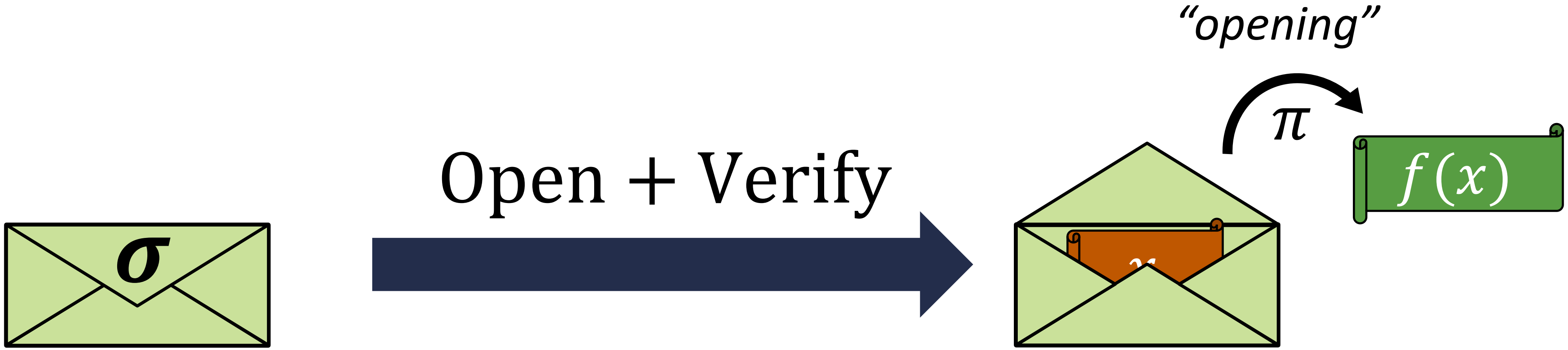
Succinct Functional Commitments for Circuits from k-Lin

Hoeteck Wee and David Wu

June 2024

Functional Commitments



Functional Commitments

$$\text{Commit}(\text{crs}, x) \rightarrow (\sigma, \text{st})$$

Takes a common reference string and commits to an input x

Outputs commitment σ and commitment state st

Functional Commitments

$\text{Commit}(\text{crs}, x) \to (\sigma, \text{st})$

$\text{Open}(\text{st}, f) \to \pi$

Takes the commitment state and a function f and outputs an opening π

$\text{Verify}(\text{crs}, \sigma, (f, y), \pi) \to 0/1$

Checks whether π is valid opening of σ to value y with respect to f

Functional Commitments

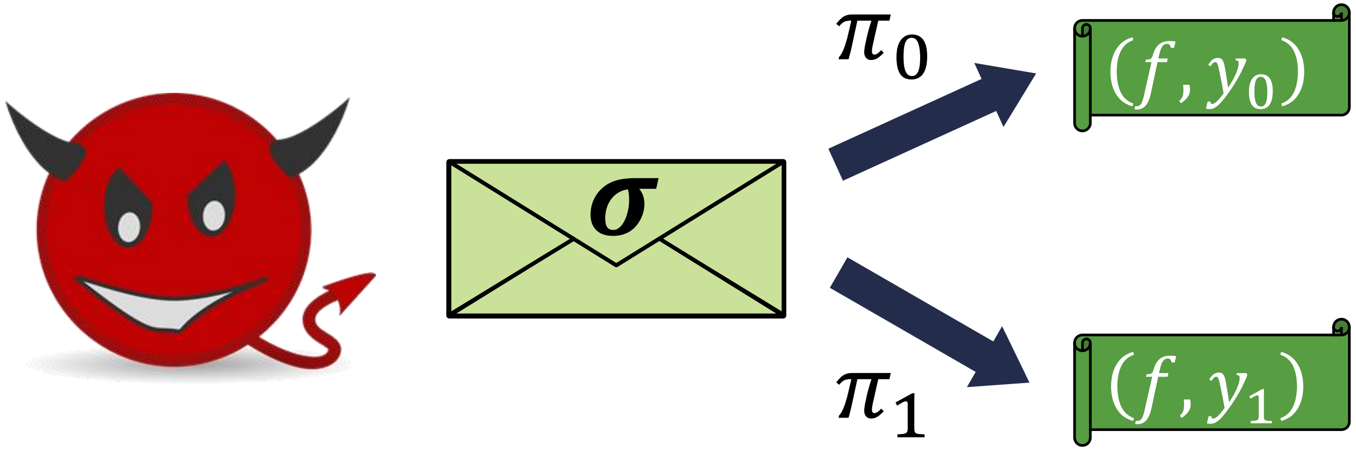
Correctness: if $(\sigma, \mathrm{st}) \leftarrow \mathrm{Commit}(\mathrm{crs}, x)$ and $\pi \leftarrow \mathrm{Open}(\mathrm{st}, f)$

then $\mathrm{Verify}\big(\mathrm{crs}, \sigma, \big(f, f(x)\big), \pi\big) = 1$

Can open commitment to x to value $y = f(x)$ for any function f

Functional Commitments

Binding: efficient adversary **cannot** open σ to two different values with respect to the **same** f



$\mathrm{Verify}(\mathrm{crs}, \sigma, (f, y_0), \pi_0) = 1$

$\mathrm{Verify}(\mathrm{crs}, \sigma, (f, y_1), \pi_1) = 1$

Functional Commitments

Succinctness: commitments and openings should be short

- **Short commitment:** $|\sigma| = \text{poly}(\lambda, \log |x|)$
- **Short opening:** $|\pi| = \text{poly}(\lambda, \log|x|\ )$

Special Cases of Functional Commitments

Vector commitments:

$$[x_1, x_2, \ldots, x_n] \xrightarrow{\mathrm{ind}_i(x_1, \ldots, x_n) \coloneqq x_i} x_i$$

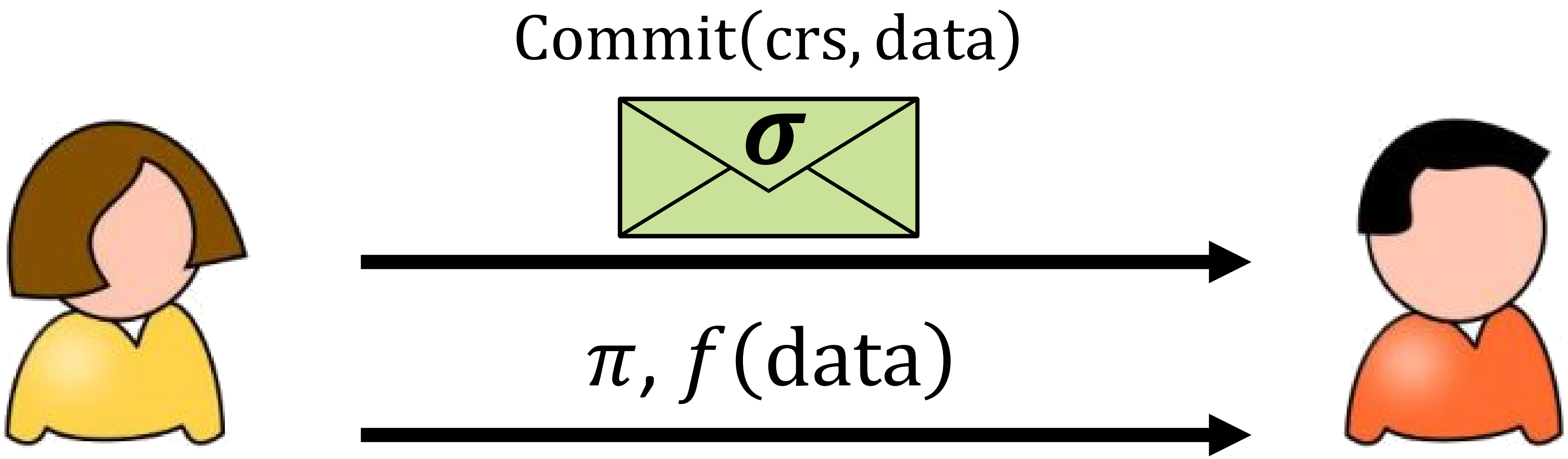
commit to a vector, open at an index

Polynomial commitments:

$$[\alpha_0, \alpha_1, \ldots, \alpha_d] \xrightarrow{f_x(\alpha_0, \ldots, \alpha_d) \coloneqq \alpha_0 + \alpha_1 x + \cdots + \alpha_d x^d} y$$

commit to a polynomial, open to the evaluation at x

Commitments as Proofs on Committed Data



π is a proof that the data satisfies some property
(e.g., committed input is in a certain range)

Succinctness: both the commitment and the proof are short

Succinct Functional Commitments

(not an exhaustive list!)

Scheme	Function Class	Assumption
[Mer87]	vector commitment	collision-resistant hash functions
[LY10, CF13, LM19, GRWZ20]	vector commitment	q-type pairing assumptions
[CF13, LM19, BBF19]	vector commitment	groups of unknown order
[PPS21]	vector commitment	short integer solutions (SIS)
[KZG10, Lee20]	polynomial commitment	q-type pairing assumptions
[BFS19, BHRRS21, BF23]	polynomial commitment	groups of unknown order
[CLM23, FLV23]	polynomial commitment	k-R-ISIS assumption (lattices)
[LRY16]	linear functions	q-type pairing assumptions
[ACLMT22, CLM23]	constant-degree polynomials	k-R-ISIS assumption (lattices)
[LRY16]	Boolean circuits	collision-resistant hash functions + SNARKs
[dCP23]	Boolean circuits	SIS (non-succinct openings in general)
[KLVW23]	Boolean circuits	batch arguments for NP
[BCFL23]	Boolean circuits	twin k-R-ISIS (lattice) / HiKER (pairing)
[WW23a, WW23b]	Boolean circuits	ℓ-succinct SIS

Pairing-Based Functional Commitments

This work: functional commitments for **general circuits** using **pairings**

Why bilinear maps? Schemes have the best **succinctness**

- Pairing-based SNARKs just have a constant number of group elements

Can we construct a functional commitment for general circuits where the size of the commitment and the opening contain a ***constant*** *number of group elements?*

Namely: match the succinctness of pairing-based SNARKs, but only using standard pairing-based assumptions (no knowledge assumptions or ideal models)

Pairing-Based Functional Commitments

This work: functional commitments for **general circuits** using **pairings**

Scheme	**Function Class**	$\lvert\mathrm{crs}\rvert$	$\lvert\sigma\rvert$	$\lvert\pi\rvert$	**Assumption**
[LRY16, Gro16]	arithmetic circuits	$O(s)$	$O(1)$	$O(1)$	generic group
[LRY16]	linear functions	$O(\ell)$	$O(1)$	$O(m)$	subgroup decision
[LM19]	linear functions	$O(\ell m)$	$O(1)$	$O(1)$	generic group
[LP20]	μ-sparse polynomials	$O(\mu)$	$O(m)$	$O(1)$	über assumption
[CFT22]	degree-d polynomials	$O(\ell^d m)$	$O(d)$	$O(d)$	ℓ^d-Diffie-Hellman exponent
[BCFL23]	arithmetic circuits	$O(s^5)$	$O(1)$	$O(d)$	hinted kernel (q-type)
[KLVW23]	arithmetic circuits	$\mathrm{poly}(\lambda)$	$O(1)$	$\mathrm{poly}(\lambda)$	k-Lin
This work	**arithmetic circuits**	$\boldsymbol{O(s^5)}$	$\boldsymbol{O(1)}$	$\boldsymbol{O(1)}$	**bilateral k-Lin**

ℓ = input length, m = output length, s = circuit size

metrics in # group elements

This Work

This work: functional commitments for **general circuits** using **pairings**

Scheme	Function Class	$\lvert \mathrm{crs} \rvert$	$\lvert \sigma \rvert$	$\lvert \pi \rvert$	Assumption
This work	arithmetic circuits	$\boldsymbol{O(s^5)}$	$\boldsymbol{O}(1)$	$\boldsymbol{O}(1)$	bilateral $\boldsymbol{k}$-Lin

