INSTITUT
DE RECHERCHE
EN INFORMATIQUE
FONDAMENTALE

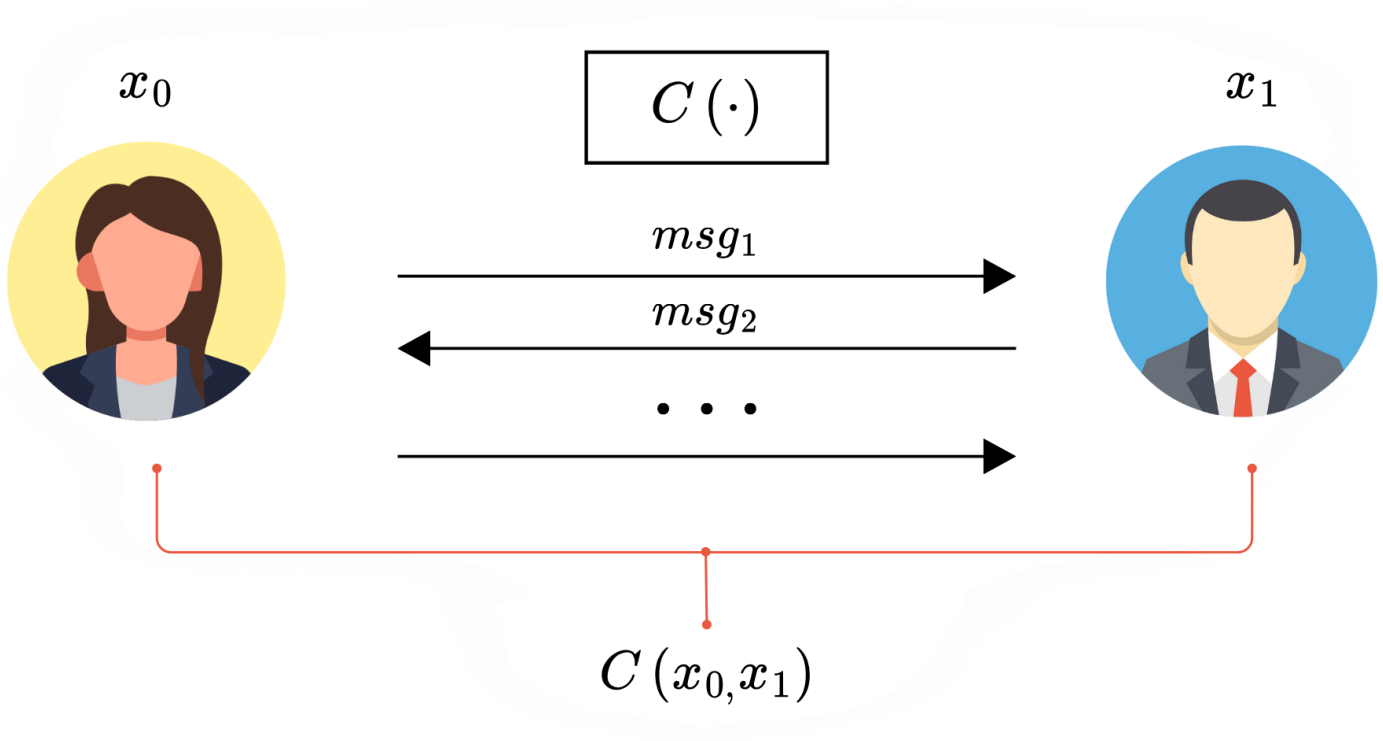
Oregon State
University

10-Party Sublinear Secure Computation from Standard Assumptions

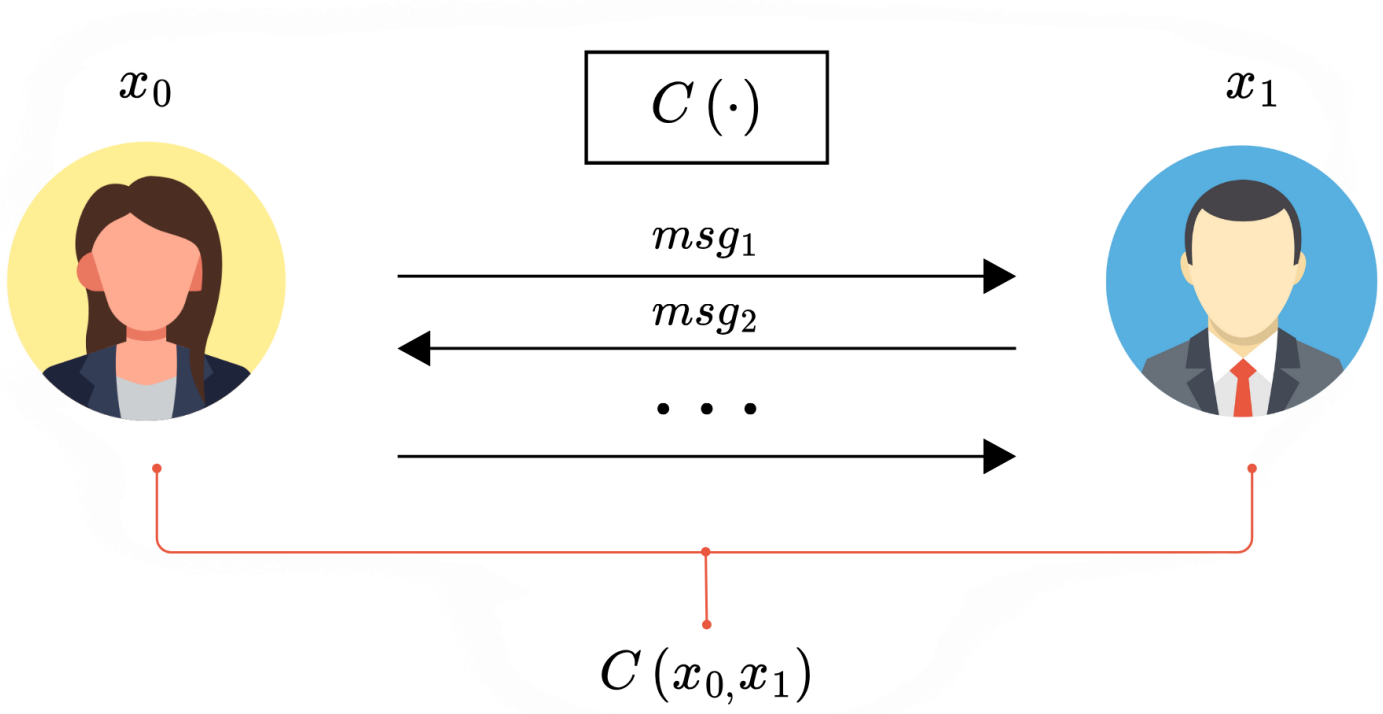
Authors: Geoffroy Couteau, Naman Kumar

Presented by: Naman Kumar

Sublinear Secure Computation - Motivation

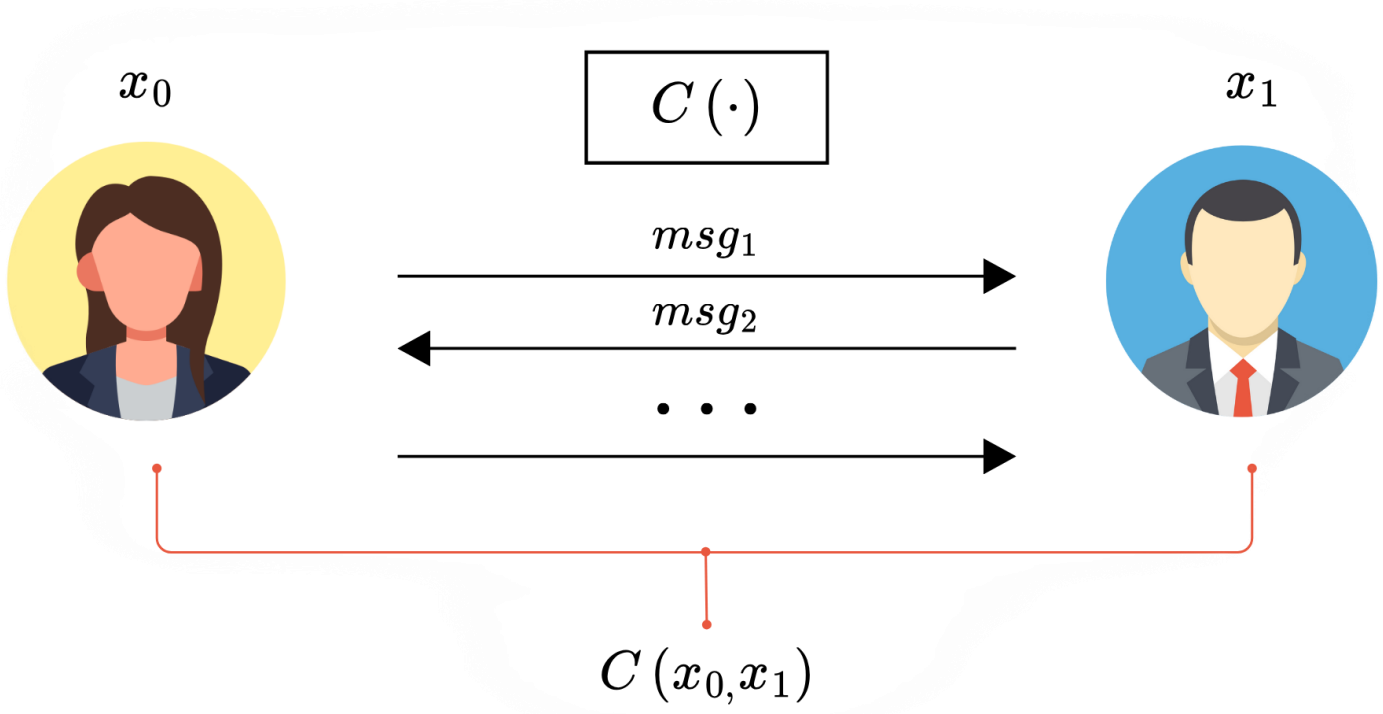


Sublinear Secure Computation - Motivation



Can the total communication be $o(|C|) + O(\lambda) + O(|x_0| + |x_1|)$?

Sublinear Secure Computation - Motivation



Setting:

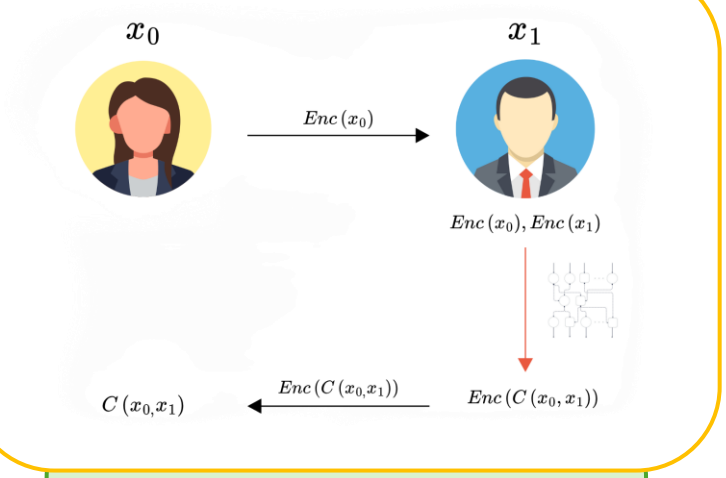
- Semi-Honest
- Dishonest Majority
- Static
- Multi-Party Setting

Circuit class:

- P/Poly (goal)
- Layered circuits

Can the total communication be $o(|C|) + O(\lambda) + O(|x_0| + |x_1|)$?

