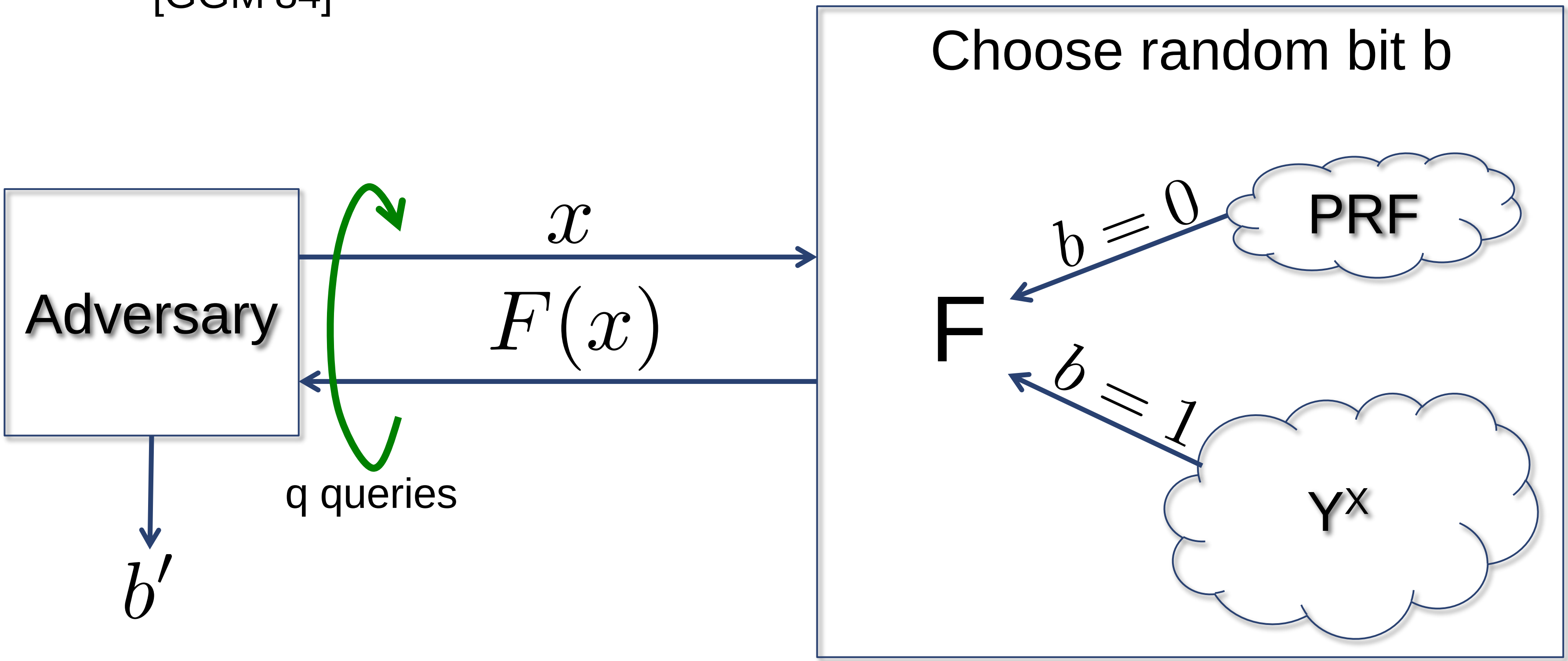
HOW TO CONSTRUCT QUANTUM RANDOM FUNCTIONS

Mark Zhandry – Stanford University

(Classical) Pseudorandom Functions

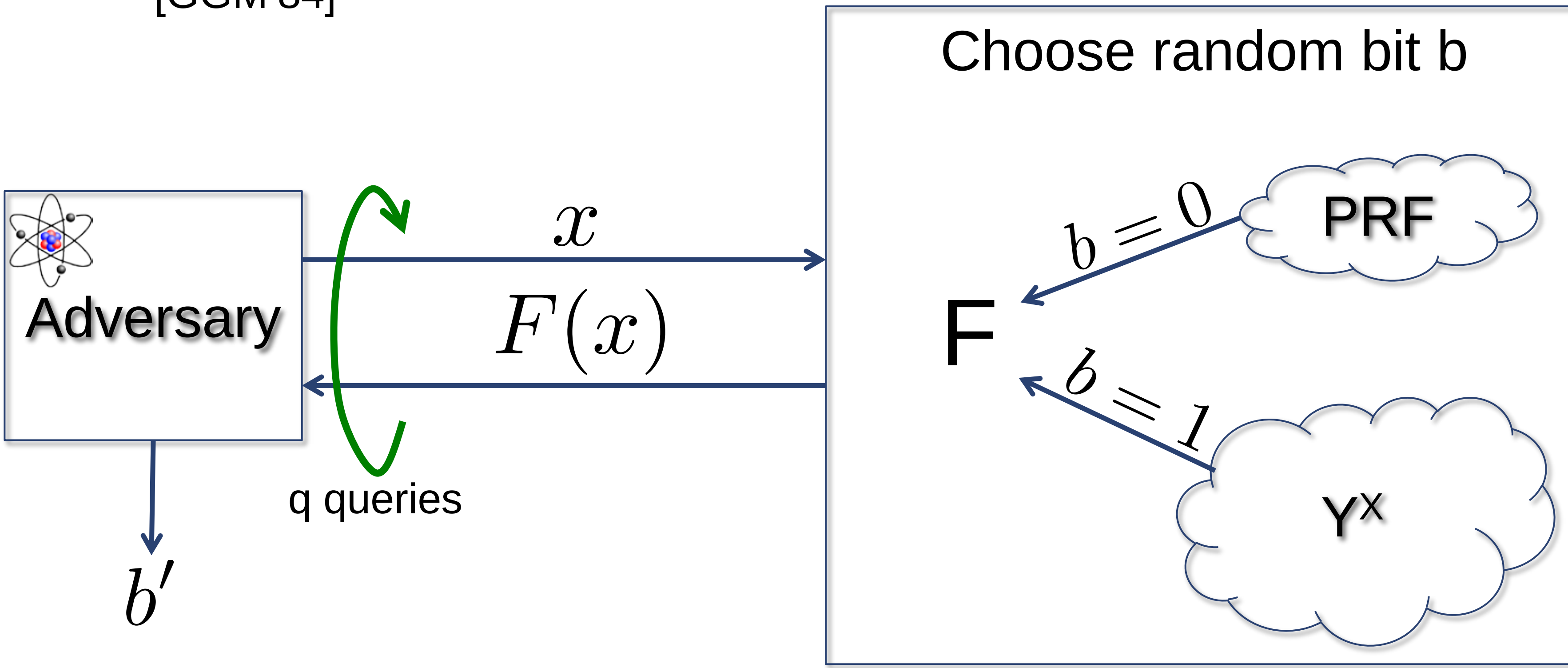
[GGM'84]



PRF is secure if $\left|\Pr[b = b'] - \frac{1}{2}\right| < \texttt{negl}$

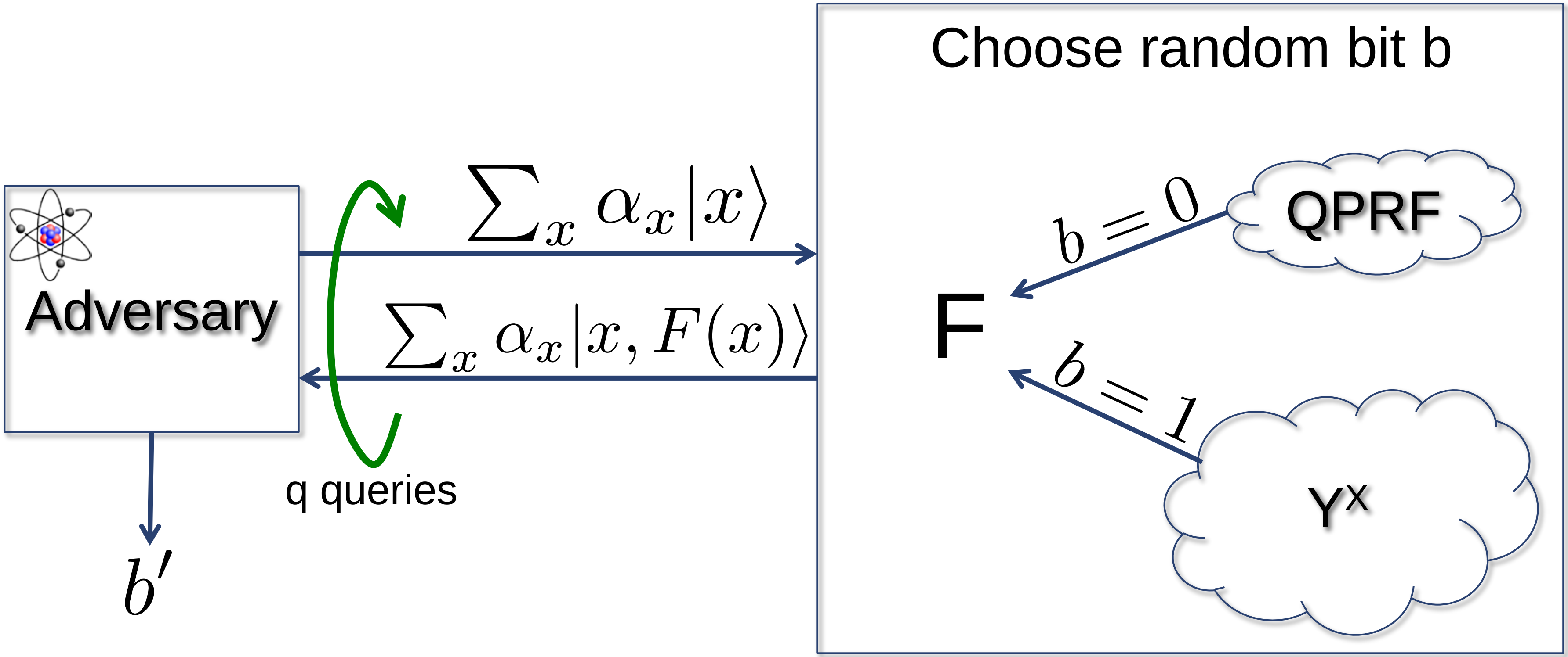
(Classical) Pseudorandom Functions

[GGM'84]