- First pairing-based construction for general **circuits** based on **falsifiable** assumptions where commitment and openings contain **constant** number of group elements
  - **Previously:** needed SNARKs (non-falsifiable assumptions)
- First scheme that only makes **black-box** use of cryptographic primitives/algorithms where the commitment + opening size is $\mathrm{poly}(\lambda)$ bits
  - **Previously:** need non-black-box techniques (e.g., SNARKs or BARGs for NP)

This Work

This work: functional commitments for **general circuits** using **pairings**

Scheme	Function Class	$\lvert\mathrm{crs}\rvert$	$\lvert\sigma\rvert$	$\lvert\pi\rvert$	Assumption
This work	**arithmetic circuits**	$\boldsymbol{O(s^5)}$	$\boldsymbol{O(1)}$	$\boldsymbol{O(1)}$	**bilateral $\boldsymbol{k}$-Lin**

Constant number of group elements

Additional implications (for free!):

- SNARG for P/poly with a **universal** setup with constant-size proofs (CRS only depends on the size of the circuit)
  - **Previously (from pairings):** SNARG for P/poly with circuit-dependent CRS [GZ21]
- Homomorphic signature for general (bounded-size) circuits with constant-size signatures
  - **Previously (from pairings):** Signature size scaled with the *depth* of the circuit [BCFL23]

(all results without relying on knowledge assumptions or ideal models)

Starting Point: Chainable Commitment

Chainable commitment [BCFL23]

Let $f\colon \mathbb{Z}_p^{\ell} \to \mathbb{Z}_p^{d}$ be a vector-valued function

Can think of commitment as a subset product:

$$\sigma = \prod_{i \in [\ell]} h_i^{x_i}$$

where h_i are in the CRS

x_1
x_2
$\vdots$
x_ℓ

succinct commitment to vector $\boldsymbol{x}$

$\sigma_{\boldsymbol{x}}$

succinct opening π

y_1
y_2
$\vdots$
y_d

$\sigma_{\boldsymbol{y}}$

succinct commitment to vector $\boldsymbol{y} = f(\boldsymbol{x})$

Instead of committing to $\boldsymbol{x}$ and opening to $\boldsymbol{y} = f(\boldsymbol{x})$

Open to **commitment** to $\boldsymbol{y} = f(\boldsymbol{x})$

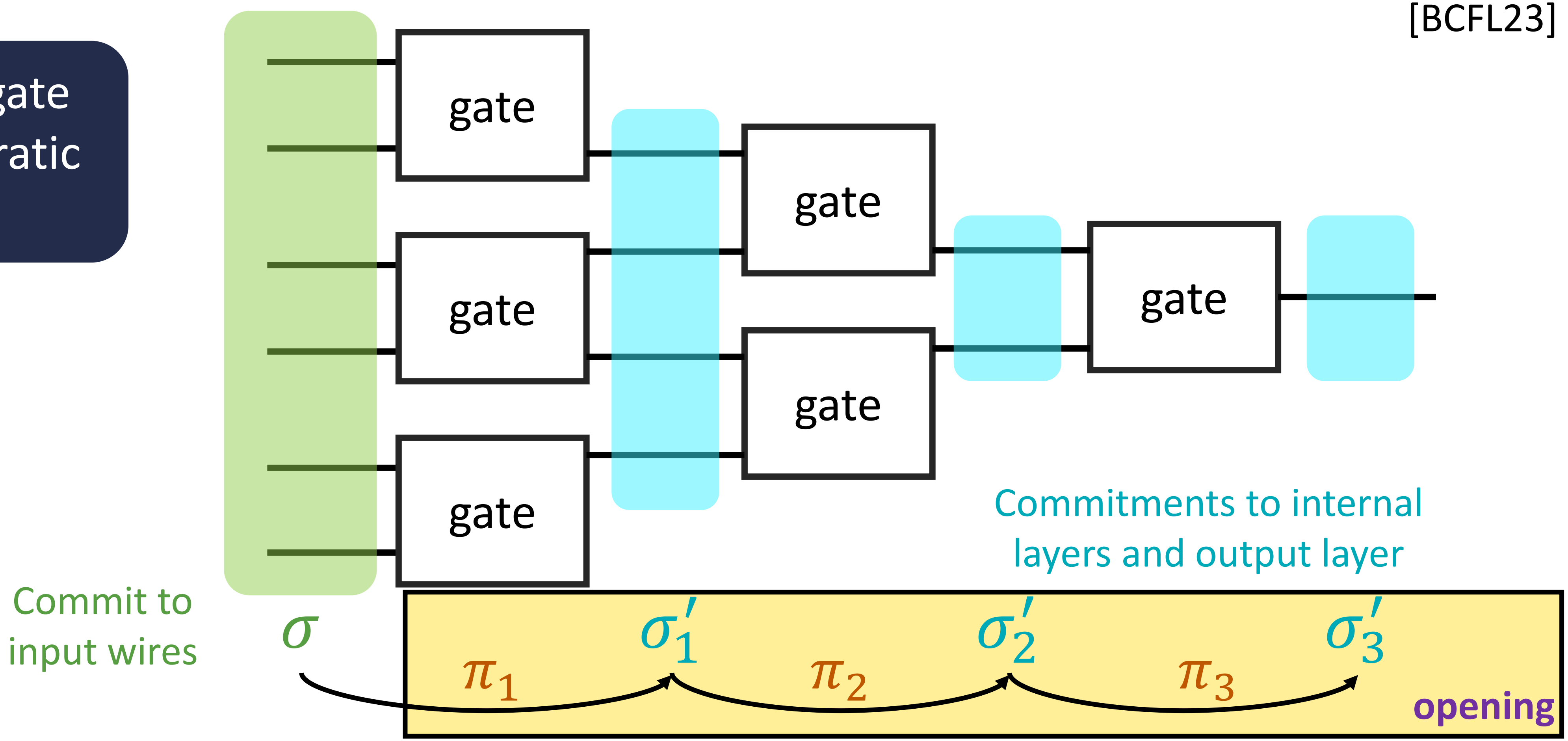
Chain binding: cannot open σ_{in} to two distinct commitments $\sigma_{\text{out}}, \sigma'_{\text{out}}$

Starting Point: Chainable Commitment

Chainable commitment for **quadratic functions** ⇒ functional commitment for **circuits**

[BCFL23]

Assume: each gate computes quadratic function



Chainable commitment openings for each layer

Starting Point: Chainable Commitment

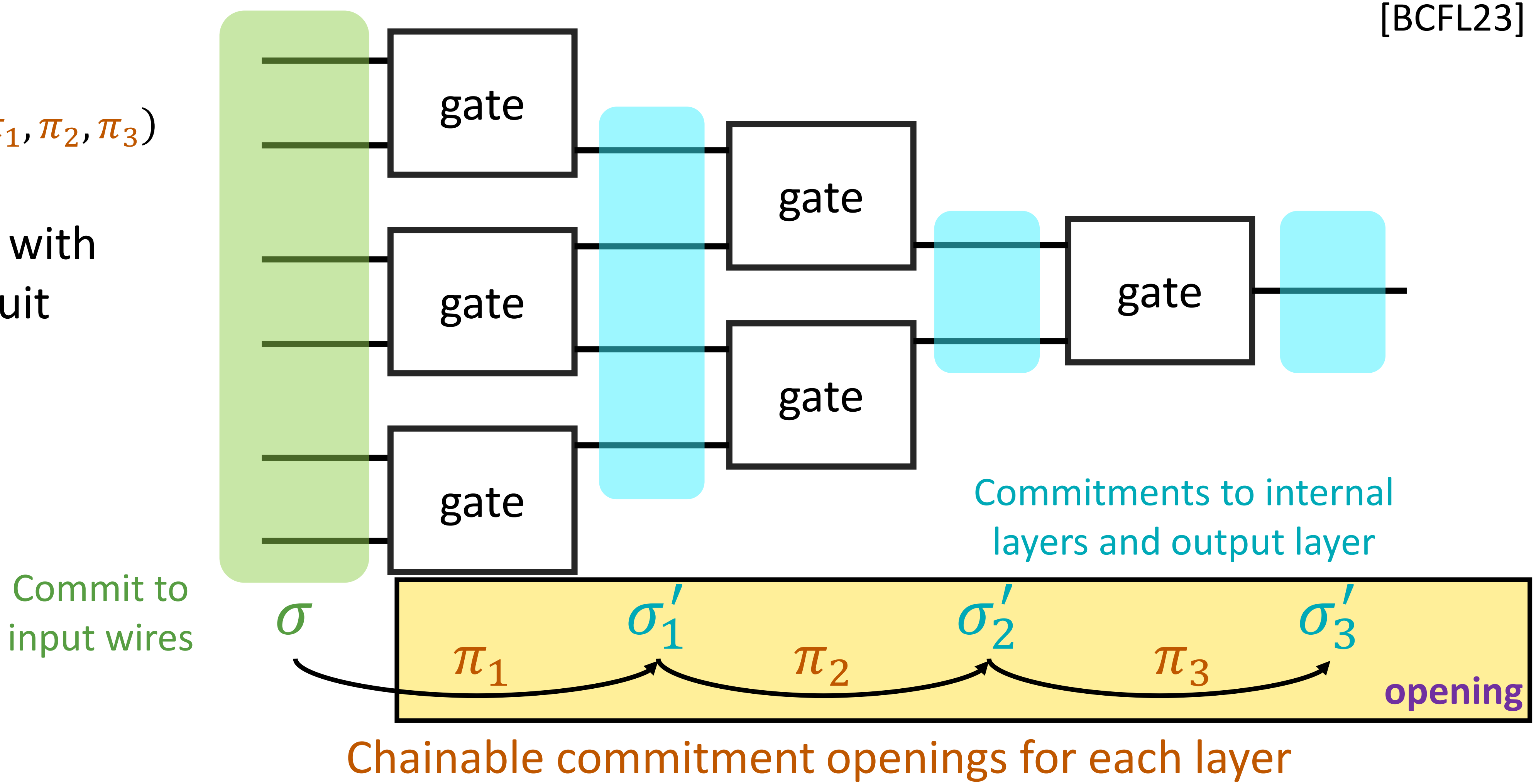
Chainable commitment for **quadratic functions** ⇒ functional commitment for **circuits**

[BCFL23]

Commitment: σ

Opening: $(\sigma_1', \sigma_2', \sigma_3', \pi_1, \pi_2, \pi_3)$

Opening scales with depth of circuit



Our Approach: Commit to All Wires

Goal: Constant number of group elements for commitment **and** openings

Commitment: (same as before)

$$\boxed{x_1} \boxed{x_2} \boxed{\cdots} \boxed{x_\ell} \longrightarrow \sigma_{\mathrm{in}}$$

Verifier know output $(z_1, \ldots, z_d)$:

$$\boxed{z_1} \boxed{z_2} \boxed{\cdots} \boxed{z_d} \longrightarrow \sigma_{\mathrm{out}}$$

Opening: commit to **all** wires (i.e., concatenated together) **twice**

$$\underbrace{\boxed{x_1} \boxed{x_2} \boxed{\cdots} \boxed{x_\ell}}_{\text{Input layer}} \underbrace{\boxed{y_1} \boxed{y_2} \boxed{\cdots} \boxed{y_t}}_{\text{Intermediate layer}} \underbrace{\boxed{z_1} \boxed{z_2} \boxed{\cdots} \boxed{z_d}}_{\text{Output layer}} \longrightarrow \sigma_1$$

$$\overbrace{\boxed{x_1} \boxed{x_2} \boxed{\cdots} \boxed{x_\ell}} \overbrace{\boxed{y_1} \boxed{y_2} \boxed{\cdots} \boxed{y_t}} \overbrace{\boxed{z_1} \boxed{z_2} \boxed{\cdots} \boxed{z_d}} \longrightarrow \sigma_2$$

Our Approach: Commit to All Wires

Goal: Constant number of group elements for commitment **and** openings

Commitment: (same as before)

$$\boxed{x_1}\boxed{x_2}\boxed{\cdots}\boxed{x_\ell} \longrightarrow \sigma_{\text{in}}$$

Verifier know output $(z_1, \ldots, z_d)$:

$$\boxed{z_1}\boxed{z_2}\boxed{\cdots}\boxed{z_d} \longrightarrow \sigma_{\text{out}}$$

Opening: commit to **all** wires (i.e., concatenated together) **twice**

$$\underbrace{\boxed{x_1}\boxed{x_2}\boxed{\cdots}\boxed{x_\ell}}_{\text{Input layer}}\underbrace{\boxed{y_1}\boxed{y_2}\boxed{\cdots}\boxed{y_t}}_{\text{Intermediate layer}}\underbrace{\boxed{z_1}\boxed{z_2}\boxed{\cdots}\boxed{z_d}}_{\text{Output layer}} \longrightarrow \sigma_1$$

$$\overbrace{\boxed{x_1}\boxed{x_2}\boxed{\cdots}\boxed{x_\ell}}\overbrace{\boxed{y_1}\boxed{y_2}\boxed{\cdots}\boxed{y_t}}\overbrace{\boxed{z_1}\boxed{z_2}\boxed{\cdots}\boxed{z_d}} \longrightarrow \sigma_2$$

Everything is short, but how do we argue binding?

