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

Jonathan Lee

Presentation: Marianna Spyrakou

Advanced Topics in Cryptography
Spring 2023
National Technical University of Athens

June 12, 2023

Table of Contents

Table of Contents

In this work...

We will see Dory,

- a transparent setup,
- public coin,
- interactive argument,
- for inner-pairing products between commited vectors of elements of two source groups

In this work...

For a product of vectors of length n:

- **Proofs** are:
  - $6 \log n$ target group elements and
  - $O(1)$ additional elements.
- **Verifier work** is dominated by:
  - $O(\log n)$ multiexponentiation in the target group and
  - $O(1)$ pairings
- **Security** is reduced to the standard SXDH assumption in the standard model.

In this work...

Apply Dory to build a **multivariate polynomial commitment scheme** via the Fiat-Shamir transform.
For a dense polynomial with n coefficients

- **Prover work** to compute a **commitment** is donimated by a multiexponentiation 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
  - $O(n^{1/2})$ for univariate or multilinear polynomials
- **Communication Complexity:** $O(\log(n))$
- **Verifier work:** $O(\log(n))$

In this work...

These arguments can be **batched**!

- to validate ℓ polynomial evaluations for polynomials of size at most n
- $O(\ell + \log n)$ exponentiations
- $O(\ell \log n)$ field operations

In this work...

Dory is inspired by "Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting" of Bootle et al. but applies new techniques to achieve logarithmic verifier complexity . It can be applied to:

- give polynomial commitments for arbitraty number of variables
- matrix commitment strategy
- give commitment to univariate and bivariate polynomials

Operations for transparent polynomial commitment schemes

1. $\mathcal{P}$ and $\mathcal{V}$ generate public parameters

2. $\mathcal{P}$ must commit to a polynomial and transmit that commitment to $\mathcal{V}$

3,4. $\mathcal{P}$ and $\mathcal{V}$ must compute, transmit and verify a proof of evaluation og the polynomial

Comparison of Polynomial Commitment Schemes

	Transparent Setup?	Communication Complexity		Time Complexity			
		Commit	Eval	Gen	Commit	Eval ($\mathcal{P}$)	Eval ($\mathcal{V}$)
LCC-DLOG	✓	$n^{1/2}\ \lvert\mathbb{G}\rvert$	$\log n\ \lvert\mathbb{G}\rvert$	$n^{1/2}\mathbb{H}$	$n\ \mathbb{G}$	$n^{1/2}\ \mathbb{G}$	$n^{1/2}\ \mathbb{G}$
RS-IOP	✓	$1\ \lvert\mathbb{H}\rvert$	$\log^2 n\ \lvert\mathbb{H}\rvert$	1	$n \log n\ \mathbb{H}$	$n \log n\ \mathbb{H}$	$\log^2 n\ \mathbb{H}$
DARK-GUO	✓	$1\ \lvert\mathbb{G}_U\rvert$	$\log n\ \lvert\mathbb{G}_U\rvert$	$n \log n\ \mathbb{G}_U$	$n\ \mathbb{G}_U$	$n \log n\ \mathbb{G}_U$	$\log n\ \mathbb{G}_U$
KZG [32,36]	✗	$1\ \lvert\mathbb{G}_T\rvert$	$\log n\ \lvert\mathbb{G}_T\rvert$	$n\ \mathbb{G}_1$	$n\ \mathbb{G}_1$	$n\ \mathbb{G}_1$	$r\ P$
GIPP [20]	✗	$1\ \lvert\mathbb{G}_T\rvert$	$\log n\ \lvert\mathbb{G}_T\rvert$	$n^{1/2}\ \mathbb{G}_1$	$n\ \mathbb{G}_1$	$n^{1/2}\ P$	$\log n\ \mathbb{G}_T$
This work	✓	$1\ \lvert\mathbb{G}_T\rvert$	$\log n\ \lvert\mathbb{G}_T\rvert$	$n^{1/2}P$	$n\ \mathbb{G}_1$	$n^{1/2}P$	$\log n\ \mathbb{G}_T$