PRF is secure if $\left| \Pr[b = b'] - \frac{1}{2} \right| < \texttt{negl}$

Quantum Pseudorandom Functions



Single query evaluates F on exponentially-many inputs

Quantum Pseudorandom Functions

PRFs: building block for most of symmetric crypto

Quantum PRFs: may be needed when end-users are quantum

Specific applications:

- Proofs in the Quantum Random Oracle Model [BDFLSZ'11]
- Needed for MACs secure against quantum chosen message attacks [BZ'12]
- Step towards quantum PRP (e.g. Luby-Rackoff)

Separation

PRF —✗→ Quantum PRF

Theorem: If PRFs exist, then there are PRFs that are not quantum PRFs

- Construct a PRF that is periodic with large, secret period
- Cannot find period with classical queries
- Easy with quantum queries

How to Construct Quantum PRFs

We prove security for some classical PRF constructions:

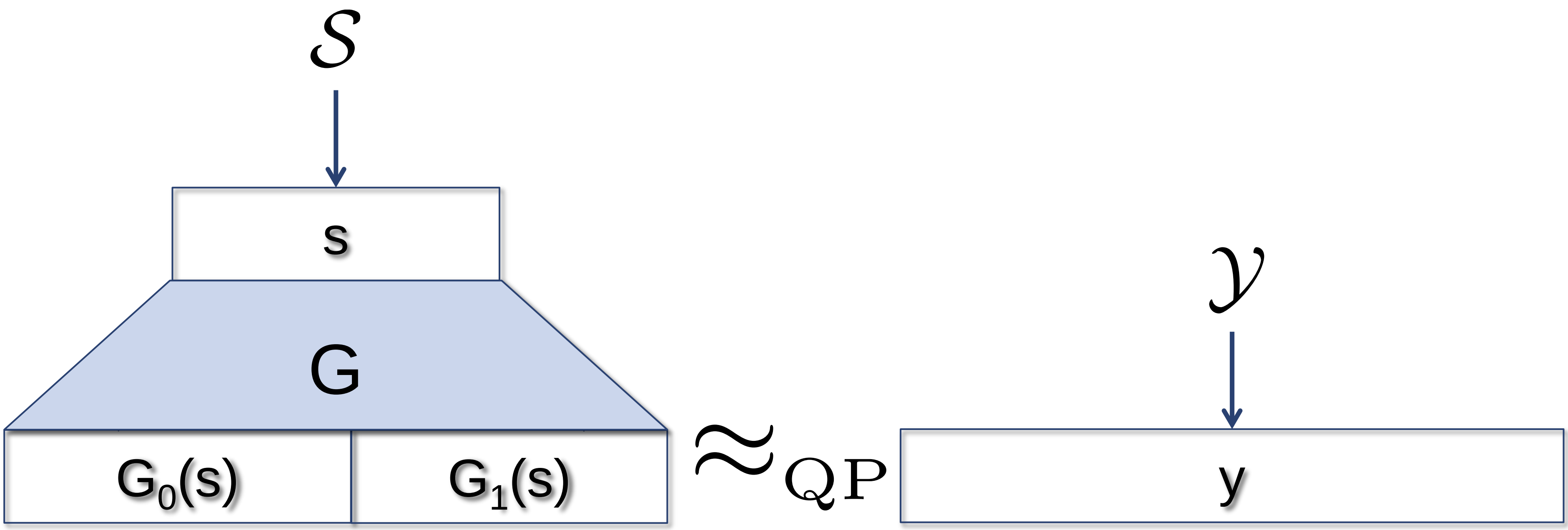
- From quantum-secure pseudorandom generators [GGM'84]
- From quantum-secure pseudorandom synthesizers [NR'95]
- Directly from lattices [BPR'11]