Our Approach: Commit to All Wires

Goal: Constant number of group elements for commitment **and** openings

Commitment: (same as before)

$$\boxed{x_1 \mid x_2 \mid \cdots \mid x_\ell} \longrightarrow \sigma_{\text{in}}$$

Verifier know output $(z_1, \ldots, z_t)$:

$$\boxed{z_1 \mid z_2 \mid \cdots \mid z_d} \longrightarrow \sigma_{\text{out}}$$

Opening: commit to **all** wires (i.e., concatenated together) **twice**

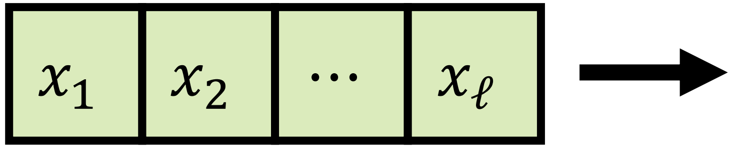
$$\boxed{x_1 \mid x_2 \mid \cdots \mid x_\ell \mid y_1 \mid y_2 \mid \cdots \mid y_t \mid z_1 \mid z_2 \mid \cdots \mid z_d} \longrightarrow \sigma_1$$

Neither σ_1 nor σ_2 is a quadratic function of σ_{input}

With bilinear maps, we only know how to check quadratic functions

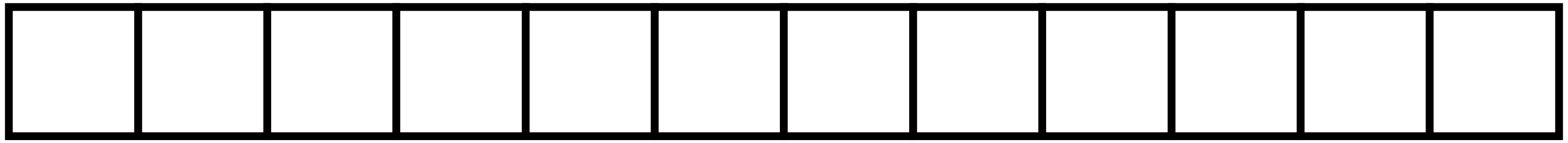
$$\boxed{x_1 \mid x_2 \mid \cdots \mid x_\ell \mid y_1 \mid y_2 \mid \cdots \mid y_t \mid z_1 \mid z_2 \mid \cdots \mid z_d} \longrightarrow \sigma_2$$

Approach Overview



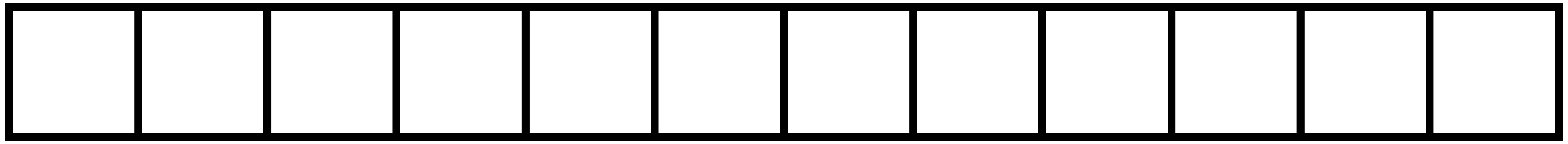
σ_{in}

Consider two different openings: $(\sigma_1, \sigma_2, \sigma_{\text{out}}, \pi)$ and $(\sigma_1', \sigma_2', \sigma_{\text{out}}', \pi')$



σ_1, σ_1'

Initially: no guarantees on what $\sigma_1, \sigma_1', \sigma_2, \sigma_2'$ commit to



σ_2, σ_2'

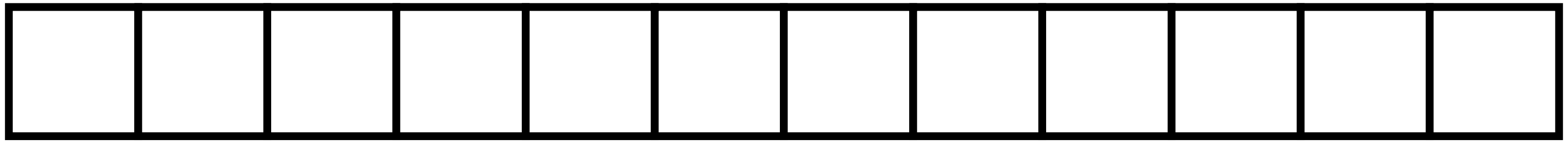
Cannot use chain binding to argue that σ_1 and σ_1' are equal since they are not a quadratic function of σ_{in}

Our approach: argue that a **prefix** of σ_1, σ_1' are still equal

Approach Overview

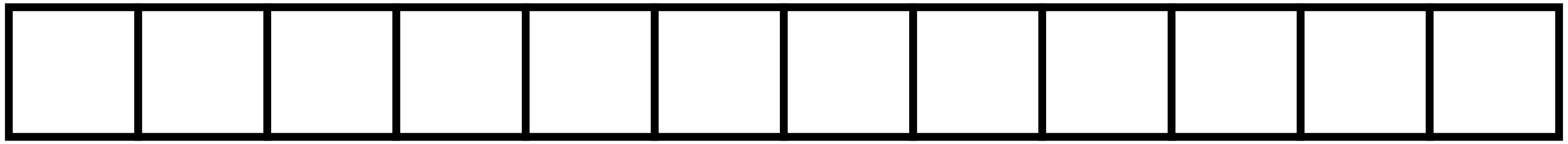
$\boxed{x_1}\boxed{x_2}\boxed{\cdots}\boxed{x_\ell} \longrightarrow \sigma_{\text{in}}$

Consider two different openings: $(\sigma_1, \sigma_2, \sigma_{\text{out}}, \pi)$ and $(\sigma_1', \sigma_2', \sigma_{\text{out}}', \pi')$



σ_1, σ_1'

Initially: no guarantees on what $\sigma_1, \sigma_1', \sigma_2, \sigma_2'$ commit to



σ_2, σ_2'

Input consistency: π, π' includes an opening that asserts that the first ℓ components of σ_1, σ_1' are consistent with σ_{in}

Approach Overview

$$\boxed{x_1}\boxed{x_2}\boxed{\cdots}\boxed{x_\ell} \longrightarrow \sigma_{\text{in}}$$

Consider two different openings: $(\sigma_1, \sigma_2, \sigma_{\text{out}}, \pi)$ and $(\sigma_1', \sigma_2', \sigma_{\text{out}}', \pi')$

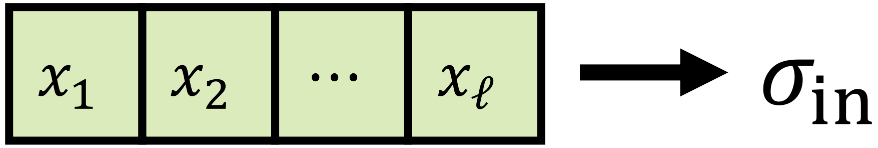
$$\boxed{\hat{x}_1}\boxed{\hat{x}_2}\boxed{\cdots}\boxed{\hat{x}_\ell}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{} \qquad \sigma_1, \sigma_1'$$

Close to a chain binding property: prover is opening σ_{in} to output commitments σ_1, σ_1'

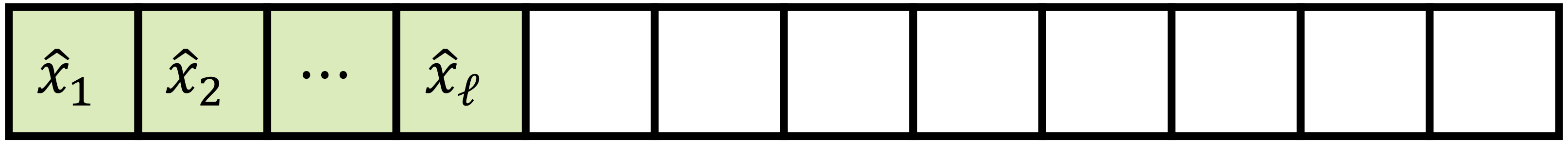
Caveat: Only reasoning about the first ℓ components of σ_1 and σ_1' (*not* the entire vector)

Input consistency: π, π' includes an opening that asserts that the first ℓ components of σ_1, σ_1' are consistent with σ_{in}

Approach Overview

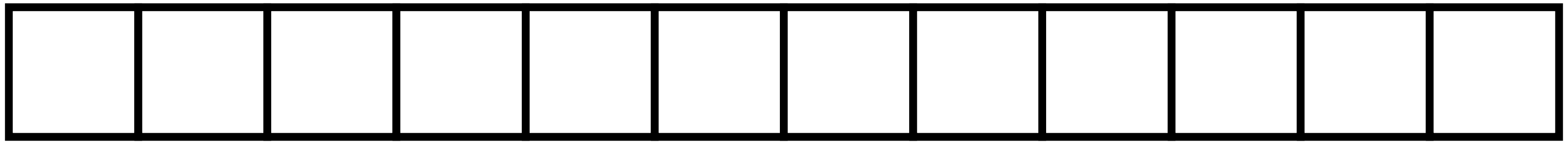


Consider two different openings: $(\sigma_1, \sigma_2, \sigma_{\text{out}}, \pi)$ and $(\sigma_1', \sigma_2', \sigma_{\text{out}}', \pi')$



σ_1, σ_1'

If we establish that the first ℓ components of σ_1, σ_1' agree, we can try to argue that the first $\ell + 1$ components of σ_2, σ_2' also agree

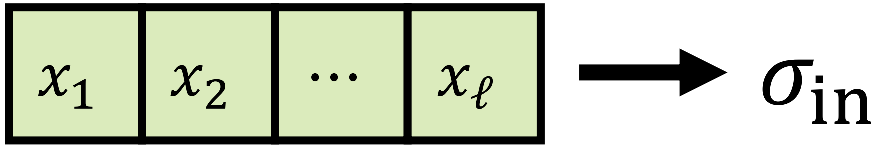


σ_2, σ_2'

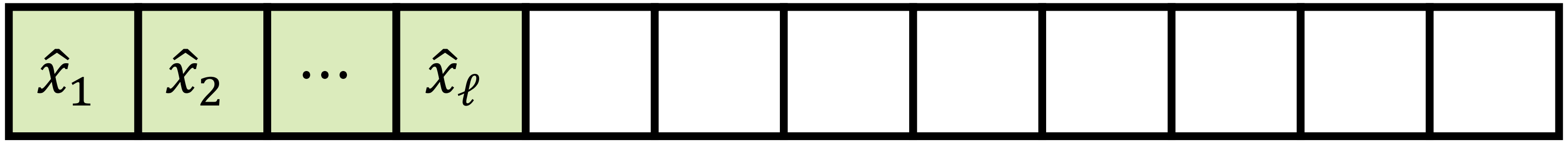
corresponds to a single gate

Observation: first $\ell + 1$ components of σ_2, σ_2' is a quadratic function of the first ℓ components of σ_1, σ_1'