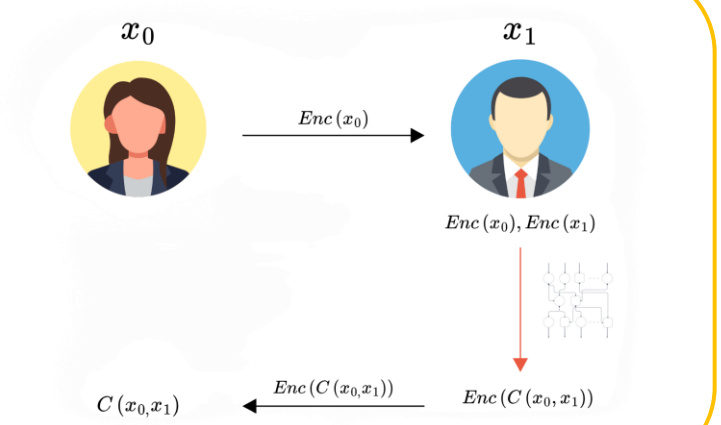
How to achieve Sublinear MPC?



FHE

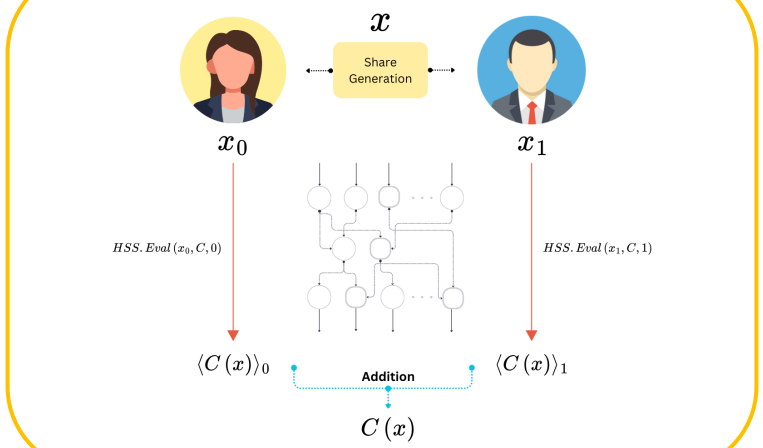
- Works for Lattice Assumptions
- P/Poly
- Comm. Independent

How to achieve Sublinear MPC?



FHE

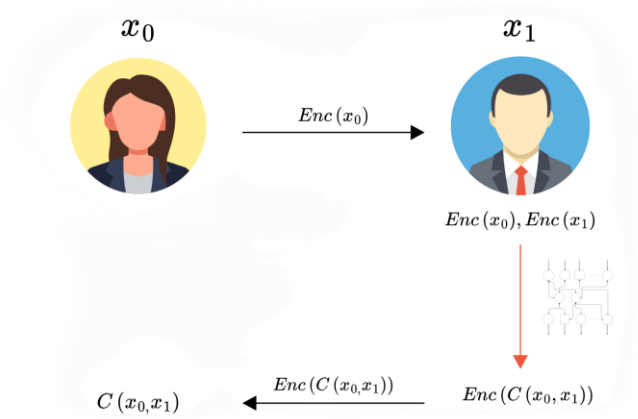
- Works for Lattice Assumptions
- P/Poly
- Comm. Independent



HSS

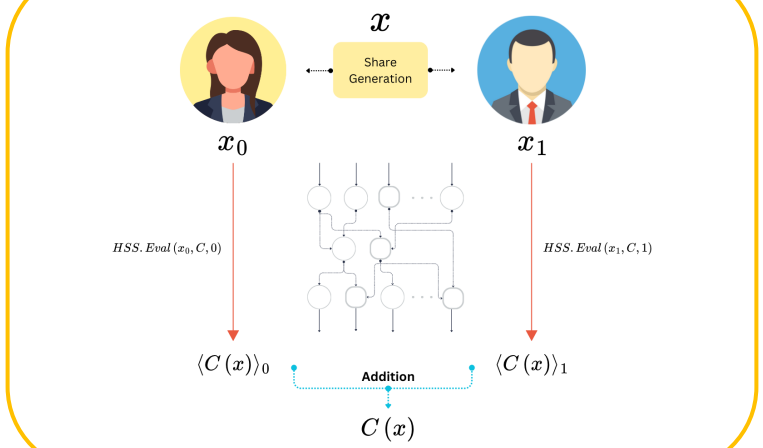
- DDH/DCR/LPN/LWE
- Log or loglog depth, layered
- Comm. O(n/log n) or O(n/loglog n)

How to achieve Sublinear MPC?



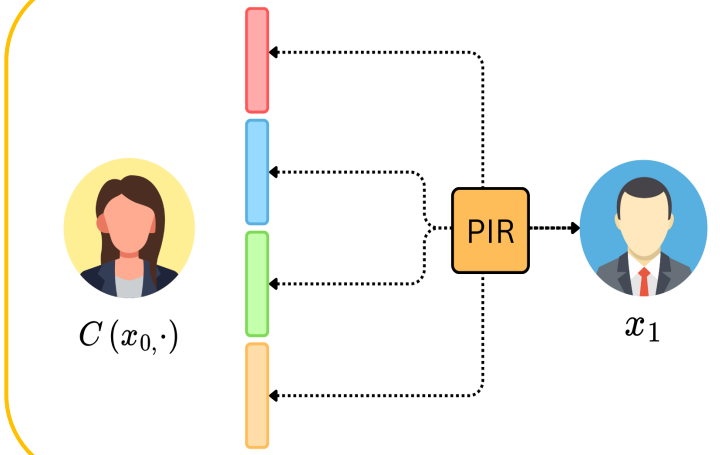
FHE

- Works for Lattice Assumptions
- P/Poly
- Comm. Independent



HSS

- DDH/DCR/LPN/LWE
- Log or loglog depth, layered
- Comm. O(n/log n) or O(n/loglog n)



Correlated SPIR

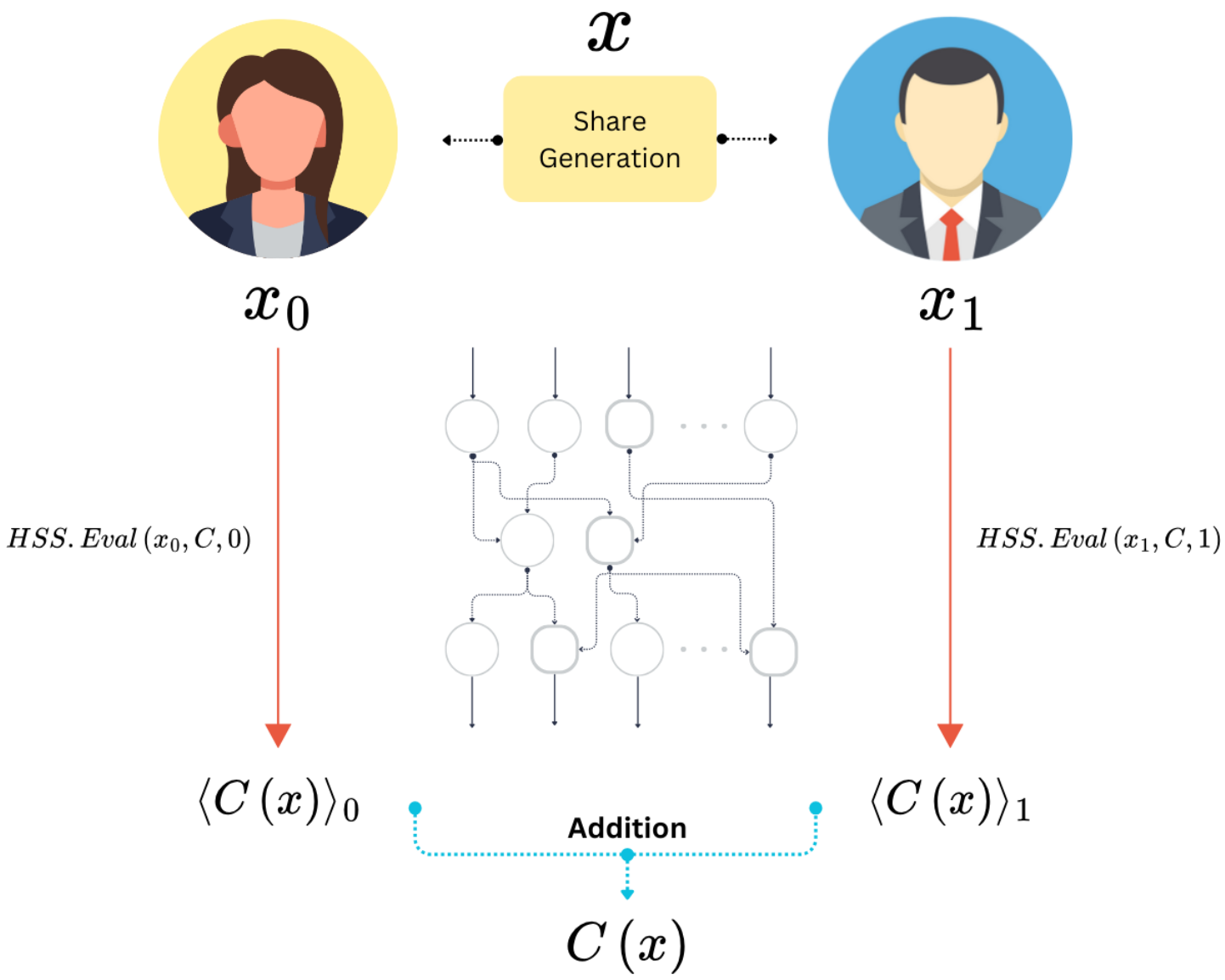
- QR/DDH + LPN
- Loglog depth, layered circuits
- Comm. O(n/loglog n)

Homomorphic Secret Sharing

- Correctness:

$$\langle C(x)\rangle_0 + \langle C(x)\rangle_1 = C(x)$$

- Privacy: x_0 and x_1 hide x.



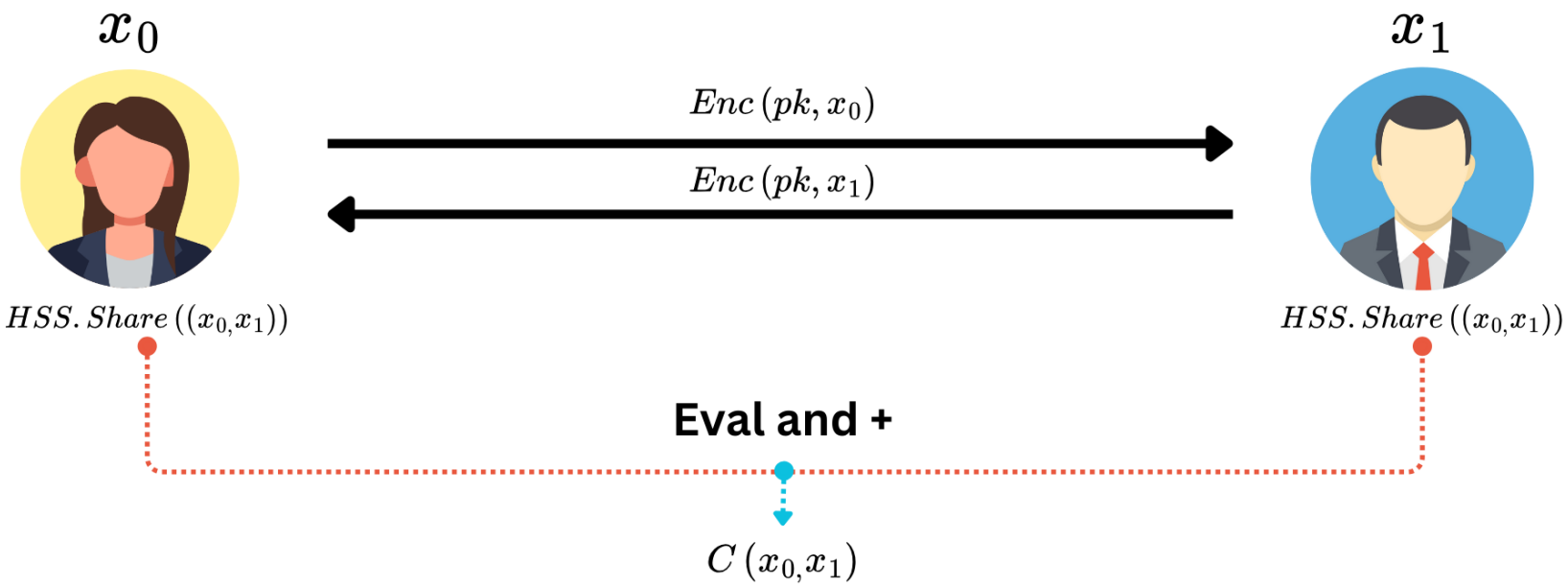
Homomorphic Secret Sharing

	Assumptions	Circuit Class	Error Probability	Message Space
[DHRW15], [BGI15]	LWE (Spooky Encryption)	P/Poly	negligible	exponential
[BGI16]	DDH, DCR	RMS Programs*	$1/poly$	polynomial
[BKS19]	LWE	RMS Programs*	negligible	exponential
[BCGIKS19], [CM21]	(superpoly) LPN	Low-Degree Polynomials**	negligible	exponential
[OSY21], [RS21]	DCR	RMS Programs*	negligible	exponential

*RMS Programs encapsulate branching programs as well as NC^1 (log-depth) circuits.

