad D_2' \leftarrow D_2 + e(\gamma^{-1}\langle\Gamma_1,s_2\rangle,H_2)$

$\mathcal{V}$**:** Accept if $(C',D_1',D_2') \in \mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}$

$\mathcal{P}$ **witness:** $(v_1',v_2',r_C',r_{D_1}',r_{D_2}')$

Theorem 4.1. *For $\Gamma_i \leftarrow_\$ \mathbb{G}_i^n$, $H_i \leftarrow_\$ G_i$, Fold-Scalars is an HVSZK, succinct, interactive argument of knowledge for $\mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}(s_1,s_2)$ with witness extended emulation under SXDH.*

Proof. Completeness, Succinctness and the Public-Coin property are immediate. Since all $\mathcal{P}$ messages are independent and uniformly random zero-knowledge is straightforward.

To show witness extended emulation, apply Lemma 2.3, with $\mu = 1$ and $w_1 = 3$. For $i \in \{1,2\}$, from the 3 values D_i' we either find a non trivial relationship between Γ_i, H_i for some i, contradicting Lemma 2.2, or: $v_1'(\gamma) = v_1 + \gamma s_1 H_1$, $v_2'(\gamma) = v_2 + \gamma^{-1} s_2 H_2$. We interpolate $r_C'(\gamma) = r_C' + \gamma r_{E_2} + \gamma^{-1} r_{E_1}$, and get:

$$\begin{aligned} C'(\gamma) &= C + \langle s_1,s_2\rangle H_T + \gamma e(H_1,E_2) + \gamma^{-1} e(E_1,H_2) \\ &= \langle v_1'(\gamma), v_2'(\gamma)\rangle + r_C'(\gamma) H_T \\ &= \langle v_1,v_2\rangle + (r_C' + \langle s_1,s_2\rangle) H_T \\ &\quad + \gamma e(H_1, \langle s_1,v_2\rangle + r_{E_2} H_2) + \gamma^{-1} e(\langle v_1,s_2\rangle + r_{E_1} H_1, H_2). \end{aligned}$$

Since this holds for 3 values of γ, the $1, \gamma, \gamma^{-1}$ coefficients must be equal, which immediately implies we have extracted the required witness. □

4.2 Extending Dory-Reduce

We modify the argument so that $\mathcal{P}$'s first message additionally contains: $E_{1\beta} = \langle\Gamma_1, s_2\rangle$, $E_{2\beta} = \langle s_1, \Gamma_2\rangle$. Prior to their second message, $\mathcal{P}$ samples $r_{E_{\{1,2\}\{+,-\}}} \leftarrow_\$ \mathbb{F}$ and their

second message additionally contains:

$$E_{1+} = \langle v_{1L}, s_{2R} \rangle + r_{E_{1+}} H_1, \qquad E_{1-} = \langle v_{1R}, s_{2L} \rangle + r_{E_{1-}} H_1,$$
$$E_{2+} = \langle s_{1L}, v_{2R} \rangle + r_{E_{2+}} H_2, \qquad E_{2-} = \langle s_{1R}, v_{2L} \rangle + r_{E_{2-}} H_2.$$

After their second message, $\mathcal{P}$ additionally computes:

$$r'_{E_1} \leftarrow r_{E_1} + \alpha r_{E_{1+}} + \alpha^{-1} r_{E_{2-}}, \qquad r'_{E_2} \leftarrow r_{E_2} + \alpha r_{E_{2+}} + \alpha^{-1} r_{E_{2-}}.$$

$\mathcal{V}$ additionally computes:

$$E'_1 \leftarrow E_1 + \beta E_{1\beta} + \alpha E_{1+} + \alpha^{-1} E_{2-}, \quad E'_2 \leftarrow E_2 + \beta^{-1} E_{2\beta} + \alpha E_{2+} + \alpha^{-1} E_{2-},$$

and both $\mathcal{P}$, $\mathcal{V}$ compute the new vectors: $s'_1 \leftarrow \alpha s_{1L} + s_{1R}$, $s'_2 \leftarrow \alpha^{-1} s_{2L} + s_{2R}$

Theorem 4.2. *For $\Gamma'_2 \leftarrow_\$ \mathbb{G}_2^{m-1}$, $H_2 \leftarrow_\$ \mathbb{G}_2$, $\Gamma_1 \leftarrow_\$ \mathbb{G}_2^{m-1}$, $H_1 \leftarrow_\$ \mathbb{G}_1$, the extended Dory-Reduce is an HVSZK, succinct interactive argument of knowledge for $\mathcal{L}_{2^m,\Gamma_1,\Gamma_2,H_1,H_2}(s_1,s_2)$ with witness extended emulation under SXDH.*

Proof. Succinctness and the Public Coin properties are immediate. Completeness and HVSZK holds as in the proof of Theorem 3.2. Witness extended emulation follows directly from Theorem 3.2 as the witnesses for the witness for $(C, D_1, D_2) \in \mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}$ is the witness we require. □

4.3 Extending Dory-Innerproduct

We use the extended Dory-Reduce, and apply Fold-Scalars at $n = 1$:

Dory-Innerproduct$_{\Gamma_{ij},H_i}(C, D_1, D_2, E_1, E_2, s_1, s_2)$

Precompute: $H_T = e(H_1, H_2)$, for all $i \in 0 \ldots m$ compute $\chi_i = \langle \Gamma_{1i}, \Gamma_{2i} \rangle$, and for all $i \in 0 \ldots m-1$ compute:

$$\Delta_{1L,i} = \langle (\Gamma_{1,i})_L, \Gamma_{2,i+1} \rangle, \qquad \Delta_{1R,i} = \langle (\Gamma_{1,i})_R, \Gamma_{2,i+1} \rangle,$$
$$\Delta_{2L,i} = \langle \Gamma_{1,i+1}, (\Gamma_{2,i})_L \rangle, \qquad \Delta_{2R,i} = \langle \Gamma_{1,i+1}, (\Gamma_{2,i})_R \rangle,$$