Fig. 1: Asymptotic comparisons for dense univariate polynomials of degree n, neglecting Pippenger-type savings in groups. We report the most expensive dominant operations for the most efficient instantiations of each class. $\mathbb{H}$ 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. $\mathbb{G}_U$ is a group of unknown order. These schemes all generalise to multivariate polynomials with degree sequence $(d_1, \ldots, d_r)$, setting $n = \prod_i (d_i + 1)$

Key idea of LCC-DLOG techniques

For any vectors $\overrightarrow{u}_L, \overrightarrow{u}_R, \overrightarrow{v}_L, \overrightarrow{v}_R$ and any non-zero scalar α

$$\langle \overrightarrow{u}, \overrightarrow{v} \rangle = \langle \overrightarrow{u}_L || \overrightarrow{u}_R, \overrightarrow{v}_L || \overrightarrow{v}_R \rangle =$$
$$= \langle \alpha \overrightarrow{u}_L + \overrightarrow{u}_R, \alpha^{-1} \overrightarrow{v}_L + \overrightarrow{v}_R \rangle - \alpha \langle \overrightarrow{u}_L, \overrightarrow{v}_R \rangle - \alpha^{-1} \langle \overrightarrow{u}_R, \overrightarrow{v}_L \rangle$$

- Hence a claim about the inner product $\langle \overrightarrow{u}, \overrightarrow{v} \rangle$ of length n can be **reduced** to some claims about the inner products of vectors of length $n/2$
- This procedure is applied recursively to obtain a claim about vectors of length 1.

Core techniques of Dory

The key ideas of Dory are

- Symmetry of messages and commitment keys

AFGHO structure preserving commitments have symmetry between messages and the commitment key.
For some pairing group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ if the **message** is a vector in $\mathbb{G}_1$ then **commitment key** is a vector in $\mathbb{G}_2$ (and vice versa) and the **commitment** is in $\mathbb{G}_T$

- Commitment key and all Verifier challenges are **public** so we can **outsource** computation on the commitment key to the Prover

- Structured Verifier computation
- Structured public scalars
- Public parameters
- Batching
- Application to Polynomial commitments

Core techniques of Dory

The key ideas of Dory are

- Symmetry of messages and commitment keys
- Structured Verifier computation

The computations of the Verifier are highly structured. Given the first challenge α the verifier

- must turn commitment key $\overrightarrow{\Gamma} = (\overrightarrow{\Gamma}_L || \overrightarrow{\Gamma}_R)$ into $\overrightarrow{\Gamma'} = (f(\alpha)\overrightarrow{\Gamma}_L + g(\alpha)\overrightarrow{\Gamma}_R)$ where f, g are cheap to compute
- If the verifier holds structure preserving commitments to $\overrightarrow{\Gamma}_L, \overrightarrow{\Gamma}_R$ they can quicly compute a commitment to $\overrightarrow{\Gamma'}$

Hence if we have structure preserving commitments to the commited key, the **Verifier** can apply one or more challenges to **shrink the commited key** and have the **Prover** do the **linear work** of computing the inner product.

- Structured public scalars
- Public parameters

Core techniques of Dory

The key ideas of Dory are

- Symmetry of messages and commitment keys
- Structured Verifier computation
- Structured public scalars

For polynomial commitments the polynomial size vector of scalars has multiplicative structure, as it is the evaluation of monomials for fixed values of variables.
Inner products of vectors of this form can be computed in only **logarithmically** many operations.

- Public parameters
- Batching
- Application to Polynomial commitments

Core techniques of Dory

The key ideas of Dory are

- Symmetry of messages and commitment keys
- Structured Verifier computation
- Structured public scalars
- Public parameters

Dory public parameters contain **commitment keys** for every power of 2 length less than n in both $\mathbb{G}_1, \mathbb{G}_2$ and commitments to the left and right halves of each commitment key.
In this way the online proof generation and verification is **accelerated**, as pp are computed once during setup with linear-size computation.

