# Batch Arguments for NP from Standard Bilinear Group Assumptions

Brent Waters and David Wu

# Batch Arguments for NP

Boolean circuit satisfiability

$$\mathcal{L}_C = \{x \in \{0,1\}^n : C(x, w) = 1 \text{ for some } w\}$$

prover

$(x_1, \ldots, x_m)$

prover has $m$ statements and wants to convince verifier that $x_i \in \mathcal{L}_C$ for all $i \in [m]$

verifier

# Batch Arguments for NP

Boolean circuit satisfiability

$$\mathcal{L}_C = \{x \in \{0,1\}^n : C(x, w) = 1 \text{ for some } w\}$$

prover

$(x_1, \ldots, x_m)$

$\pi = (w_1, \ldots, w_m)$

verifier

**Naïve solution:** send witnesses $w_1, \ldots, w_m$ and verifier checks $C(x_i, w_i) = 1$ for all $i \in [m]$

Can the proof size be sublinear in the number of instances $m$?

# Goal: Amortize the Cost of NP Verification

Boolean circuit satisfiability

$$\mathcal{L}_C = \{x \in \{0,1\}^n : C(x, w) = 1 \text{ for some } w\}$$

prover

$(x_1, \ldots, x_m)$

$\pi$

verifier

**Proof size:** $|\pi| = |C| \cdot \mathrm{poly}(\log m, \lambda)$

*"Proof size for a single instance"*

$\lambda$ : security parameter

Proof size scales *sublinearly* with the number of instances

# Goal: Amortize the Cost of NP Verification

Boolean circuit satisfiability

$$\mathcal{L}_C = \{x \in \{0,1\}^n : C(x, w) = 1 \text{ for some } w\}$$

prover

$(x_1, \ldots, x_m)$

$\pi$

verifier

**Proof size:** $|\pi| = |C| \cdot \text{poly}(\log m, \lambda)$

Similar* requirement on verification time

*Verifier does need to read statements so we do allow a $\text{poly}(\lambda, m, n)$ dependence

# Batch Arguments for NP

## Special case of succinct non-interactive arguments for NP (SNARGs)

Constructions rely on idealized models or knowledge assumptions or indistinguishability obfuscation

## Batch arguments from correlation intractable hash functions

Sub-exponential DDH (in pairing-free groups) + QR (with $\sqrt{m}$ size proofs) [CJJ21a]

Learning with errors (LWE) [CJJ21b]

## Batch arguments from pairing-based assumptions

Non-standard, but falsifiable $q$-type assumption on bilinear groups [KPY19]

# This Work

New constructions of non-interactive batch arguments for NP

Batch arguments for NP from standard assumptions over bilinear maps

- $k$-Linear assumption (for any $k \geq 1$) in prime-order bilinear groups
- Subgroup decision assumption in composite-order bilinear groups

**Key feature:** Construction is "low-tech"

- No heavy tools like correlation-intractable hash functions or probabilistically-checkable proofs
- Direct "commit-and-prove" approach à la classic NIZK construction of Groth-Ostrovsky-Sahai

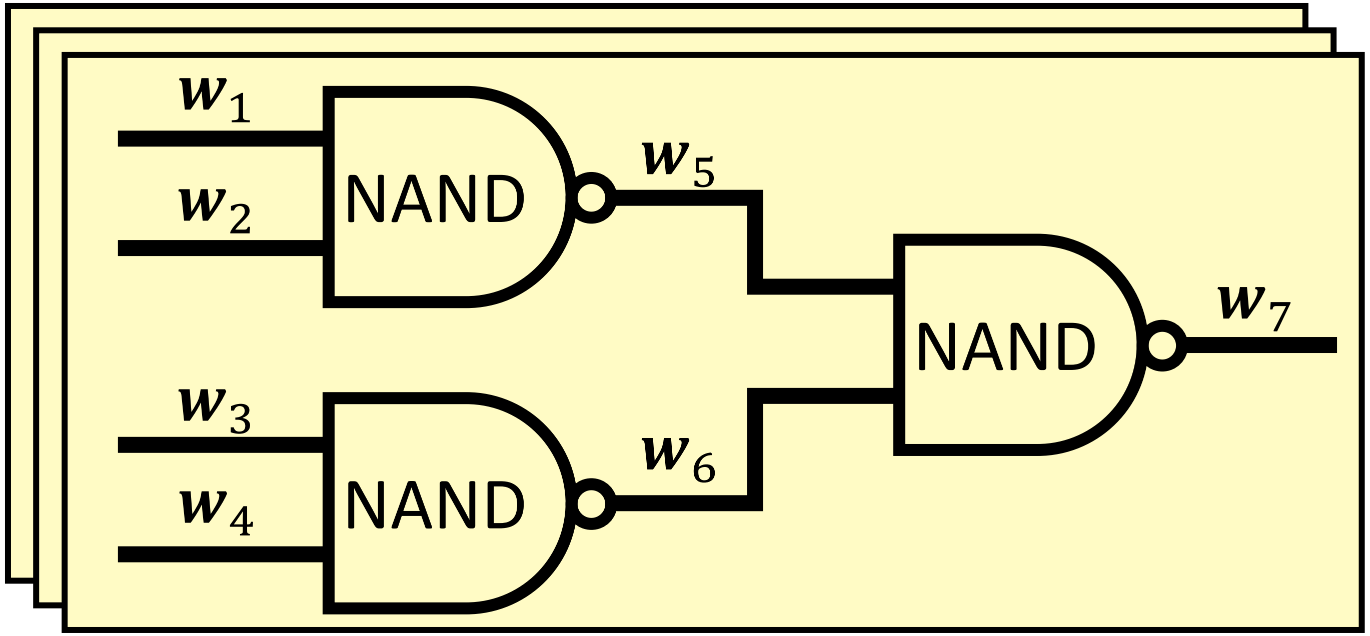
**Corollary:** RAM delegation (i.e., "SNARG for P") with sublinear CRS from standard bilinear map assumptions

- **Previous bilinear map constructions:** need non-standard assumptions [KPY19] or have long CRS [GZ21]

**Corollary:** Aggregate signature with bounded aggregation from standard bilinear map assumptions

- **Previous bilinear map constructions:** random oracle based [BGLS03]