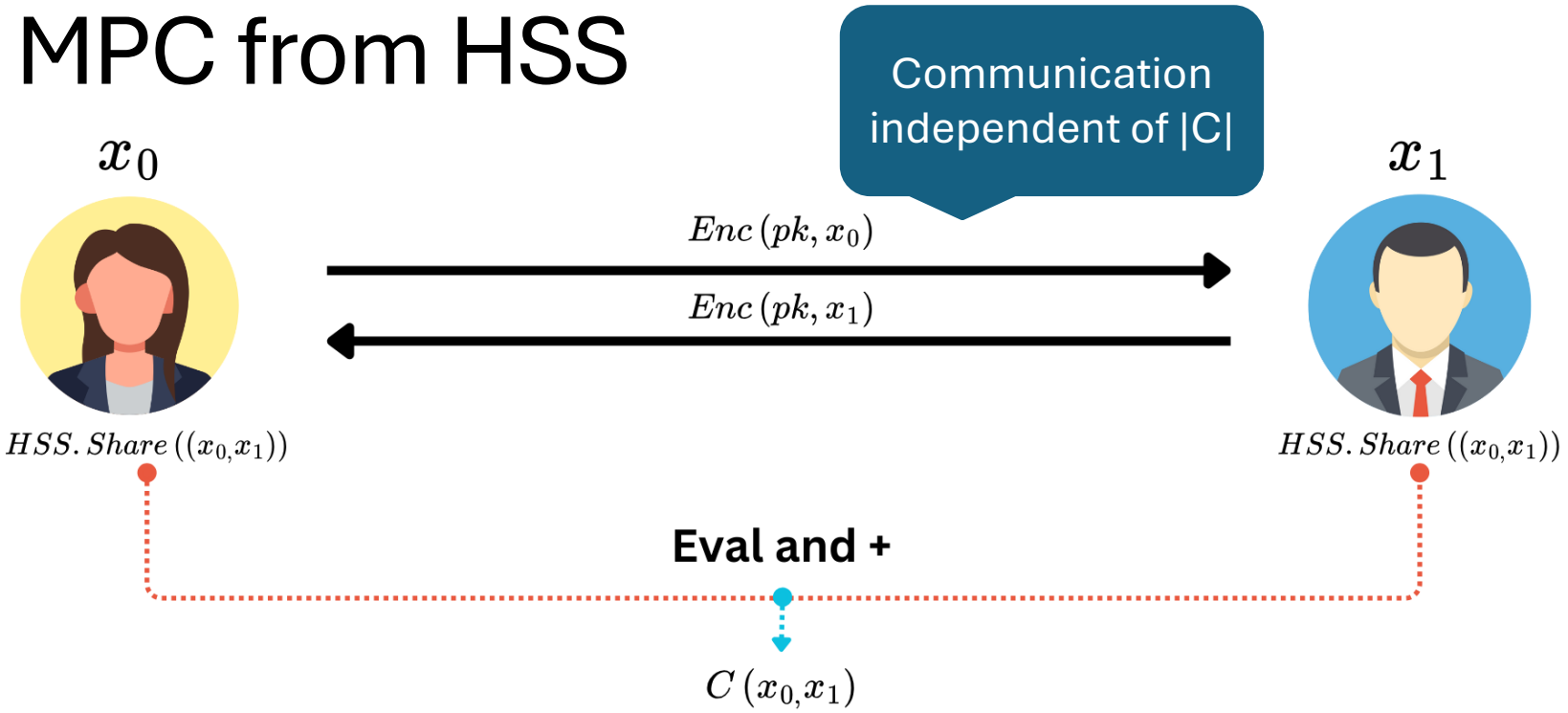
** Via complexity leveraging, it is possible to gain slightly superconstant (loglog-degree) polynomials assuming superpolynomial LPN.