- Batching
- Application to Polynomial commitments

Core techniques of Dory

The key ideas of Dory are

- Symmetry of messages and commitment keys
- Structured Verifier computation
- Structured public scalars
- Public parameters
- Batching

The arguments can be batched to reduce verification time further.
The cost of evaluating each additional polynomial commitment is:

- $O(1)$ group operations and
- $O(\log n)$ additional operations in $\mathbb{F}$

- Application to Polynomial commitments

Core techniques of Dory

The key ideas of Dory are

- Symmetry of messages and commitment keys
- Structured Verifier computation
- Structured public scalars
- Public parameters
- Batching
- Application to Polynomial commitments

Construct a polynomial commitment from a two-tiered homomorphic commitment to matrices. Evaluation of dense univariate or multilinear polynomials with n coefficients is reduced to two inner producs of size $O(n^{1/2})$

Table of Contents

Notation

- Use additive group notation
- prime field $\mathbb{F} = \mathbb{F}_p$
- 3 groups of order p: $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$
- bilinear map $e : (\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T)$
- generators $G_1 \in \mathbb{G}_1,\ G_2 \in \mathbb{G}_2$ such that $e(G_1, G_2)$ generates $\mathbb{G}_T$
- $\langle , \rangle$ denotes generalised inner products

Computationally hard problems

SXDH assumption

$(\mathbb{F}_p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$ satisfies the **symmetric external Diffie-Hellman** assumption (SXDH) if the Decisional Diffie-Hellman (DDH) assumption holds for $(\mathbb{F}_p, \mathbb{G}_1, G_1)$ and $(\mathbb{F}_p, \mathbb{G}_2, G_2)$

Succinct interactive arguments of knowledge

For an $\mathcal{NP}$ language $\mathcal{L}$ there is a deterministic polynomial time $Sat_{\mathcal{L}}$ s.t.

$$\{\exists w : \ Sat\mathcal{L}(\mathbf{x}, w) = 1\} \Leftrightarrow \mathrm{x} \in \mathcal{L}$$

Public-coin succinct interactive argument of knowledge:

- Completeness

If $\mathrm{x} \in \mathcal{L}$ for any witness w and $r \in \{0,1\}^*$

$$Pr[\langle \mathrm{P}(pp, w), \mathrm{V}(pp, r)\rangle(\mathrm{x}) = 1 | Sat_{\mathcal{L}}(\mathrm{x}, w) = 1] \geq 1 - negl(\lambda)$$

- Soundness
- Knowledge soundness
- succinctness
- public coin

Succinct interactive arguments of knowledge

For an $\mathcal{NP}$ language $\mathcal{L}$ there is a deterministic polynomial time $Sat_{\mathcal{L}}$ s.t.

$$\{\exists w : \ Sat\mathcal{L}(\mathbf{x}, w) = 1\} \Leftrightarrow \mathrm{x} \in \mathcal{L}$$

Public-coin succinct interactive argument of knowledge:

- Completeness
- Soundness

For $\mathrm{x} \notin \mathcal{L}$, any PPT P^* and for all $r \in \{0,1\}^*$

$$Pr[\langle \mathrm{P}^*(pp), \mathrm{V}(pp, r)\rangle(\mathrm{x}) = 1] \leq negl(\lambda)$$

- Knowledge soundness
- succinctness
- public coin

Succinct interactive arguments of knowledge

For an $\mathcal{NP}$ language $\mathcal{L}$ there is a deterministic polynomial time $Sat_{\mathcal{L}}$ s.t.

$$\{\exists w : \ Sat\mathcal{L}(\mathbf{x}, w) = 1\} \Leftrightarrow \mathrm{x} \in \mathcal{L}$$

Public-coin succinct interactive argument of knowledge:

- Completeness
- Soundness
- Knowledge soundness

For any PPT adversary $\mathcal{A}$, there exists a PPT extractor Ext such that $\forall \mathrm{x} \in \mathcal{L}, \forall r \in \{0,1\}^*$ if

$$Pr[\langle \mathcal{A}(pp), \mathrm{V}(pp, r)\rangle(\mathrm{x}) = 1] \geq negl(\lambda)$$

then

$$Pr[Sat_{\mathcal{L}}(\mathrm{x}, \mathrm{Ext}^{\mathcal{A}}(pp, \mathrm{x})) = 1] \geq negl(\lambda)$$