# A Commit-and-Prove Strategy for Batch Arguments



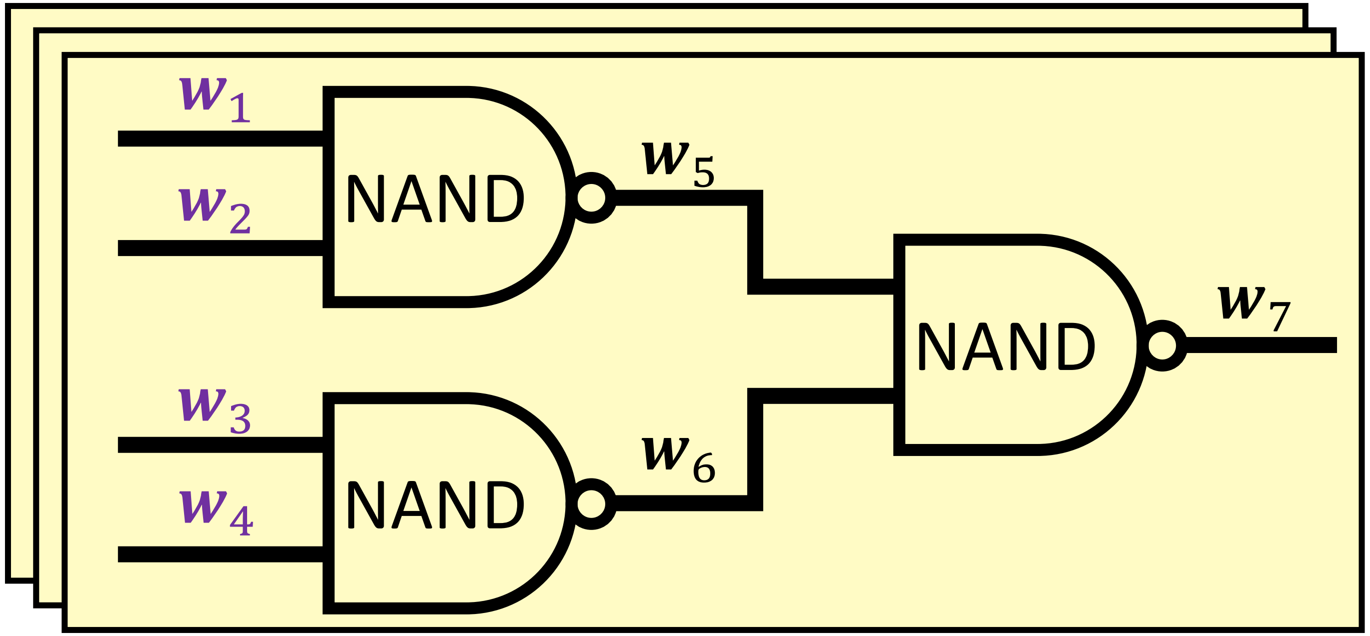

Let $\boldsymbol{w}_i = (w_{i,1}, \dots, w_{i,m})$ be vector of wire labels associated with wire $i$ across the $m$ instances

1 Prover commits to each vector of wire assignments


**Requirement:** $|\sigma_i| = \text{poly}(\lambda, \log m)$

**Our construction:** $|\sigma_i| = \text{poly}(\lambda)$

# A Commit-and-Prove Strategy for Batch Arguments




Let $\boldsymbol{w}_i = (w_{i,1}, \dots, w_{i,m})$ be vector of wire labels associated with wire $i$ across the $m$ instances

1 Prover commits to each vector of wire assignments

$\boldsymbol{w}_i =$ $w_{i,1}$ $w_{i,2}$ $\cdots$ $w_{i,m}$ → $\sigma_i$

**Requirement:** $|\sigma_i| = \text{poly}(\lambda, \log m)$

**Our construction:** $|\sigma_i| = \text{poly}(\lambda)$

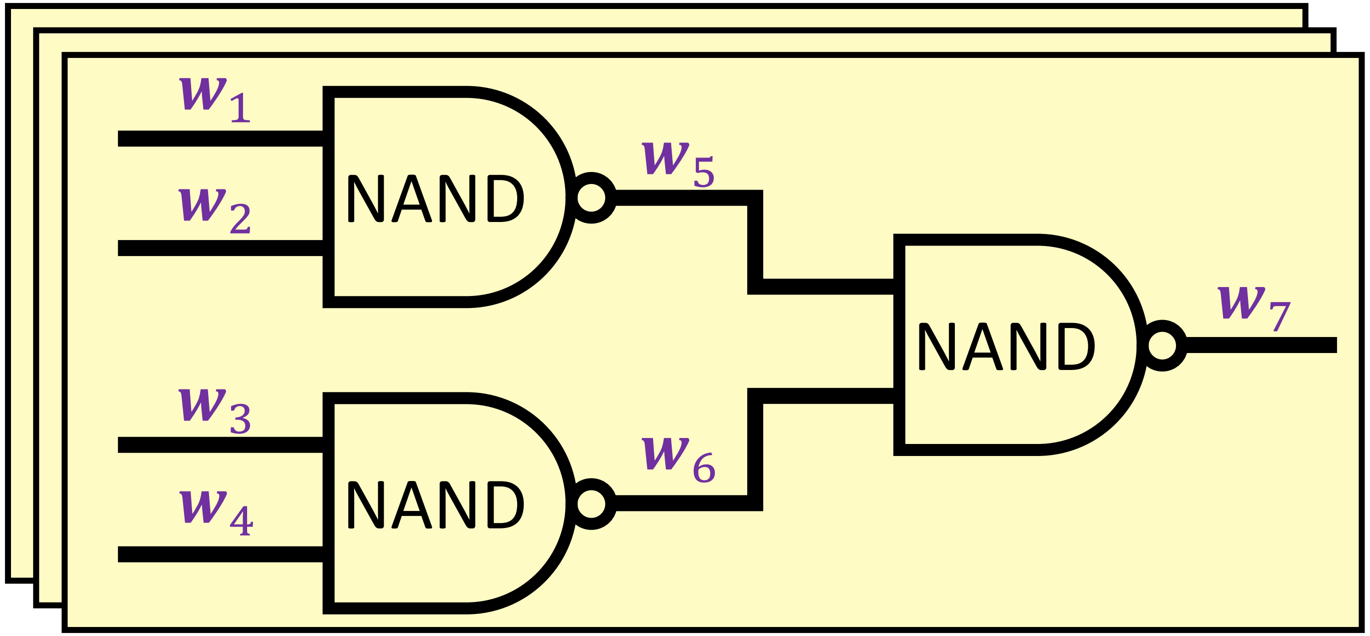
2 Prover constructs the following proofs:

**Input validity**

Commitments to the statement wires are correctly computed

Commitments in our scheme are *deterministic*, so verifier can directly check

# A Commit-and-Prove Strategy for Batch Arguments




Let $\boldsymbol{w}_i = (w_{i,1}, \dots, w_{i,m})$ be vector of wire labels associated with wire $i$ across the $m$ instances