Approach Overview

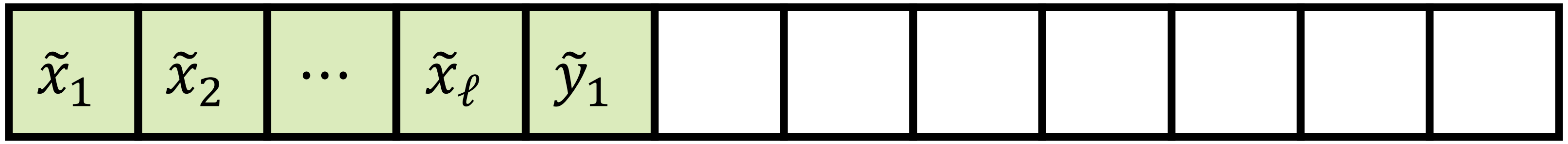


Consider two different openings: $(\sigma_1, \sigma_2, \sigma_{\text{out}}, \pi)$ and $(\sigma_1', \sigma_2', \sigma_{\text{out}}', \pi')$



σ_1, σ_1'

If we establish that the first ℓ components of σ_1, σ_1' agree, we can try to argue that the first $\ell + 1$ components of σ_2, σ_2' also agree

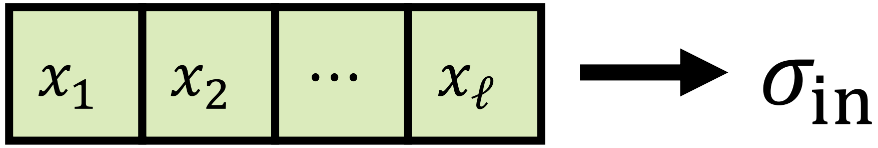


σ_2, σ_2'

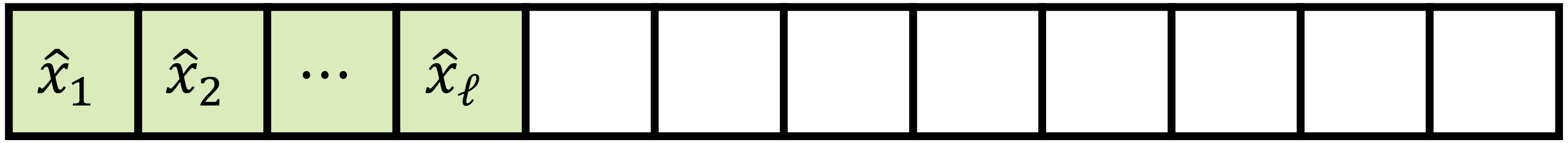
corresponds to a single gate

Observation: first $\ell + 1$ components of σ_2, σ_2' is a quadratic function of the first ℓ components of σ_1, σ_1'

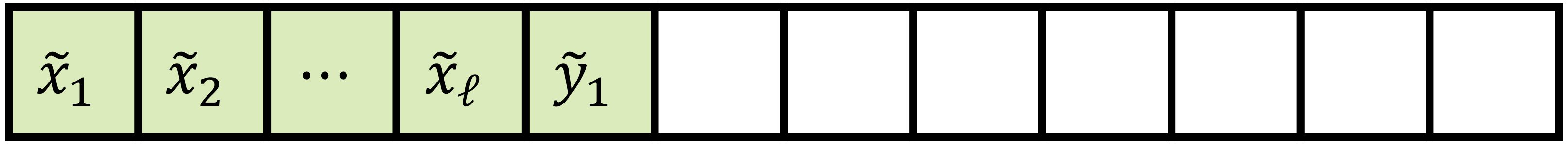
Approach Overview



Consider two different openings: $(\sigma_1, \sigma_2, \sigma_{\text{out}}, \pi)$ and $(\sigma_1', \sigma_2', \sigma_{\text{out}}', \pi')$



σ_1, σ_1'



σ_2, σ_2'

Repeat this process: if σ_2, σ_2' agree on the first $\ell + 1$ values, then σ_1, σ_1' agree on the first $\ell + 1$ values

Approach Overview

x_1	x_2	$\cdots$	x_ℓ

$\longrightarrow \sigma_{\text{in}}$

Consider two different openings: $(\sigma_1, \sigma_2, \sigma_{\text{out}}, \pi)$ and $(\sigma_1', \sigma_2', \sigma_{\text{out}}', \pi')$

$\hat{x}_1$	$\hat{x}_2$	$\cdots$	$\hat{x}_\ell$	$\hat{y}_1$							

σ_1, σ_1'

$\tilde{x}_1$	$\tilde{x}_2$	$\cdots$	$\tilde{x}_\ell$	$\tilde{y}_1$							

σ_2, σ_2'

Repeat this process: if σ_2, σ_2' agree on the first $\ell + 1$ values, then σ_1, σ_1' agree on the first $\ell + 1$ values

Approach Overview

x_1	x_2	$\cdots$	x_ℓ	$\longrightarrow \sigma_{\text{in}}$

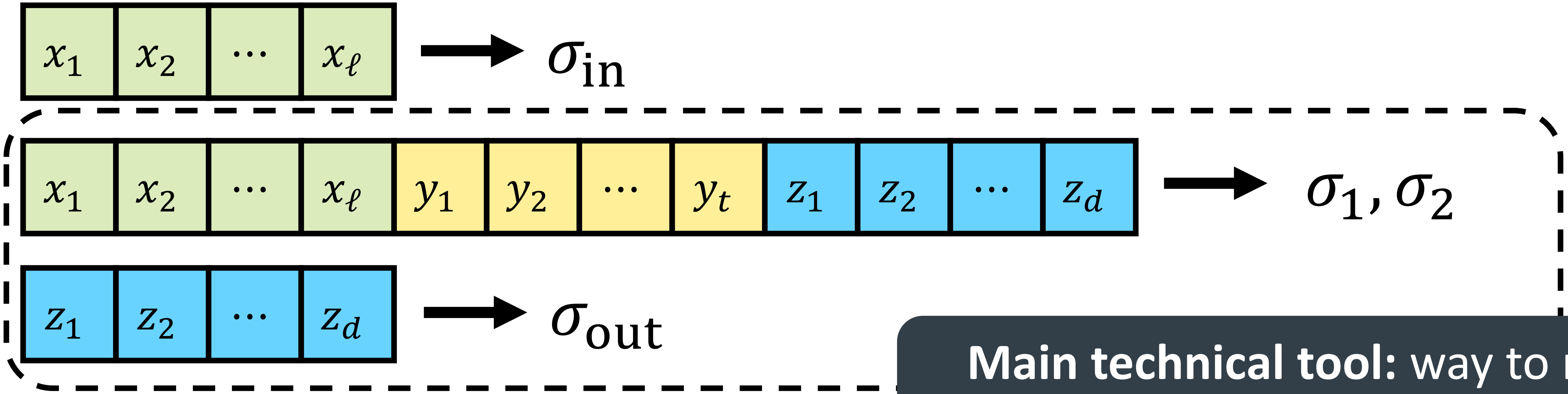
Consider two different openings: $(\sigma_1, \sigma_2, \sigma_{\text{out}}, \pi)$ and $(\sigma_1', \sigma_2', \sigma_{\text{out}}', \pi')$

$\hat{x}_1$	$\hat{x}_2$	$\cdots$	$\hat{x}_\ell$	$\hat{y}_1$	$\hat{y}_2$	$\cdots$	$\hat{y}_t$	$\hat{z}_1$	$\hat{z}_2$	$\cdots$	$\hat{z}_d$	σ_1, σ_1'

$\tilde{x}_1$	$\tilde{x}_2$	$\cdots$	$\tilde{x}_\ell$	$\tilde{y}_1$	$\tilde{y}_2$	$\cdots$	$\tilde{y}_t$	$\tilde{z}_1$	$\tilde{z}_2$	$\cdots$	$\tilde{z}_d$	σ_2, σ_2'

Iterate to conclude that σ_1, σ_1' actually agree on **all** values (including the outputs), which implies binding

Approach Overview

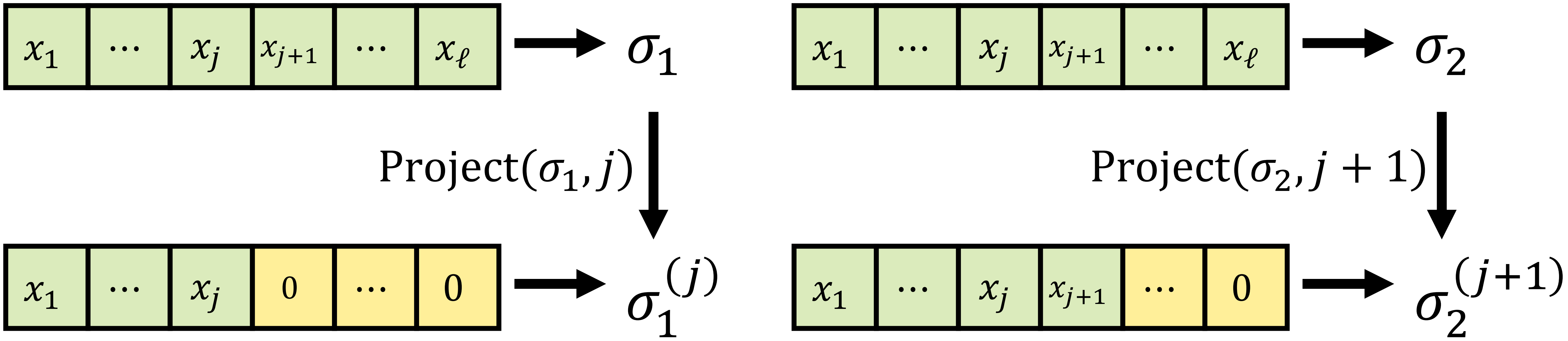


Main technical tool: way to reason about prefixes of a committed vector

Prove statements of the following form:

- **Input consistency:** first ℓ wires in σ_1 is consistent with σ_{in}
- **Gate consistency:** first $j + 1$ wires in σ_2 is consistent with first j wires in σ_1
- **Internal consistency:** first j wires in σ_1 is consistent with first j wires in σ_2
- **Output consistency:** last t wires in σ_1 are consistent with σ_{out}

Projective Chainable Commitments



Intuitively: can associate CRS with an index j that allows projecting a commitment σ_1 onto a commitment to the first j indices

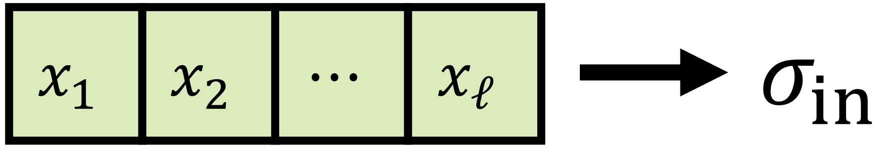
Projective chain binding: given $(\sigma_1, \sigma_2, \pi)$ and $(\sigma_1', \sigma_2', \pi')$

If $\mathrm{Project}(\mathrm{td}, \sigma_1, j) = \mathrm{Project}(\mathrm{td}, \sigma_1', j)$ and

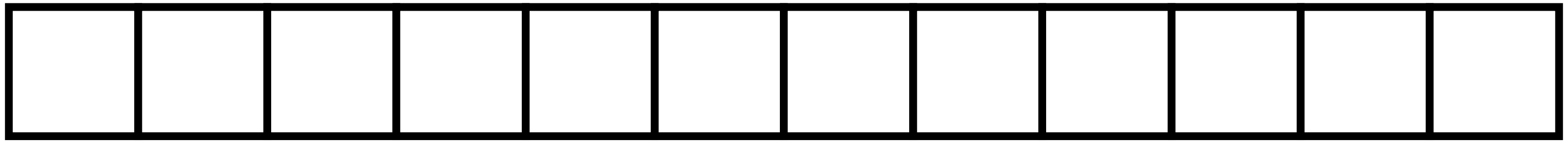
- (σ_2, π, f) is a valid opening for σ_1
- (σ_2', π', f) is a valid opening for σ_1'

Then, $\mathrm{Project}(\mathrm{td}, \sigma_2, j+1) = \mathrm{Project}(\mathrm{td}, \sigma_2', j+1)$