$\mathcal{P}$ **witness:** $(v_1, v_2, r_C, r_{D_1}, r_{D_2}, r_{E_1}, r_{E_2})$ for $(C, D_1, D_2, E_1, E_2) \in \mathcal{L}_{2^m,\Gamma_{1,0},\Gamma_{2,0},H_1,H_2}(s_1, s_2)$

For $j = 0 \ldots m-1$

$\mathcal{P}, \mathcal{V}$: $(C, D_1, D_2, E_1, E_2, s_1, s_2) \leftarrow$ Dory-Reduce$_{m-j,\Gamma_{1,j}\Gamma_{2,j},\Gamma_{1,j+1},\Gamma_{2,j+1},H_1,H_2}(C, D_1, D_2, E_1, E_2, s_1, s_2)$

$\mathcal{P}, \mathcal{V}$: $(C, D_1, D_2) \leftarrow$ Fold-Scalars$_{\Gamma_{1,m},\Gamma_{2,m},H_1,H_2}(C, D_1, D_2, E_1, E_2, s_1, s_2)$

$\mathcal{P}, \mathcal{V}$: Scalar-Product$_{\Gamma_{1,m},\Gamma_{2,m},H_1,H_2}(C, D_1, D_2)$

Theorem 4.3. *If each Γ_{ij} is uniformly random in $\mathbb{G}_i^{2^{m-j}}$ and $H_i \leftarrow_\$ \mathbb{G}_i$, $\Gamma_1, H_1 \leftarrow_\$ \mathbb{G}_1$, then the extended Dory-Innerproduct is an HVSZK, succinct interactive argument of knowledge for $\mathcal{L}_{2^m,\Gamma_1,\Gamma_2,H_1,H_2}(s_1,s_2)$ with witness extended emulation under SXDH.*

Proof. Succinctness and the Public Coin properties are immediate. Completeness and HVSZK holds as in the proof of Theorem 3.3. Witness extended emulation follows directly from Theorem 3.3 as the witnesses for the witness for $(C, D_1, D_2) \in \mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}$ is the witness we require. □

4.3.1 Concrete costs of the extended Dory-Innerproduct

$\mathcal{P}$ sends 3 additional elements of $\mathbb{G}_1$ and $\mathbb{G}_2$ in each invocation of Dory-Reduce. $\mathcal{P}$ also computes exponentiations of total size $2 \times 2^{m-j}$ exponentiations in $\mathbb{G}_1$ and $\mathbb{G}_2$, and $O(2^{m-j})$ additional field arithmetic. So in total, $\mathcal{P}$'s work is: $(6P+4\mathbb{G}_2+4\mathbb{G}_1+O(1)\mathbb{F}) \times n + o(n)$ which is dominated by the $6n$ pairings, especially as multi-exponentiations in $\mathbb{G}_1, \mathbb{G}_2$ can be accelerated with variants of Pippenger's algorithm. The total size of $\mathcal{P}$'s messages is: $(6|G_T| + 3|G_2| + 3|G_1|) \log n + 4|G_T| + |G_2| + |G_1| + 5|\mathbb{F}|$. As before, $\mathcal{V}$ defers computation to reduce their costs. To compute the C, E_1, E_2 passed to Fold-Scalars requires, respectively, a multi-exponentiation in $\mathbb{G}_T$ of size $9m+9$, a multi-exponentiation in $\mathbb{G}_1$ of size $4m$ and a multi-exponentiation in $\mathbb{G}_2$ of size $4m$. The computation of the final D_1, D_2 and verification of Fold-Scalars and Scalar-Product require 3 additional pairings and $O(1)$ exponentiations. Whilst naively there are 5 pairings, 2 of them are pairings with H_1 and 2 are pairings with H_2, which can be combined in the final check of Scalar-Product.

$\mathcal{V}$ must also compute the final s_1, s_2 used as arguments to Fold-Scalars. In particular, these are the scalars: $\langle s_1, \otimes_{i=0}^{m-1}(\alpha_i, 1)\rangle$, $\langle s_2, \otimes_{i=0}^{m-1}(\alpha_i^{-1}, 1)\rangle$. For general vectors s_1, s_2, these require $O(n)$ operations in $\mathbb{F}$. However, when the vectors s_i themselves have multiplicative structure, we have the identity:

$$\langle \otimes_{i=0}^{m-1}(\ell_i, r_i), \otimes_{i=0}^{m-1}(X_i, 1)\rangle = \prod_{i=0}^{m-1}(\ell_i X_i + r_i),$$

which allows the computation of the product in $O(m)$ operations in $\mathbb{F}$. Similarly, a vector that can be written as a sum of ℓ vectors with multiplicative structure can have this inner product computed in $O(\ell m)$ operations in $\mathbb{F}$ (as in Section 4.4).

4.4 Extending Batch-Innerproduct

$\mathcal{P}$ samples $r_{Y_1}, r_{Y_2} \leftarrow_\$ \mathbb{F}$, and their first message contains in addition to X:

$$Y_1 = \langle v_1, s_2'\rangle + \langle v_1', s_2\rangle + r_{Y_1} H_1, \qquad Y_2 = \langle s_1', v_2\rangle + \langle s_1, v_2'\rangle + r_{Y_2} H_2.$$

After receiving γ, $\mathcal{P}$ computes: $r_{E_1}'' \leftarrow \gamma^2 r_{E_1} + \gamma r_{Y_1} + r_{E_1}'$, $r_{E_2}'' \leftarrow \gamma^2 r_{E_2} + \gamma r_{Y_2} + r_{E_2}'$, and $\mathcal{V}$ computes: $E_1'' \leftarrow \gamma^2 E_1 + \gamma Y_1 + E_1'$, $E_2'' \leftarrow \gamma^2 E_2 + \gamma Y_2 + E_2'$. Both $\mathcal{P}$ and $\mathcal{V}$ compute: $s_1'' \leftarrow \gamma s_2 + s_2'$, $s_2'' \leftarrow \gamma s_1 + s_1'$.