Succinct interactive arguments of knowledge

For an $\mathcal{NP}$ language $\mathcal{L}$ there is a deterministic polynomial time $Sat_{\mathcal{L}}$ s.t.

$$\{\exists w : \ Sat\mathcal{L}(\mathbf{x}, w) = 1\} \Leftrightarrow \mathrm{x} \in \mathcal{L}$$

Public-coin succinct interactive argument of knowledge:

- Completeness
- Soundness
- Knowledge soundness
- succinctness

Communication between P and V is sublinear in $|w|$

- public coin

Succinct interactive arguments of knowledge

For an $\mathcal{NP}$ language $\mathcal{L}$ there is a deterministic polynomial time $Sat_{\mathcal{L}}$ s.t.

$$\{\exists w : \ Sat\mathcal{L}(\mathbf{x}, w) = 1\} \Leftrightarrow \mathrm{x} \in \mathcal{L}$$

Public-coin succinct interactive argument of knowledge:

- Completeness
- Soundness
- Knowledge soundness
- succinctness
- public coin

Each V message $\mathcal{M} \leftarrow\$ \mathcal{C}$, for $\mathcal{C}$ for some fixed set.

Honest-Verifier Statistical Zero-Knowledge

HVZK

An interactive argument $(Gen, \mathsf{P}, \mathsf{V})$ for $\mathcal{L}$ is **Honest-Verifier Statistical Zero-Knowledge** (HVZK) if there exists a PPT algorithm $\mathsf{Sim}(\mathsf{x}, z)$ called the **simulator**, running in poly time in $|\mathsf{x}|$, such that for every $\mathsf{x} \in \mathcal{L}, w \in \mathcal{R}_{\mathsf{x}}$ and $z \in \{0,1\}^*$ the statistical distance between the distributions

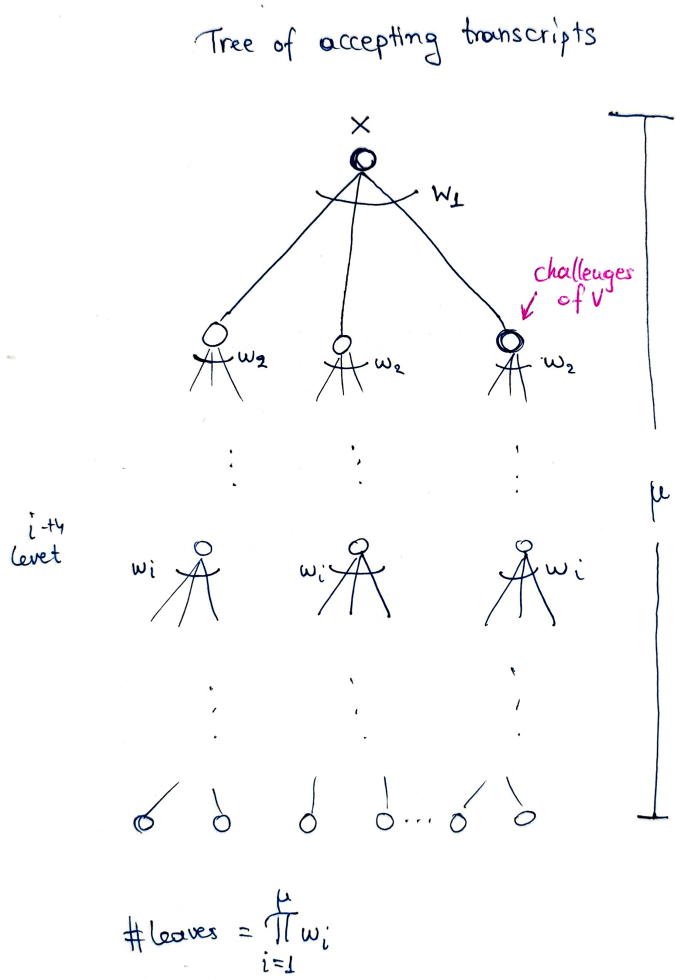
$$\langle \mathsf{P}(w), \mathsf{V}(z) \rangle(\mathsf{x}) \qquad \mathsf{Sim}(\mathsf{x}, z)$$

is $negl(\lambda)$.