**1** Prover commits to each vector of wire assignments


**Requirement:** $|\sigma_i| = \text{poly}(\lambda, \log m)$

**Our construction:** $|\sigma_i| = \text{poly}(\lambda)$

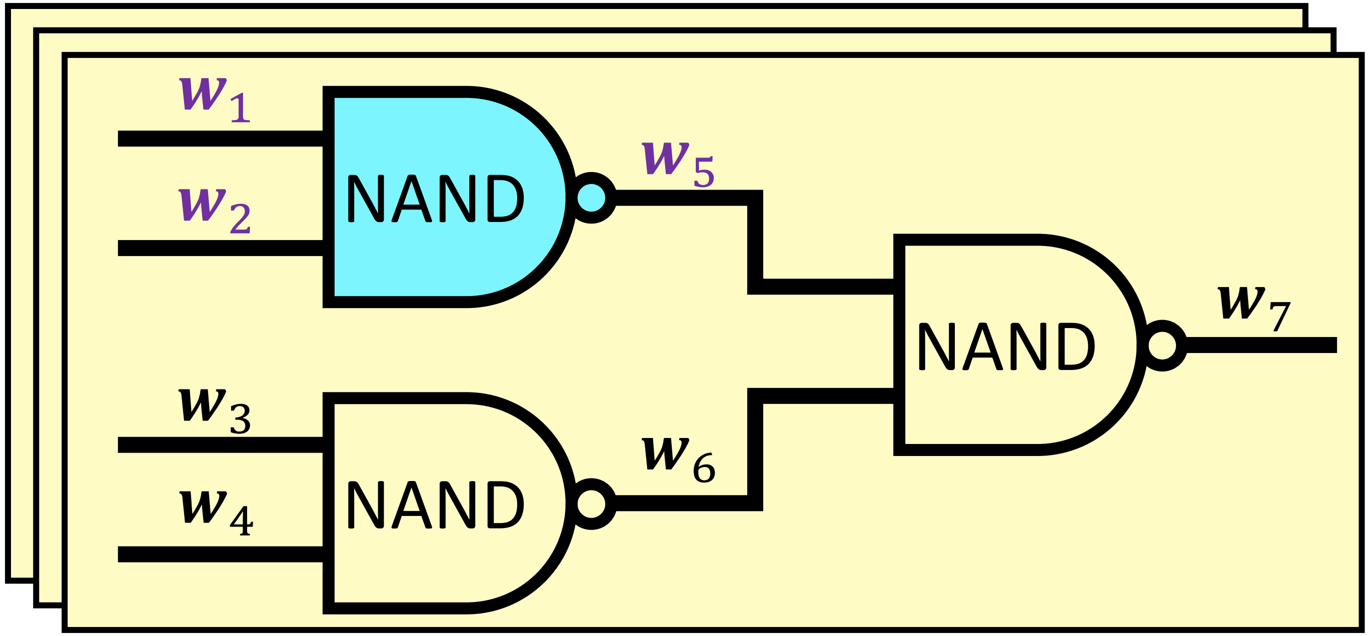
**2** Prover constructs the following proofs:

**Input validity**

**Wire validity**

Commitment for each wire is a commitment to a 0/1 vector

# A Commit-and-Prove Strategy for Batch Arguments




Let $\boldsymbol{w}_i = (w_{i,1}, \ldots, w_{i,m})$ be vector of wire labels associated with wire $i$ across the $m$ instances

**1** Prover commits to each vector of wire assignments

$$\boldsymbol{w}_i = \boxed{w_{i,1}}\ \boxed{w_{i,2}}\ \boxed{\cdots}\ \boxed{w_{i,m}} \rightarrow \boxed{\sigma_i}$$

**Requirement:** $|\sigma_i| = \mathrm{poly}(\lambda, \log m)$

**Our construction:** $|\sigma_i| = \mathrm{poly}(\lambda)$

**2** Prover constructs the following proofs:

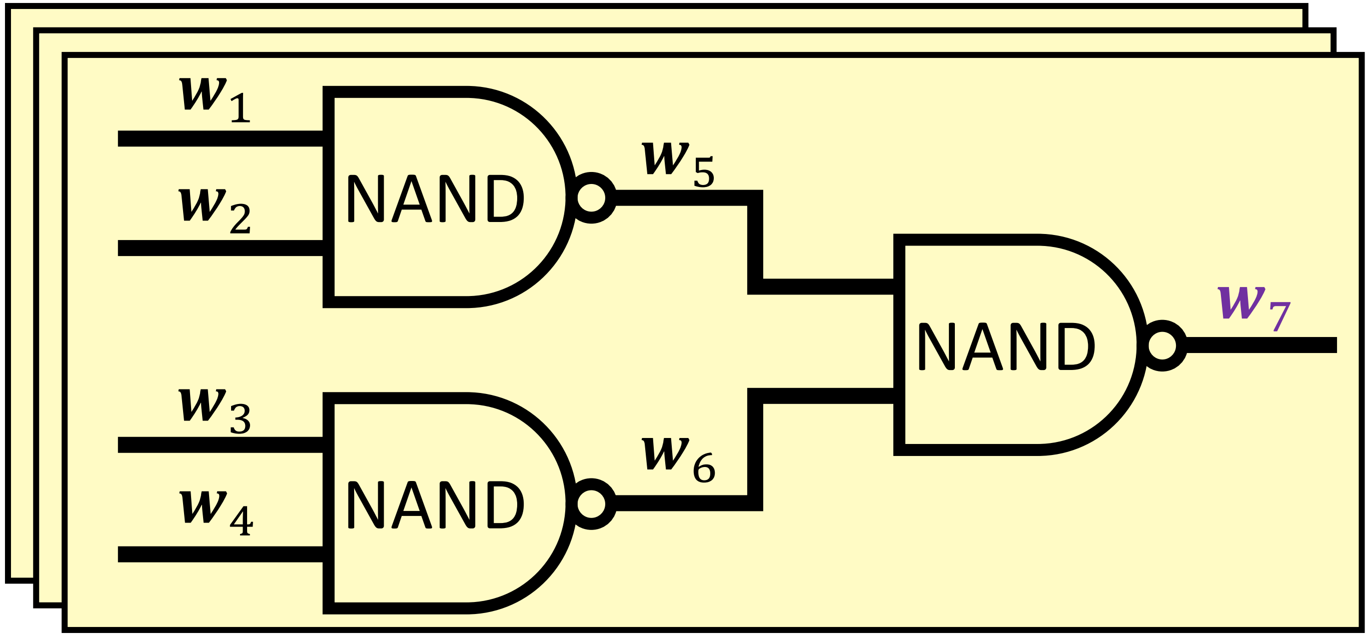
**Input validity**

**Wire validity**

**Gate validity**

For each gate, commitment to output wires is consistent with gate operation and commitment to input wires

# A Commit-and-Prove Strategy for Batch Arguments




Let $\boldsymbol{w}_i = (w_{i,1}, \dots, w_{i,m})$ be vector of wire labels associated with wire $i$ across the $m$ instances