Sublinear MPC from HSS



Sublinear MPC from HSS

HSS for circuit class = Sublinear MPC for circuit class

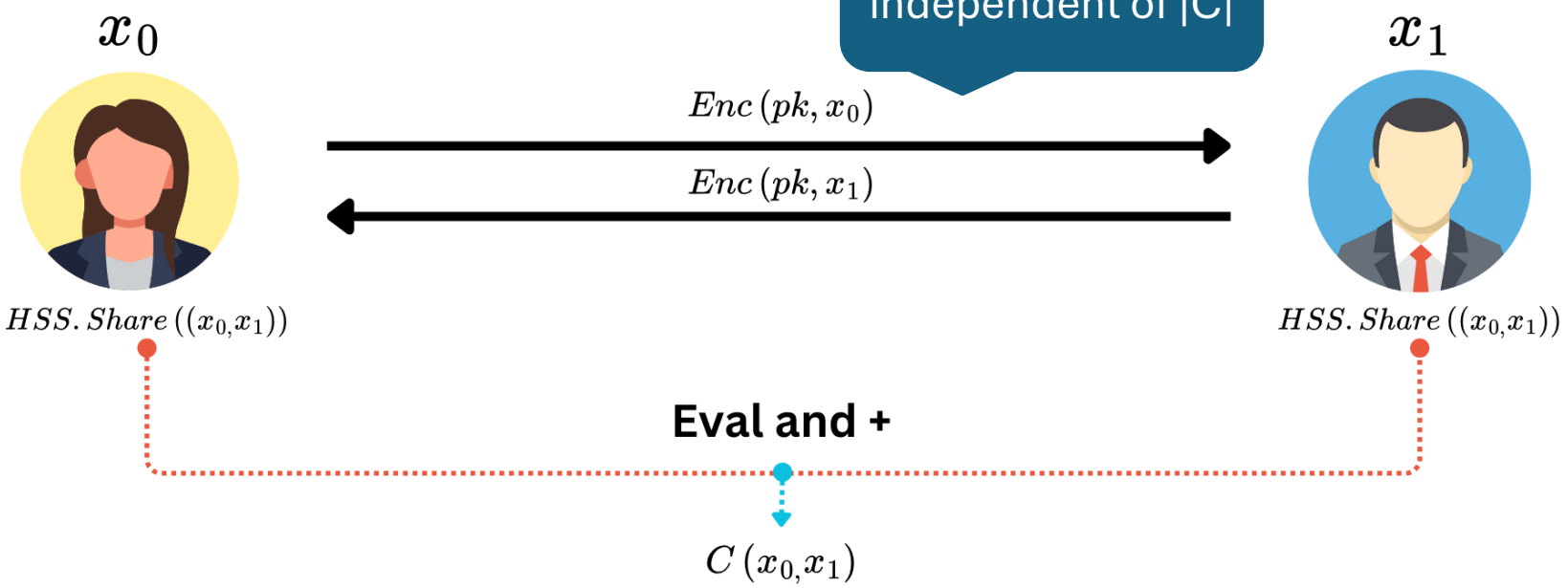


N-party HSS = N-party sublinear MPC

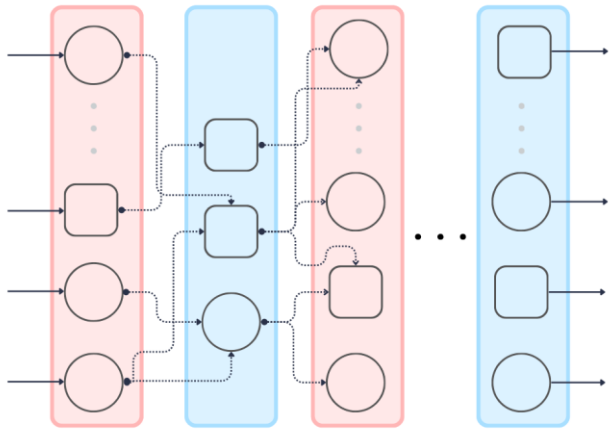
Sublinear MPC from HSS

HSS for circuit class = Sublinear MPC for circuit class

Communication independent of |C|



N-party HSS = N-party sublinear MPC



- Extends to **Layered Circuits**
- Let depth of each layer be

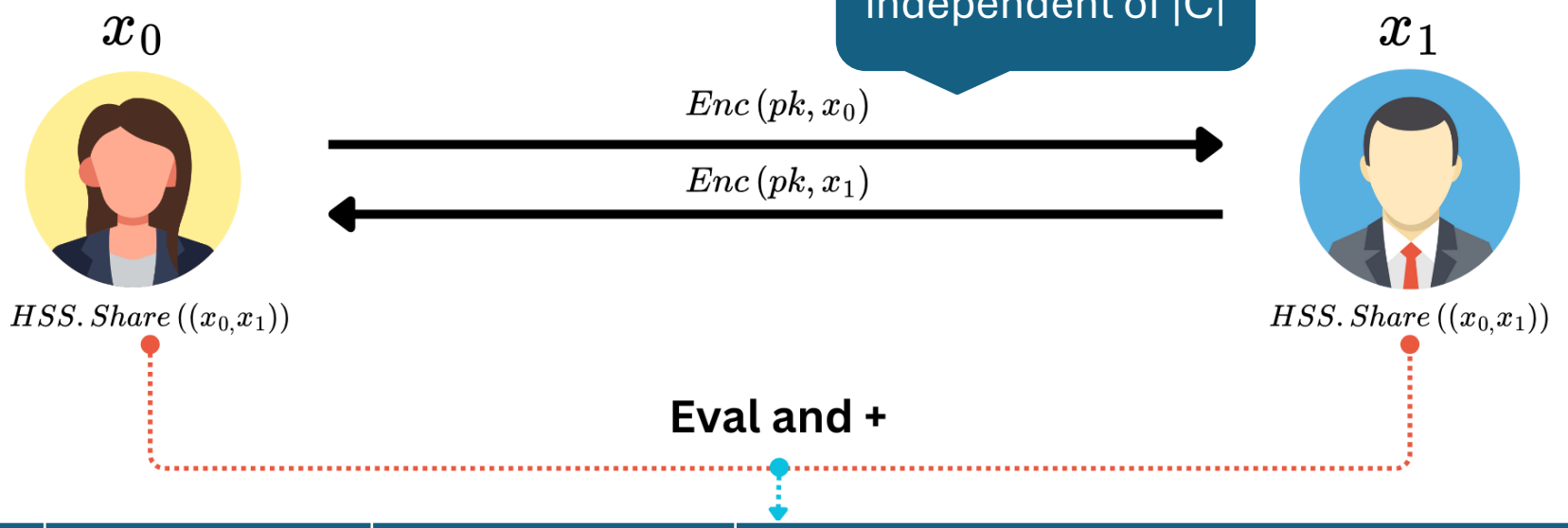
$$d = \omega(1)$$

- Then refresh after each layer
- Total communication = s/d

Sublinear MPC from HSS

HSS for circuit class = Sublinear MPC for circuit class

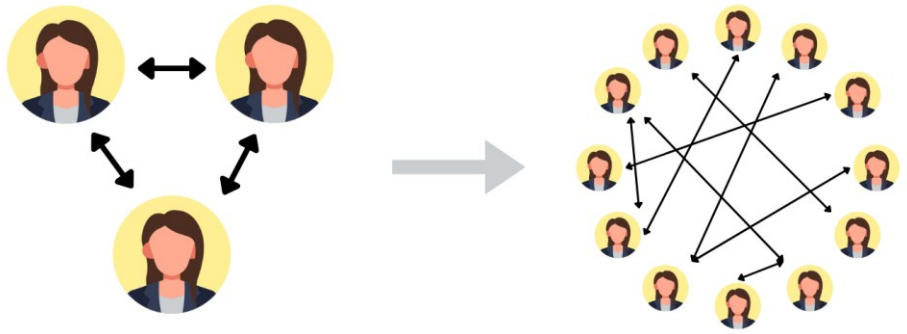
Communication independent of |C|



N-party HSS = N-party sublinear MPC

	Assumptions	Circuit Class	Communication	No. of Parties
[Gen09]	LWE	P/Poly	$O(n+m) + poly(\lambda)$	Arbitrary
[BGI16]	DDH	Layered Circuits	$O(n+m) + O(s/\log s)$	2
[OSY21], [RS21]	DCR	Layered Circuits	$O(n+m) + O(s/\log s)$	2
[CM21]	Superpoly LPN	Layered Circuits	$O(n+m) + O(s/\log\log s)$	2
[BCM22]	QR+LPN	Layered Circuits	$O(n+m+d^{\frac{1}{3}}s^{\frac{2(1+\epsilon)}{3}} \cdot poly(\lambda) + s/\log\log s)$	2
[BCM23]	**DDH/DCR + LPN**	**Layered Circuits**	$O(n+m+d^{\frac{1}{3}}s^{\frac{2(1+\epsilon)}{3}} \cdot poly(\lambda) + s/\log\log s)$	**3 / 5**

This Work

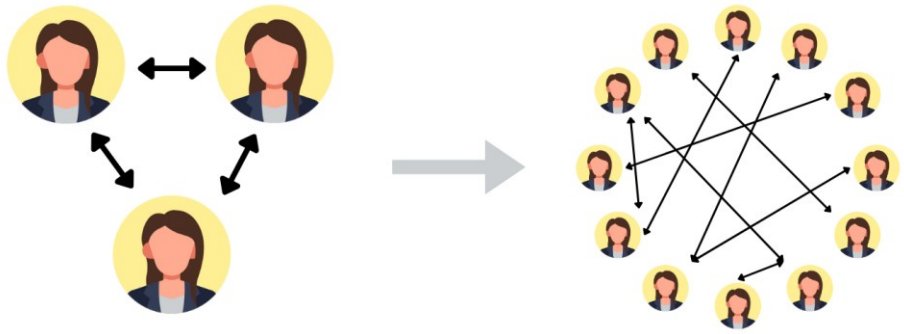


More Parties

DDH/DCR/LPN + Low Depth PRG

More Assumptions

This Work



More Parties

DDH/DCR/LPN + Low Depth PRG

More Assumptions

	Assumptions	Circuit Class	Communication	No. of Parties
[BCM23]	(DDH + LPN) / (DCR + LPN)	Layered Circuits	$O(s/\log\log s)$	3 / 5
This Work	DCR/DDH + LD-PRG*	Layered Circuits	$O(s/\log\log s)$	4
This Work	DCR/DDH + LPN + LD-PRG*	Layered Circuits	$O(s/\log\log s)$	5
This Work	DCR/DDH + superpoly LPN + LD-PRG*	Layered Circuits	$O(s/\log\log s)$	8
This Work	Superpoly (DCR/DDH + LPN) + LD-PRG*	Layered Circuits	$O(s/\log\log\log s)$	9
This Work	Superpoly (DCR/DDH + LPN) + LD-PRG*	Layered Circuits	$O(s/\log\log\log s)$	**10**

* Low-Depth PRG in the class XOR-AND of constant-degree polynomials, which can be instantiated based on the security of Goldreich's PRG, one-wayness of random local functions or from the multivariate quadratic family of assumptions.

Concurrent Work

- Dao, Ishai, Jain, Lin [DIJL23] – CRYPTO 2023
  - **N-party HSS** from **Sparse-LPN**
  - **N-party sublinear MPC** for layered circuits – O(s/log log s) communication
  - 1/poly **error**!

- Abram, Roy, Scholl [ARS24] – Eurocrypt 2024
  - **N-party sublinear MPC** for layered circuits from **DCR**
  - O(s/log log s) communication
  - No error

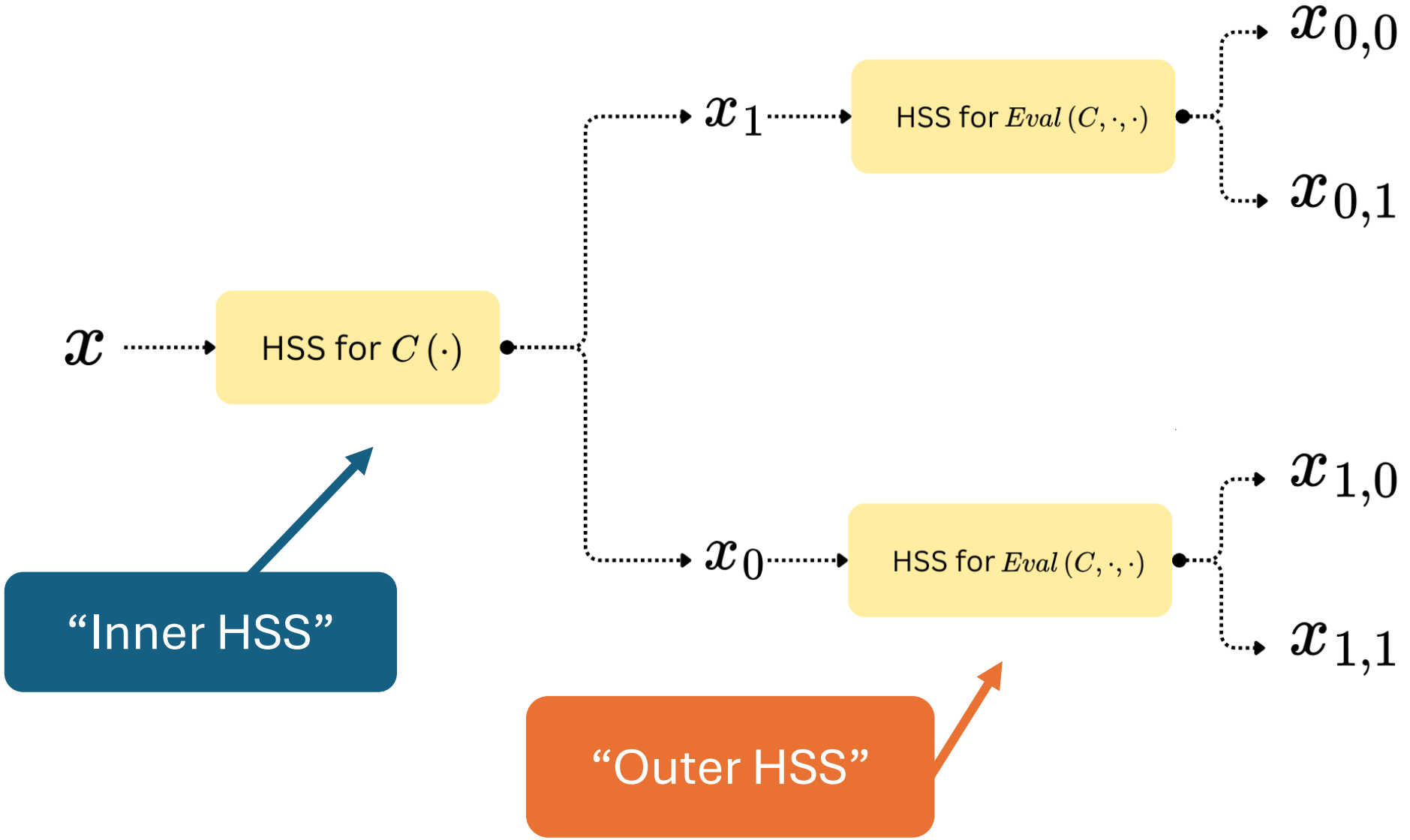
Concurrent Work

- Dao, Ishai, [illegible]3] – CRYPTO 2023
  - **N-party** [illegible]**PN**
  - **N-party su**[illegible]or layered circuits – O(s/log log s) communication
  - 1/poly **error**!

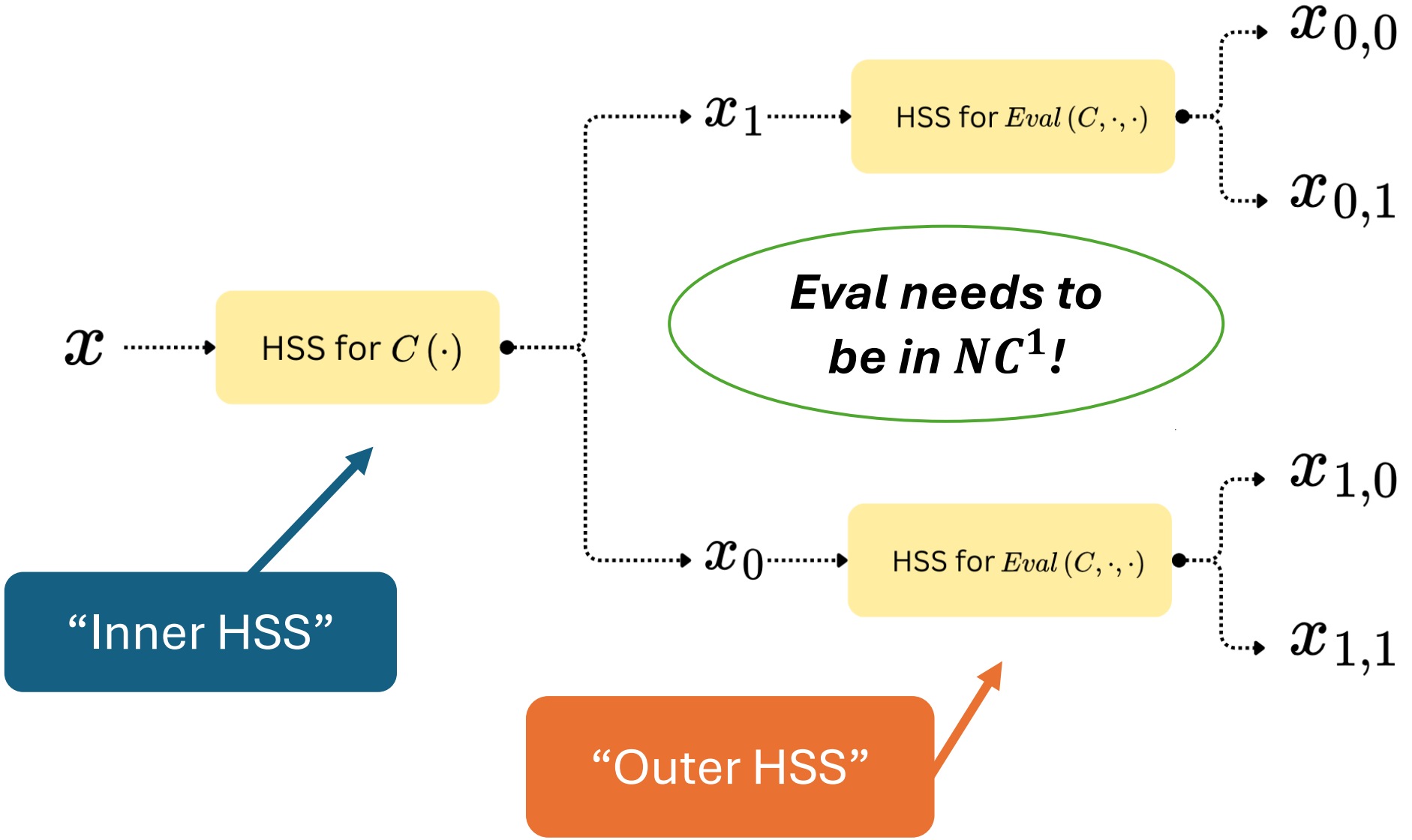
We show how to fix this error!