Using Projective Chainable Commitments

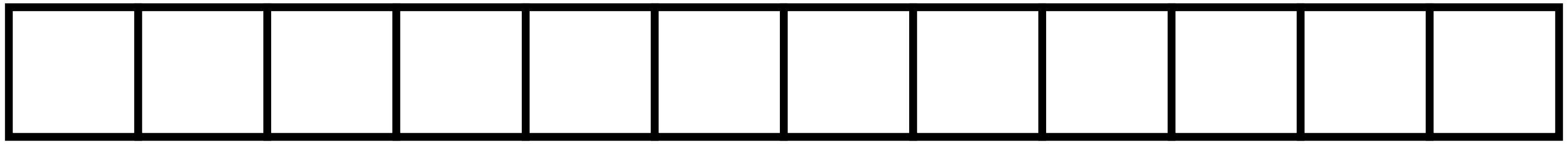


Consider two different openings: $(\sigma_1, \sigma_2, \sigma_{\text{out}}, \pi)$ and $(\sigma_1', \sigma_2', \sigma_{\text{out}}', \pi')$



σ_1, σ_1'

Initially: no guarantees on what $\sigma_1, \sigma_1', \sigma_2, \sigma_2'$ commit to

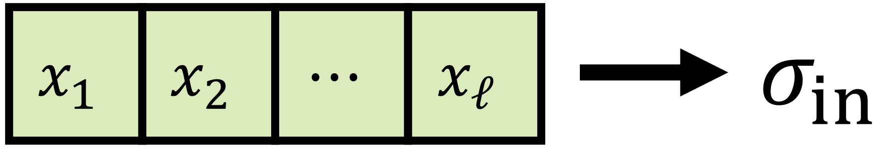


σ_2, σ_2'

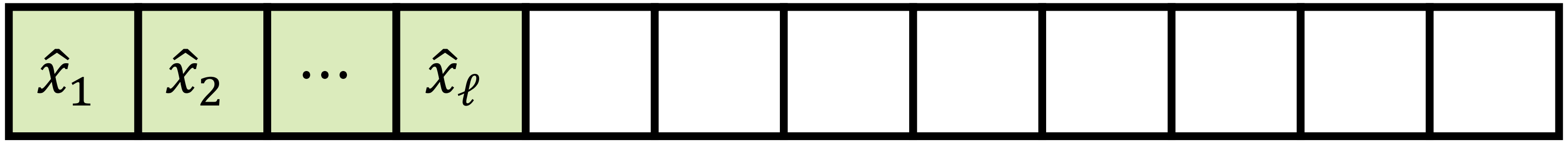
Step 1: Input consistency between σ_{in} and σ_1, σ_1'

Projective chain binding: σ_1, σ_1' are both openings for σ_{in} so $\text{Project}(\sigma_1, \ell) = \text{Project}(\sigma_1', \ell)$

Using Projective Chainable Commitments



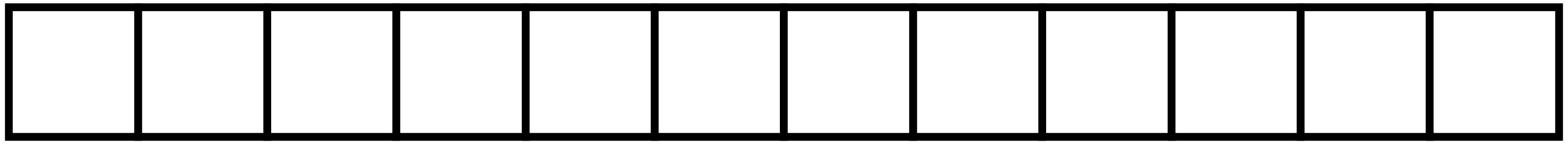
Consider two different openings: $(\sigma_1, \sigma_2, \sigma_{\text{out}}, \pi)$ and $(\sigma_1', \sigma_2', \sigma_{\text{out}}', \pi')$



σ_1, σ_1'

σ_1 and σ_1' **agree** on first ℓ components:
$\text{Project}(\sigma_1, \ell) = \text{Project}(\sigma_1', \ell)$

Note: we do **not** know what values they have, only that they agree

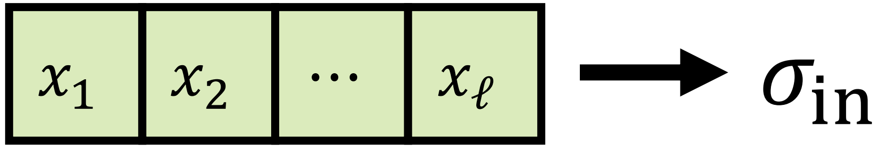


σ_2, σ_2'

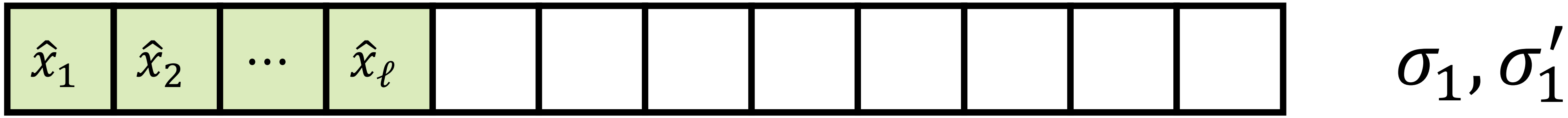
Step 1: Input consistency between σ_{in} and σ_1, σ_1'

Projective chain binding: σ_1, σ_1' are both openings for σ_{in} so $\text{Project}(\sigma_1, \ell) = \text{Project}(\sigma_1', \ell)$

Using Projective Chainable Commitments

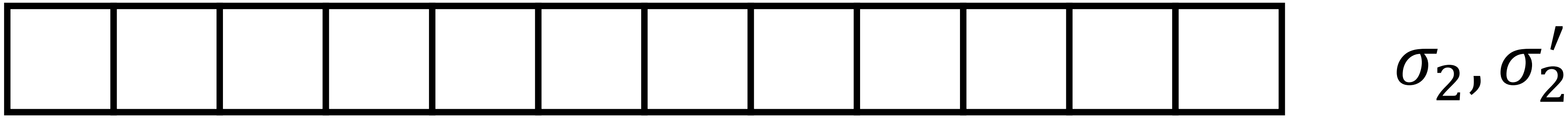


Consider two different openings: $(\sigma_1, \sigma_2, \sigma_{\text{out}}, \pi)$ and $(\sigma_1', \sigma_2', \sigma_{\text{out}}', \pi')$



σ_1 and σ_1' **agree** on first ℓ components:
$\text{Project}(\sigma_1, \ell) = \text{Project}(\sigma_1', \ell)$

Note: we do **not** know what values they have, only that they agree



Step 2: Gate consistency between first ℓ wires in σ_1, σ_1' with first $\ell + 1$ wires in σ_2, σ_2'

Since $\text{Project}(\sigma_1, \ell) = \text{Project}(\sigma_1', \ell)$, projective chain binding implies $\text{Project}(\sigma_2, \ell + 1) = \text{Project}(\sigma_2', \ell + 1)$

Using Projective Chainable Commitments

x_1	x_2	$\cdots$	x_ℓ

$\longrightarrow \sigma_{\text{in}}$

Consider two different openings: $(\sigma_1, \sigma_2, \sigma_{\text{out}}, \pi)$ and $(\sigma_1', \sigma_2', \sigma_{\text{out}}', \pi')$

$\hat{x}_1$	$\hat{x}_2$	$\cdots$	$\hat{x}_\ell$								

σ_1, σ_1'

σ_2 and σ_2' agree on first $\ell + 1$ components:
$\text{Project}(\sigma_2, \ell + 1) = \text{Project}(\sigma_2', \ell + 1)$

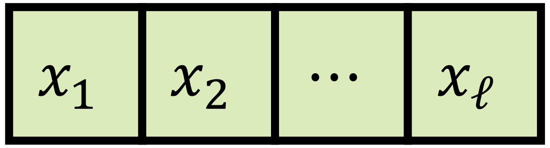
$\tilde{x}_1$	$\tilde{x}_2$	$\cdots$	$\tilde{x}_\ell$	$\tilde{y}_1$							

σ_2, σ_2'

Step 2: Gate consistency between first k wires in σ_1, σ_1'
with first $\ell + 1$ wires in σ_2, σ_2'

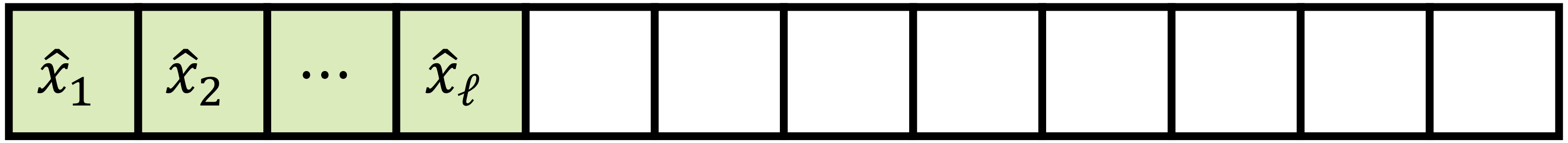
Since $\text{Project}(\sigma_1, \ell) = \text{Project}(\sigma_1', \ell)$, projective chain binding implies $\text{Project}(\sigma_2, \ell + 1) = \text{Project}(\sigma_2', \ell + 1)$

Using Projective Chainable Commitments



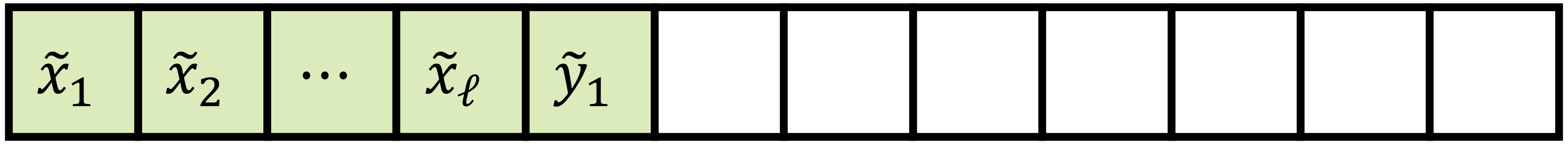
σ_{in}

Consider two different openings: $(\sigma_1, \sigma_2, \sigma_{\text{out}}, \pi)$ and $(\sigma_1', \sigma_2', \sigma_{\text{out}}', \pi')$



σ_1, σ_1'

σ_2 and σ_2' agree on first $\ell + 1$ components:
$\text{Project}(\sigma_2, \ell + 1) = \text{Project}(\sigma_2', \ell + 1)$

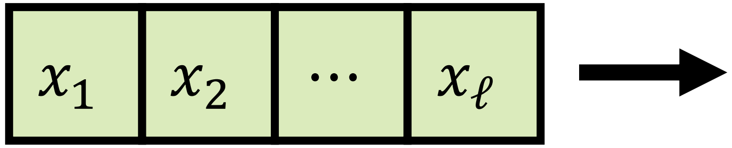


σ_2, σ_2'

Step 3: Internal consistency between first $\ell + 1$ wires in σ_2, σ_2' with first $\ell + 1$ wires in σ_1, σ_1'

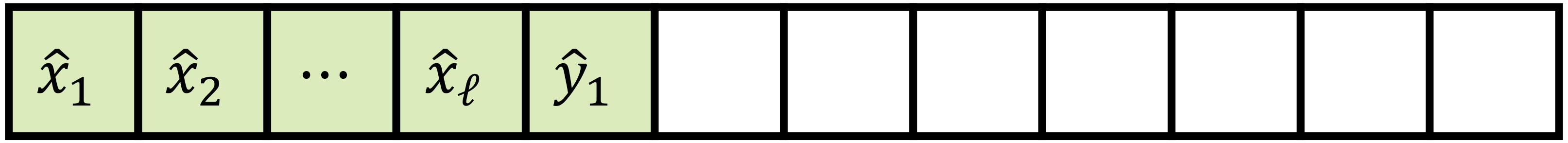
Since $\text{Project}(\sigma_2, \ell + 1) = \text{Project}(\sigma_2', \ell + 1)$, projective chain binding implies $\text{Project}(\sigma_1, \ell + 1) = \text{Project}(\sigma_1', \ell + 1)$