Classical proofs do not carry over into the quantum setting

$\Rightarrow$ Need new proof techniques

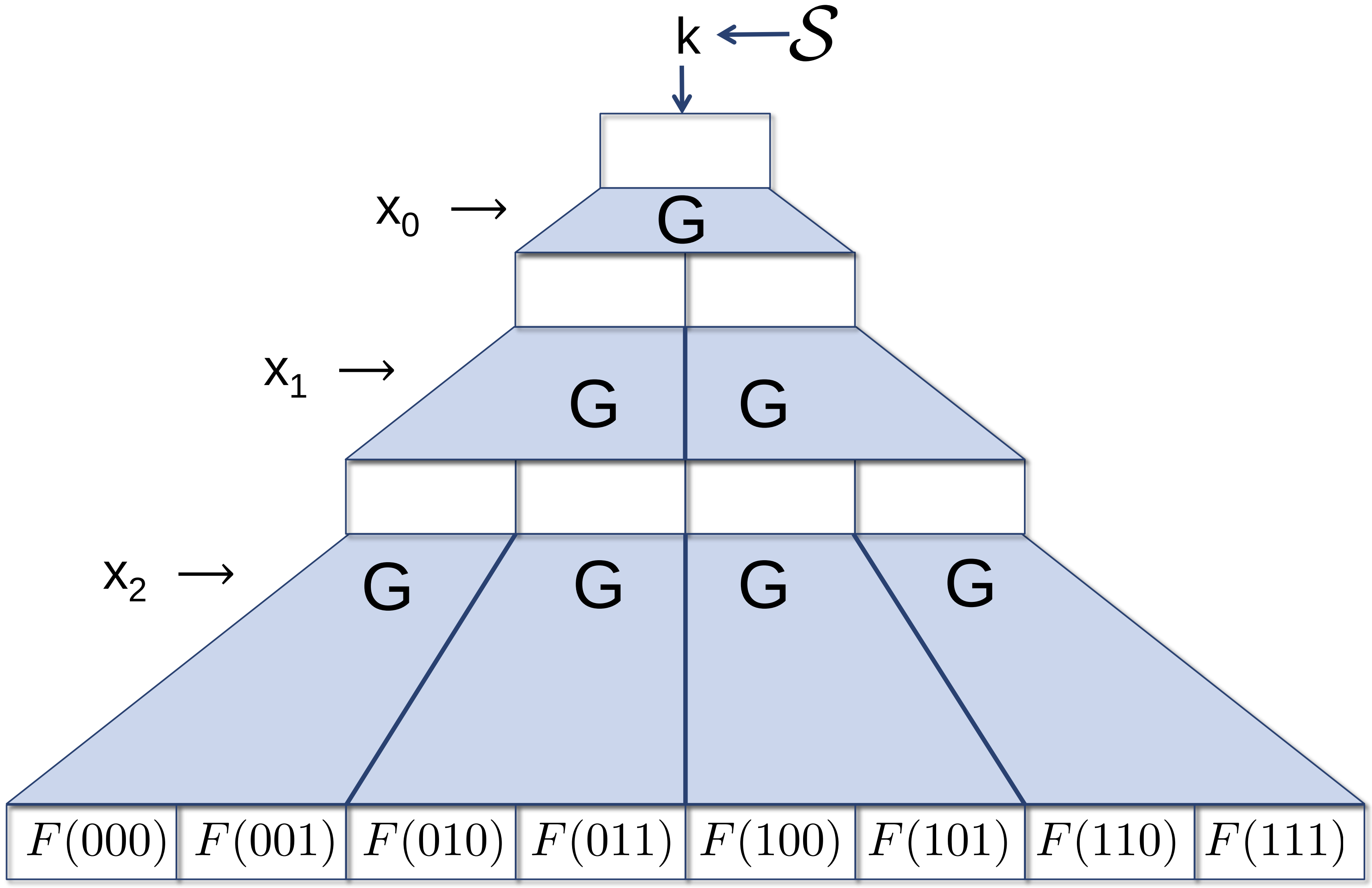
Example: GGM

Pseudorandom Generators



Indistinguishable for Quantum Machines

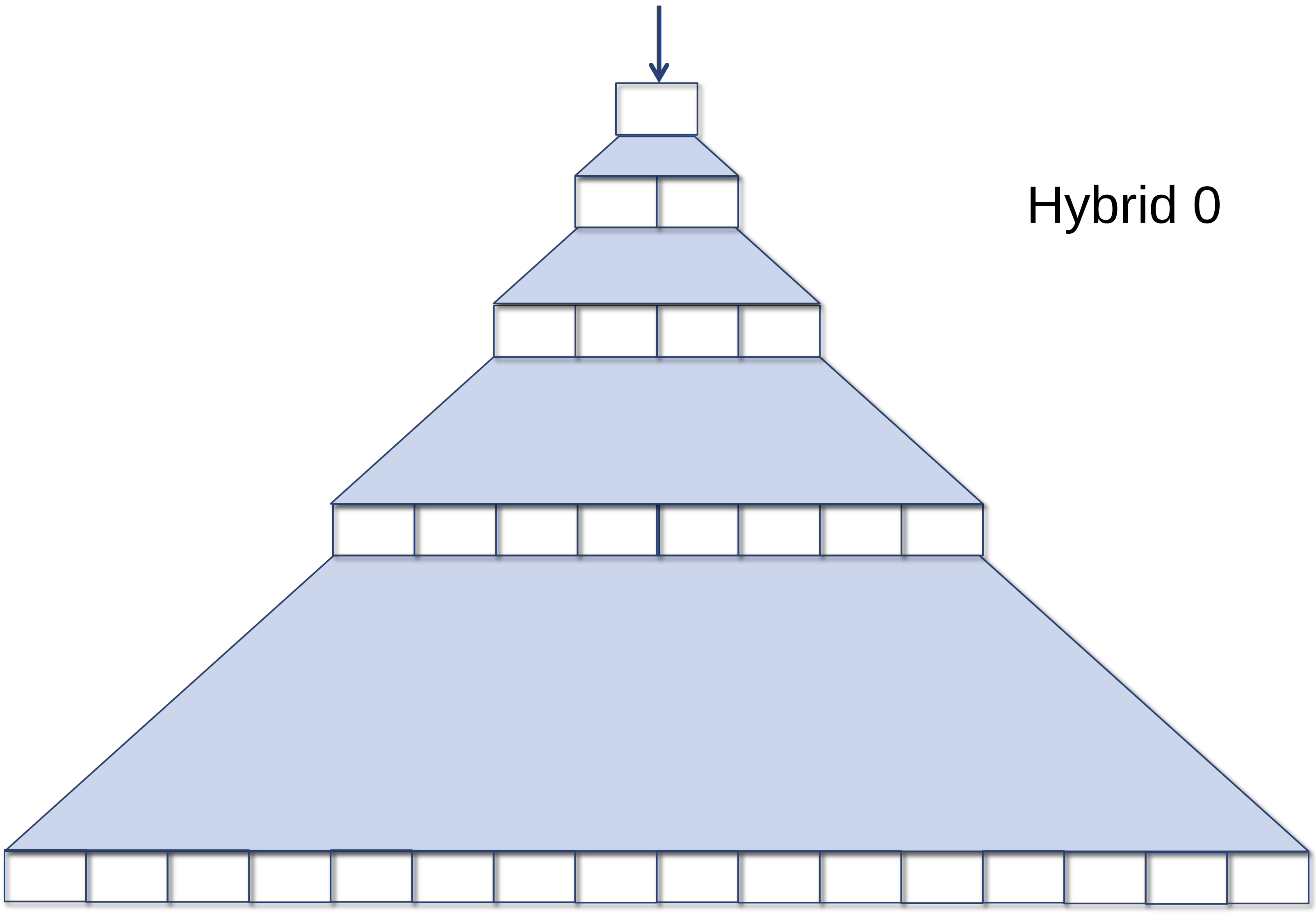
The GGM Construction



Original Security Proof

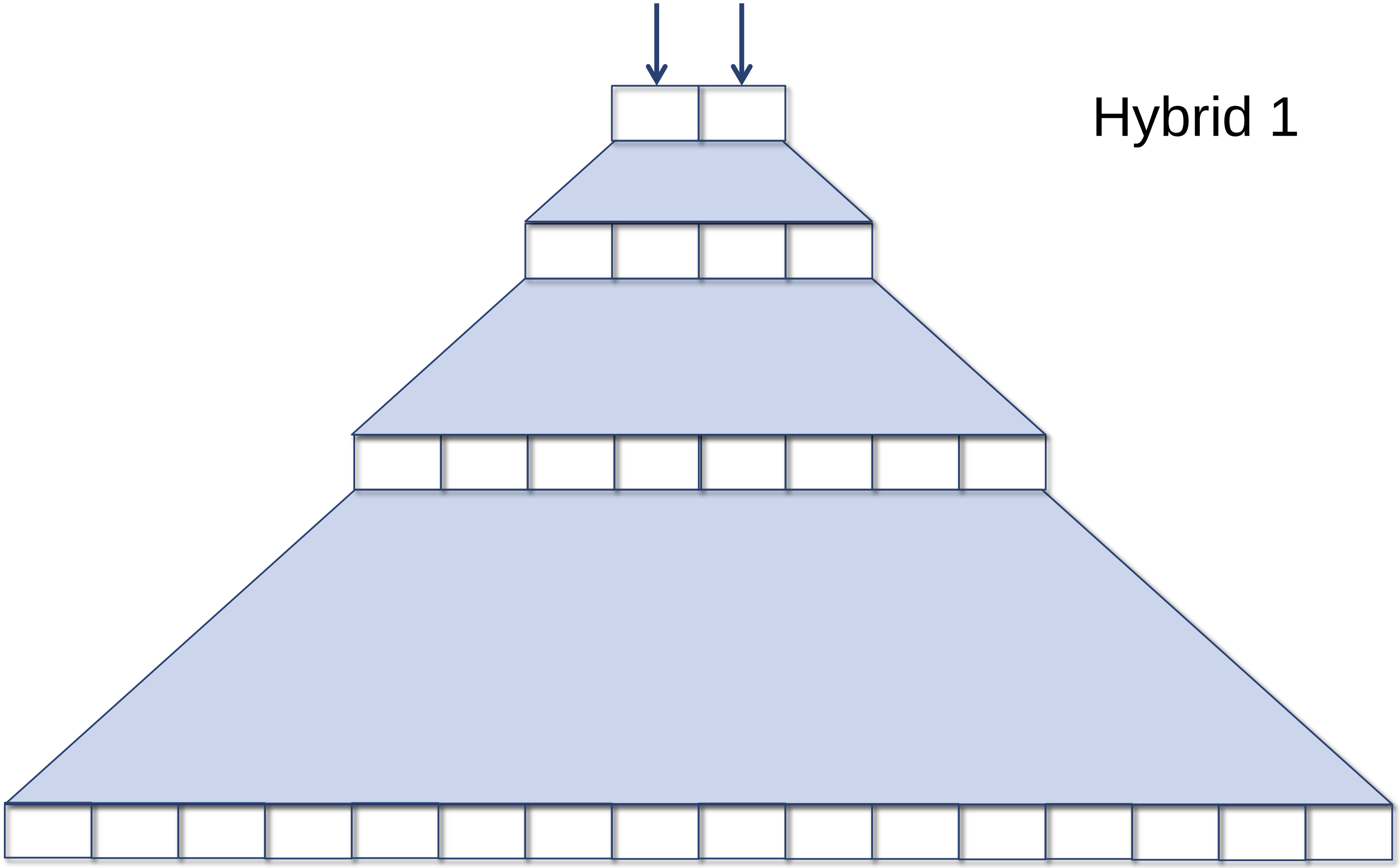