Theorem 4.4. *For $\Gamma_i \leftarrow_\$ \mathbb{G}_i^n$, $H_i \leftarrow_\$ G_i$, the extended Batch-Innerproduct is an HVSZK, succinct interactive argument of knowledge for $\mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}(s_1, s_2) \times \mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}(s_1', s_2')$ with witness extended emulation under SXDH.*

Proof. Succinctness and the Public Coin properties are immediate. Completeness and HVSZK holds as in the proof of Theorem 3.4. Witness extended emulation follows directly from Theorem 3.4 as the witnesses for the witness for $\mathcal{L}^2_{n,\Gamma_1,\Gamma_2,H_1,H_2}$ is the witness we require. □

$\mathcal{P}$'s messages to $\mathcal{V}$ have size $|\mathbb{G}_T| + |\mathbb{G}_2| + |\mathbb{G}_1|$. As before, $\mathcal{P}$'s computation is dominated by a $2n$-size multi-pairing and $\mathcal{V}$'s group operations are $O(1)$ exponentiations. For general vectors s_1, s_2, $\mathcal{V}$ must perform $O(n)$ operations in $\mathbb{F}$. However, if s_i, s_i' are some linear combination of ℓ, ℓ' vectors with multiplicative structure, then s_i'' is a linear combination of $\ell + \ell'$ vectors with multiplicative structure; this representation can be computed in $O(m)$ operations in $\mathbb{F}$.

5 Vector-Matrix-Vector products

Let $n = 2^m$. Fix some commitment scheme for $\mathbb{F}$ and $\mathbb{F}^{n\times n}$, and define:

$$
\begin{aligned}
(\mathcal{C}_M, \mathcal{C}_y, L, R) \in \mathcal{L}_{VMV} \subset \mathbb{G}_T \times \mathbb{G}_T \times \mathbb{F}^n \times \mathbb{F}^n \\
\Leftrightarrow \exists (M \in \mathbb{F}^{n\times n}, y \in \mathbb{F}, \mathcal{S}_M, \mathcal{S}_y) : \\
\mathit{Open}(pp_{\mathbb{F}^{n\times n}}, \mathcal{C}_M, M, \mathcal{S}_M) = 1, \quad \mathit{Open}(pp_{\mathbb{F}}, \mathcal{C}_y, y, \mathcal{S}_y) = 1, \quad y = L^T M R
\end{aligned}
$$

This is a stepping stone to polynomial commitments, which will take L, R to have multiplicative structure. For a batch of ℓ evaluations these vectors will be linear combinations of ℓ vectors with multiplicative structure. We require public parameters pp_{VMV}, generated by the public coin Gen_{VMV}:

$$
\Gamma_{1,0}, \Gamma_{1,fin}, H_1 \leftarrow_\$ \mathbb{G}_1^{2^m} \times \mathbb{G}_1 \times \mathbb{G}_1, \qquad \Gamma_{2,0}, \Gamma_{2,fin}, H_2 \leftarrow_\$ \mathbb{G}_2^{2^m} \times \mathbb{G}_2 \times \mathbb{G}_2,
$$

$$
\begin{aligned}
&\forall i \in 1, \ldots, m: && \Gamma_{1,i} = (\Gamma_{1,i})_L, && \Gamma_{2,i} = (\Gamma_{2,i})_L, \\
&\forall i \in 0, \ldots, m-1: && \Delta_{1L,i} = \Delta_{2L,i} = \langle \Gamma_{1,i+1}, \Gamma_{2,i+1} \rangle, \\
&\forall i \in 0, \ldots, m-1: && \Delta_{1R,i} = \langle (\Gamma_{1,i})_R, \Gamma_{2,i+1} \rangle, && \Delta_{2R,i} = \langle \Gamma_{1,i+1}, (\Gamma_{2,i})_R \rangle, \\
& && \chi = \sum_{j=0}^{m-1} \langle \Gamma_{1,j}, \Gamma_{2,j} \rangle, && \chi_{fin} = \langle \Gamma_{1m}, \Gamma_{2m} \rangle \\
& && H_T = e(H_1, H_2) && \Upsilon = e(H_1, \Gamma_{2,fin})
\end{aligned}
$$

Recall the matrix commitment from Section 2.4. Explicitly, we fix parameters $pp_{\mathbb{F}^{n\times n}} = \{\Gamma_{1,0}, H_1, \Gamma_{2,0}, H_2\}$ for this commitment. As noted in Section 2.4, if $\mathsf{Commit}(pp, M) = (T, (r_{rows}, r_{fin}, T'))$, then $T' \in \mathbb{G}_1^n$ is a vector of hiding Pedersen commitments to the rows of M with generators $(\Gamma_{1,0}\ ;\ H_1)$, and T is a hiding AFGHO commitment to T' with generators $(\Gamma_{2,0}\ ;\ H_2)$. So T is a hiding commitment to M. The alert reader may note that T' depends only on M and r_{rows}; since it is in the opening hint retained by $\mathcal{P}$ it need not be recomputed in Eval-VMV.

The general strategy for Eval-VMV is as follows. The commitment to the evaluation $y = L^T M R$ will be a Pedersen commitment with parameters $pp_{\mathbb{F}} = (\Gamma_{1,fin}, H_1)$, so the commitment $y_{com} = y\Gamma_{1,fin} + r_y H_1$. Now, $\mathcal{P}$ can compute the vector $v = L^T M$, and by construction $y = L^T M R = \langle v, R \rangle$. Since Pedersen commitments are linearly homomorphic: $v_{com} = \langle L, C' \rangle = C_{\Gamma_{1,0}:H_1}(v\ ;\ \langle L, r_{\text{rows}} \rangle)$ is a hiding, binding commitment to v with blind $r_v = \langle L, r_{\text{rows}} \rangle$. Recall also that T is a hiding, binding commitment to

$T' \in \mathbb{G}_1^n$. So to prove that T, y_{com} are commitments to M, y and that $y = L^T MR$, we will have $\mathcal{P}$ prove knowledge of:

$$\begin{aligned} T' \in \mathbb{G}_1^n, v \in \mathbb{F}^n, r_v, r_{\text{fin}}, r_y \in \mathbb{F} : \\ T = \langle T', \Gamma_2 \rangle + r_{\text{fin}} H_T \\ \langle L, T' \rangle = \langle v, \Gamma_1 \rangle + r_v H_1 \quad (= v_{com}) \\ y_{com} = \langle v, R \rangle \Gamma_{1,fin} + r_y H_1 \end{aligned}$$

To reduce proving knowledge of this to $\mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2}(L,R)$, we will implicitly use the vector $v\Gamma_{2,fin} \in \mathbb{G}_2^n$. Whilst this prevents witness-extracting v directly, we will show that M can still be extracted if $\mathcal{P}$ shows evaluations at many L, R. We also make use of auxiliary Σ-protocols to prove knowledge of logarithms.