**1** Prover commits to each vector of wire assignments


**Requirement:** $|\sigma_i| = \text{poly}(\lambda, \log m)$

**Our construction:** $|\sigma_i| = \text{poly}(\lambda)$

**2** Prover constructs the following proofs:

**Input validity**

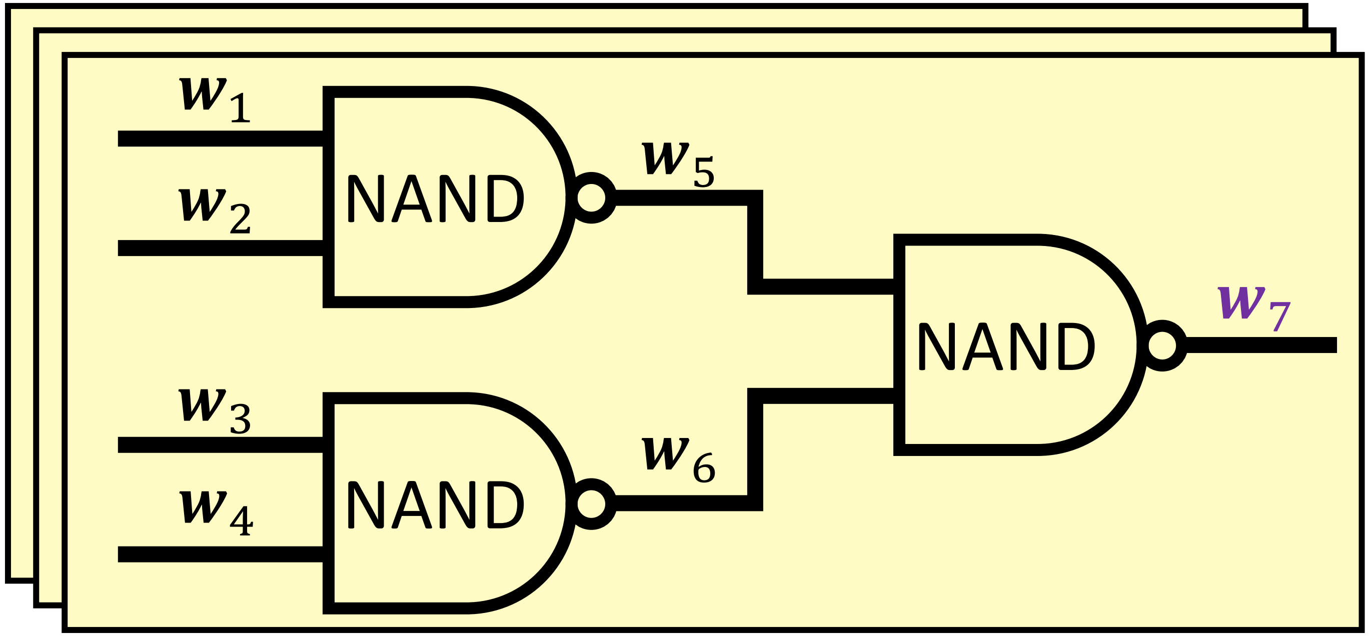
**Wire validity**

**Gate validity**

**Output validity**

Commitment to output wire is a commitment to the all-ones vector

# A Commit-and-Prove Strategy for Batch Arguments




Let $\boldsymbol{w}_i = (w_{i,1}, \dots, w_{i,m})$ be vector of wire labels associated with wire $i$ across the $m$ instances

**1** Prover commits to each vector of wire assignments


**Requirement:** $|\sigma_i| = \text{poly}(\lambda, \log m)$

**Our construction:** $|\sigma_i| = \text{poly}(\lambda)$

**2** Prover constructs the following proofs:

- **Input validity**
- **Wire validity**
- **Gate validity**
- **Output validity**

**Key idea:** Validity checks are quadratic and can be checked in the exponent

# Construction from Composite-Order Groups

Pedersen multi-commitments: (*without* randomness)

Let $\mathbb{G}$ be a group of order $N = pq$ (composite order)

Let $\mathbb{G}_p \subset \mathbb{G}$ be the subgroup of order $p$ and let $g_p$ be a generator of $\mathbb{G}_p$

crs: sample $\alpha_1, \dots, \alpha_m \leftarrow \mathbb{Z}_N$
output $A_1 \leftarrow g_p^{\alpha_1}, \dots, A_m \leftarrow g_p^{\alpha_m}$

denotes encodings in $\mathbb{G}_p$

$[\alpha_1]$ $[\alpha_2]$ $[\cdots]$ $[\alpha_m]$

commitment to $\boldsymbol{x} = (x_1, \dots, x_m) \in \{0,1\}^m$:

$$\sigma_{\boldsymbol{x}} = A_1^{x_1} A_2^{x_2} \cdots A_m^{x_m}$$

(subset product of the $A_i$'s)

$$[\sigma_{\boldsymbol{x}}] = [\Sigma_{i \in [m]} \alpha_i x_i]$$

# Proving Relations on Committed Values

common reference string

$[\alpha_1]$ $A_1 = g_p^{\alpha_1}$

$[\vdots]$

$[\alpha_m]$ $A_m = g_p^{\alpha_m}$

commitment to $(x_1, \ldots, x_m)$

$[\Sigma_{i\in[m]}\alpha_i x_i]$

$$\sigma_x = A_1^{x_1} A_2^{x_2} \cdots A_m^{x_m} = g_p^{\alpha_1 x_1 + \cdots + \alpha_m x_m}$$

**Wire validity**

Commitment for each wire is a commitment to a 0/1 vector

$x \in \{0,1\}$ if and only if $x^2 = x$

**Key idea:** Use pairing to check quadratic relation in the exponent

**Recall:** pairing is an efficiently-computable bilinear map on $\mathbb{G}$:

$$e(g^x, g^y) = e(g,g)^{xy}$$

$$e([x], [y]) \longrightarrow [xy]$$

*Multiplies exponents in the target group*

# Proving Relations on Committed Values

common reference string

$[\alpha_1]$ $A_1 = g_p^{\alpha_1}$

$[\vdots]$

$[\alpha_m]$ $A_m = g_p^{\alpha_m}$

commitment to $(x_1, \dots, x_m)$

$[\Sigma_{i \in [m]} \alpha_i x_i]$

$$\sigma_x = A_1^{x_1} A_2^{x_2} \cdots A_m^{x_m}$$
$$= g_p^{\alpha_1 x_1 + \cdots + \alpha_m x_m}$$

**Wire validity**

Commitment for each wire is a commitment to a 0/1 vector

$x \in \{0,1\}$ if and only if $x^2 = x$

**Approach:** consider the following pairing relations:

$e(\sigma_x, \sigma_x)$ and $e\left(\sigma_x, \Pi_{i \in [m]} A_i\right)$

$$A = \Pi_{i \in [m]} A_i = g_p^{\Sigma_{i \in [m]} \alpha_i}$$

*(commitment to all-ones vector)*

# Proving Relations on Committed Values

common reference string

$[\alpha_1]$ $A_1 = g_p^{\alpha_1}$

$[\vdots]$

$[\alpha_m]$ $A_m = g_p^{\alpha_m}$

commitment to $(x_1, \dots, x_m)$

$[\Sigma_{i \in [m]} \alpha_i x_i]$

$$\sigma_x = A_1^{x_1} A_2^{x_2} \cdots A_m^{x_m}$$
$$= g_p^{\alpha_1 x_1 + \cdots + \alpha_m x_m}$$

**Wire validity**

Commitment for each wire is a commitment to a 0/1 vector

$x \in \{0,1\}$ if and only if $x^2 = x$

**Approach:** consider the following pairing relations:

$e(\sigma_x, \sigma_x)$ and $e\left(\sigma_x, \Pi_{i \in [m]} A_i\right)$

$$e\left([\Sigma_{i \in [m]} \alpha_i x_i], [\Sigma_{i \in [m]} \alpha_i x_i]\right)$$

$$= [\Sigma_{i \in [m]} \alpha_i^2 x_i^2] \times [\Sigma_{i \neq j} \alpha_i \alpha_j x_i x_j]$$

*non-cross terms* *cross terms*

# Proving Relations on Committed Values

common reference string

$[\alpha_1]$ $A_1 = g_p^{\alpha_1}$

$[\vdots]$

$[\alpha_m]$ $A_m = g_p^{\alpha_m}$

**Wire validity**

Commitment for each wire is a commitment to a 0/1 vector

$x \in \{0,1\}$ if and only if $x^2 = x$

**Approach:** consider the following pairing relations:

$e(\sigma_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}})$ and $e\left(\sigma_{\boldsymbol{x}}, \Pi_{i\in[m]} A_i\right)$

$$e\left([\Sigma_{i\in[m]}\alpha_i x_i], [\Sigma_{i\in[m]}\alpha_i]\right) = [\Sigma_{i\in[m]}\alpha_i^2 x_i] \times [\Sigma_{i\neq j}\alpha_i\alpha_j x_i]$$