Step 1: Hybridize over levels of tree

Original Security Proof: Step 1



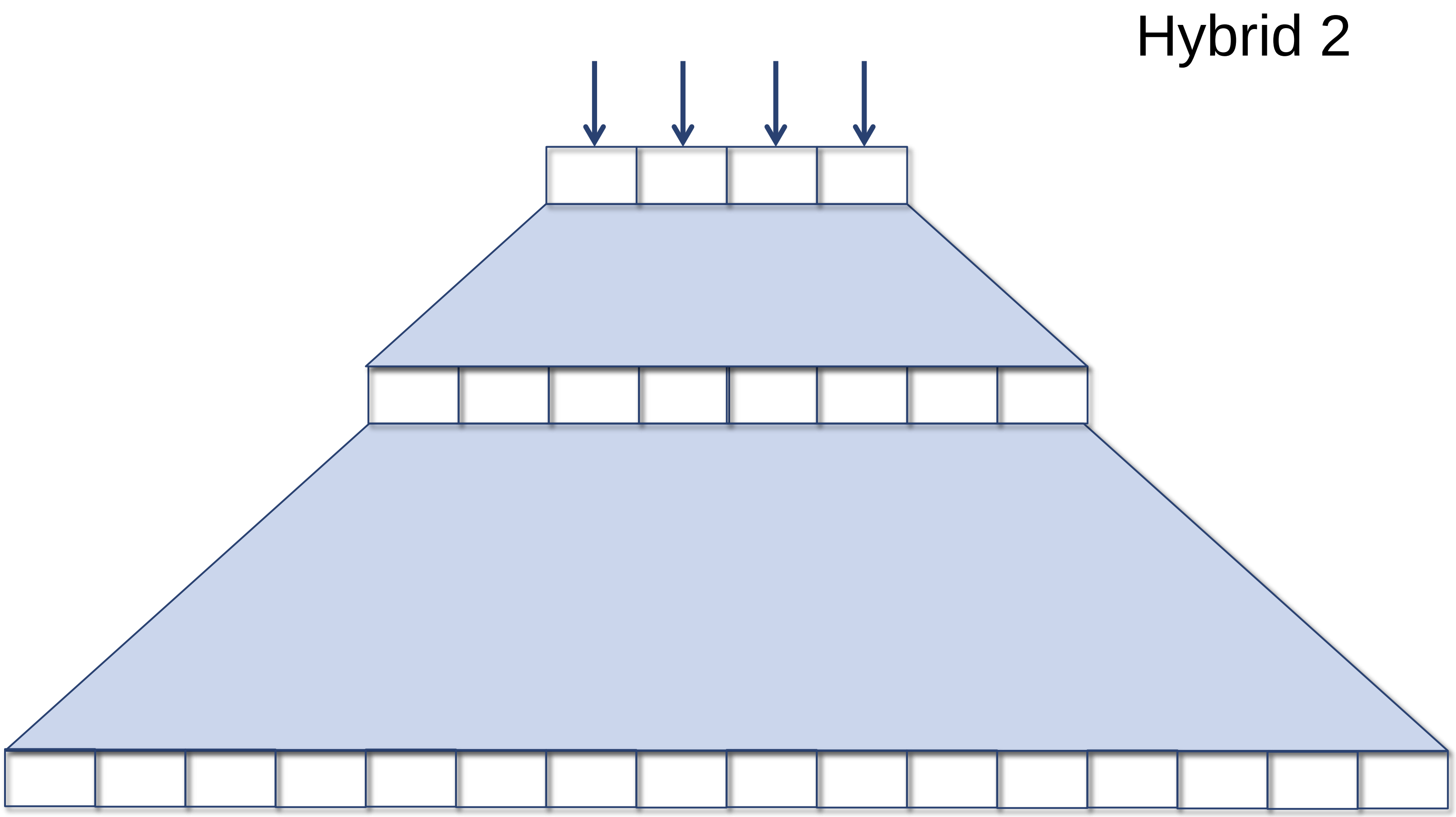
Hybrid 0

Original Security Proof: Step 1



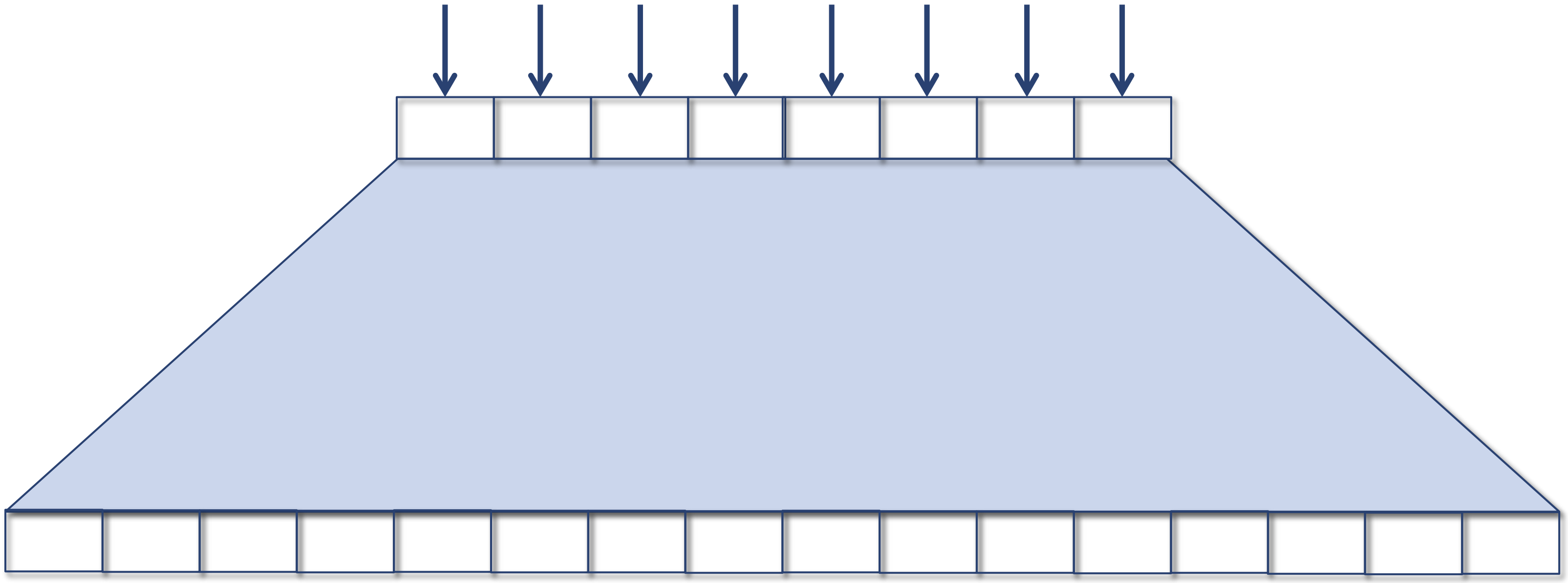
Hybrid 1

Original Security Proof: Step 1



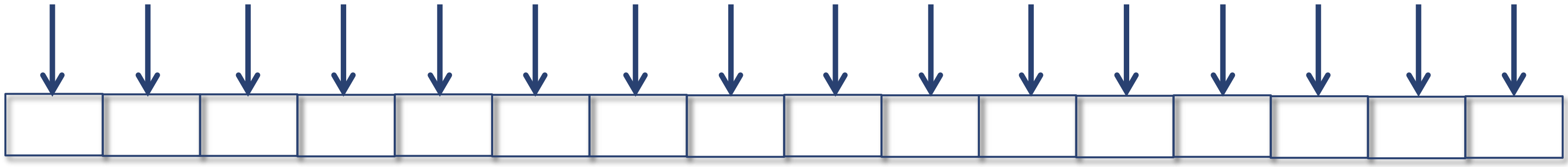
Original Security Proof: Step 1

Hybrid 3