Using Projective Chainable Commitments



σ_{in}

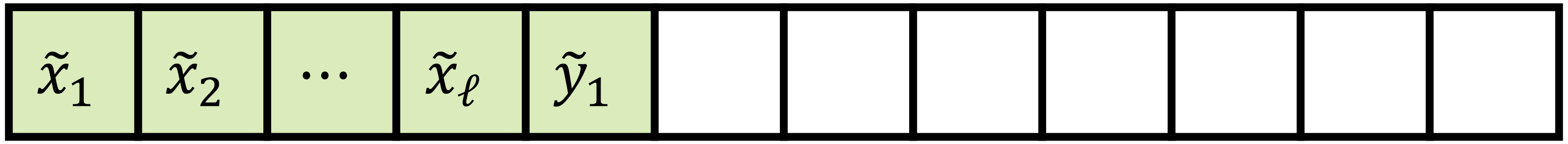
Consider two different openings: $(\sigma_1, \sigma_2, \sigma_{\text{out}}, \pi)$ and $(\sigma_1', \sigma_2', \sigma_{\text{out}}', \pi')$



σ_1, σ_1'

σ_1 and σ_1' agree on first $\ell + 1$ components:

$\text{Project}(\sigma_1, \ell + 1) = \text{Project}(\sigma_1', \ell + 1)$



σ_2, σ_2'

Step 3: Internal consistency between first $\ell + 1$ wires in σ_2, σ_2' with first $\ell + 1$ wires in σ_1, σ_1'

Since $\text{Project}(\sigma_2, \ell + 1) = \text{Project}(\sigma_2', \ell + 1)$, projective chain binding implies $\text{Project}(\sigma_1, \ell + 1) = \text{Project}(\sigma_1', \ell + 1)$

Using Projective Chainable Commitments

x_1	x_2	$\cdots$	x_ℓ

$\longrightarrow \sigma_{\text{in}}$

Consider two different openings: $(\sigma_1, \sigma_2, \sigma_{\text{out}}, \pi)$ and $(\sigma_1', \sigma_2', \sigma_{\text{out}}', \pi')$

$\hat{x}_1$	$\hat{x}_2$	$\cdots$	$\hat{x}_\ell$	$\hat{y}_1$							

σ_1, σ_1'

σ_1 and σ_1' agree on first $\ell + 1$ components:
$\text{Project}(\sigma_1, \ell + 1) = \text{Project}(\sigma_1', \ell + 1)$

$\tilde{x}_1$	$\tilde{x}_2$	$\cdots$	$\tilde{x}_\ell$	$\tilde{y}_1$							

σ_2, σ_2'

Observe: we have established that $\text{Project}(\sigma_1, \ell + 1) = \text{Project}(\sigma_1', \ell + 1)$
Can iterate this strategy for each index $\ell + 1, \ell + 2, \ldots$ to argue that σ_1, σ_1' agree on **all** components

Using Projective Chainable Commitments

x_1	x_2	$\cdots$	x_ℓ

$\longrightarrow \sigma_{\text{in}}$

Consider two different openings: $(\sigma_1, \sigma_2, \sigma_{\text{out}}, \pi)$ and $(\sigma_1', \sigma_2', \sigma_{\text{out}}', \pi')$

$\hat{x}_1$	$\hat{x}_2$	$\cdots$	$\hat{x}_\ell$	$\hat{y}_1$	$\hat{y}_2$	$\cdots$	$\hat{y}_t$	$\hat{z}_1$	$\hat{z}_2$	$\cdots$	$\hat{z}_d$

σ_1, σ_1'

$\tilde{x}_1$	$\tilde{x}_2$	$\cdots$	$\tilde{x}_\ell$	$\tilde{y}_1$	$\tilde{y}_2$	$\cdots$	$\tilde{y}_t$	$\tilde{z}_1$	$\tilde{z}_2$	$\cdots$	$\tilde{z}_d$

σ_2, σ_2'

Observe: we have established that $\mathrm{Project}(\sigma_1, \ell + 1) = \mathrm{Project}(\sigma_1', \ell + 1)$
Can iterate this strategy for each index $\ell + 1, \ell + 2, \ldots$ to argue that σ_1, σ_1' agree on **all** components

Using Projective Chainable Commitments

x_1	x_2	$\cdots$	x_ℓ

$\longrightarrow \sigma_{\text{in}}$

Consider two different openings: $(\sigma_1, \sigma_2, \sigma_{\text{out}}, \pi)$ and $(\sigma_1', \sigma_2', \sigma_{\text{out}}', \pi')$

$\hat{x}_1$	$\hat{x}_2$	$\cdots$	$\hat{x}_\ell$	$\hat{y}_1$	$\hat{y}_2$	$\cdots$	$\hat{y}_t$	$\hat{z}_1$	$\hat{z}_2$	$\cdots$	$\hat{z}_d$

σ_1, σ_1'

$\tilde{x}_1$	$\tilde{x}_2$	$\cdots$	$\tilde{x}_\ell$	$\tilde{y}_1$	$\tilde{y}_2$	$\cdots$	$\tilde{y}_t$	$\tilde{z}_1$	$\tilde{z}_2$	$\cdots$	$\tilde{z}_d$

σ_2, σ_2'

If $\sigma_1 = \sigma_1'$, then final output commitment check ensures $\sigma_{\text{out}} = \sigma_{\text{out}}'$

Similar proof strategy as [GZ21, CJJ21, KLVW23]

Constructing Projective Chainable Commitments

Starting point: Kiltz-Wee [KW15] proof system for proving membership in linear spaces

Suppose we want to support openings to a *fixed* linear function

$$\boldsymbol{x} \in \mathbb{Z}_p^{\ell} \mapsto \boldsymbol{M}\boldsymbol{x} \in \mathbb{Z}_p^{d} \text{ where } \boldsymbol{M} \in \mathbb{Z}_p^{d \times \ell}$$

Let $(\mathbb{G}, \mathbb{G}_T, e)$ be a pairing group and let g be a generator of $\mathbb{G}$

Common reference string contains two vectors $g^{\boldsymbol{t}}$ and $g^{\hat{\boldsymbol{t}}}$ where $\boldsymbol{t} \leftarrow \mathbb{Z}_p^{\ell}$ and $\hat{\boldsymbol{t}} \leftarrow \mathbb{Z}_p^{d}$

Vector $\boldsymbol{t}$ is used to commit to the inputs and vector $\hat{\boldsymbol{t}}$ is used to commit to outputs

Commitment to input $\boldsymbol{x} \in \mathbb{Z}_p^{\ell}$ is $\sigma_{\text{in}} = g^{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}}$

Commitment to output $\boldsymbol{y} \in \mathbb{Z}_p^{d}$ is $\sigma_{\text{out}} = g^{\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{y}}$

Basically a Pedersen (vector) commitment:
if $g^{\boldsymbol{t}} = [h_1, \dots, h_\ell]$, then $\sigma = \prod_{i \in [\ell]} h_i^{x_i}$

Chainable Commitments for Linear Functions

Suppose we want to support openings to a *fixed* linear function

$$\boldsymbol{x} \in \mathbb{Z}_p^{\ell} \mapsto \boldsymbol{M}\boldsymbol{x} \in \mathbb{Z}_p^{d} \text{ where } \boldsymbol{M} \in \mathbb{Z}_p^{d \times \ell}$$

Commitment to input $\boldsymbol{x} \in \mathbb{Z}_p^{\ell}$ is $\sigma_{\text{in}} = g^{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}}$

Commitment to output $\boldsymbol{y} \in \mathbb{Z}_p^{d}$ is $\sigma_{\text{out}} = g^{\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{y}}$

To support openings to the linear function $\boldsymbol{M}$ ($\boldsymbol{x} \mapsto \boldsymbol{M}\boldsymbol{x}$), we also include in the CRS $g^{\boldsymbol{z}^{\mathrm{T}}}$ where

$$\boldsymbol{z}^{\mathrm{T}} = w\boldsymbol{t}^{\mathrm{T}} - r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M} \in \mathbb{Z}_p^{\ell} \quad \text{and} \quad r, w \leftarrow \mathbb{Z}_p$$

Chainable Commitments for Linear Functions

Suppose we want to support openings to a *fixed* linear function

$$\boldsymbol{x} \in \mathbb{Z}_p^{\ell} \mapsto \boldsymbol{M}\boldsymbol{x} \in \mathbb{Z}_p^{d} \text{ where } \boldsymbol{M} \in \mathbb{Z}_p^{d \times \ell}$$

Intuitively: $\boldsymbol{z}$ "recodes" an input commitment with respect to $\boldsymbol{t}$ to an output commitment with respect to $\hat{\boldsymbol{t}}$

Commitment to output $\boldsymbol{y} \in \mathbb{Z}_p^d$ is $\sigma_{\text{out}} = g^{\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{y}}$

$(\boldsymbol{x} \mapsto \boldsymbol{M}\boldsymbol{x})$, we also include in the CRS $g^{\boldsymbol{z}^{\mathrm{T}}}$ where

$$\boldsymbol{z}^{\mathrm{T}} = w\boldsymbol{t}^{\mathrm{T}} - r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M} \in \mathbb{Z}_p^{\ell} \quad \text{and} \quad r, w \leftarrow \mathbb{Z}_p$$

Chainable Commitments for Linear Functions

Suppose we want to support openings to a *fixed* linear function

$$\boldsymbol{x} \in \mathbb{Z}_p^{\ell} \mapsto \boldsymbol{M}\boldsymbol{x} \in \mathbb{Z}_p^{d} \text{ where } \boldsymbol{M} \in \mathbb{Z}_p^{d\times\ell}$$

Commitment to input $\boldsymbol{x} \in \mathbb{Z}_p^{\ell}$ is $\sigma_{\text{in}} = g^{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}}$ Commitment to output $\boldsymbol{y} \in \mathbb{Z}_p^{d}$ is $\sigma_{\text{out}} = g^{\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{y}}$

To support openings to the linear function $\boldsymbol{M}$ ($\boldsymbol{x} \mapsto \boldsymbol{M}\boldsymbol{x}$), we also include in the CRS $g^{\boldsymbol{z}^{\mathrm{T}}}$ where

$$\boldsymbol{z}^{\mathrm{T}} = w\boldsymbol{t}^{\mathrm{T}} - r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M} \in \mathbb{Z}_p^{\ell} \quad \text{and} \quad r, w \leftarrow \mathbb{Z}_p$$

For now, we consider the **designated-verifier** setting where **secret key** needed to check proofs

Opening: $\pi = g^{\boldsymbol{z}^{\mathrm{T}}\boldsymbol{x}}$

Verification relation: Check that $\pi = \dfrac{\sigma_{\text{in}}^{w}}{\sigma_{\text{out}}^{r}}$

Secret verification key: r, w

Correctness: $\dfrac{\sigma_{\text{in}}^{w}}{\sigma_{\text{out}}^{r}} = \dfrac{g^{w\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}}}{g^{r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{y}}} = \dfrac{g^{w\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}}}{g^{r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{x}}} = g^{(w\boldsymbol{t}^{\mathrm{T}} - r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M})\boldsymbol{x}} = g^{\boldsymbol{z}^{\mathrm{T}}\boldsymbol{x}} = \pi$

Security for Linear Functions

Suppose we want to support openings to a *fixed* linear function

$$\boldsymbol{x} \in \mathbb{Z}_p^\ell \mapsto \boldsymbol{M}\boldsymbol{x} \in \mathbb{Z}_p^d \text{ where } \boldsymbol{M} \in \mathbb{Z}_p^{d\times\ell}$$

Common reference string: $g^{\boldsymbol{t}}, g^{\hat{\boldsymbol{t}}}, g^{w\boldsymbol{t}^{\mathrm{T}} - r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M}}$

Verification relation: Check that $\pi = \frac{\sigma_{\text{in}}^w}{\sigma_{\text{out}}^r}$

Suppose adversary produces the following:

Input commitment $\sigma_{\text{in}} = g^c$

Output commitments $\sigma_{\text{out}} = g^{\hat{c}}, \sigma'_{\text{out}} = g^{\hat{c}'}$

Openings $\pi = g^v, \pi' = g^{v'}$

If the openings are valid, then

$$v = wc - r\hat{c}$$
$$v' = wc - r\hat{c}'$$

Thus, $v - v' = r(\hat{c} - \hat{c}')$

Non-zero since $\hat{c} \neq \hat{c}'$

Security for Linear Functions

Suppose we want to support openings to a *fixed* linear function

Under DDH, $\boldsymbol{wt}$ computationally hides value of r

Common reference string: $g^{\boldsymbol{t}}, g^{\hat{\boldsymbol{t}}}, g^{\boldsymbol{w}\boldsymbol{t}^{\mathsf{T}} - r\hat{\boldsymbol{t}}^{\mathsf{T}}\boldsymbol{M}}$