Generalization of Special Soundness

For a $2\mu + 1$ move interactive protocol a $(w_1, \ldots w_\mu)$-tree of accepting transcripts is a tree of depth μ in which:

- the root is labelled with x and the initial *prover* message
- each node at depth i has w_i children labelled with distinct V challenges and subsequent P message
- the concatenation of the labels on any path from the root to a leaf of the tree is an accepting transcript for the protocol



Generalization of Special Soundness

Tree extractability (arguments)

A $(2\mu + 1)$ move interactive protocol (P, V) with Verifier message spave $\mathcal{C}$ is (W, ϵ) extractable if:

- $\exists$ a PPT algorithm extracting a witness from $(w_1, ..., w_\mu)$-tree of accepting trancripts

with failure probability $\leq \epsilon, \prod_{i=1}^{\mu} w_i \leq W$ and $\max_i(w_i) \leq \epsilon|\mathcal{C}|$

Tree extractability (reductions)

An interactive protocol reducing $x \in \mathcal{L}$ to $x' \in \mathcal{L}'$ is tree extractable if

- the composition of this argument with a final Prover message revealing a witness w' for $x' \in \mathcal{L}'$ is a (W, ϵ) tree extractable argument for $\mathcal{L}$

Polynomial Commitments

Commitment Scheme

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

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

Pedersen and AFGHO commitments

For messages $\mathcal{X} = \mathbb{F}^n$ the Pedersen commitment scheme is defined by:

Pedersen commitments

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

AFGHO commitments are structure preserving commitments to group elements, where for $\mathcal{X} = \mathbb{G}_i^n$ for $i \in \{1, 2\}$ we have that:

AFGHO commitments

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

Properties of AFGHO commitments

AFGHO commitments

$$pp \leftarrow Gen(1^\lambda) = (g \leftarrow\$ \mathbb{G}_{3-i}^n, H_1 \leftarrow\$ \mathbb{G}_1, H_2 \leftarrow\$ \mathbb{G}_2)$$
$$(\mathcal{C}, \mathcal{S}) \leftarrow Commit(pp; x) = \{r \leftarrow\$ \mathbb{F}; (\langle x, g \rangle + r \cdot e(H_1, H_2), r)\}$$
$$Open(pp, \mathcal{C}, x, \mathcal{S}) = (\langle x, g \rangle + \mathcal{S} \cdot e(H_1, H_2) \stackrel{?}{=} \mathcal{C})$$

AFGHO commitments are:

- **hiding**: since $r \cdot e(H_1, H_2)$ is uniformly random in $\mathbb{G}_T$
- it is a commitment conditional on SXDH problem
- AFGHO commitments are additively **homomorphic**

Table of Contents

Inner Product argument with a logarithmic Verifier

- argument for inner products between vectors $\overrightarrow{v}_1 \in \mathbb{G}_1^n,\ \overrightarrow{v}_2 \in \mathbb{G}_2^n$ commited with AFGHO commitments with generators $(\Gamma_2, e(H_1, H_2)) \in \mathbb{G}_2^n \times \mathbb{G}_T$ and $(\Gamma_1, e(H_1, H_2)) \in \mathbb{G}_1^n \times \mathbb{G}_T$
- We define a language:

$$(C, D_1, D_2) \in \mathcal{L}_{n,\Gamma_1,\Gamma_2,H_1,H_2} \subset \mathbb{G}_T^3 \Leftrightarrow$$
$$\exists(\overrightarrow{v}_1 \in \mathbb{G}_1^n,\ \overrightarrow{v}_2 \in \mathbb{G}_2^n, r_c \in \mathbb{F}, r_{D_1} \in \mathbb{F}, r_{D_2} \in \mathbb{F}) :$$
$$D_1 = \langle \overrightarrow{v}_1, \Gamma_2 \rangle + r_{D_1} \cdot e(H_1, H_2)$$
$$D_2 = \langle \Gamma_1, \overrightarrow{v}_2 \rangle + r_{D_2} \cdot e(H_1, H_2)$$
$$C = \langle \overrightarrow{v}_1, \overrightarrow{v}_2 \rangle + r_C \cdot e(H_1, H_2)$$

Scalar-Product

- Interactive argument of knowledge for $\mathcal{L}_{1,\Gamma_1,\Gamma_2,H_1,H_2}$
- Prove that the product of two elements $v_1 \in \mathbb{G}_1$ and $v_2 \in \mathbb{G}_2$ under AFGHO commitments
- Note that pairings are more expensive than multiplication in $\mathbb{G}_1$ or $\mathbb{G}_2$

$Scalar - Product_{\Gamma_1,\Gamma_2,H_1,H_2}(C, D_1, D_2)$

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

Prover

Witness $(v_1, v_2, r_c, r_{D_1}, r_{D_2})$

Verifier

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

$P_1 = e(d_1, \Gamma_2) + r_{P_1} H_T$

$P_2 = e(\Gamma_1, d_2) + r_{P_2} H_T$

$Q = e(d_1, v_2) + e(v_1, d_2) + r_Q H_T$

$R = e(d_1, d_2) + r_R H_T$

$P_1, P_2, Q, R \rightarrow$

$\leftarrow c \leftarrow_\$ \mathbb{F}$

$E_1 \leftarrow d_1 + cv_1$

$E_2 \leftarrow d_2 + cv_2$

$r_1 \leftarrow r_{P_1} + cr_{D_1}$

$r_2 \leftarrow r_{P_2} + cr_{D_2}$

$r_3 \leftarrow r_R + cr_Q + c^2 r_C$

$E_1, E_2, r_1, r_2, r_3 \rightarrow$

Accept if: for $d \leftarrow_\$ \mathbb{F}$

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

Dory-Reduce

- Interactive Argument to reduce membership of $\mathcal{L}_{2^m,\Gamma_1,\Gamma_2,H_1,H_2}$ to membership of $\mathcal{L}_{2^{m-1},\Gamma'_1,\Gamma'_2,H_1,H_2}$
- If we neglect zero-knowledge, we start with 3 claims about 2^m length vectors:

$$D_1 = \langle \overrightarrow{v}_1, \Gamma_2 \rangle \qquad D_2 = \langle \Gamma_1, \overrightarrow{v}_2 \rangle \qquad C = \langle \overrightarrow{v}_1, \overrightarrow{v}_2 \rangle$$

- fold each into claims about 2^{m-1} length vectors $(\overrightarrow{v}_{i\alpha}, \Gamma'_i)$, using a challenge α from the Verifier:

$$D'_1 = \langle \overrightarrow{v}_{1\alpha}, \Gamma_{2\alpha} \rangle \qquad D'_2 = \langle \Gamma_{1\alpha}, \overrightarrow{v}_{2\alpha} \rangle \qquad C' = \langle \overrightarrow{v}_{1\alpha}, \overrightarrow{v}_{2\alpha} \rangle$$

- Prover and Verifier would seperately compute commitments from α and precomputed data:

$$\Delta_1 = \langle \overrightarrow{v}_{1\alpha}, \Gamma'_2 \rangle \qquad \Delta_2 = \langle \Gamma'_1, \overrightarrow{v}_{2\alpha} \rangle$$

Dory – Reduce

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

Precompute: $H_T = e(H_1, H_2)$, $\Delta_{1L} = \langle \Gamma_{1L}, \Gamma_2' \rangle$, $\Delta_{1R} = \langle \Gamma_{1R}, \Gamma_2' \rangle$, $\Delta_{2L} = \langle \Gamma_1', \Gamma_{2L} \rangle$, $\Delta_{2R} = \langle \Gamma_1', \Gamma_{2R} \rangle$, and $\chi = \langle \Gamma_1, \Gamma_2 \rangle$