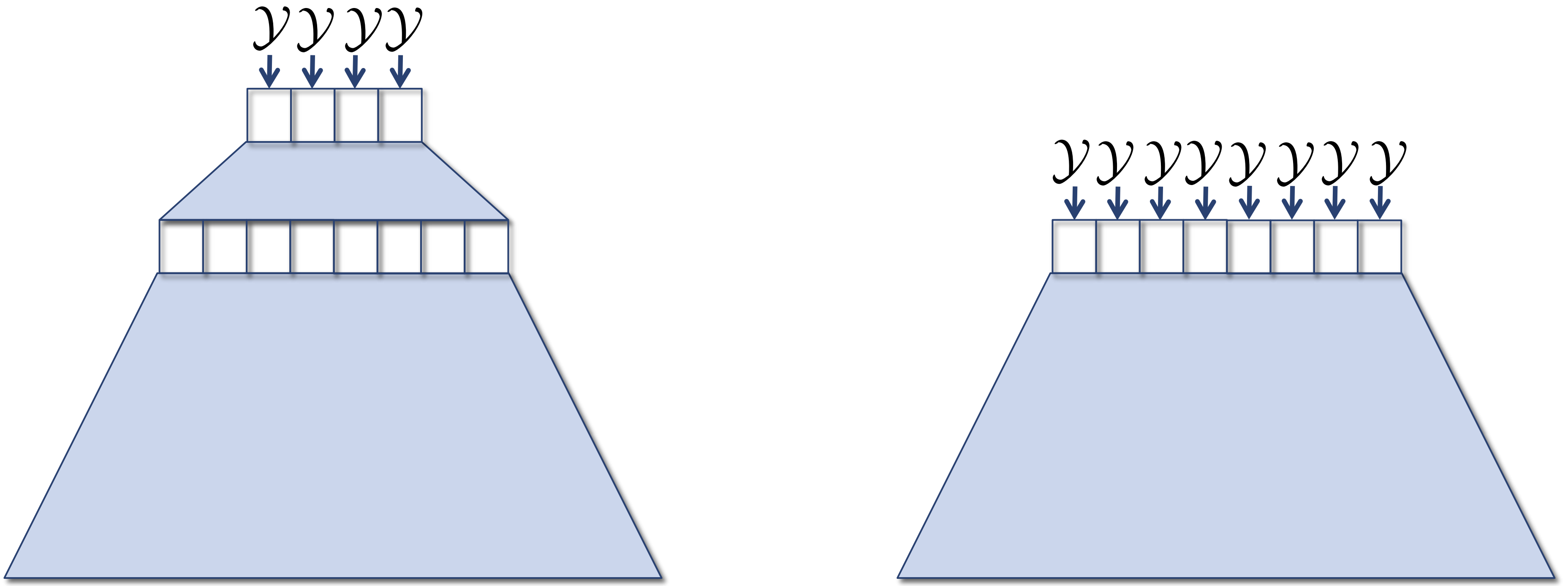
Original Security Proof: Step 1

Hybrid n



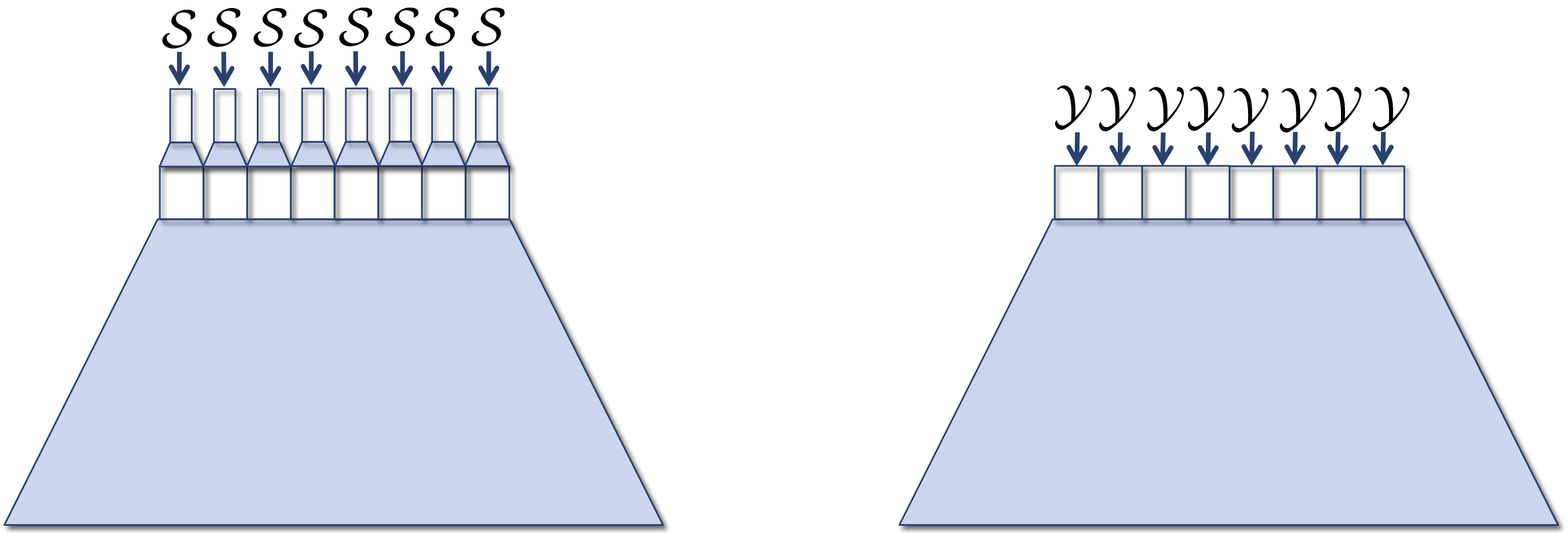
Original Security Proof: Step 1

PRF distinguisher will distinguish two adjacent hybrids



Original Security Proof: Step 1

PRF distinguisher will distinguish two adjacent hybrids

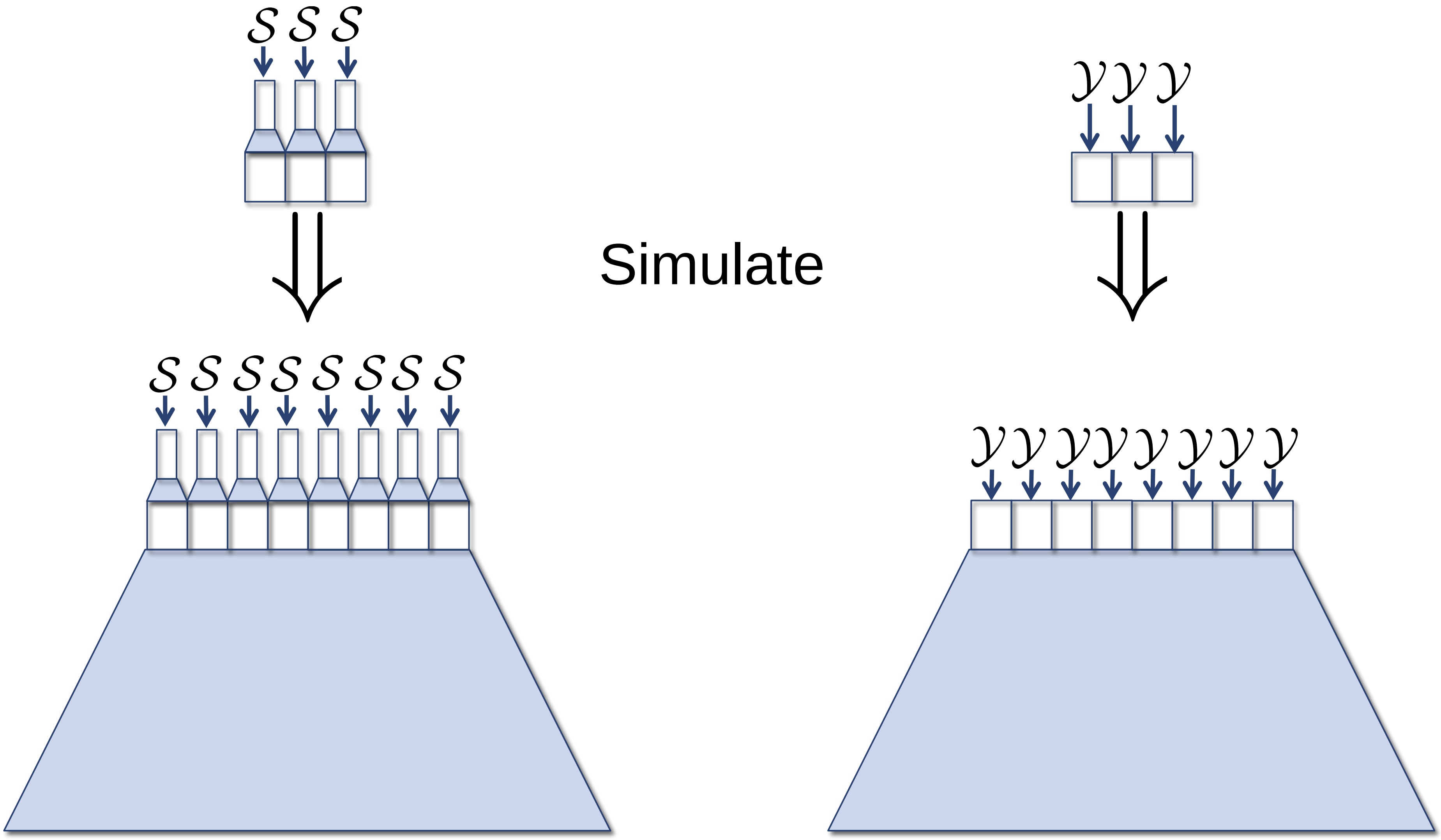