- Abram, Roy, Scholl [ARS24] – Eurocrypt 2024
  - **N-party sublinear MPC** for layered circuits from **DCR**
  - O(s/log log s) communication
  - No error

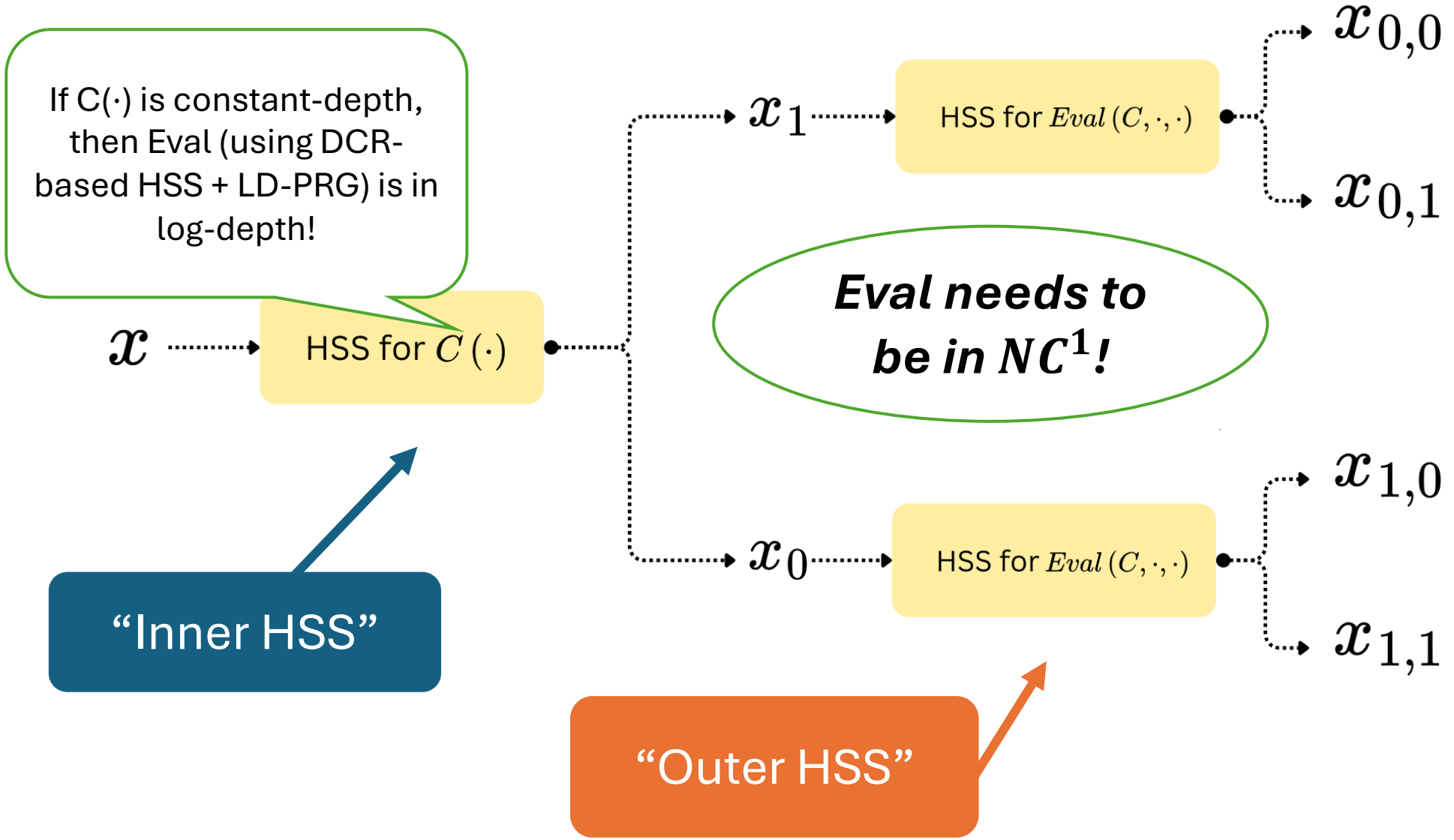
Multi-Party HSS – Nesting



Multi-Party HSS – Nesting

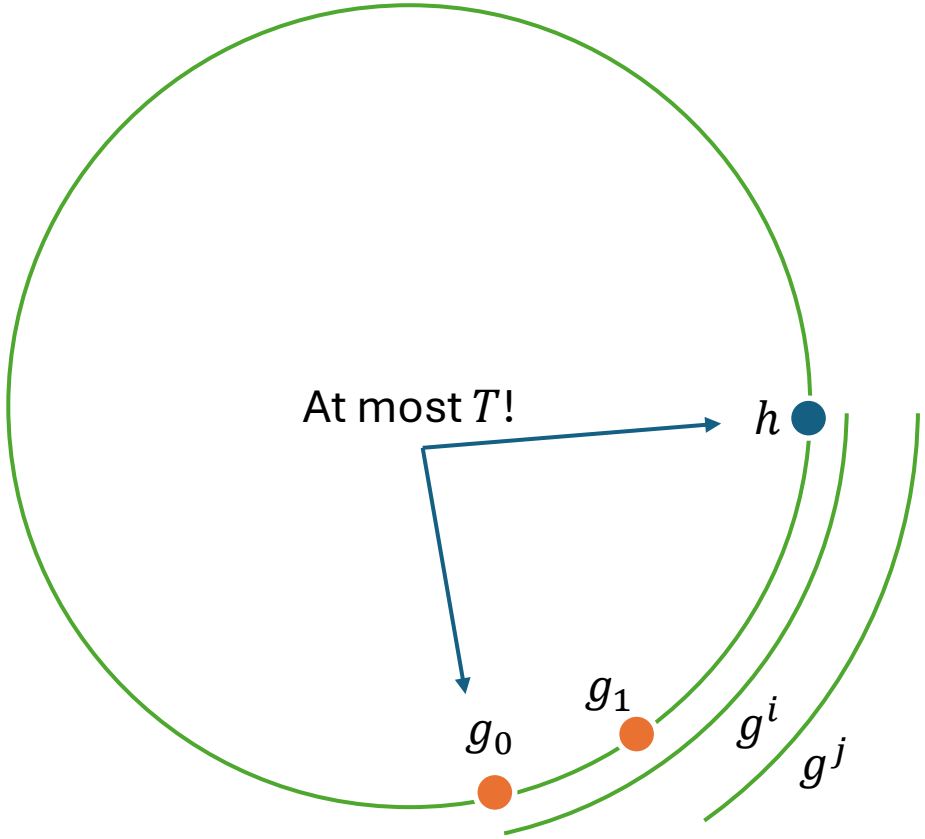


Multi-Party HSS – Nesting



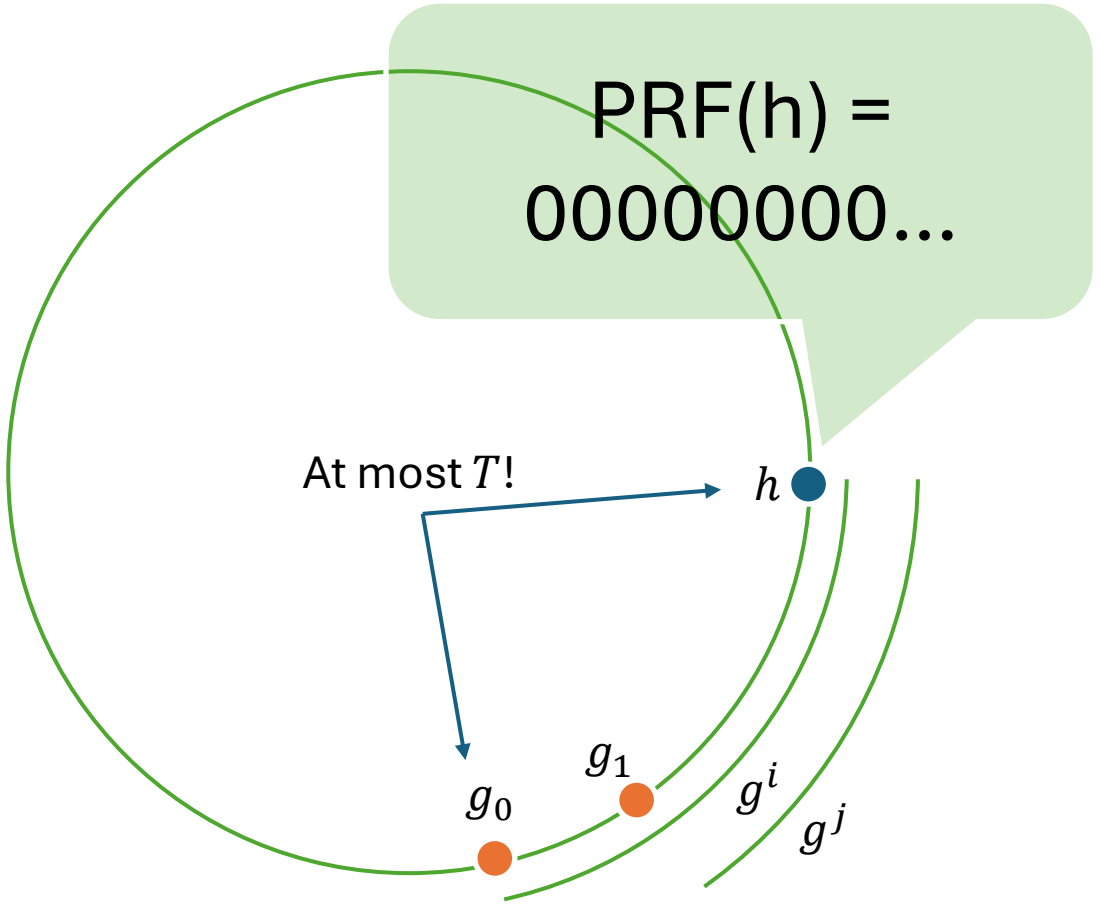
First Contribution – Nesting Imperfect HSS

- Want to do Discrete Log (from DDH) in log-depth
- i can be size $poly(\lambda)$, might need to take $poly(\lambda)$ steps



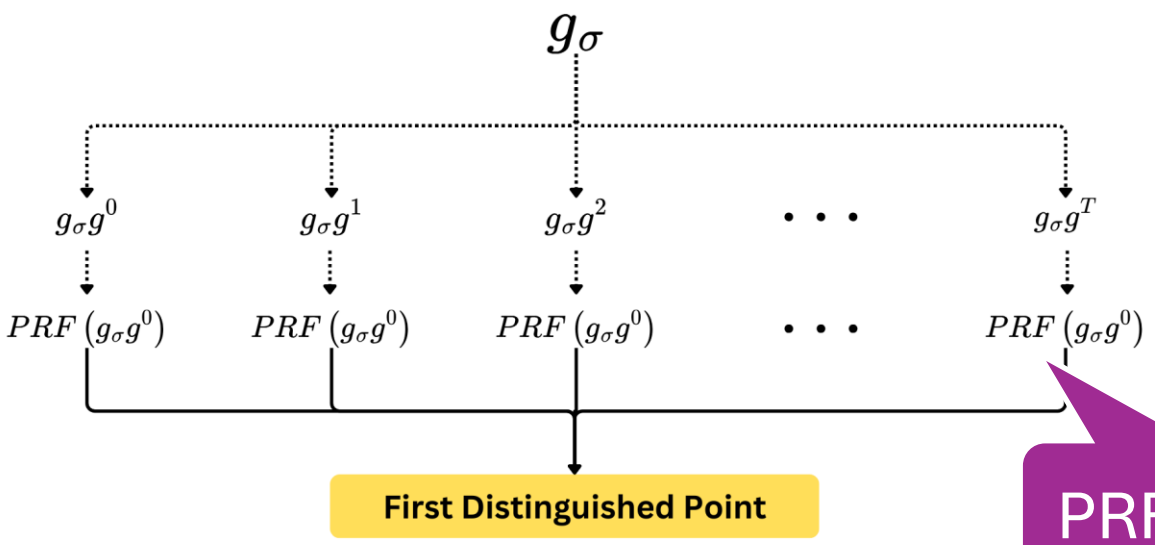
First Contribution – Nesting Imperfect HSS