$\mathsf{Eval\text{-}VMV}_{pp_{VMV}}(T, y_{com}, L, R)$

$\mathcal{P}$ **witness:** $M, (T', r_{rows}, r_{fin}), r_y$

$\mathcal{P}$**:** $v = L^T M,\ r_v = \langle L, r_{rows} \rangle,\ y = \langle v, R \rangle,\ r_C, r_{D_2}, r_{E_1}, r_{E_2} \leftarrow_{\$} \mathbb{F}$

$\mathcal{P} \to \mathcal{V}$**:** $C = e(\langle v, T' \rangle, \Gamma_{2,fin}) + r_C H_T,$
$D_2 = e(\langle \Gamma_1, v \rangle, \Gamma_{2,fin}) + r_{D_2} H_T$
$E_1 = \langle L, C' \rangle + r_{E_1} H_1,$
$E_2 = y\Gamma_{2,fin} + r_{E_2} H_2,$

$\mathcal{P}, \mathcal{V}$**:** Σ-protocol showing $\mathcal{P}$ knows $s \in \mathbb{F}^3$:
$E_2 = s_1 \Gamma_{2,fin} + s_2 H_2 \wedge y_C = s_1 \Gamma_{1,fin} + s_3 H_1$
$\mathcal{P}$ **witness:** $s = (y, r_{E_2}, r_y)$

$\mathcal{P}, \mathcal{V}$**:** Σ-protocol showing $\mathcal{P}$ knows $t \in \mathbb{F}^2$:
$e(E_1, \Gamma_{2,fin}) - D_2 = e(H_1, t_1 \Gamma_{2,fin} + t_2 H_2)$
$\mathcal{P}$ **witness:** $t = (r_{E_1} + r_v, -r_{D_2})$

$\mathcal{P}, \mathcal{V}$**:** $\mathsf{Dory\text{-}Innerproduct}_{\Gamma_{ij}, H_i}(C, T, D_2, E_1, E_2, L, R)$.

$\mathcal{P}$ **witness:** $(T', v\Gamma_{2,fin}, r_C, r_{\text{fin}}, r_{D_2}, r_{E_1}, r_{E_2})$

Theorem 5.1. *For pp_{VMV} generated as above, if $\mathcal{P}$ commits to M as above, then for any $L, R \in \mathbb{F}^{n \times n}$, $\mathcal{P}$ can send a commitment y_{com} to $L^T MR$ and use Eval-VMV to argue that $\mathcal{P}$ knows M this is true. This argument is a complete, HVSZK interactive argument of knowledge with witness extended emulation under SXDH.*

Proof. Completeness is straightforward from the definition of $\mathcal{P}$'s witnesses. Succinctness, the Public Coin property and honest-verifier statistical zero-knowledge of Eval-VMV follow straightforwardly from the same properties for the two auxiliary Σ-protocols and Dory-Innerproduct.

We apply Lemma 2.3, taking $\mu = 1$ and $w_1 = 2n^2$. We then witness extract Dory-Innerproduct and the two sigma proofs in Eval-VMV. Since we have $2n^2$ pairs L^i, R^i,

the maps $RL^T \in (\mathbb{F}^{n\times n})^*$ do not span the dual space with negl(λ) probability. For each of $2n^2$ cases, we have:

$$\begin{aligned}
&v_1 \in \mathbb{G}_1^n, v_2 \in \mathbb{G}_2^n, S, y_C \in \mathbb{G}_1, D_2 \in \mathbb{G}_2, E_1, E_2 \in \mathbb{G}_T,\\
&y, s_2, r_y, t_1, t_2, r_C, r_{D_1}, r_{D_2}, r_{E_1}, r_{E_2} \in \mathbb{F} :\\
E_2 &= y\Gamma_2' + s_2 H_2, && (1)\\
y_C &= y\Gamma_1' + r_y H_1 && (2)\\
e(E_1, \Gamma_2') - D_2 &= e(H_1, t_1\Gamma_2' - t_2 H_2) && (3)\\
C &= \langle v_1, v_2\rangle + r_C e(H_1, H_2) && (4)\\
T &= \langle v_1, \Gamma_2\rangle + r_{D_1} e(H_1, H_2) && (5)\\
D_2 &= \langle \Gamma_1, v_2\rangle + r_{D_2} e(H_1, H_2) && (6)\\
E_1 &= \langle L, v_1\rangle + r_{E_1} H_1 && (7)\\
E_2 &= \langle R, v_2\rangle + r_{E_2} H_2 && (8)
\end{aligned}$$

Since T is constant in (5), v_1, r_{D_1} must be constant as functions of L^i, R^i, as otherwise we obtain a non-trivial relationship between Γ_2, H_2, contradicting Lemma 2.2. Then substituting (6, 7) into (3) we have:

$$e(\langle L, v_1\rangle, \Gamma_2') = \langle \Gamma_1, v_2\rangle + e(H_1, (r_{D_2} - t_2)H_2 - (r_{E_1} - t_1)\Gamma_2') \quad (9)$$