*non-cross terms* *cross terms*

$$e\left([\Sigma_{i\in[m]}\alpha_i x_i], [\Sigma_{i\in[m]}\alpha_i x_i]\right) = [\Sigma_{i\in[m]}\alpha_i^2 x_i^2] \times [\Sigma_{i\neq j}\alpha_i\alpha_j x_i x_j]$$

*non-cross terms* *cross terms*

# Proving Relations on Committed Values

If $x_i^2 = x_i$ for all $i$, then

$$[\Sigma_{i\in[m]}\alpha_i^2 x_i] = [\Sigma_{i\in[m]}\alpha_i^2 x_i^2]$$

**Wire validity**

Commitment for each wire is a commitment to a 0/1 vector

$x \in \{0,1\}$ if and only if $x^2 = x$

**Approach:** consider the following pairing relations:

$$e(\sigma_x, \sigma_x) \text{ and } e\left(\sigma_x, \Pi_{i\in[m]} A_i\right)$$

$$e\left([\Sigma_{i\in[m]}\alpha_i x_i], [\Sigma_{i\in[m]}\alpha_i]\right) = [\Sigma_{i\in[m]}\alpha_i^2 x_i] \times [\Sigma_{i\neq j}\alpha_i\alpha_j x_i]$$

*non-cross terms* — *cross terms*

$$e\left([\Sigma_{i\in[m]}\alpha_i x_i], [\Sigma_{i\in[m]}\alpha_i x_i]\right) = [\Sigma_{i\in[m]}\alpha_i^2 x_i^2] \times [\Sigma_{i\neq j}\alpha_i\alpha_j x_i x_j]$$

*non-cross terms* — *cross terms*

# Proving Relations on Committed Values

If $x_i^2 = x_i$ for all $i$, then

$$\left[\Sigma_{i\in[m]}\alpha_i^2 x_i\right] = \left[\Sigma_{i\in[m]}\alpha_i^2 x_i^2\right]$$

**Wire validity**

Commitment for each wire is a commitment to a 0/1 vector

$x \in \{0,1\}$ if and only if $x^2 = x$

**Approach:** consider the following pairing relations:

$e(\sigma_x, \sigma_x)$ and $e\left(\sigma_x, \Pi_{i\in[m]} A_i\right)$

$$e\left(\left[\Sigma_{i\in[m]}\alpha_i x_i\right], \left[\Sigma_{i\in[m]}\alpha_i\right]\right) \qquad e\left(\left[\Sigma_{i\in[m]}\alpha_i x_i\right], \left[\Sigma_{i\in[m]}\alpha_i x_i\right]\right)$$

When $x_i^2 = x_i$, difference between these terms is

$$\left[\Sigma_{i\neq j}\alpha_i\alpha_j\left(x_i - x_i x_j\right)\right]$$

Give prover ability to <u>eliminate</u> cross-terms *only* →

Augment CRS with cross-terms

$$\left[\alpha_i\alpha_j\right] \quad B_{i,j} = g_p^{\alpha_i\alpha_j} \quad \forall i \neq j$$

# Proving Relations on Committed Values

Prover now computes additional group component in the *base* group

$[\Sigma_{i \neq j} \alpha_i \alpha_j (x_i - x_i x_j)]$ —Pair with $g_p$→ $[\Sigma_{i \neq j} \alpha_i \alpha_j (x_i - x_i x_j)]$

$V = B_{i,j}^{x_i - x_i x_j}$  $e(g_p, V)$

$$e\left([\Sigma_{i \in [m]} \alpha_i x_i], [\Sigma_{i \in [m]} \alpha_i]\right) \qquad e\left([\Sigma_{i \in [m]} \alpha_i x_i], [\Sigma_{i \in [m]} \alpha_i x_i]\right)$$

When $x_i^2 = x_i$, difference between these terms is

$[\Sigma_{i \neq j} \alpha_i \alpha_j (x_i - x_i x_j)]$ —Give prover ability to eliminate cross-terms *only*→

Augment CRS with cross-terms

$[\alpha_i \alpha_j]$ $B_{i,j} = g_p^{\alpha_i \alpha_j} \quad \forall i \neq j$

# Proving Relations on Committed Values

Prover now computes additional group component in the *base* group

$[\Sigma_{i\neq j}\alpha_i\alpha_j(x_i - x_i x_j)]$ —Pair with $g_p$→ $[\Sigma_{i\neq j}\alpha_i\alpha_j(x_i - x_i x_j)]$

$V = B_{i,j}^{x_i - x_i x_j}$ $\qquad$ $e(g_p, V)$

**Overall verification relation:** $e(\sigma_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}}) = e(\sigma_{\boldsymbol{x}}, A)e(g_p, V)$ $\qquad$ $A = \Pi_{i\in[m]} A_i$

# Proving Relations on Committed Values

Prover now computes additional group component in the *base* group

$[\Sigma_{i \neq j} \alpha_i \alpha_j (x_i - x_i x_j)]$ —Pair with $g_p$→ $[\Sigma_{i \neq j} \alpha_i \alpha_j (x_i - x_i x_j)]$

$V = B_{i,j}^{x_i - x_i x_j}$ $e(g_p, V)$

**Overall verification relation:** $e(\sigma_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}}) = e(\sigma_{\boldsymbol{x}}, A) e(g_p, V)$ $A = \Pi_{i \in [m]} A_i$

Non-cross terms ensure that $x_i^2 = x_i$

# Proving Relations on Committed Values

Prover now computes additional group component in the *base* group

$[\Sigma_{i \neq j} \alpha_i \alpha_j (x_i - x_i x_j)]$ — Pair with $g_p$ → $[\Sigma_{i \neq j} \alpha_i \alpha_j (x_i - x_i x_j)]$

$V = B_{i,j}^{x_i - x_i x_j}$ $\quad\quad$ $e(g_p, V)$

**Overall verification relation:** $e(\sigma_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}}) = e(\sigma_{\boldsymbol{x}}, A) e(g_p, V)$ $\quad\quad$ $A = \Pi_{i \in [m]} A_i$

Non-cross terms ensure that $x_i^2 = x_i$

Correction factor to correct for cross terms

# Proving Relations on Committed Values

**Common reference string:**

$[\alpha_1]$ $[\cdots]$ $[\alpha_m]$

$A_1 = g_p^{\alpha_1}$ $A_m = g_p^{\alpha_m}$

$[\alpha_1 + \cdots \alpha_m]$ $A = \Pi_{i \in [m]} A_i$

$[\alpha_i \alpha_j]$ $B_{i,j} = g_p^{\alpha_i \alpha_j} \; \forall i \neq j$

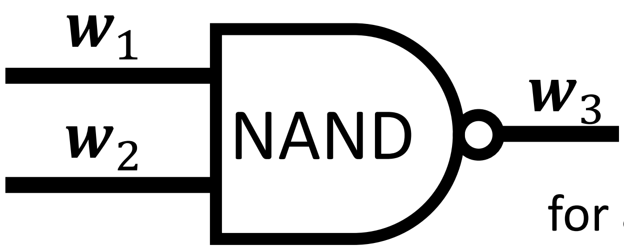
**Commitment to $(x_1, \dots, x_m)$:**

$[\Sigma_{i \in [m]} \alpha_i x_i]$

$$\sigma_x = A_1^{x_1} A_2^{x_2} \cdots A_m^{x_m} = g_p^{\alpha_1 x_1 + \cdots + \alpha_m x_m}$$

**Gate validity**

For each gate, commitment to output wires is consistent with gate operation and commitment to input wires