Original Security Proof

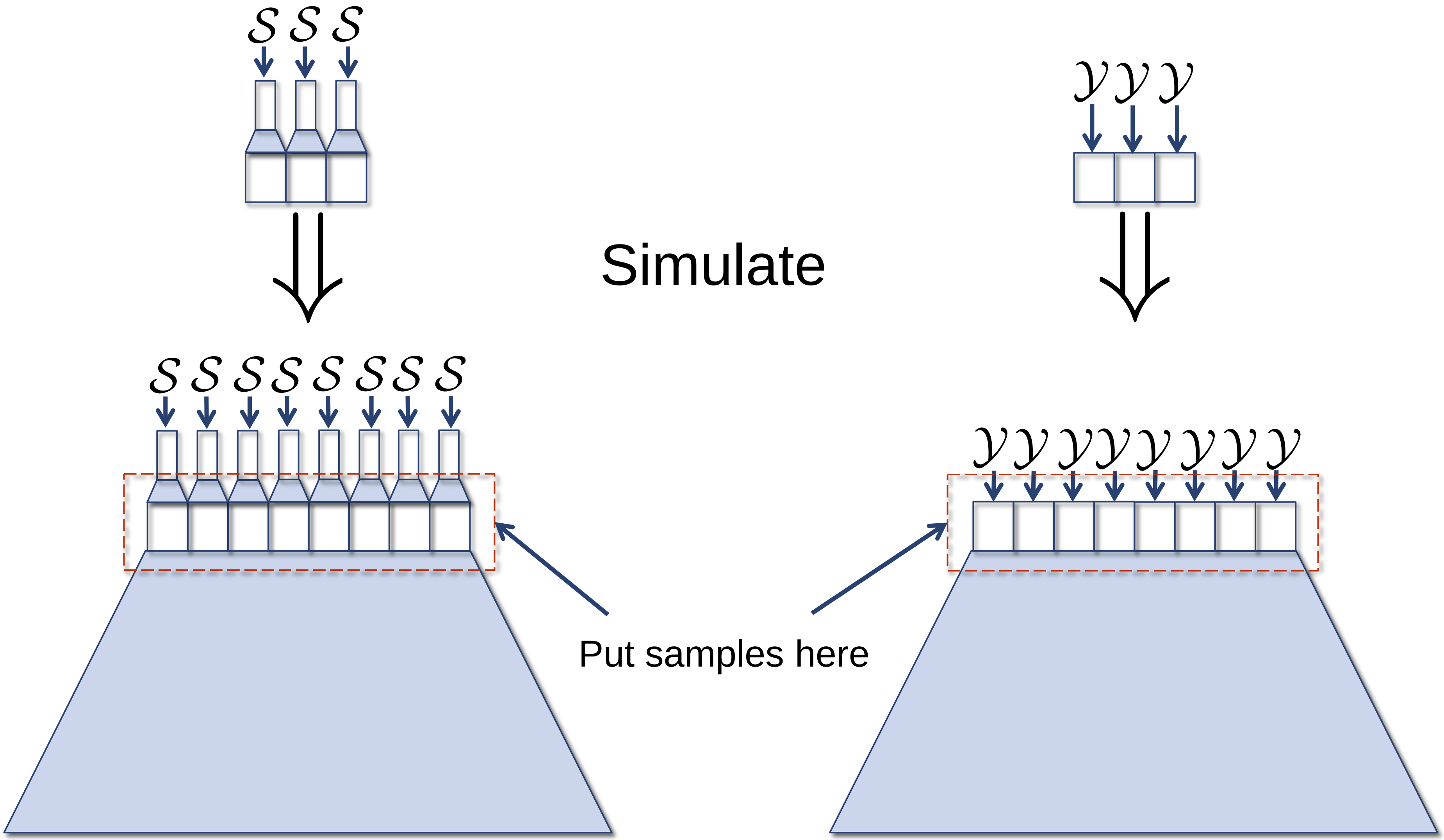
Step 1: Hybridize over levels of tree ✓

Step 2: Simulate hybrids using q samples

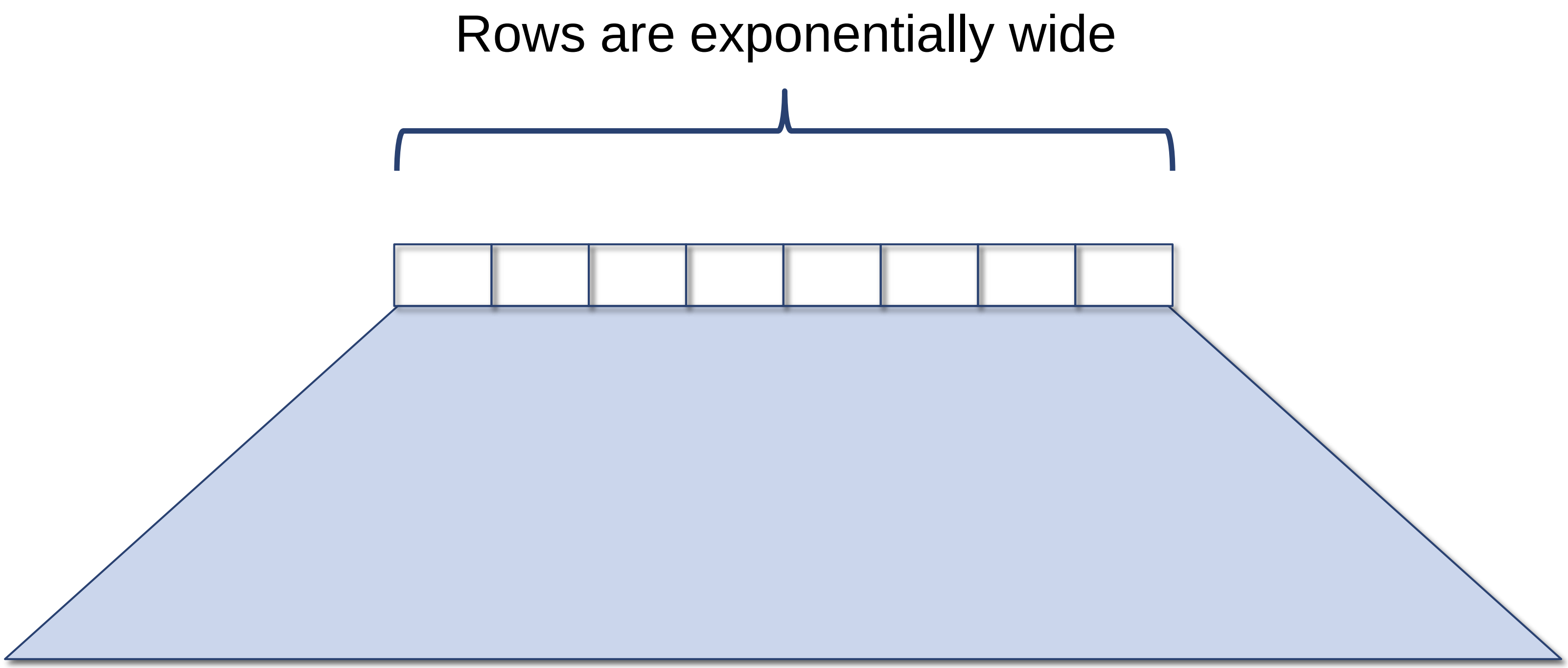
Original Security Proof: Step 2



Original Security Proof: Step 2



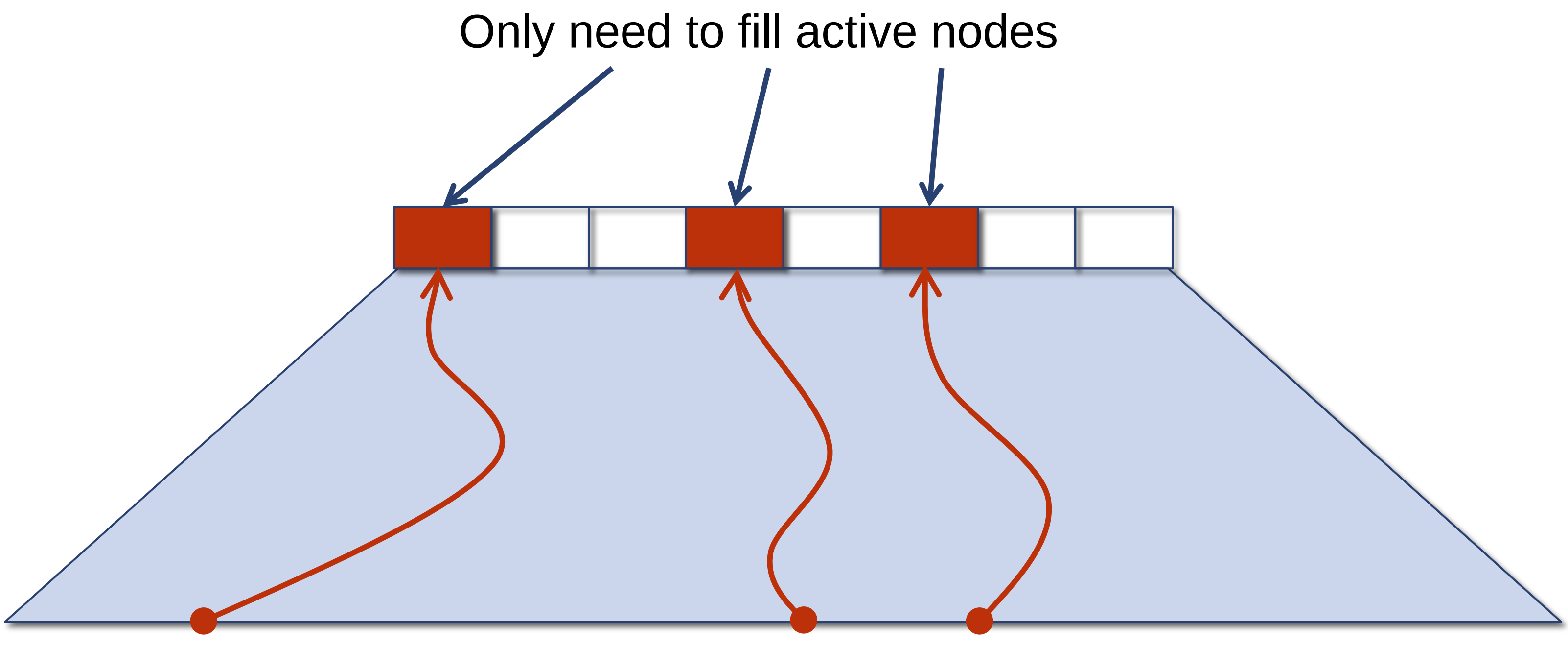
Original Security Proof: Step 2



Problem?