Then if v_2 is not linear in L and independent of R, we can eliminate L and obtain a non-trivial relationship between Γ_1, H_1, contradicting Lemma 2.2. From (1, 8) we have: $\langle R, v_2\rangle = y\Gamma_2' + (s_2 - r_{E_2})H_2$, and so if y and $s_2 - r_{E_2}$ are not bilinear in L, R we obtain a non-trivial relationship between Γ_2', H_2, contradicting Lemma 2.2. In particular we extract matrices $M, B \in \mathbb{F}^{n\times n}$ such that $y = L^T MR$ and $s_2 - r_{E_2} = L^T BR$ in all cases, and so $v_2 = L^T M\Gamma_2' + L^T BH_2$. Substituting into (9), we have:

$$e(\langle L, v_1 - M\Gamma_1\rangle + (r_{E_1} - t_1)H_1, \Gamma_2') = e(\langle L^T B, \Gamma_1\rangle + (r_{D_2} - t_2)H_1, H_2)$$

and so either we find a non-trivial pairing relationship between Γ_2', H_2, contradicting Lemma 2.2, or:

$$\begin{aligned}
0 &= \langle L, v_1 - M\Gamma_1\rangle + (r_{E_1} - t_1)H_1\\
0 &= \langle L^T B, \Gamma_1\rangle + (r_{D_2} - t_2)H_1
\end{aligned}$$

We apply a similar argument to look for a non-trivial scalar relationship between the Γ_1, H_1, which would violate Lemma 2.1. If one is not found, then from the second equation $r_{D_2} = t_2$ and $L^T B$ is identically 0, so $B = 0$. From the first we deduce that $r_{E_1} - t_1$ must be a linear function of L and independent of R, so we have some $r_{\text{rows}} \in \mathbb{F}^n$ such that $r_{E_1} - t_1 = L^T r_{\text{rows}}$, which implies that $v_1 = M\Gamma_1 + r_{\text{rows}} H_1$. So from (5) we have:

$$T = \langle M\Gamma_1, \Gamma_2\rangle + e(H_1, r_{D_2} H_2 + \langle r_{\text{rows}}, \Gamma_2\rangle),$$

which precisely states that T is a commitment to M with opening hint $(r_{\text{rows}}, r_{\text{fin}} = r_{D_2}, T' = v_1)$. Furthermore in each case $y_C = (L^T MR)\Gamma_1' + r_y H_1$, so is a commitment to the desired evaluation. So we have extracted a matrix M and evaluations y consistent with the claimed evaluations. Since the vectors $L_i{}^*(R_i{}^*)$ span A, the projection of M onto A^* is uniquely defined. □

Remark 5.1. *In the above, it suffices for the matrices RL^T to span $\mathbb{F}^{n\times n}$. This means that $\mathcal{V}$ can restrict their choices of L, R substantially.*

Remark 5.2. *As observed in Remark 2.4, extractability in this case requires that RL^T is not fixed by some adversary. If required, this randomization can be achived here. Concretely, $\mathcal{V}$ can sample some $L', R' \in e_1, \ldots, e_n$. Then $\mathcal{P}$ responds with claimed commitments $\mathcal{C}_2, \mathcal{C}_1$ to ${L'}^T M R'$ and to ${L'}^T M R + L^T M R'$, and $\mathcal{V}$ samples some $\gamma \leftarrow_\$ \mathbb{F}$. The standard argument is applied to $L = L + \gamma L'$, $R = R + \gamma R'$, $y_{com} \leftarrow y_{com} + \gamma \mathcal{C}_1 + \gamma \mathcal{C}_2$. Plainly the matrix RL^T then has a α^2 contribution only at one point, so across $O(n^2 \log n)$ samples spans $\mathbb{F}^{n\times n}$.*

5.0.1 Batching

From Section 4.4, we can batch multiple invocations of Dory-Innerproduct and so we similarly have an argument for a batch of Eval-VMV.

We can further optimize this batch argument by observing that the Sigma proofs in Eval-VMV show knowledge of logarithms with respect to fixed bases $\Gamma_{2,fin}, H_2, \Gamma_{1,fin}, H_1$. So by usual arguments these proofs can be linearly combined with random challenges supplied by $\mathcal{V}$, without altering soundness.

5.0.2 Concrete costs

For an $n \times n$ matrix M, the size of the public parameters is $(n+2)|\mathbb{G}_1| + (n+2)|\mathbb{G}_2| + (3\log n + 4)|\mathbb{G}_T|$, and running Gen requires sampling $n+2$ elements of $\mathbb{G}_1$, $n+2$ elements of $\mathbb{G}_2$, $3n$ pairings and $\log n$ additions in $\mathbb{G}_T$.

Running Commit on the matrix M requires sampling $n+1$ elements of $\mathbb{F}$, n multi-exponentiations of size $n+1$ in $\mathbb{G}_1$, a multi-pairing of size n, and an exponentiation and addition in $\mathbb{G}_T$. Since the n multi-exponentiations in $\mathbb{G}_1$ are over fixed generators $(\Gamma_1 || H_1)$, Pippenger-type savings of a factor $2\log n$ are available.

In addition to Dory-Innerproduct, computing Eval-VMV requires $\mathcal{P}$ perform three exponentiations in $\mathbb{G}_1$ of size n and $O(1)$ additional exponentiations in $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$. The messages from $\mathcal{P}$ to $\mathcal{V}$ have size $5|\mathbb{F}| + 2|\mathbb{G}_1| + 2|\mathbb{G}_2| + 3|G_T|$, and $\mathcal{V}$'s computation is 5 exponentiations in $|\mathbb{G}_2|$, and 3 exponentiations in $|\mathbb{G}_1|$, an exponentiation in $\mathbb{G}_T$ and 2 pairings.

6 Dory-PC

We recall the discussion in Section 2.5. Concretely, the evaluation of any multivariate polynomial in $x_1 \ldots x_\ell$ of degrees $d_1, \ldots, d_\ell$ at some point r can be replaced by the evaluation of a multilinear polynomial in $r = \sum_i \lceil \log(d_i + 1) \rceil$ variables, where the coefficients of the two polynomials are equal. Given a multilinear polynomial f on r variables, we observe that:

$$f(\vec{x}) = \sum_{b \in \{0,1\}^r} f_b \prod_i x_i^{b_i} = \sum_{v \in \{0,1\}^r} f(v) \prod_i (x_i v_i + (1 - x_i)(1 - v_i))$$

which provides for the evaluation of the polynomial as a contraction of a $2 \times \ldots \times 2$ tensor $T^{i_1 \ldots i_r}$ with vectors given either the coefficients of f or its evaluations on $\{0,1\}^r$.