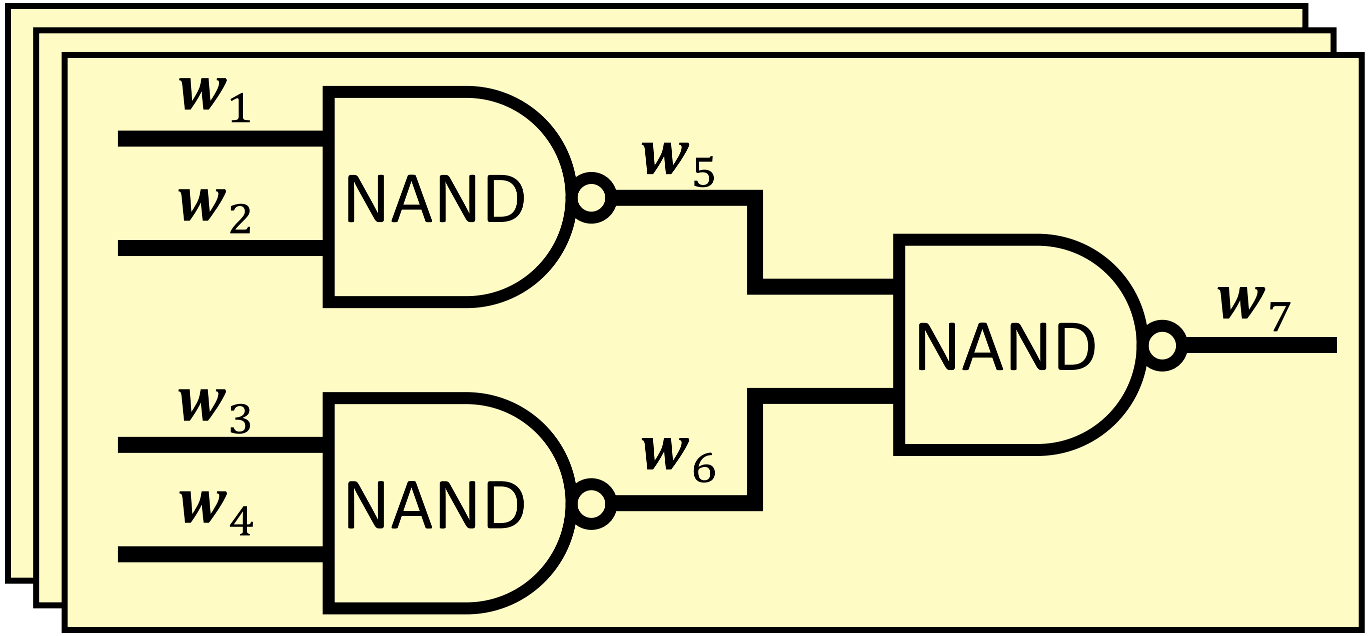



for all $i \in [m]$: $w_{3,i} = 1 - w_{1,i} w_{2,i}$

Relation is <u>quadratic</u> in the inputs

Can leverage similar approach as before

# Proof Size




Let $\boldsymbol{w}_i = (w_{i,1}, \ldots, w_{i,m})$ be vector of wire labels associated with wire $i$

1 Prover commits to each vector of wire assignments

$\boldsymbol{w}_i =$ $w_{i,1}$ $w_{i,2}$ $\cdots$ $w_{i,m}$ → $\sigma_i$

**Commitment size:** $|\sigma_i| = \text{poly}(\lambda)$

Single group element

2 Prover constructs the following proofs:

| | |
|---|---|
| **Input validity** | |
| **Wire validity** | One group element |
| **Gate validity** | One group element |
| **Output validity** | |

**Overall proof size ($t$ wires, $s$ gates):**

$(2t + s) \cdot \text{poly}(\lambda) = |C| \cdot \text{poly}(\lambda)$

# Is This Sound?

**Common reference string:**

$[\alpha_1]$ $[\cdots]$ $[\alpha_m]$

$A_1 = g_p^{\alpha_1}$ $A_m = g_p^{\alpha_m}$

$[\alpha_1 + \cdots \alpha_m]$ $A = \Pi_{i \in [m]} A_i$

$[\alpha_i \alpha_j]$ $B_{i,j} = g_p^{\alpha_i \alpha_j} \; \forall i \neq j$

**Commitment to $(x_1, \dots, x_m)$:**

$[\Sigma_{i \in [m]} \alpha_i x_i]$

$$\sigma_x = A_1^{x_1} A_2^{x_2} \cdots A_m^{x_m}$$
$$= g_p^{\alpha_1 x_1 + \cdots + \alpha_m x_m}$$

Soundness requires some care:

Groth-Ostrovsky-Sahai NIZK based on similar commit-and-prove strategy

Soundness in GOS is possible by ***extracting*** a witness from the commitment

For a false statement, no witness exists

**Our setting:** commitments are *succinct* – cannot extract a full witness

**Solution:** "local extractability" [KPY19] or "somewhere extractability" [CJJ21]

# Somewhere Soundness

CRS will have two modes:

**Normal mode:** used in the real scheme

**Extracting on index $\boldsymbol{i}$:** supports witness extraction for instance $i$ (given a trapdoor)

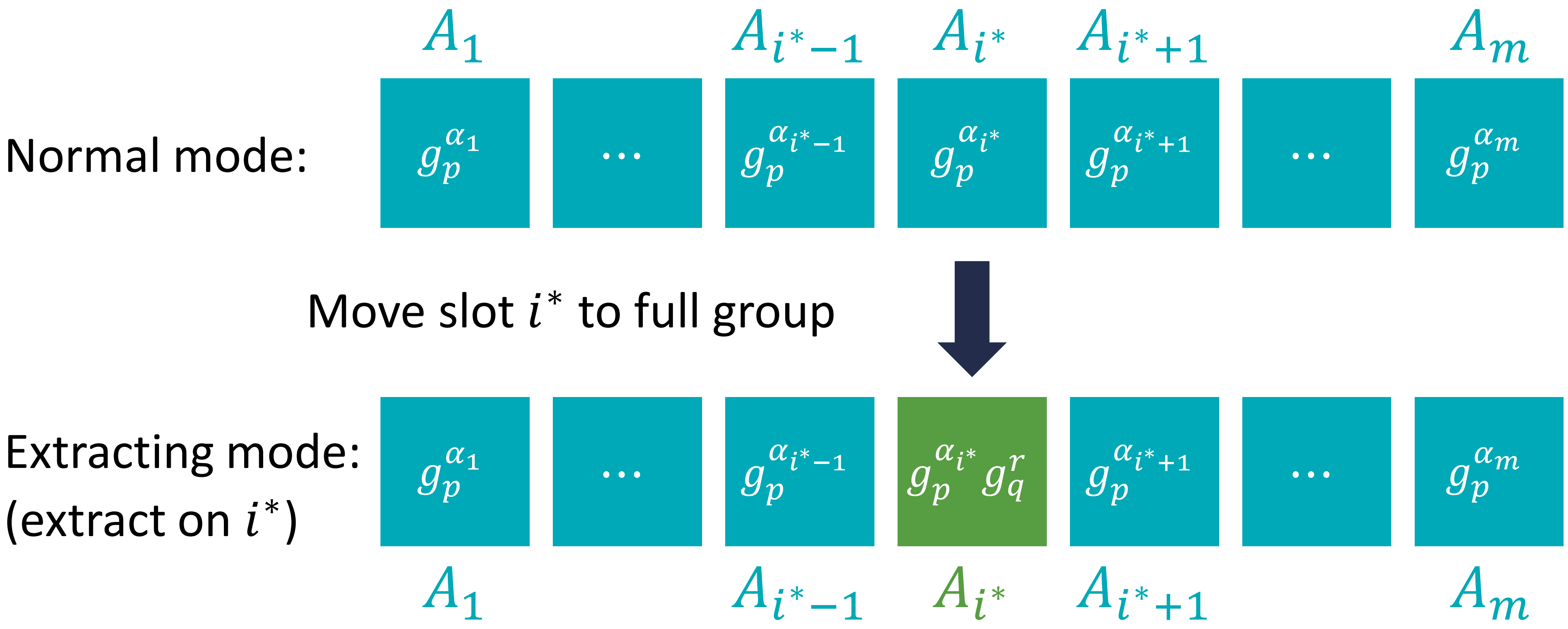
If proof $\pi$ verifies, then we can extract a witness $w_i$ such that $C(x_i, w_i) = 1$