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

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

$\mathcal{P} \to \mathcal{V}$: $D_{1L} = \langle \vec{v_{1L}}, \Gamma_2' \rangle + r_{D_{1L}} H_T, \quad D_{1R} = \langle \vec{v_{1R}}, \Gamma_2' \rangle + r_{D_{1R}} H_T$

$D_{2L} = \langle \Gamma_1', \vec{v_{2L}} \rangle + r_{D_{2L}} H_T, \quad D_{2R} = \langle \Gamma_1', \vec{v_{2R}} \rangle + r_{D_{2R}} H_T$

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

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

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

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

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

$\mathcal{P}(**)$: $\vec{v_1}' \leftarrow \alpha \vec{v_{1L}} + \vec{v_{1R}}, \qquad \vec{v_2}' \leftarrow \alpha^{-1} \vec{v_{2L}} + \vec{v_{2R}}$

$r'_{D_1} \leftarrow \alpha r_{D_{1L}} + r_{D_{1R}}, \qquad r'_{D_2} \leftarrow \alpha^{-1} r_{D_{2L}} + r_{D_{2R}},$

$r'_C \leftarrow r_C + \alpha r_{C_+} + \alpha^{-1} r_{C_-}$

$\mathcal{V}(**)$: $C' \leftarrow C + \chi + \beta D_2 + \beta^{-1} D_1 + \alpha C_+ + \alpha^{-1} C_-$

$D_1' \leftarrow \alpha D_{1L} + D_{1R} + \alpha\beta \Delta_{1L} + \beta \Delta_{1R}$

$D_2' \leftarrow \alpha^{-1} D_{2L} + D_{2R} + \alpha^{-1}\beta^{-1} \Delta_{2L} + \beta^{-1} \Delta_{2R}$

$\mathcal{V}$: Accept if $(C', D_1', D_2') \in \mathcal{L}_{2^{m-1},\Gamma_1',\Gamma_2',H_1,H_2}$

$\mathcal{P}$ **witness:** $(\vec{v_1}', \vec{v_2}', r'_C, r'_{D_1}, r'_{D_2})$

Dory-Innerproduct

- Apply Dory-Reduce iteratively to shrink an inner-product to a product
- and then apply Scalar-Product

Dory-Innerproduct$_{\Gamma_{1,0},\Gamma_{2,0},H_1,H_2}(C, D_1, D_2)$

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

$\Gamma_{1,j+1} = (\Gamma_{1,j})_L$, $\Gamma_{2,j+1} = (\Gamma_{2,j})_L$, for all $j \in 0, \dots, m-1$,

$\chi_i = \langle \Gamma_{1,i}, \Gamma_{2,i} \rangle$, for all $i \in 0, \dots, m$,

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

Prover

Witness $(\overrightarrow{v_1}, \overrightarrow{v_2}, r_C, r_{D_1}, r_{D_2})$

For $(C, D_1, D_2) \in \mathcal{L}_{2^m,\Gamma_{1,0},\Gamma_{2,0},H_1,H_2}$

Verifier

For $j = 0, \dots, m-1$:

$(C, D_1, D_2) \leftarrow Dory - Reduce_{m-j,\Gamma_{1,j},\Gamma_{2,j},\Gamma_{1,j+1},\Gamma_{2,j+1},H_1,H_2}(C, D_1, D_2)$

$Scalar - Product_{\Gamma_{1,m},\Gamma_{2,m},H_1,H_2}(C, D_1, D_2)$

Costs of Dory-Innerproduct for the Prover

Communication Complexity:

- **Prover:** sends 6 elements of $\mathbb{G}_T$ to verifier

Computational Complexity:

- Prover:
  For Dory-Reduce:
  - 6 multi-pairings of size 2^{m-j-1}
  - $O(2^{m-j})$ operations in $\mathbb{F}$
  - $O(1)$ operations in $\mathbb{G}_T$

  For Scalar-Product:
  - $O(1)$ pairings and exponentiations in $\mathbb{G}_T$

  Hence, the overall cost to the prover is dominated by multipairings of size 6×2^m, $O(m)$ group operations and $O(2^m)$ field arithmetic.