- Want to do Discrete Log (from DDH) in log-depth
- i can be size $poly(\lambda)$, might need to take $poly(\lambda)$ steps

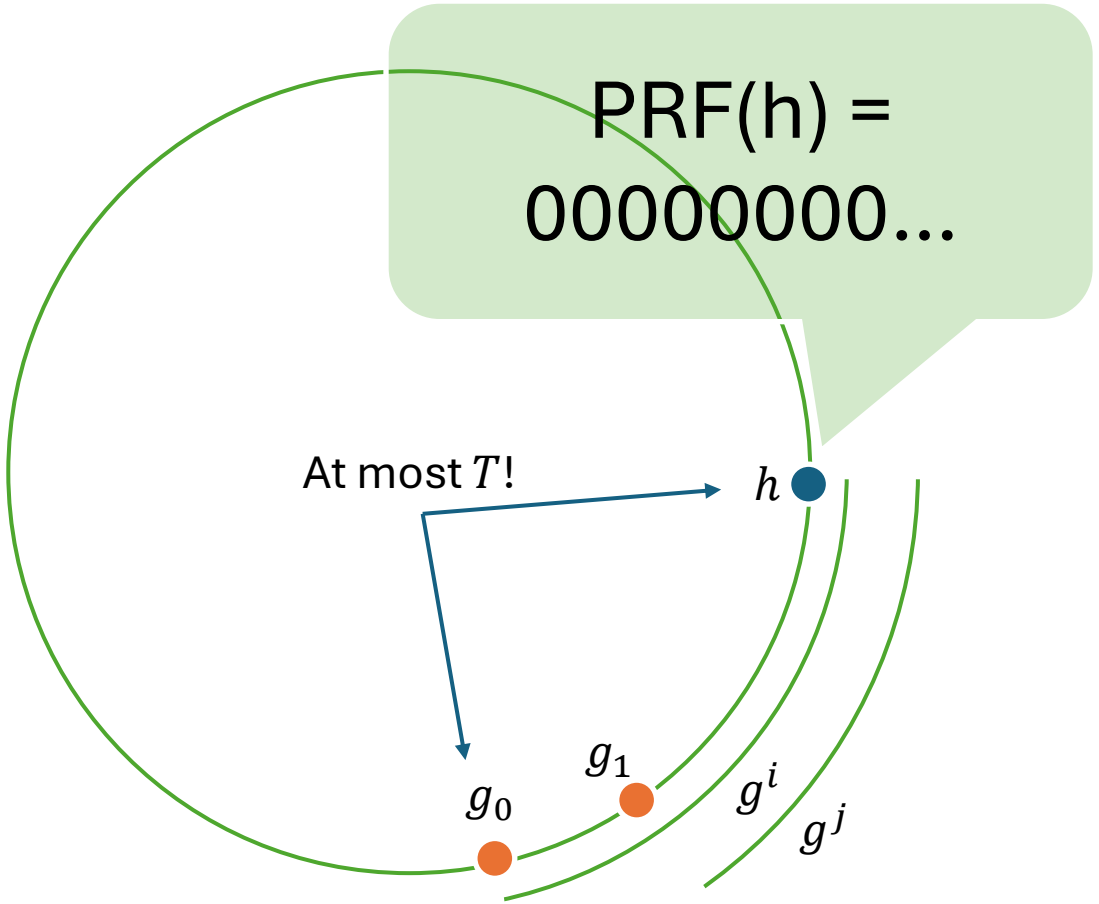


First Contribution – Nesting Imperfect HSS

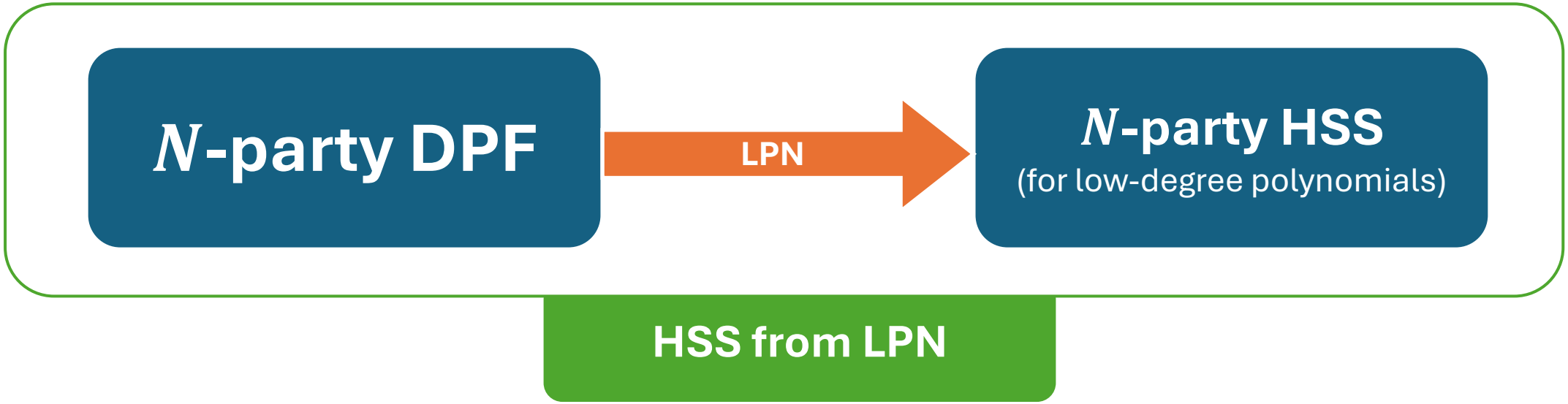
- Want to do Discrete Log (from DDH) in log-depth
- i can be size $poly(\lambda)$, might need to take $poly(\lambda)$ steps



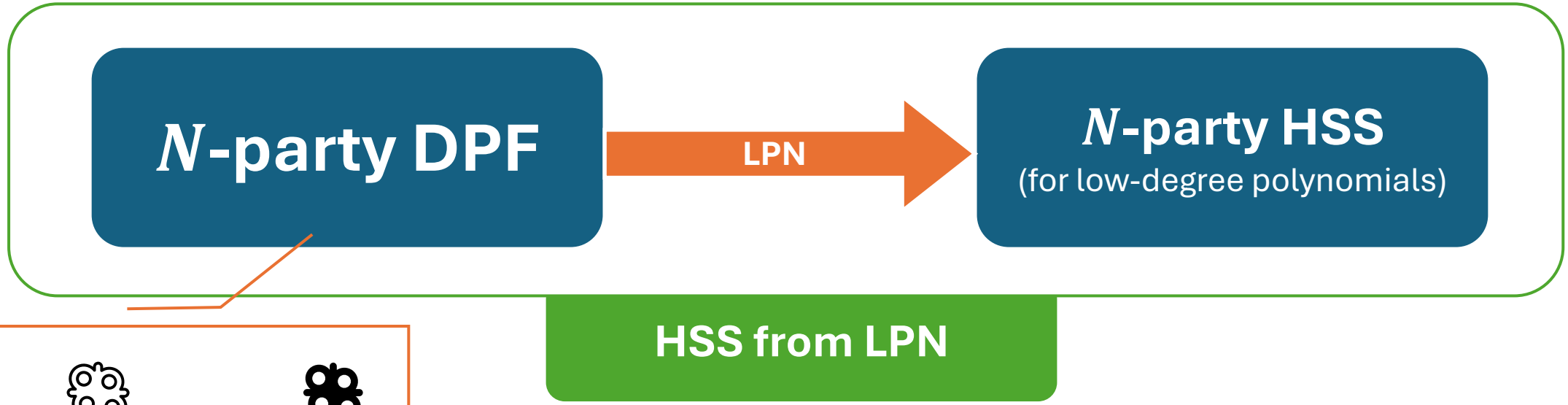
PRF in NC^1!



Second Contribution – 8-party HSS



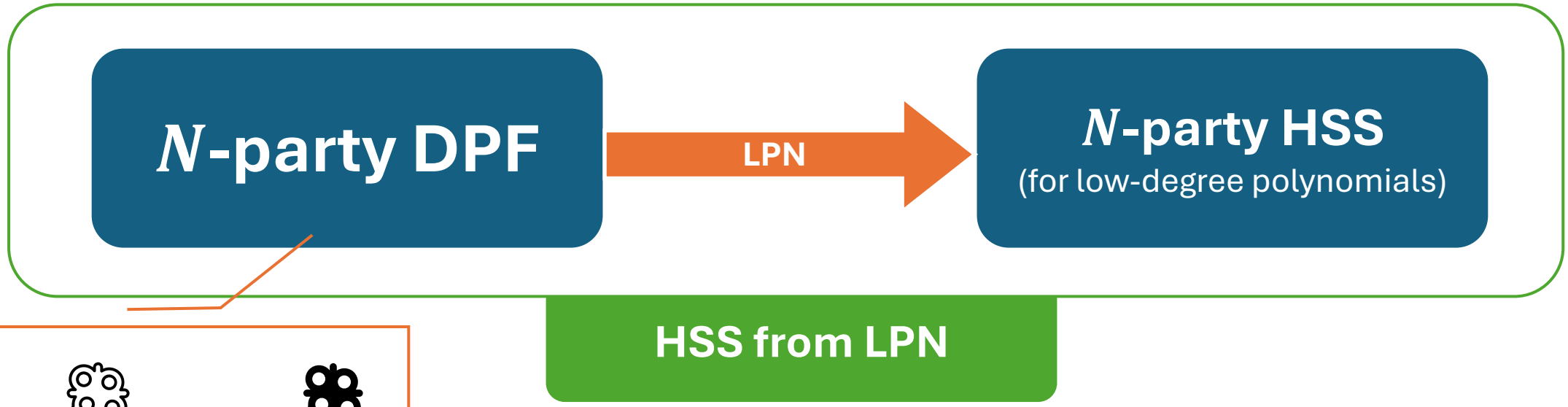
Second Contribution – 8-party HSS



$Eval(k_i, x)$

- **Privacy:** Keys hide function
- **Output:** Additive shares of a point function