CRS in the two modes are computationally indistinguishable

Similar to "dual-mode" proof systems and somewhere statistically binding hash functions

Implies non-adaptive soundness

# Local Extraction




**Subgroup decision assumption [BGN05]:**

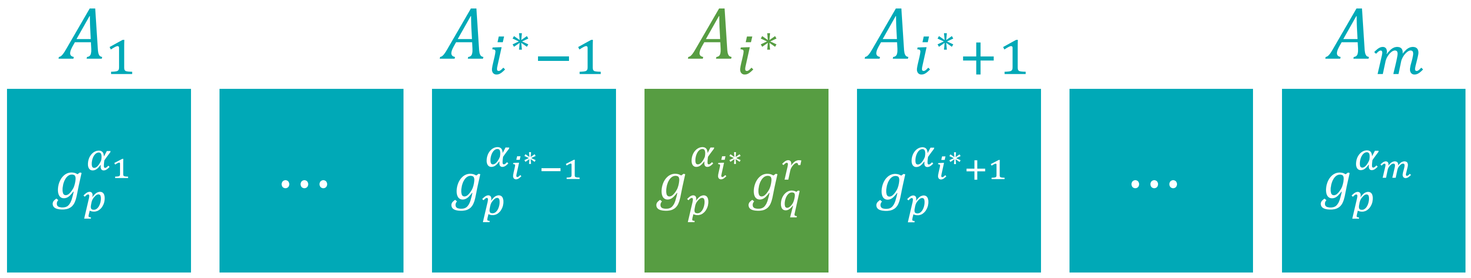
Random element in subgroup ($\mathbb{G}_p$)

$\approx$

Random element in full group ($\mathbb{G}$)

# Local Extraction

CRS in extraction mode (for index $i^*$):




**Trapdoor:** $g_q$ (generator of $\mathbb{G}_q$)

Can extract by projecting into $\mathbb{G}_q$

- Extracted bit for a commitment $\boldsymbol{\sigma}$ is 1 if $\boldsymbol{\sigma}$ has a (non-zero) component in $\mathbb{G}_q$

# Correctness of Extraction

Consider wire validity check:

$$e(\sigma_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}}) = e(\sigma_{\boldsymbol{x}}, A)e(g_p, V)$$

# Correctness of Extraction

Consider wire validity check:

$$e(\sigma_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}}) = e(\sigma_{\boldsymbol{x}}, A)e(g_p, V)$$

Adversary chooses commitment $\sigma_{\boldsymbol{x}}$ and proof $V$

# Correctness of Extraction

Consider wire validity check:

$$e(\sigma_x, \sigma_x) = e(\sigma_x, A)e(g_p, V)$$

Adversary chooses commitment $\sigma_x$ and proof $V$

Generator $g_p$ and aggregated component $A$ part of the CRS (honestly-generated)

If this relation holds, it must hold in **both**
the order-$p$ subgroup **and** the order-$q$ subgroup of $\mathbb{G}_T$

**Key property:** $e(g_p, V)$ is **always** in the order-$p$ subgroup; adversary **cannot** influence the verification relation in the order-$q$ subgroup

Write $\sigma_x = g_p^s g_q^t$

Write $A = g_p^{\sum_{i \in [m]} \alpha_i} g_q^r$

In the order-$q$ subgroup, exponents must satisfy:

$$t^2 = tr \bmod q$$

# Correctness of Extraction

Consider wire validity check:

$$e(\sigma_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}}) = e(\sigma_{\boldsymbol{x}}, A) e(g_p, V)$$

Adversary chooses commitment $\sigma_{\boldsymbol{x}}$ and proof $V$

Generator $g_p$ and aggregated component $A$ part of the CRS (honestly-generated)

If this relation holds, it must hold in **both**
the order-$p$ subgroup **and** the order-$q$ subgroup of $\mathbb{G}$

**Key property:** $e(g_p, V)$ is **alwa**
verification relation in the ord

If wire validity checks pass, then $t = b_i r$ where $b_i \in \{0,1\}$

**Observe:** $b_i \in \{0,1\}$ is also the extracted bit

Write $\sigma_{\boldsymbol{x}} = g_p^s g_q^t$

Write $A = g_p^{\sum_{i \in [m]} \alpha_i} g_q^r$

In the order-$q$ subgroup, exponents must satisfy:

$$t^2 = tr \bmod q$$

# Correctness of Extraction

Consider gate validity check:

$$e(\sigma_{w_3}, A)e(\sigma_{w_1}, \sigma_{w_2}) = e(A, A)e(g_p, W)$$

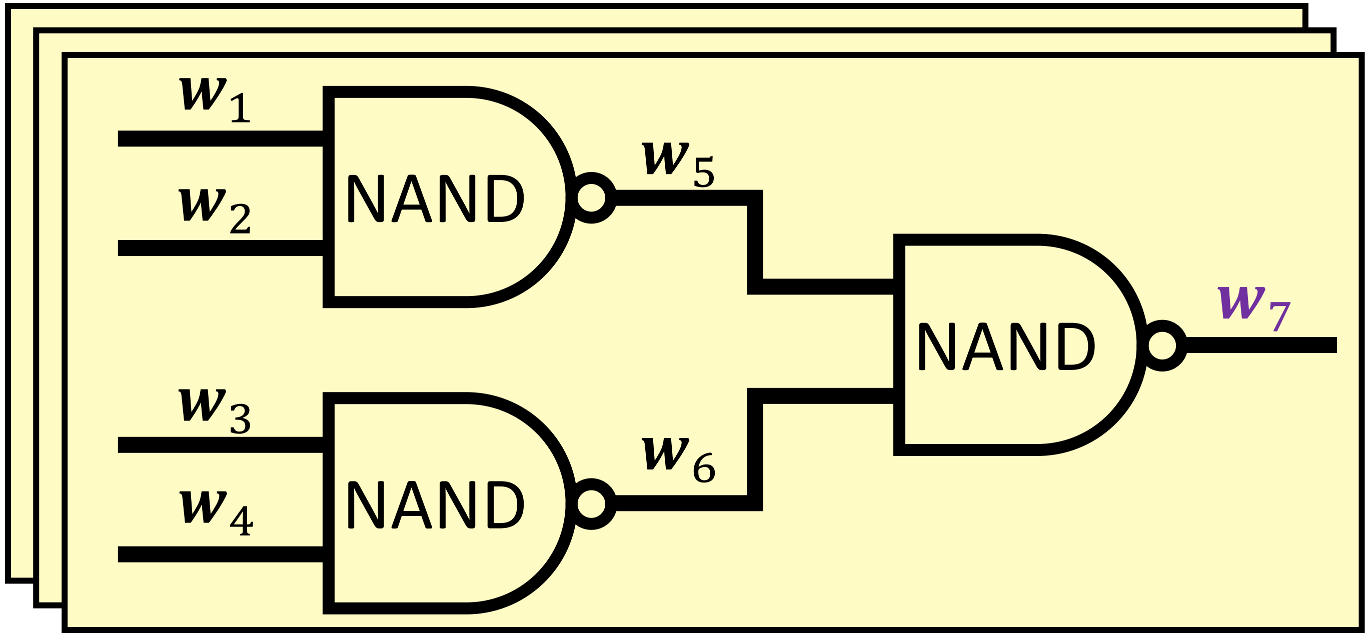
Adversary chooses commitment $\sigma_{w_1}, \sigma_{w_2}, \sigma_{w_3}$ and proof $W$