Costs of Dory-Innerproduct for Verifier

Computational Complexity:

- **Verifier**:
  For Dory-Reduce:
  - 10 exponentiations in $\mathbb{G}_T$
  - 2 inversions and 2 multiplications in $\mathbb{F}$
  - $O(1)$ additional operations in $\mathbb{G}_T$
  - $O(1)$ additions in $\mathbb{F}$

  For Scalar-Product:
  - 1 pairing
  - 7 exponentiations in $\mathbb{G}_T$
  - 1 inversion and 5 multiplications in $\mathbb{F}$
  - $O(1)$ additional operations in $\mathbb{G}_T$ and additions in $\mathbb{F}$

Table of Contents

Dual Dory

Dual Dory

Dual Dory is a:

- (Linkable) ring signature scheme
- with logarithmic signature size and
- with logarithmic verifier
- **does not require** trusted setup

Is based on Dory (discrete-log type assumptions + bilinear pairing)

Security is based on the *SXDH* assumption.

Comparison

	Sign	Verify	Sig. size	Assumptions and model		KGen Authority	Transparent setup	Malicious pk	Linkable
Ateniese et al. [5]	$O(1)$	$O(1)$	$O(1)$	strong RSA, DDH	RO	●	❍	❍	(●)
Rivest et al. [24]	$O(n)$	$O(n)$	$O(n)$	TD-OWP	RO	❍	●	❍	❍
Liu et al. [22]	$O(n)$	$O(n)$	$O(n)$	DDH	RO	❍	●	❍	●
BBS Signatures [11]	$O(1)$	$O(1)$	$O(1)$	q-SDH, DLin	RO	●	❍	❍	(●)
Dodis et al. [18]	$O(1)$	$O(1)$	$O(1)$	strong RSA	RO	❍	❍	❍	❍
Au et al. [6]	$O(1)$	$O(1)$	$O(1)$	strong RSA,DDH,LD-RSA	RO	❍	❍	❍	●
Chandran et al. [15]	$O(n)$	$O(n)$	$O(\sqrt{n})$	strong DDH, SUB	CRS	❍	❍	●	❍
Groth et al. [20]	$O(n \log n)$	$O(n)$	$O(\log n)$	DLOG	RO	❍	●	●	❍
CLSAG [19]	$O(n)$	$O(n)$	$O(n)$	OM-LC-DLOG,DDH	RO	❍	●	●	●
DualRing-EC [29]	$O(n)$	$O(n)$	$O(\log n)$	DLOG	RO	❍	●	❍	❍
DualDory, **this work**	$O(n)$	$O(n)+O(\log n)$	$O(\log n)$	SXDH	RO	❍	●	❍	●

Table 1: Development of the asymptotic efficiency of practical RSA- and DLOG-based signature schemes that allow signing on behalf of a group with n members. If applicable, linking costs are negligible. Costs depict exponentiations in the group for Sign and Verify, and number of group elements for Signature size. In DualDory, verification time is split into preprocessing effort per group, plus verification effort per signature. ● means applicable/required, ❍ means not applicable/required. (●) means linkable only by the key generation authority.

Adding Linkability

Idea:

- A signer can use DualRing to prove that they know a secret key sk_i corresponding to one of the pk's from a list
- the tag is computed as $tag = H'(prfx)^{sk_i}$
- signer produces a Pedersen commitment $com = P^{sk}Q^{r}$ to their secret key and use a **tag proof** based on standard Σ-protocols to show that tag and com use same secret key
- Use an idea from [Groth & Kohlweiss] to prove that they know how to open exactly one of the commitments:

$$(\frac{com}{pk_1}, \ldots, \frac{com}{pk_n})$$


Thank You