Technically: DDH does not hold in a symmetric pairing group, but can use asymmetric group (or k-Lin)

Verification relation: Check that $\pi = \dfrac{\sigma_{\text{in}}^{w}}{\sigma_{\text{out}}^{r}}$

Suppose adversary produces the following:

Input commitment $\sigma_{\text{in}} = g^{c}$

Output commitments $\sigma_{\text{out}} = g^{\hat{c}}, \sigma'_{\text{out}} = g^{\hat{c}'}$

Openings $\pi = g^{v}, \pi' = g^{v'}$

Distribution of $r(\hat{c} - \hat{c}')$ is pseudorandom from the perspective of the adversary, so this check passes with probability $1/p$

Thus, $v - v' = r(\hat{c} - \hat{c}')$

Non-zero since $\hat{c} \neq \hat{c}'$

Chainable Commitments for Linear Functions

Suppose we want to support openings to a *fixed* linear function

$$\boldsymbol{x} \in \mathbb{Z}_p^{\ell} \mapsto \boldsymbol{M}\boldsymbol{x} \in \mathbb{Z}_p^{d} \text{ where } \boldsymbol{M} \in \mathbb{Z}_p^{d \times \ell}$$

Common reference string: $g^{\boldsymbol{t}}, g^{\hat{\boldsymbol{t}}}, g^{w\boldsymbol{t}^{\mathrm{T}} - r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M}}$

$\sigma_{\mathrm{in}} = g^{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}}$

Verification relation: Check that $\pi = \frac{\sigma_{\mathrm{in}}^{w}}{\sigma_{\mathrm{out}}^{r}}$

$\sigma_{\mathrm{out}} = g^{\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{y}}$

Lots of caveats:

Only supports **fixed** functions

Only supports **linear** functions

Only **designated-verifier**

Chainable Commitments for Linear Functions

Suppose we want to support openings to a *fixed* linear function

$$\boldsymbol{x} \in \mathbb{Z}_p^\ell \mapsto \boldsymbol{M}\boldsymbol{x} \in \mathbb{Z}_p^d \text{ where } \boldsymbol{M} \in \mathbb{Z}_p^{d \times \ell}$$

Common reference string: $g^{\boldsymbol{t}}, g^{\hat{\boldsymbol{t}}}, g^{w\boldsymbol{t}^{\mathrm{T}} - r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M}}$

Verification relation: Check that $\pi = \frac{\sigma_{\mathrm{in}}^w}{\sigma_{\mathrm{out}}^r}$

$\sigma_{\mathrm{in}} = g^{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}}$

$\sigma_{\mathrm{out}} = g^{\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{y}}$

Caveat: Only supports **fixed** functions

Extend to arbitrary functions by relying on **linear homomorphism**

Suppose we publish $g^{\boldsymbol{z}_1^{\mathrm{T}}} = g^{w_1\boldsymbol{t}^{\mathrm{T}} - r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M}_1}$ and $g^{\boldsymbol{z}_2^{\mathrm{T}}} = g^{w_2\boldsymbol{t}^{\mathrm{T}} - r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M}_2}$ in the CRS

$\sigma_{\mathrm{in}} = g^{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}}$

$\sigma_{\mathrm{out}} = g^{\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{y}}$

$g^{\alpha_1\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{x}}$ is an opening to $\boldsymbol{y} = \alpha_1\boldsymbol{M}_1\boldsymbol{x}$

$$\frac{\sigma_{\mathrm{in}}^{\alpha_1 w_1}}{\sigma_{\mathrm{out}}^r} = g^{\alpha_1 w_1\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x} - r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{y}} = g^{\alpha_1 w_1\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x} - \alpha_1 r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M}_1\boldsymbol{x}} = g^{\alpha_1\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{x}}$$

Chainable Commitments for Linear Functions

Suppose we want to support openings to a *fixed* linear function

$$\boldsymbol{x} \in \mathbb{Z}_p^\ell \mapsto \boldsymbol{M}\boldsymbol{x} \in \mathbb{Z}_p^d \text{ where } \boldsymbol{M} \in \mathbb{Z}_p^{d\times\ell}$$

Common reference string: $g^{\boldsymbol{t}}, g^{\hat{\boldsymbol{t}}}, g^{w\boldsymbol{t}^{\mathrm{T}} - r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M}}$

Verification relation: Check that $\pi = \frac{\sigma_{\mathrm{in}}^w}{\sigma_{\mathrm{out}}^r}$

$\sigma_{\mathrm{in}} = g^{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}}$

$\sigma_{\mathrm{out}} = g^{\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{y}}$

Caveat: Only supports **fixed** functions

Extend to arbitrary functions by relying on **linear homomorphism**

Suppose we publish $g^{\boldsymbol{z}_1^{\mathrm{T}}} = g^{w_1\boldsymbol{t}^{\mathrm{T}} - r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M}_1}$ and $g^{\boldsymbol{z}_2^{\mathrm{T}}} = g^{w_2\boldsymbol{t}^{\mathrm{T}} - r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M}_2}$ in the CRS

$\sigma_{\mathrm{in}} = g^{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}}$ $\qquad g^{\alpha_1\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{x}}$ is an opening to $\alpha_1\boldsymbol{M}_1\boldsymbol{x}$

$\sigma_{\mathrm{out}} = g^{\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{y}}$ $\qquad g^{\alpha_2\boldsymbol{z}_2^{\mathrm{T}}\boldsymbol{x}}$ is an opening to $\alpha_2\boldsymbol{M}_2\boldsymbol{x}$

Chainable Commitments for Linear Functions

$$\frac{\sigma_{\text{in}}^{\alpha_1 w_1}}{g^{r\hat{\boldsymbol{t}}^{\mathrm{T}}(\alpha_1 \boldsymbol{M}_1 \boldsymbol{x})}} = g^{\alpha_1 \boldsymbol{z}_1^{\mathrm{T}} \boldsymbol{x}} \qquad \frac{\sigma_{\text{in}}^{\alpha_2 w_2}}{g^{r\hat{\boldsymbol{t}}^{\mathrm{T}}(\alpha_2 \boldsymbol{M}_2 \boldsymbol{x})}} = g^{\alpha_2 \boldsymbol{z}_2^{\mathrm{T}} \boldsymbol{x}}$$

Caveat: Only supports **fixed** functions

Extend to arbitrary functions by relying on **linear homomorphism**

Suppose we publish $g^{\boldsymbol{z}_1^{\mathrm{T}}} = g^{w_1 \boldsymbol{t}^{\mathrm{T}} - r\hat{\boldsymbol{t}}^{\mathrm{T}} \boldsymbol{M}_1}$ and $g^{\boldsymbol{z}_2^{\mathrm{T}}} = g^{w_2 \boldsymbol{t}^{\mathrm{T}} - r\hat{\boldsymbol{t}}^{\mathrm{T}} \boldsymbol{M}_2}$ in the CRS

$\sigma_{\text{in}} = g^{\boldsymbol{t}^{\mathrm{T}} \boldsymbol{x}}$ $\quad g^{\alpha_1 \boldsymbol{z}_1^{\mathrm{T}} \boldsymbol{x} + \alpha_2 \boldsymbol{z}_2^{\mathrm{T}} \boldsymbol{x}}$ is an opening to $\boldsymbol{y} = \alpha_1 \boldsymbol{M}_1 \boldsymbol{x} + \alpha_2 \boldsymbol{M}_2 \boldsymbol{x}$

$\sigma_{\text{out}} = g^{\hat{\boldsymbol{t}}^{\mathrm{T}} \boldsymbol{y}}$ $\quad \dfrac{\sigma_{\text{in}}^{\alpha_1 w_1 + \alpha_2 w_2}}{\sigma_{\text{out}}^{r}} = g^{\alpha_1 \boldsymbol{z}_1^{\mathrm{T}} \boldsymbol{x} + \alpha_2 \boldsymbol{z}_2^{\mathrm{T}} \boldsymbol{x}}$

Verification relation for $\boldsymbol{x} \mapsto (\alpha_1 \boldsymbol{M}_1 + \alpha_2 \boldsymbol{M}_2)\boldsymbol{x}$

Chainable Commitments for Linear Functions

$$\frac{\sigma_{\text{in}}^{\alpha_1 w_1}}{g^{r\hat{\boldsymbol{t}}^{\mathrm{T}}(\alpha_1 \boldsymbol{M}_1 \boldsymbol{x})}} = g^{\alpha_1 \boldsymbol{z}_1^{\mathrm{T}} \boldsymbol{x}} \qquad \frac{\sigma_{\text{in}}^{\alpha_2 w_2}}{g^{r\hat{\boldsymbol{t}}^{\mathrm{T}}(\alpha_2 \boldsymbol{M}_2 \boldsymbol{x})}} = g^{\alpha_2 \boldsymbol{z}_2^{\mathrm{T}} \boldsymbol{x}}$$

Caveat: Only supports **fixed** functions

Extend to arbitrary functions by relying on **linear homomorphism**

Publish components for complete basis of linear functions

$$\boldsymbol{M}_{i,j} = \begin{array}{cl} \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & 1 & \vdots \\ 0 & \cdots & 0 \end{bmatrix} & \leftarrow \text{column } j \\ \quad\;\; \uparrow \text{row } i & \end{array}$$

Any linear function $\boldsymbol{M}$ can be expressed as a linear combination of $\boldsymbol{M}_{i,j}$

Chainable Commitments for Linear Functions

Suppose we want to support openings to a *fixed* linear function

$$\boldsymbol{x} \in \mathbb{Z}_p^{\ell} \mapsto \boldsymbol{M}\boldsymbol{x} \in \mathbb{Z}_p^{d} \text{ where } \boldsymbol{M} \in \mathbb{Z}_p^{d \times \ell}$$

Common reference string: $g^{\boldsymbol{t}}, g^{\hat{\boldsymbol{t}}}, g^{w\boldsymbol{t}^{\mathrm{T}} - r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M}}$

Verification relation: Check that $\pi = \frac{\sigma_{\mathrm{in}}^{w}}{\sigma_{\mathrm{out}}^{r}}$

$\sigma_{\mathrm{in}} = g^{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}}$

$\sigma_{\mathrm{out}} = g^{\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{y}}$

Caveat: Only supports **linear** functions

Can extend to quadratic functions by linearization (and tensoring)

Quadratic function of $\boldsymbol{x}$ is a linear function of $\boldsymbol{x} \otimes \boldsymbol{x}$

[see paper for details]

Prover commits to $\boldsymbol{x} \otimes \boldsymbol{x}$ and evaluates a linear function; certify well-formedness of commitment using pairing

Chainable Commitments for Linear Functions

Suppose we want to support openings to a *fixed* linear function

$$\boldsymbol{x} \in \mathbb{Z}_p^{\ell} \mapsto \boldsymbol{M}\boldsymbol{x} \in \mathbb{Z}_p^{d} \text{ where } \boldsymbol{M} \in \mathbb{Z}_p^{d \times \ell}$$

Common reference string: $g^{\boldsymbol{t}}, g^{\hat{\boldsymbol{t}}}, g^{w\boldsymbol{t}^{\mathrm{T}} - r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M}}$

$\sigma_{\mathrm{in}} = g^{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}}$

Verification relation: Check that $\pi = \frac{\sigma_{\mathrm{in}}^{w}}{\sigma_{\mathrm{out}}^{r}}$

$\sigma_{\mathrm{out}} = g^{\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{y}}$

Caveat: Only **designated-verifier**

Solution: encode the verification key r and w in the exponent (following [KW15])

Chainable Commitments for Linear Functions

Suppose we want to support openings to a *fixed* linear function

$$\boldsymbol{x} \in \mathbb{Z}_p^{\ell} \mapsto \boldsymbol{M}\boldsymbol{x} \in \mathbb{Z}_p^{d} \text{ where } \boldsymbol{M} \in \mathbb{Z}_p^{d \times \ell}$$