Second Contribution – 8-party HSS



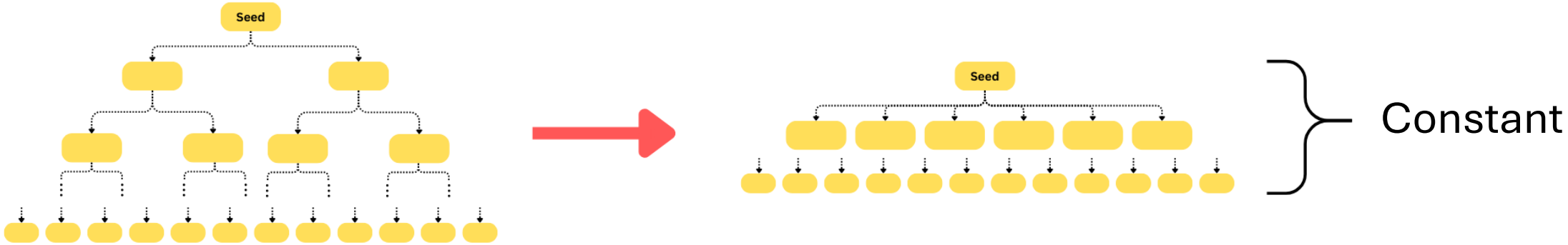
$Eval(k_i, x)$

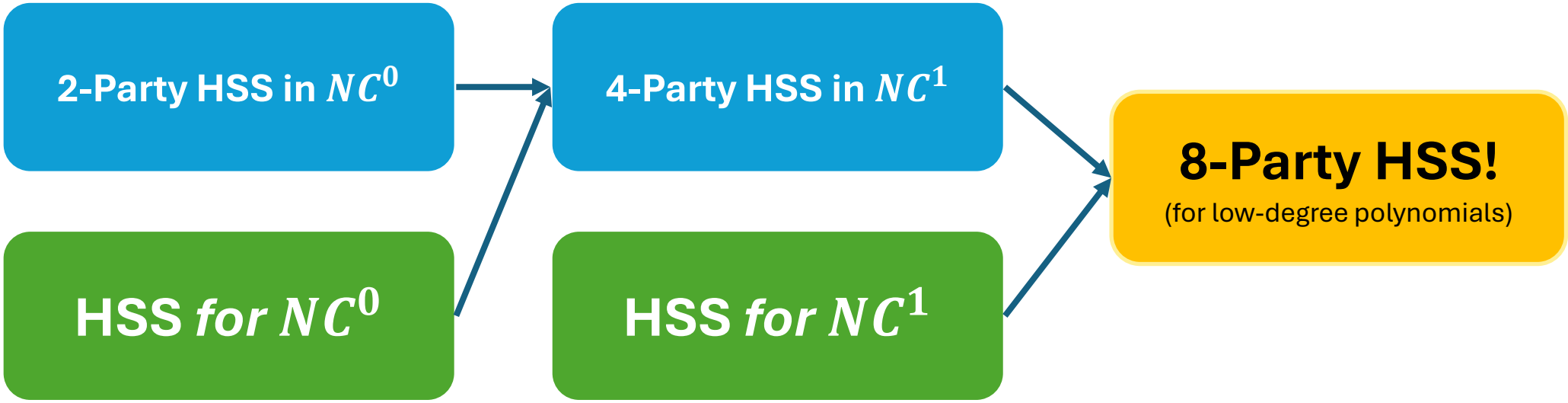
- **Privacy:** Keys hide function
- **Output:** Additive shares of a point function

- 2-Party DPF can be constructed from OWF
- **Complexity of HSS.Eval = Complexity of DPF!**

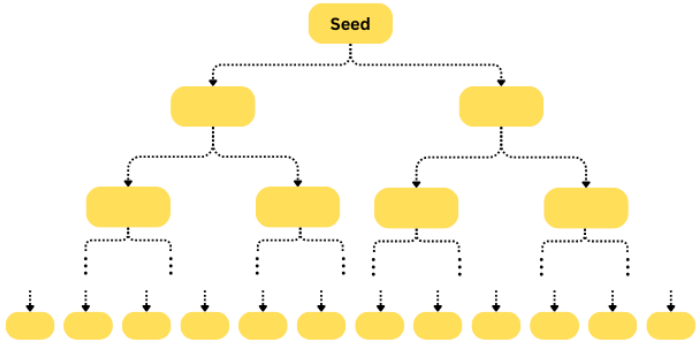
$$\boldsymbol{DPF\ in\ NC^0 \Rightarrow HSS.Eval\ in\ NC^0!}$$

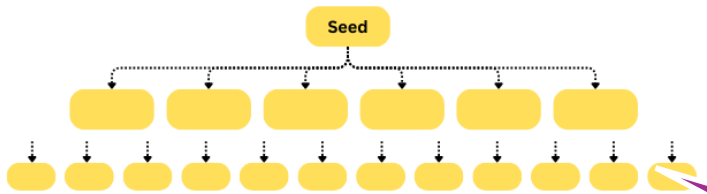
Second Contribution – 8-party HSS





Second Contribution – 8-party HSS





Constant

PRG in NC^0!

2-Party HSS in NC^0

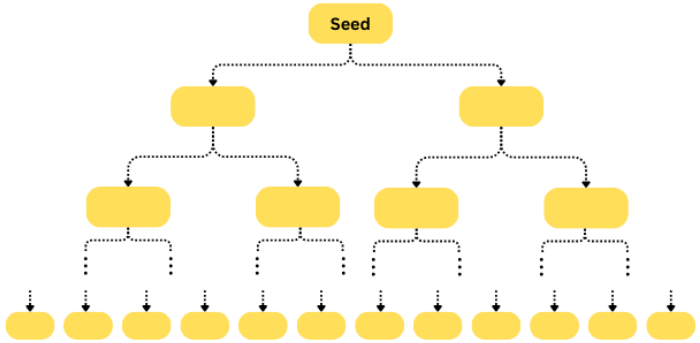
4-Party HSS in NC^1

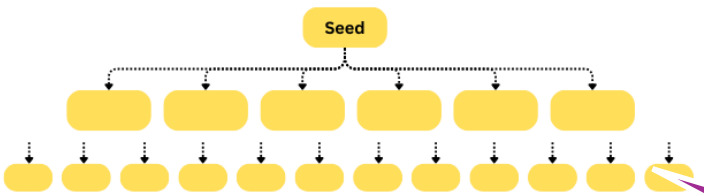
8-Party HSS!
(for low-degree polynomials)

HSS *for* NC^0

HSS *for* NC^1

Second Contribution – 8-party HSS





Constant

PRG in NC^0!

(superpoly) LPN + low-degree PRG

2-Party HSS in NC^0

4-Party HSS in NC^1

8-Party HSS!
(for low-degree polynomials)