Let $m = \lceil r/2 \rceil$. In either case, we can define a $2^m \times 2^m$ matrix M_{ij} by:

$$M_{ij} = T^{i_1 \dots i_r} \text{ if } i2^r + j = \sum_{k=1}^{r} i_k 2^{r-k}, \qquad M_{ij} = 0 \text{ otherwise.}$$

and then $f(\vec{x}) = (1-z)L^T MR$ where $L = \otimes_{i=1}^{m}(\ell_i, r_i)$ and $z = 0$ for r even, $L = (1,z) \otimes \left(\otimes_{i=1}^{m-1}(\ell_i, r_i)\right)$ for $z \leftarrow_\$ \mathbb{F}$ for r odd, and $R = \otimes_{i=r-m+1}^{r}(\ell_i, r_i)$. In the case where we are given the coefficients of f, we take $\ell_i = 1, r_i = x_i$. If we are given the evaluations of f on the $\{0,1\}$ cube, we take $\ell_i = 1 - x_i$, $r_i = x_i$. Note that the implicit extension to a polynomial in $2m$ variables has no impact, as the additional variable is unconditionally set to 0. So we have reduced polynomial evaluation to a vector-matrix-vector product, where the vectors L, R have multiplicative structure.

Dory − PC directly uses the commitment scheme of Section 2.4, and uses Eval-VMV as Eval.

Theorem 6.1. *Dory − PC is an honest-verifier, statistical zero-knowledge, extractable polynomial commitment scheme for r-variable multilinear polynomials.*

Proof. Note that for uniformly random $x \in \mathbb{F}^\ell$ (and $z \leftarrow_\$ \mathbb{F}$ for r odd), the vectors L, R are random vectors with multiplicative structure. By the Schwartz-Zippel lemma, the outer products RL^T span $\mathbb{F}^{n \times n}$, as otherwise there would be some non-zero polynomial vanishing for all x.

Theorem 5.1 and Remark 5.1 then complete the proof. □

Since the vectors L, R have multiplicative structure, the remarks made in Section 4.3.1 apply; $\mathcal{V}$'s use of the vectors L, R are restricted to computing products:

$$\langle L, \otimes_{i=0}^{m-1}(\alpha_i, 1)\rangle, \qquad \langle R, \otimes_{i=0}^{m-1}(\alpha_i^{-1}, 1)\rangle$$

which can be computed in $O(m)$ operations in $\mathbb{F}$ given x, α_i, α_i^{-1}.

6.1 Concrete costs

Let $n = \prod_i (d_i + 1)$, and let $|M| = O(n)$ be the number of non-zero entries in the matrix M. In the worst case $d_i = 4$ and $m = \frac{3}{2\log 5}\log n + O(1)$. For multilinear or univariate polynomials $m = \frac{1}{2}\log n + O(1)$.

Using the fact that the $2^m \times 2^m$ matrix has at most $|M|$ non-zero entries, $\mathcal{P}$'s time to run Commit is dominated by $|M| + 2^m$ exponentiations in $\mathbb{G}_1$ and 2^m pairings. From Section 5.0.2, $\mathcal{P}$'s time to run Eval is dominated by $O(2^m)$ pairings.

In the interactive setting, the size of the messages from the Prover to the Verifier is $(6m+7)|G_T| + (3m+3)(|\mathbb{G}_2| + |\mathbb{G}_1|) + 8|\mathbb{F}|$, and the Verifier computation is an $9m + O(1)$ sized multi-exponentiation in $\mathbb{G}_T$ and $O(1)$ additional exponentiations and pairings. To construct a polynomial commitment, we compile with the Fiat-Shamir heuristic, in which case the communication complexity is the size of the $\mathcal{P}$ to $\mathcal{V}$ messages, and both $\mathcal{P}, \mathcal{V}$ must do $O(m)$ additional work to compute the Verifier challenges.

6.2 Batching

Given a batch of ℓ polynomials with individual $m_i \leq m$, we can use batching to reduce the communication complexity and computational work of $\mathcal{V}$ substantially. Using the approach of Section 5.0.1, the total size of the messages from $\mathcal{P}$ to $\mathcal{V}$ is:

$$(6m + 3\ell + 5)|\mathbb{G}_T| + (3m + 2\ell + 2)(|\mathbb{G}_2| + |\mathbb{G}_1|) + 8|\mathbb{F}|,$$

The complexity of $\mathcal{P}$ remains $O(\ell \times 2^m)$ pairings, though concretely a factor of 3 is saved in the implied constant. However, deferring $\mathcal{V}$'s computations as before, the $\mathcal{V}$'s group operations can be reduced to an exponentiation in $\mathbb{G}_T$ of size $9m+3\ell+6$, exponentiations in $\mathbb{G}_1$ and $\mathbb{G}_2$ of size $3m + 2\ell + 2$, and a multi-pairing of size 4. Unfortunately, the computations with vectors L, R cannot be efficiently batched, and so $\mathcal{V}$ still performs $2\ell m$ multiplications and additions in $\mathbb{F}$.

7 Implementation

We implemented Dory to provide polynomial commitments for dense multilinear polynomials, building on framework for non-interactive arguments and dense multilinear polynomials in the Spartan library [34]. This took $\sim$ 3400 LOC. Our implementation used the BLS12-381 curve as implemented in RELIC [3]. Our Rust wrapper for RELIC implements fast algorithms for computing (multiple) multi-exponentiations and torus based compression for serialization of elements of $\mathbb{G}_T$ in $\sim$ 3200 LOC.