Original Security Proof: Step 2

Active node: value used to answer query



Adversary only queries polynomial number of points

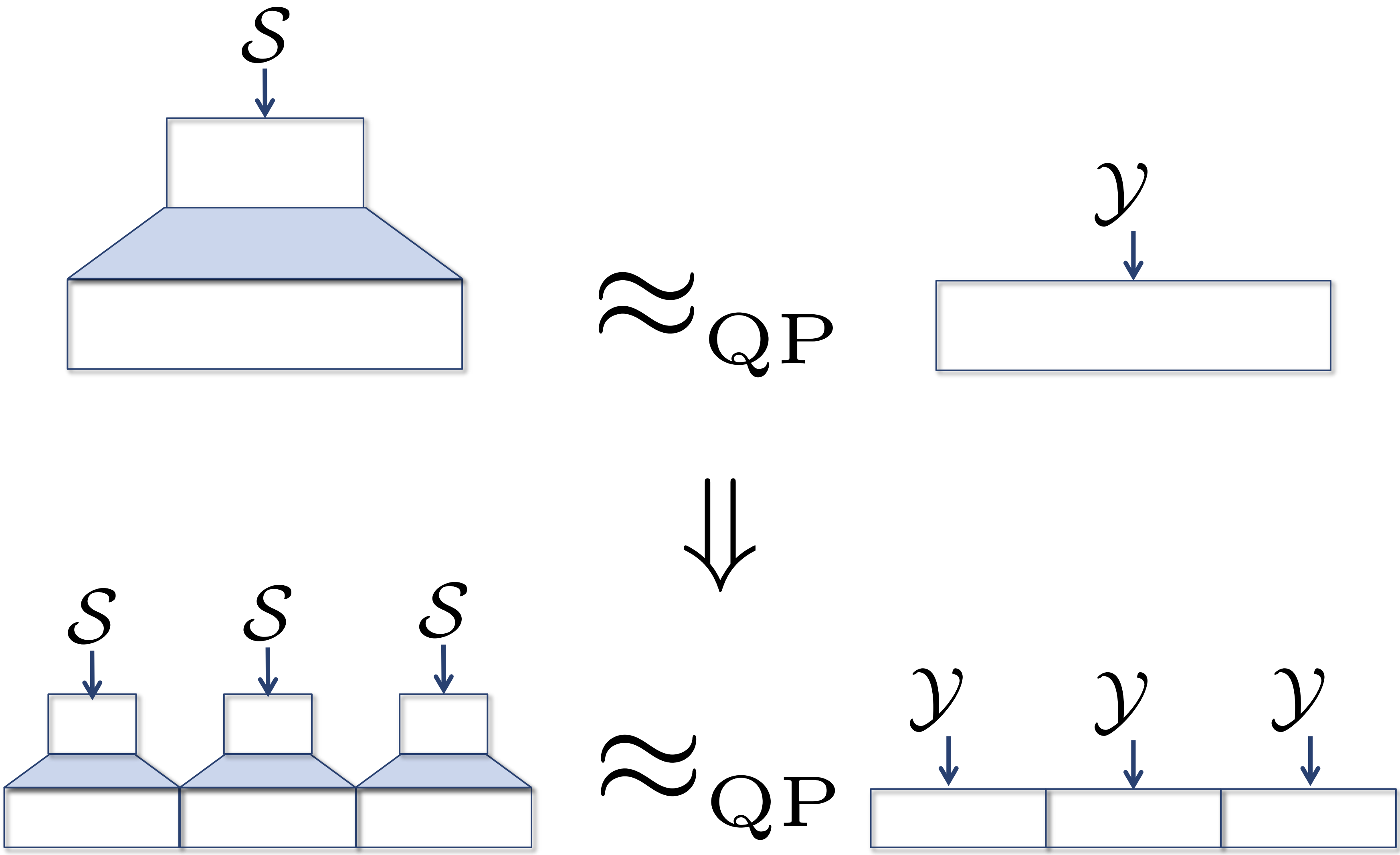
Original Security Proof

Step 1: Hybridize over levels of tree ✓

Step 2: Simulate hybrids using q samples ✓

Step 3: Pseudorandomness of one PRG sample implies pseudorandomness of q samples

Original Security Proof: Step 3



Original Security Proof

Step 1: Hybridize over levels of tree ✓

Step 2: Simulate hybrids using q samples ✓

Step 3: Pseudorandomness of one PRG sample implies pseudorandomness of q samples ✓

Quantum Security Proof Attempt

Step 1: Hybridize over levels of tree ✓

Quantum Security Proof Attempt

Step 1: Hybridize over levels of tree ✓

Step 3: Quantum pseudorandomness of one PRG sample implies quantum pseudorandomness of q samples ✓

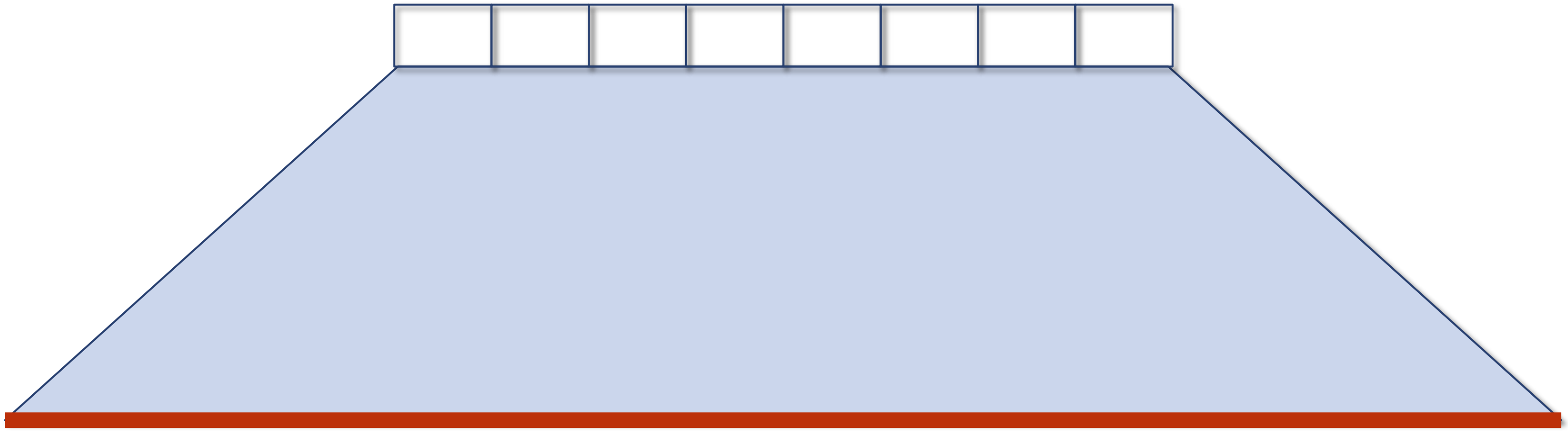
Quantum Security Proof Attempt

Step 1: Hybridize over levels of tree ✓

Step 2: Simulate hybrids using q samples X

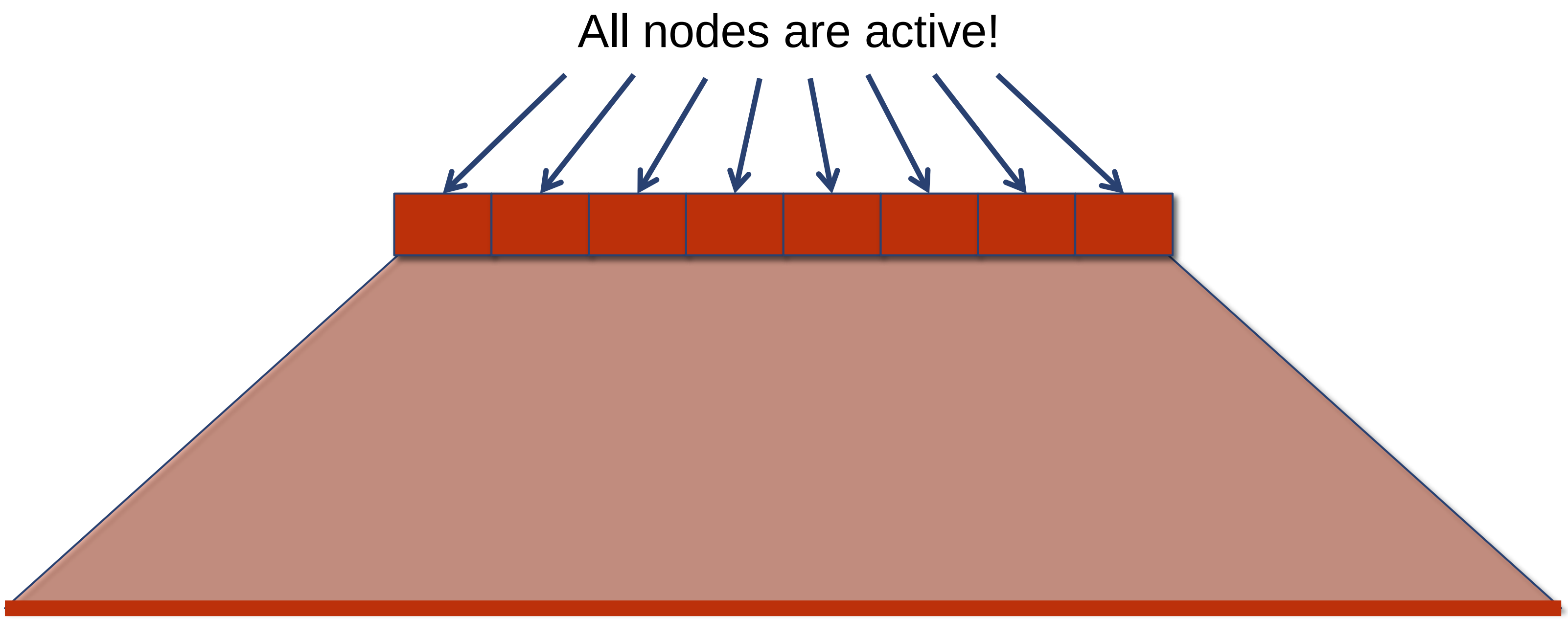
Step 3: Quantum pseudorandomness of one PRG sample implies quantum pseudorandomness of q samples ✓