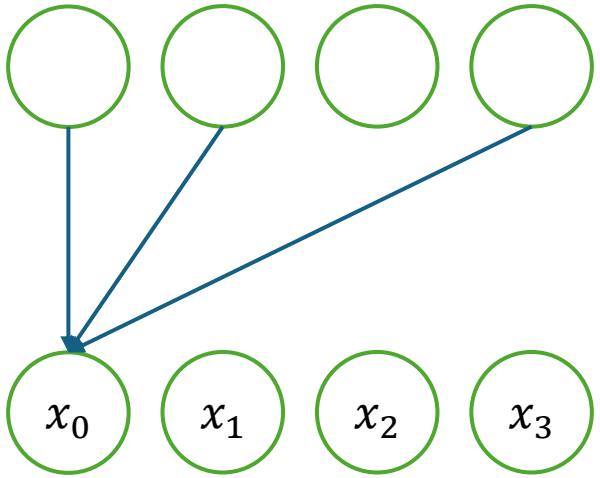
HSS *for* NC^0

HSS *for* NC^1

DDH or DCR

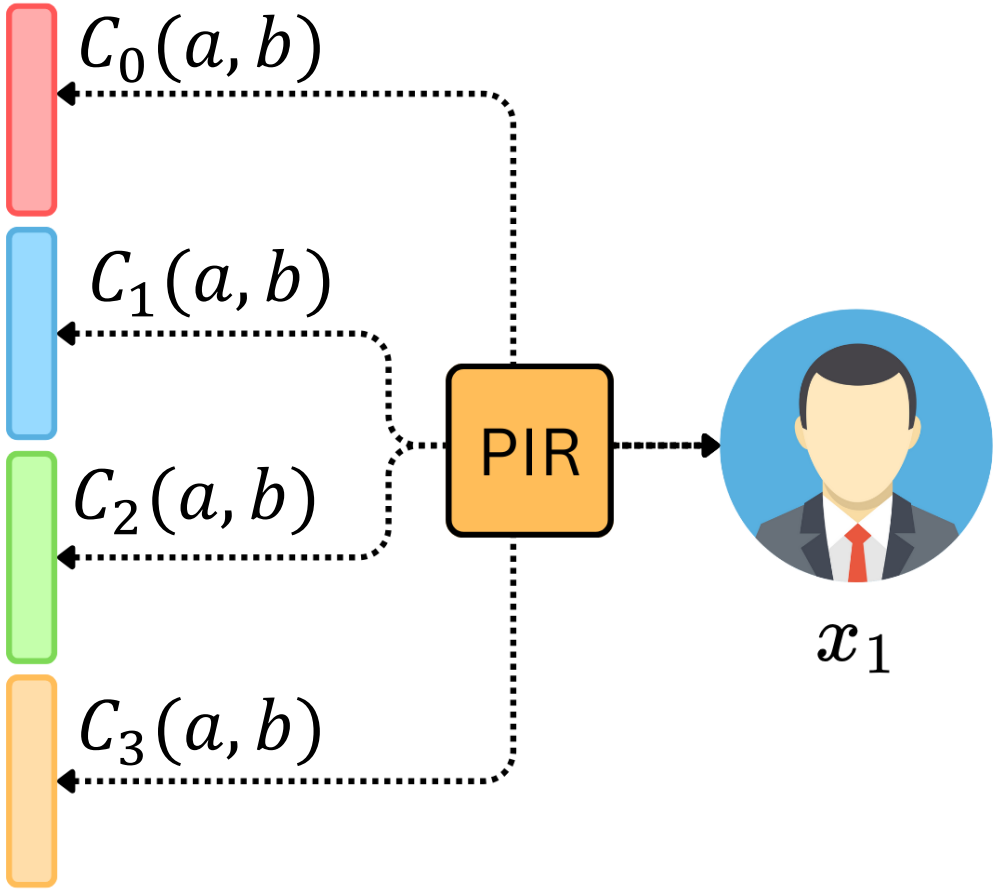
Third Contribution – N + 2-party MPC

- Starting point: **Correlated SPIR**



$C_0(\cdot)$

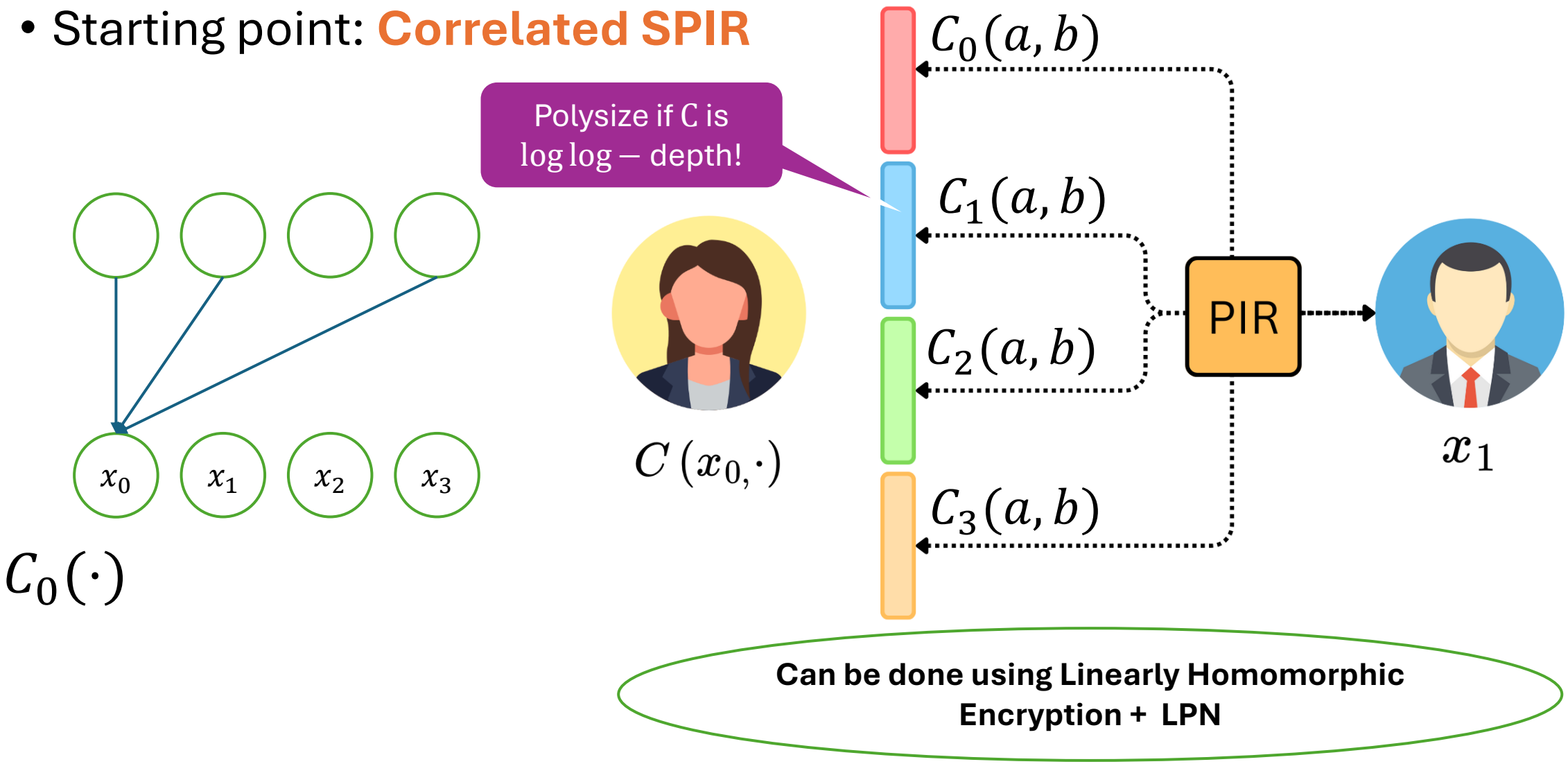
$C(x_0,\cdot)$



Can be done using Linearly Homomorphic Encryption + LPN

Third Contribution – N + 2-party MPC

- Starting point: **Correlated SPIR**



N+1 Party MPC

- Consider the function

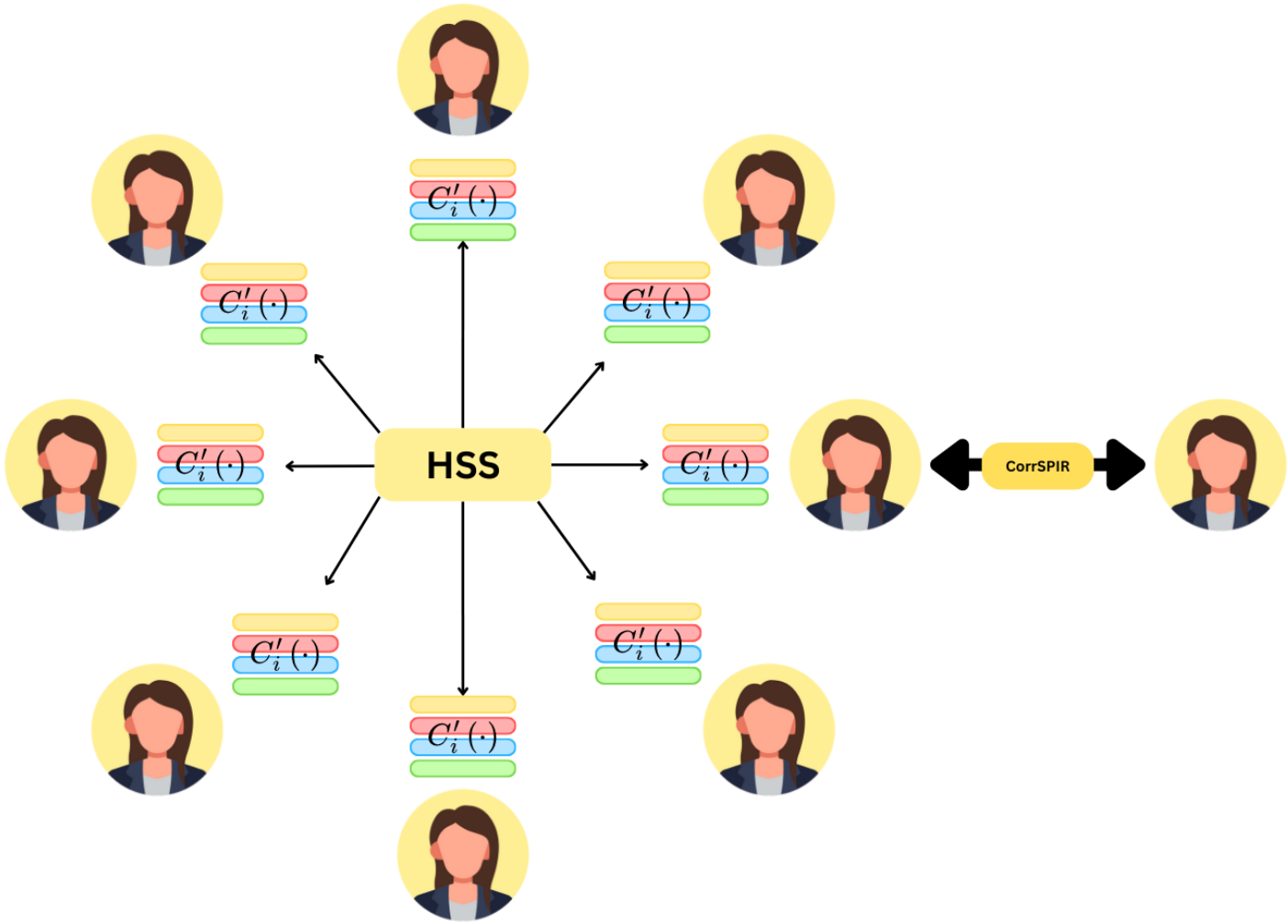
$$C'(\cdot) = C(\cdot, x_0, x_1, \ldots, x_N)$$

- Parties use HSS to evaluate shares of **truth table of C'**

- Perform SPIR with each party

- Total communication:

$$O(Ns/\log\log s)$$



N+2 Parties

In NC^1!

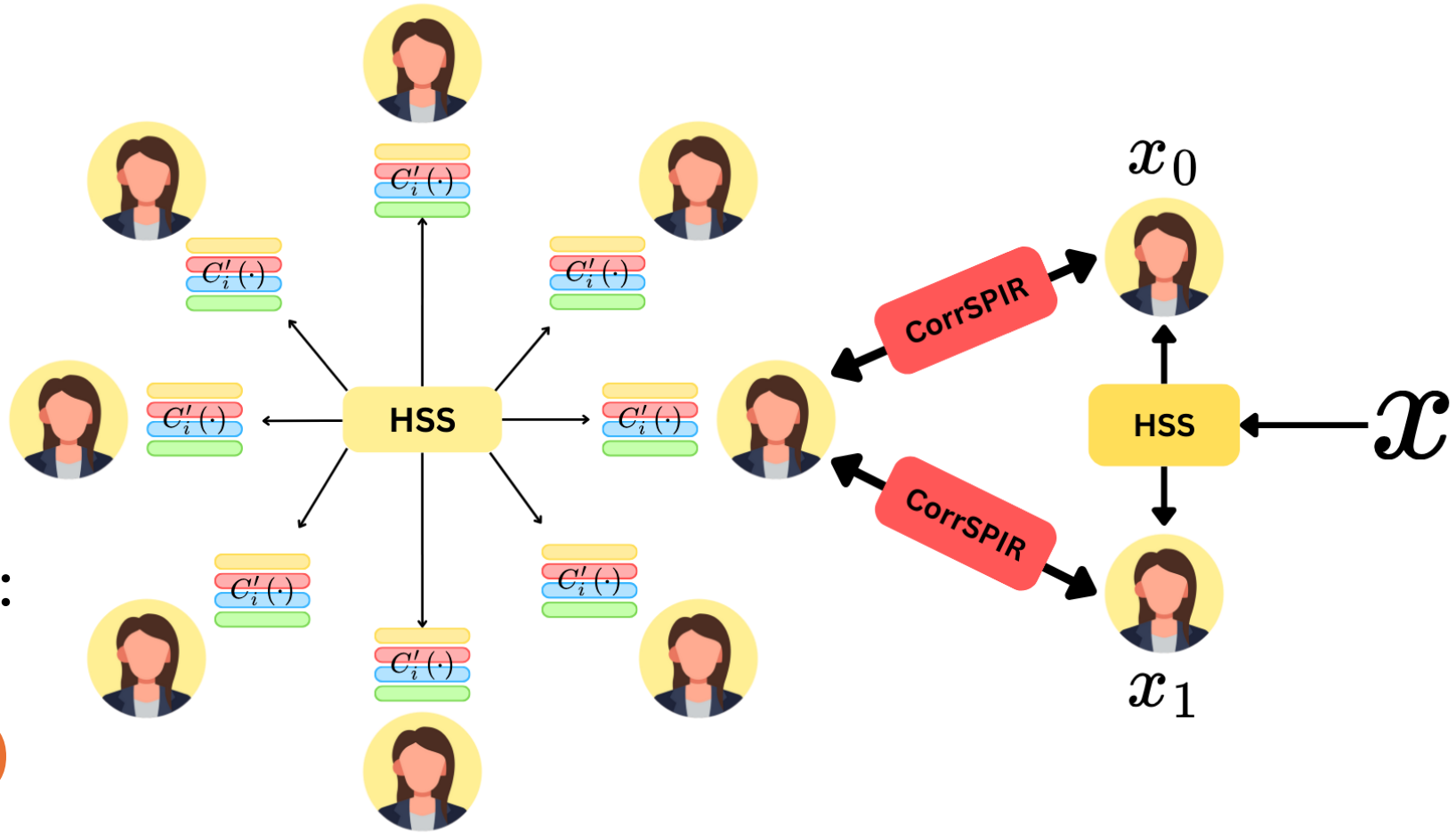
- $CorrSPIR = (Query, Answer, Decode)$

- Superpolynomial Assumptions

- Even smaller database size

- Total Communication:

$$O(Ns/\log\log\log s)$$

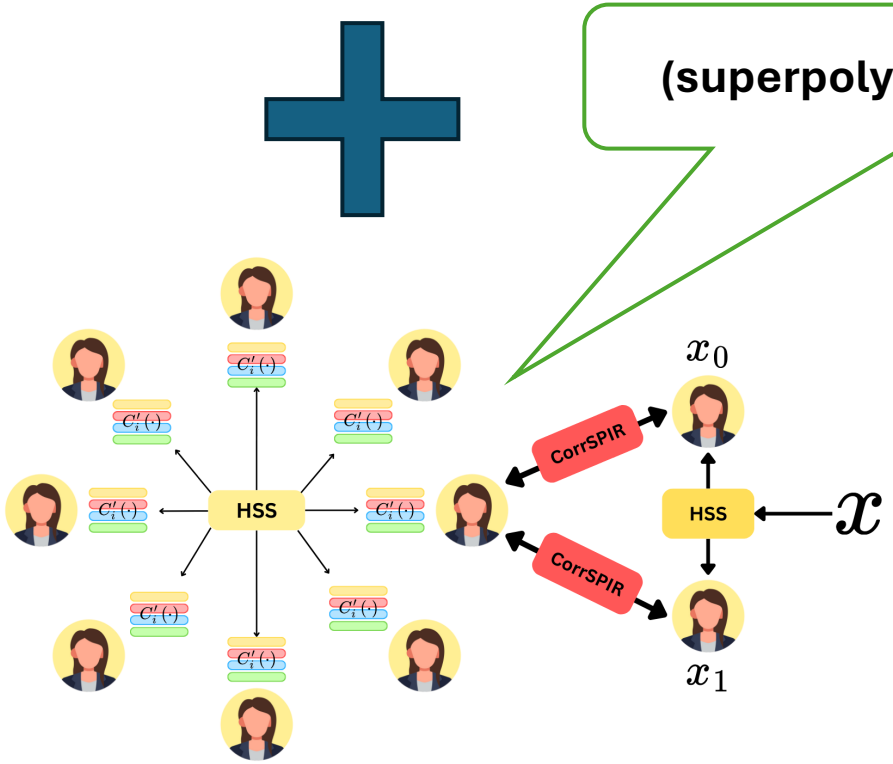


Wrapping up: putting the schemes together

8-Party HSS
(for low-degree polynomials)

DDH/DCR + (superpoly) LPN + Low-Degree PRG

(superpoly) DDH/DCR + LPN



10-Party Sublinear MPC

with communication:

$$O(s/\log\log\log s)$$

Thank You!