The implementation was evaluated on a machine with an Intel Xeon E5–1660 v4 CPU at 3.2 GHz. All measurements were taken for a single core. We compare with Spartan-PC, a discrete-log based extractable polynomial commitment scheme implemented in the Spartan library [34], which is a highly optimized derivative of the commitment scheme in [37] using Curve25519 as implemented by curve25519-dalek for its curve arithmetic.

Prover Time for Commit: We report results for a variety of polynomial sizes in Figure 3. As can be seen, Dory is slower than the baseline, by a consistent factor ~ 5, matching the relative speed of $\mathbb{G}_1$ arithmetic on the implementations of Curve25519 and BLS12-381 as seen in Figure 2.

	2^{10}	2^{12}	2^{14}	2^{16}	2^{18}	2^{20}	2^{22}	2^{24}	2^{26}	2^{28}	2^{30}
Spartan-PC	0.0123	0.0365	0.145	0.451	1.58	5.57	19.8	74.3	289	1190	4530
Dory-PC	0.607	0.157	0.525	1.89	6.57	23.7	87.4	346	1780	6650	26100

FIGURE 3—$\mathcal{P}$'s performance (in s) for varying sizes of multilinear polynomials.

Commitment Size: We report results for a variety of polynomial sizes in Figure 4. Unsurprisingly, Dory has a constant sized commitment whilst the commitment of Spartan-PC grows as $n^{1/2}$, so for all tested sizes the Dory' commitment is smaller.

	2^{10}	2^{12}	2^{14}	2^{16}	2^{18}	2^{20}	2^{22}	2^{24}	2^{26}	2^{28}	2^{30}
Spartan-PC	1032	2056	4104	8200	16392	32776	65544	131080	262152	524296	1048584
Dory-PC	192	192	192	192	192	192	192	192	192	192	192

FIGURE 4—Commitment size (in bytes) for varying sizes of multilinear polynomials.

Prover Time for Eval: We report results for a variety of polynomial sizes in Figure 5. Note that in both cases, the $O(n)$ time to evaluate the polynomials is included; for Spartan-PC this linear scaling becomes dominant for $n \geq 2^{20}$, whilst for $2^{20} \geq n \geq 2^{12}$ the $O(n^{1/2})$ scaling of the group operations is apparent. Across the range 2^{12}–2^{20}, Dory is concretely less efficient by factors of $\sim$ 40–100. This should be unsurprising, as Dory performs 6 pairings and 3 multiplications in each of $\mathbb{G}_1, \mathbb{G}_2$ for each pair of multiplications Spartan-PC performs in $\mathbb{G}_1$; scaling from the micro-benchmarks in Figure 2 would suggest that Dory should be $\sim 100\times$ slower in this context.

	2^{10}	2^{12}	2^{14}	2^{16}	2^{18}	2^{20}	2^{22}	2^{24}	2^{26}	2^{28}	2^{30}
Spartan-PC	4.23ms	5.03ms	10.1ms	22.0ms	53.6ms	0.157	0.623	2.63	10.3	54.1	248.7
Dory-PC	0.247	0.475	0.853	1.62	3.11	5.97	11.8	24.5	53.5	132	419

FIGURE 5—$\mathcal{P}$'s performance (in s) to prove the evaluation of varying sizes of multilinear polynomials.

Proof size: We report results for a variety of polynomial sizes in Figure 6. Dory' proofs are consistently larger than those of Spartan-PC by a factor ~ 24. This is this is the ratio between $6|\mathbb{G}_T| + 3(|\mathbb{G}_2| + |\mathbb{G}_1|)$ in the BLS12-381 curve and $2|\mathbb{G}_1|$ in Curve25519, and so is the ratio between the $\log n$ contributions to the proof size in the two systems.

	2^{10}	2^{12}	2^{14}	2^{16}	2^{18}	2^{20}	2^{22}	2^{24}	2^{26}	2^{28}	2^{30}
Spartan-PC	464	528	592	656	720	784	848	912	976	1040	1104
Dory-PC	10048	11632	13216	14800	16384	17968	19552	21136	22720	24304	25888

FIGURE 6—Proof size (in bytes) for varying sizes of multilinear polynomials.

Verifier Time for Eval: We report results for a variety of polynomial sizes in Figure 7. Unsurprisingly, Dory's $\mathcal{V}$ shows $O(\log n)$ complexity, concretely running in $\sim 17.5(10 + \log n)ms$. The Verifier of Spartan-PC scales like $n^{1/2}$, and is concretely slower that Dory's Verifier for $n > 2^{24}$.

	2^{10}	2^{12}	2^{14}	2^{16}	2^{18}	2^{20}	2^{22}	2^{24}	2^{26}	2^{28}	2^{30}
Spartan-PC	1.65	1.72	3.05	5.21	8.96	16.4	29.7	55.9	110	216	433
Dory-PC	35.7	40.9	44.9	48.1	51.6	55.1	58.5	61.2	64.3	68.6	71.4

FIGURE 7—Verifier performance (in ms) to verify an evaluation of varying sizes of multilinear polynomial.

Batching: To validate the benefits of batching, we use the batch argument to open multiple committed polynomial evaluations. This naturally impacts the time taken for Prover to run Eval, the resulting proof size, and the Verifier's time taken to run Eval on the batch. We report results for a variety of batch sizes in Figure 8.

As can be seen, the marginal costs to increase the batch size are small; the marginal Prover time is $\sim 640ms$, the marginal contribution to the proof size is 912 bytes, and the marginal Verifier time is $\sim 2.2ms$. On the Prover, this speedup is a by a constant factor $\sim 9.5\times$ over proving each evaluation separately; for proof sizes and the Verifier this is an asymptotic saving by a factor $\log n$.

Batch size	1	2	3	4	5	6	7	8	9	10
Prover Time (s)	6.04	6.71	7.36	7.96	8.70	9.28	9.89	10.5	11.2	11.8
Proof Size (bytes)	19208	20120	21032	21944	22856	23768	24680	25592	26504	27416
Verifier Time (ms)	48.2	50.6	53.2	54.7	56.5	57.9	59.5	62.7	64.3	67.8

FIGURE 8—Prover time, proof size and verification time to validate variable sized batches of multilinear polynomial evaluations of size 2^{20}

References

[1] Rust-crypto. https://github.com/DaGenix/rust-crypto/, 2017.