Common reference string: g

Verification relation: Check t

Previous argument required that r was computationally hidden, so we cannot just give out g^r

Caveat: Only **designated-veri**

Solution: encode the verification key r and w in the exponent (following [KW15])

Chainable Commitments for Linear Functions

Suppose we want to support openings to a *fixed* linear function

$$\boldsymbol{x} \in \mathbb{Z}_p^{\ell} \mapsto \boldsymbol{M}\boldsymbol{x} \in \mathbb{Z}_p^{d} \text{ where } \boldsymbol{M} \in \mathbb{Z}_p^{d \times \ell}$$

Common reference string: $g^{\boldsymbol{t}}, g^{\hat{\boldsymbol{t}}}, g^{w\boldsymbol{t}^{\mathrm{T}} - r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M}}$ $\qquad \sigma_{\text{in}} = g^{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}}$

Verification relation: Check that $\pi = \dfrac{\sigma_{\text{in}}^{w}}{\sigma_{\text{out}}^{r}}$ $\qquad \sigma_{\text{out}} = g^{\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{y}}$

Caveat: Only **designated-verifier**

Solution: encode the verification key r and w in the exponent (following [KW15])

Sample $\boldsymbol{a} \leftarrow \mathbb{Z}_p^2$ $\qquad$ CRS: $g^{\boldsymbol{t}}, g^{\hat{\boldsymbol{t}}}, g^{\boldsymbol{a}}, g^{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{w}}, g^{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{r}}, g^{\boldsymbol{w}\boldsymbol{t}^{\mathrm{T}} - \boldsymbol{r}\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M}}$

Sample $\boldsymbol{w}, \boldsymbol{r} \leftarrow \mathbb{Z}_p^2$ $\qquad$ Verification relation is now

$\sigma_{\text{in}} = g^{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}} \qquad \sigma_{\text{out}} = g^{\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{x}} \qquad e\left(g^{\boldsymbol{a}^{\mathrm{T}}}, \boldsymbol{\pi}\right) = \dfrac{e\left(g^{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{w}}, \sigma_{\text{in}}\right)}{e\left(g^{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{r}}, \sigma_{\text{out}}\right)} \qquad \boldsymbol{\pi} = g^{\boldsymbol{w}\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{r}\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{x}}$

Chainable Commitments for Linear Functions

Suppose we want to support openings to a *fixed* linear function

$$\boldsymbol{x} \in \mathbb{Z}_p^\ell \mapsto \boldsymbol{Mx} \in \mathbb{Z}_p^d \text{ where } \boldsymbol{M} \in \mathbb{Z}_p^{d\times\ell}$$

Common reference string: $g^{\boldsymbol{t}}, g^{\hat{\boldsymbol{t}}}, g^{w\boldsymbol{t}^{\mathrm{T}} - r\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M}}$ $\sigma_{\text{in}} = g^{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}}$

Verification relation: Check that $\pi = \frac{\sigma_{\text{in}}^w}{\sigma_{\text{out}}^r}$

In this approach, $\boldsymbol{r}$ has one unit of entropy given $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{r}$, so we can still carry out a similar argument as before

Caveat: Only **designated-verifier**

Solution: encode the verification key r and w

Sample $\boldsymbol{a} \leftarrow \mathbb{Z}_p^2$

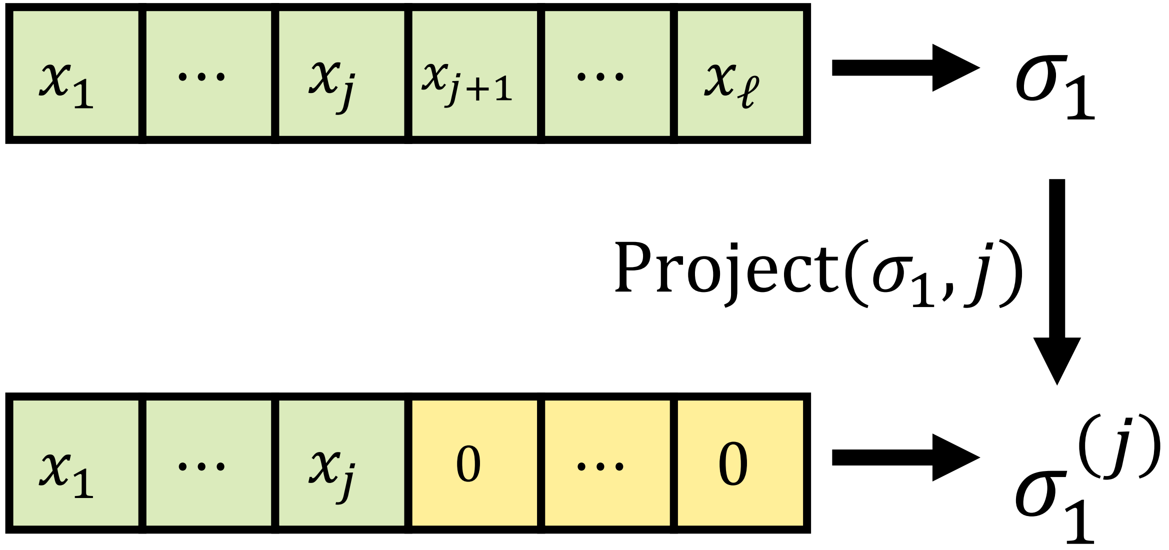
Sample $\boldsymbol{w}, \boldsymbol{r} \leftarrow \mathbb{Z}_p^2$

CRS: $g^{\boldsymbol{t}}, g^{\hat{\boldsymbol{t}}}, g^{\boldsymbol{a}}, g^{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{w}}, g^{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{r}}, g^{\boldsymbol{w}\boldsymbol{t}^{\mathrm{T}} - \boldsymbol{r}\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{M}}$

Verification relation is now

$$\sigma_{\text{in}} = g^{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}} \qquad \sigma_{\text{out}} = g^{\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{Mx}} \qquad e\left(g^{\boldsymbol{a}^{\mathrm{T}}}, \boldsymbol{\pi}\right) = \frac{e\left(g^{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{w}}, \sigma_{\text{in}}\right)}{e\left(g^{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{r}}, \sigma_{\text{out}}\right)} \qquad \boldsymbol{\pi} = g^{\boldsymbol{w}\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{r}\hat{\boldsymbol{t}}^{\mathrm{T}}\boldsymbol{Mx}}$$

Projective Chainable Commitments



Need a way to project a commitment onto a subset of its components

$$g^{\boldsymbol{t}} = [h_1, \dots, h_\ell]$$

$$\sigma = g^{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}} = \prod_{i \in [\ell]} h_i^{x_i}$$

In **composite-order groups:** introduce a subgroup for components in projection set

Suppose $\mathbb{G}$ has order $N = pq$ and let $\mathbb{G}_p, \mathbb{G}_q$ be the order-p and order-q subgroups of $\mathbb{G}$

Let g_p be a generator of $\mathbb{G}_p$ and g_q be a generator of $\mathbb{G}_q$

Replace $g^{\boldsymbol{t}}$ with $h_1 = (g_p g_q)^{t_1}, \dots, h_j = (g_p g_q)^{t_j}, h_{j+1} = g_p^{t_{j+1}}, \dots, h_\ell = g_p^{t_\ell}$

Projective Chainable Commitments

Commitment is now

$$\sigma = \prod_{i\in[\ell]} h_i^{x_i} = \prod_{i=1}^{j} (g_p g_q)^{t_i x_i} \prod_{i=j+1}^{\ell} g_p^{t_i x_i}$$

If we consider σ in the mod-q subgroup, then

$$\sigma_q = \prod_{i\in[j]} g_q^{t_i x_i}$$

This is precisely a commitment to the first j components!

Need a way to project a commitment onto a subset of its components

$$g^{\boldsymbol{t}} = [h_1, \dots, h_\ell]$$

$$\sigma = g^{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{x}} = \prod_{i\in[\ell]} h_i^{x_i}$$

In **composite-order groups:** introduce a subgroup for components in projection set

Suppose $\mathbb{G}$ has order $N = pq$ and let $\mathbb{G}_p, \mathbb{G}_q$ be the order-p and order-q subgroups of $\mathbb{G}$

Let g_p be a generator of $\mathbb{G}_p$ and g_q be a generator of $\mathbb{G}_q$

Replace $g^{\boldsymbol{t}}$ with $h_1 = (g_p g_q)^{t_1}, \dots, h_j = (g_p g_q)^{t_j}, h_{j+1} = g_p^{t_{j+1}}, \dots, h_\ell = g_p^{t_\ell}$

Projective Chainable Commitments

Commitment is now

$$\sigma = \prod_{i \in [\ell]} h_i^{x_i} = \prod_{i=1}^{j} (g_p g_q)^{t_i x_i} \prod_{i=j+1}^{\ell} g_p^{t_i x_i}$$

If we consider σ in the mod-q subgroup, then

$$\sigma_q = \prod_{i \in [j]} g_q^{t_i x_i}$$

This is precisely a commitment to the first j components!

Syntactic issue: We were considering linear/quadratic functions over $\mathbb{Z}_p$ before; when using composite-order groups, we should view it as functions over the integers

$i \in [\ell]$

Main idea: embed **two** copies of the chainable commitment scheme:

- The normal scheme is embedded in the $\mathbb{G}_p$-subgroup
- The projected scheme is embedded in the $\mathbb{G}_q$-subgroup

When reasoning about chain binding, we implement the previous proof argument within the $\mathbb{G}_q$ subgroup

Projective Chainable Commitments

Commitment is now

$$\sigma = \prod_{i \in [\ell]} h_i^{x_i} = \prod_{i=1}^{j} (g_p g_q)^{t_i x_i} \prod_{i=j+1}^{\ell} g_p^{t_i x_i}$$

If we consider σ in the mod-q subgroup, then

$$\sigma_q = \prod_{i \in [j]} g_q^{t_i x_i}$$

This is precisely a commitment to the first j components!

Syntactic issue: We were considering linear/quadratic functions over $\mathbb{Z}_p$ before; when using composite-order groups, we should view it as functions over the integers

$i \in [\ell]$

Main idea: embed **two** copies of the chainable commitment scheme:

- The normal scheme is embedded in the $\mathbb{G}_p$-subgroup
- The projected scheme is embedded in the $\mathbb{G}_q$-subgroup

In paper: use prime-order groups and consider two orthogonal subspaces (normal scheme in one subspace and projected scheme in the other); security reduces to (bilateral) k-Lin

[see paper for details; see also [GZ21] for similar projection approach]

Functional Commitments for Circuits

Goal: Constant number of group elements for commitment **and** openings

Commitment:

$[x_1 \mid x_2 \mid \cdots \mid x_\ell] \longrightarrow \sigma_{\text{in}}$

Verifier know output $(z_1, \ldots, z_d)$:

$[z_1 \mid z_2 \mid \cdots \mid z_d] \longrightarrow \sigma_{\text{out}}$

Opening: commit to **all** wires (i.e., concatenated together) **twice**

$[x_1 \mid x_2 \mid \cdots \mid x_\ell \mid y_1 \mid y_2 \mid \cdots \mid y_t \mid z_1 \mid z_2 \mid \cdots \mid z_d] \longrightarrow \sigma_1$

Use projective chain binding and
an iterative argument to argue binding

$[x_1 \mid x_2 \mid \cdots \mid x_\ell \mid y_1 \mid y_2 \mid \cdots \mid y_t \mid z_1 \mid z_2 \mid \cdots \mid z_d] \longrightarrow \sigma_2$

Summary

This work: functional commitments for **general circuits** using **pairings**

Scheme	Function Class	$\lvert\mathrm{crs}\rvert$	$\lvert\sigma\rvert$	$\lvert\pi\rvert$	Assumption
This work	arithmetic circuits	$O(s^5)$	$O(1)$	$O(1)$	bilateral k-Lin

- First pairing-based construction for general **circuits** based on **falsifiable** assumptions where commitment and openings contain **constant** number of group elements
- First scheme that only makes **black-box** use of cryptographic primitives/algorithms where the commitment + opening size is $\mathrm{poly}(\lambda)$ bits

Open problem: Construction with shorter CRS (e.g., linear-size)? Then, parameters would match state-of-the-art pairing-based SNARKs

Thank you!

https://eprint.iacr.org/2024/688