Generator $g_p$ and aggregated component $A$ part of the CRS (honestly-generated)

Similar analysis shows that extracted bits satisfy $b_3 = 1 - b_1 b_2 = \mathrm{NAND}(b_1, b_2)$

[See paper for details]

# A Commit-and-Prove Strategy for Batch Arguments




1 Prover commits to each vector of wire assignments


**Requirement:** $|\sigma_i| = \text{poly}(\lambda, \log m)$

**Our construction:** $|\sigma_i| = \text{poly}(\lambda)$

Let $\boldsymbol{w}_i = (w_{i,1}, \dots, w_{i,m})$ be vector of wire labels associated with wire $i$ across the $m$ instances

2 Prover constructs the following proofs:

**Input validity**

**Wire validity**

**Gate validity**

**Output validity**

**Key idea:** Validity checks are quadratic and can be checked in the exponent

# From Composite-Order to Prime-Order

Batch argument for NP from standard assumptions over bilinear maps

Subgroup decision assumption in composite-order bilinear groups

$$\mathbb{G} \cong \mathbb{G}_p \times \mathbb{G}_q$$

composite-order group

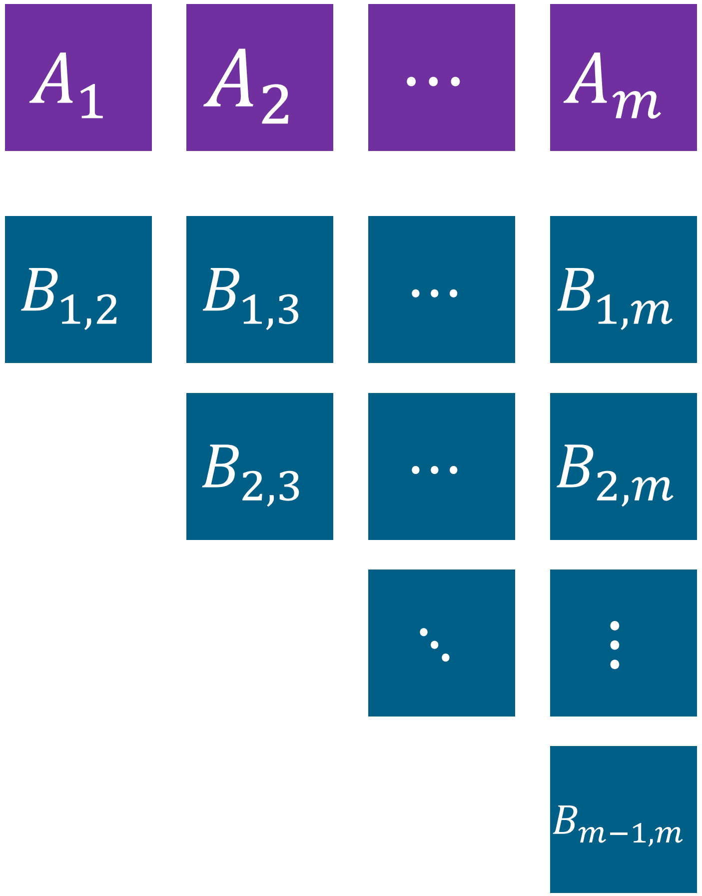
Simulate subgroups
with subspaces

**Conclusion:**

$k$-Linear assumption (for any $k \geq 1$) in prime-order asymmetric bilinear groups

# Reducing CRS Size

Common reference string:




Size of CRS is $m^2 \cdot \text{poly}(\lambda)$

Can rely on **recursive composition** to reduce CRS size:

$$m^2 \cdot \text{poly}(\lambda) \to m^\varepsilon \cdot \text{poly}(\lambda)$$

for any constant $\varepsilon > 0$

Similar approach as [KPY19]

# Application to RAM Delegation ("SNARGs for P")

Choudhuri et al. [CJJ21] showed:

*succinct argument for polynomial-time computations*

Batch argument for NP* + Somewhere extractable commitment → Delegation scheme for RAM programs

*succinct vector commitment that allows extracting on single index*

*Needs a split verification property [see paper for details]

# Application to RAM Delegation ("SNARGs for P")

Choudhuri et al. [CJJ21] showed:

*succinct argument for polynomial-time computations*

Batch argument for NP* + Somewhere extractable commitment → Delegation scheme for RAM programs

*This work (from $k$-Lin)*

*succinct vector commitment that allows extracting on single index*

*Needs a split verification property [see paper for details]

# Application to RAM Delegation ("SNARGs for P")

Choudhuri et al. [CJJ21] showed:

| | | | | |
|---|---|---|---|---|
| Batch argument for NP* | + | Somewhere extractable commitment | → | Delegation scheme for RAM programs |
| *This work (from $k$-Lin)* | | *This work + [OPWW15] (from SXDH)* | | |

*Needs a split verification property [see paper for details]

# Application to RAM Delegation ("SNARGs for P")

Choudhuri et al. [CJJ21] showed:

**Batch argument for NP*** + **Somewhere extractable commitment** → **Delegation scheme for RAM programs**

*This work (from $k$-Lin)*

*This work + [OPWW15] (from SXDH)*

**Corollary.** RAM delegation from SXDH on prime-order pairing groups

To verify a time-$T$ RAM computation:

- **CRS size:** $|\mathrm{crs}| = T^{\varepsilon} \cdot \mathrm{poly}(\lambda)$ for any constant $\varepsilon > 0$
- **Proof size:** $|\pi| = \mathrm{poly}(\lambda, \log T)$
- **Verification time:** $|\mathrm{Verify}| = \mathrm{poly}(\lambda, \log T)$

**Previous pairing constructions:** non-standard assumptions [KPY19] or quadratic CRS [GZ21]

# Summary

Batch arguments for NP from standard assumptions over bilinear maps

**Key feature:** Construction is "low-tech"

Direct "commit-and-prove" approach like classic pairing-based proof systems

**Corollary:** RAM delegation (i.e., "SNARG for P") with sublinear CRS

**Corollary:** Aggregate signature with bounded aggregation in the plain model

`https://eprint.iacr.org/2022/336`

**Thank you!**