[2] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. Structure-preserving signatures and commitments to group elements. In T. Rabin, editor, *CRYPTO*, 2010.

[3] D. F. Aranha, C. P. L. Gouvêa, T. Markmann, R. S. Wahby, and K. Liao. RELIC is an Efficient LIbrary for Cryptography. https://github.com/relic-toolkit/relic.

[4] P. S. L. M. Barreto, B. Lynn, and M. Scott. Constructing elliptic curves with prescribed embedding degrees. In S. Cimato, G. Persiano, and C. Galdi, editors, *Security in Communication Networks*, 2003.

[5] P. S. L. M. Barreto and M. Naehrig. Pairing-friendly elliptic curves of prime order. In B. Preneel and S. Tavares, editors, *Selected Areas in Cryptography*, 2006.

[6] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, 2018.

[7] E. Ben-Sasson, A. Chiesa, A. Kazorian, D. Ojha, A. Popovs, M. Riabzev, N. Spooner, M. Virza, and N. Ward. libiop: a C++ library for IOP-based zkSNARKs. https://github.com/relic-toolkit/relic.

[8] D. J. Bernstein. Curve25519: New diffie-hellman speed records. In *PKC*, 2006.

[9] D. J. Bernstein, M. Hamburg, A. Krasnova, and T. Lange. Elligator: Elliptic-curve points indistinguishable from uniform random strings. In *CCS*, 2013.

[10] J.-F. Biasse, M. J. Jacobson, Jr., and A. K. Silvester. Security estimates for quadratic field based cryptosystems. In *ACISP*, 2010.

[11] D. Boneh, B. Bünz, and B. Fisch. A survey of two verifiable delay functions. *IACR Cryptology ePrint Archive*, 2018, 2018.

[12] 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.

[13] S. Bowe. Bls12-381: New zk-snark elliptic curve construction. https://electriccoin.co/blog/new-snark-curve/, 2013.

[14] S. Bowe, J. Grigg, and D. Hopwood. Recursive proof composition without a trusted setup. Cryptology ePrint Archive, Report 2019/1021, 2019. https://eprint.iacr.org/2019/1021.

[15] B. Bunz, J. Bootle, D. Boneh, A. Poelstra, and G. Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *S&P*, 2018.

[16] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *S&P*, 2018.

[17] B. Bünz, B. Fisch, and A. Szepieniec. Transparent snarks from dark compilers. Cryptology ePrint Archive, Report 2019/1229, 2019. https://eprint.iacr.org/2019/1229.

[18] B. Bünz, M. Maller, P. Mishra, and N. Vesely. Proofs for inner pairing products and applications. Cryptology ePrint Archive, Report 2019/1177, 2019. https://eprint.iacr.org/2019/1177.

[19] A. Chiesa, D. Ojha, and N. Spooner. Fractal: Post-quantum and transparent recursive proofs from holography. Cryptology ePrint Archive, Report 2019/1076, 2019.

https://eprint.iacr.org/2019/1076.
[20] S. Dobson, S. D. Galbraith, and B. Smith. Trustless groups of unknown order with hyperelliptic curves. Cryptology ePrint Archive, Report 2020/196, 2020. https://eprint.iacr.org/2020/196.
[21] A. Gabizon, Z. J. Williamson, and O. Ciobotaru. Plonk: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. *IACR Cryptol. ePrint Arch.*, 2019, 2019.
[22] S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113 – 3121, 2008.
[23] O. Goldreich, A. Sahai, and S. Vadhan. Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge. In *STOC*, 1998.
[24] J. Groth. Efficient zero-knowledge arguments from two-tiered homomorphic commitments. In D. H. Lee and X. Wang, editors, *ASIACRYPT*, 2011.
[25] J. Groth and Y. Ishai. Sub-linear zero-knowledge argument for correctness of a shuffle. In *EUROCRYPT*, 2008.
[26] W. Hart. Antic – algebraic number theory in c. https://github.com/wbhart/antic, 2013.
[27] T. Icart. How to hash into elliptic curves. In *CRYPTO*, 2009.
[28] M. J. Jacobson, Jr. and A. J. v. d. Poorten. Computational aspects of NUCOMP. In *ANTS*, 2002.
[29] A. Kate, G. M. Zaverucha, and I. Goldberg. Constant-size commitments to polynomials and their applications. In *ASIACRYPT*, 2010.
[30] H. Lipmaa. On diophantine complexity and statistical zero-knowledge arguments. In *ASIACRYPT*, 2003.
[31] I. A. Lovecruft and H. de Valence. curve25519-dalek: A pure-rust implementation of group operations on ristretto and curve25519. https://github.com/dalek-cryptography/curve25519-dalek, 2018.
[32] M. Naehrig, P. S. L. M. Barreto, and P. Schwabe. On compressible pairings and their computation. In S. Vaudenay, editor, *AFRICACRYPT*, 2008.
[33] C. Papamanthou, E. Shi, and R. Tamassia. Signatures of correct computation. In *TCC*, 2013.
[34] S. Setty. Spartan: Efficient and general-purpose zksnarks without trusted setup. In D. Micciancio and T. Ristenpart, editors, *CRYPTO*, 2020.
[35] A. Shallue and C. E. van de Woestijne. Construction of rational points on elliptic curves over finite fields. In *ANTS*, 2006.
[36] M. Tibouchi. Elligator squared: Uniform points on elliptic curves of prime order as uniform random strings. In *FOCS*, 2014.
[37] R. S. Wahby, I. Tzialla, A. Shelat, J. Thaler, and M. Walfish. Doubly-efficient zkSNARKs without trusted setup. In *S&P*, 2018.
[38] J. Zhang, T. Xie, Y. Zhang, and D. Song. Transparent polynomial delegation and its applications to zero knowledge proof. In *S&P*, 2020.