Difficulty Simulating Hybrids



Adversary can query on all exponentially-many inputs

Difficulty Simulating Hybrids



Exact simulation requires exponentially-many samples

Need new simulation technique

A Distribution to Simulate

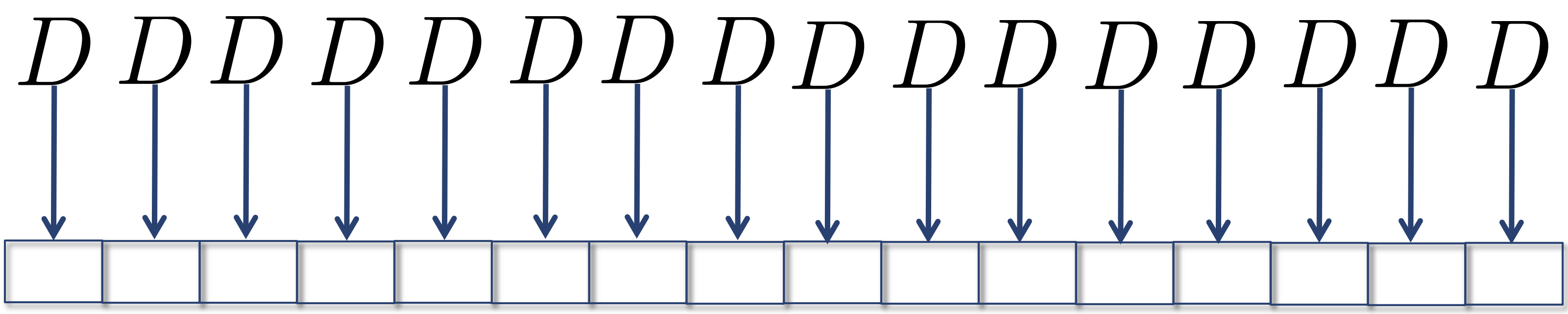
Any distribution D on values induces a distribution on functions

For all $x \in \mathcal{X}$

$$y_x \leftarrow D$$

$$H(x) = y_x$$

H:



$$D^{\mathcal{X}}$$

Main Tool: Small Range Distributions

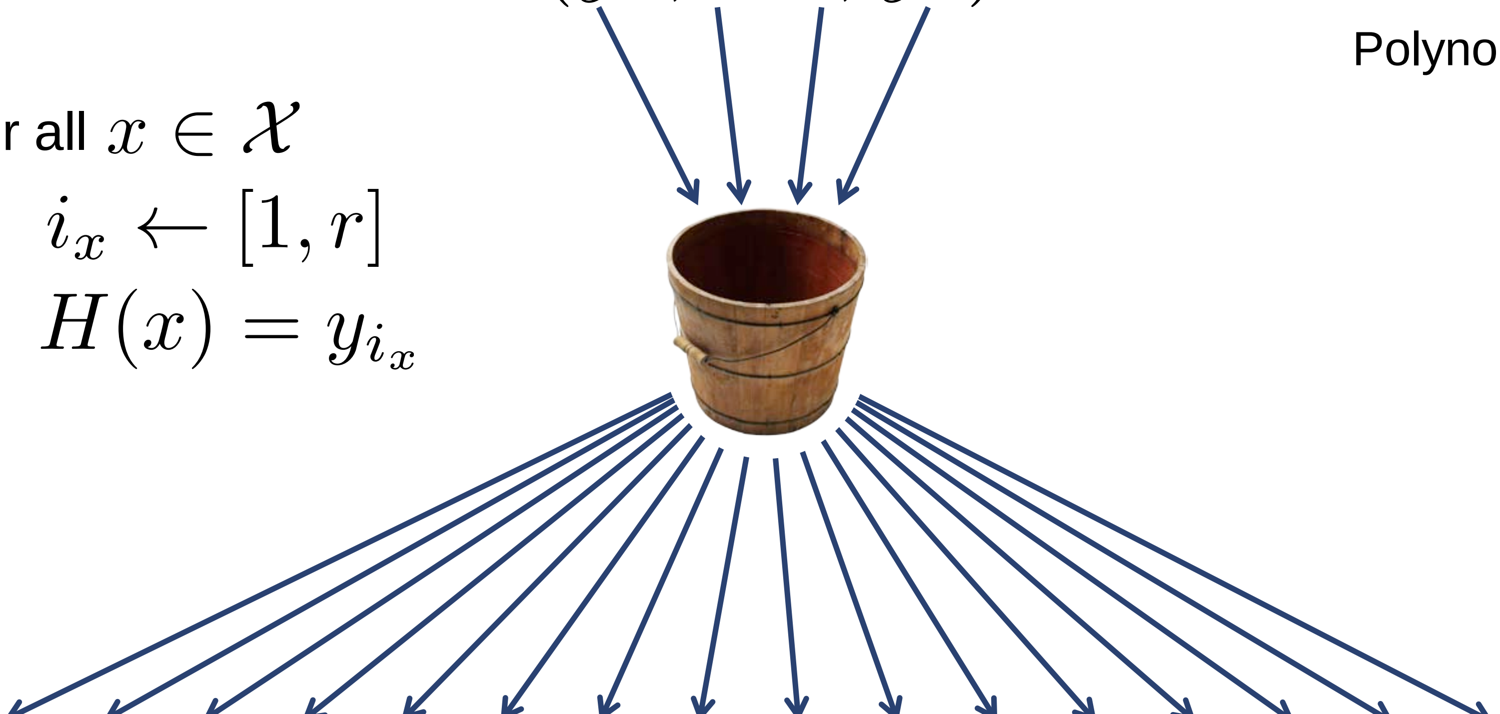
$$(y_1, \ldots, y_r) \leftarrow D^r$$

Polynomial r

For all $x \in \mathcal{X}$

$$i_x \leftarrow [1, r]$$

$$H(x) = y_{i_x}$$

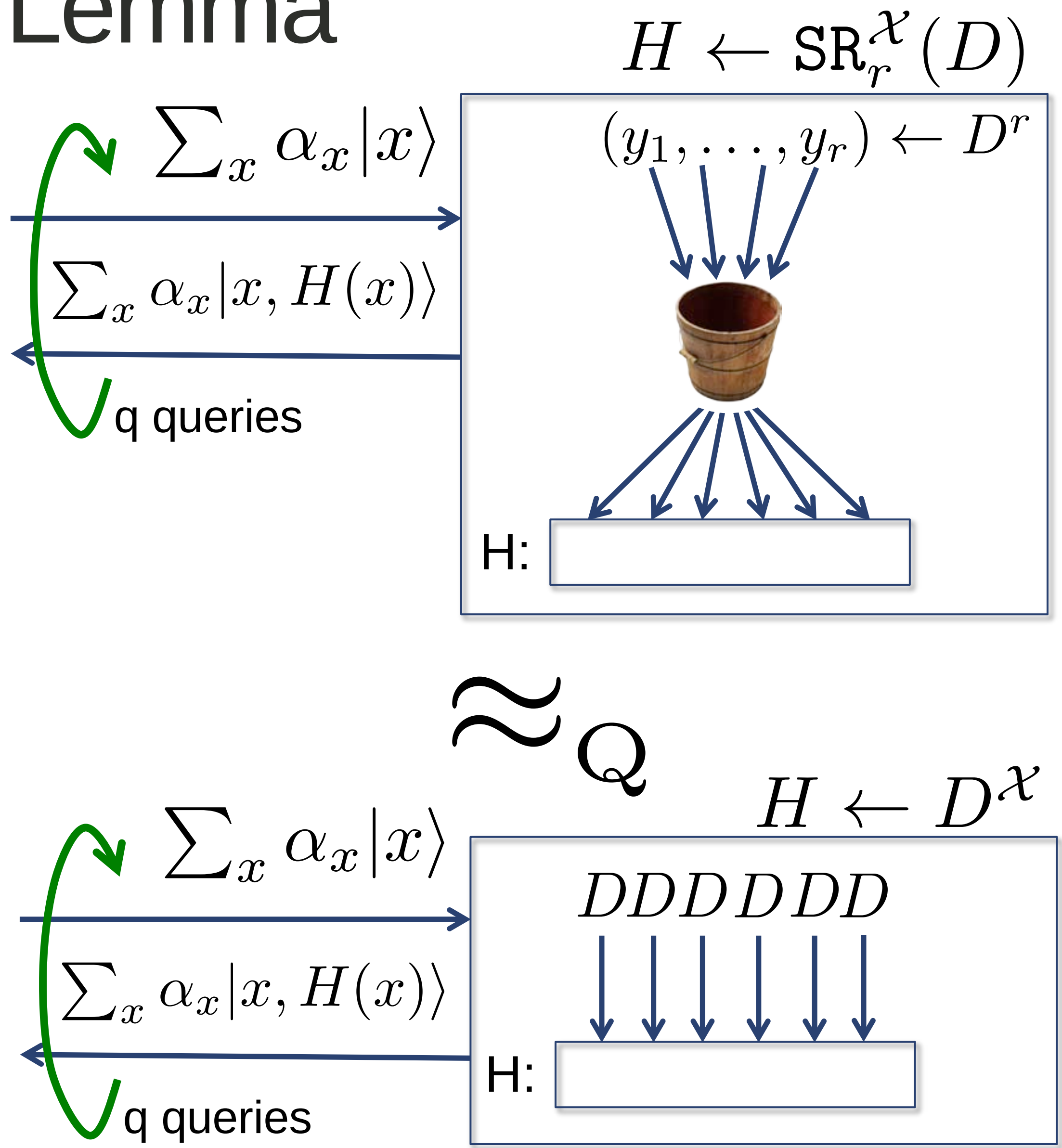


H: y_5 y_2 y_4 y_4 y_1 y_5 y_3 y_3 y_4 y_5 y_2 y_5 y_2 y_3 y_5 y_1

$$\mathrm{SR}_r^{\mathcal{X}}(D)$$

Main Technical Lemma

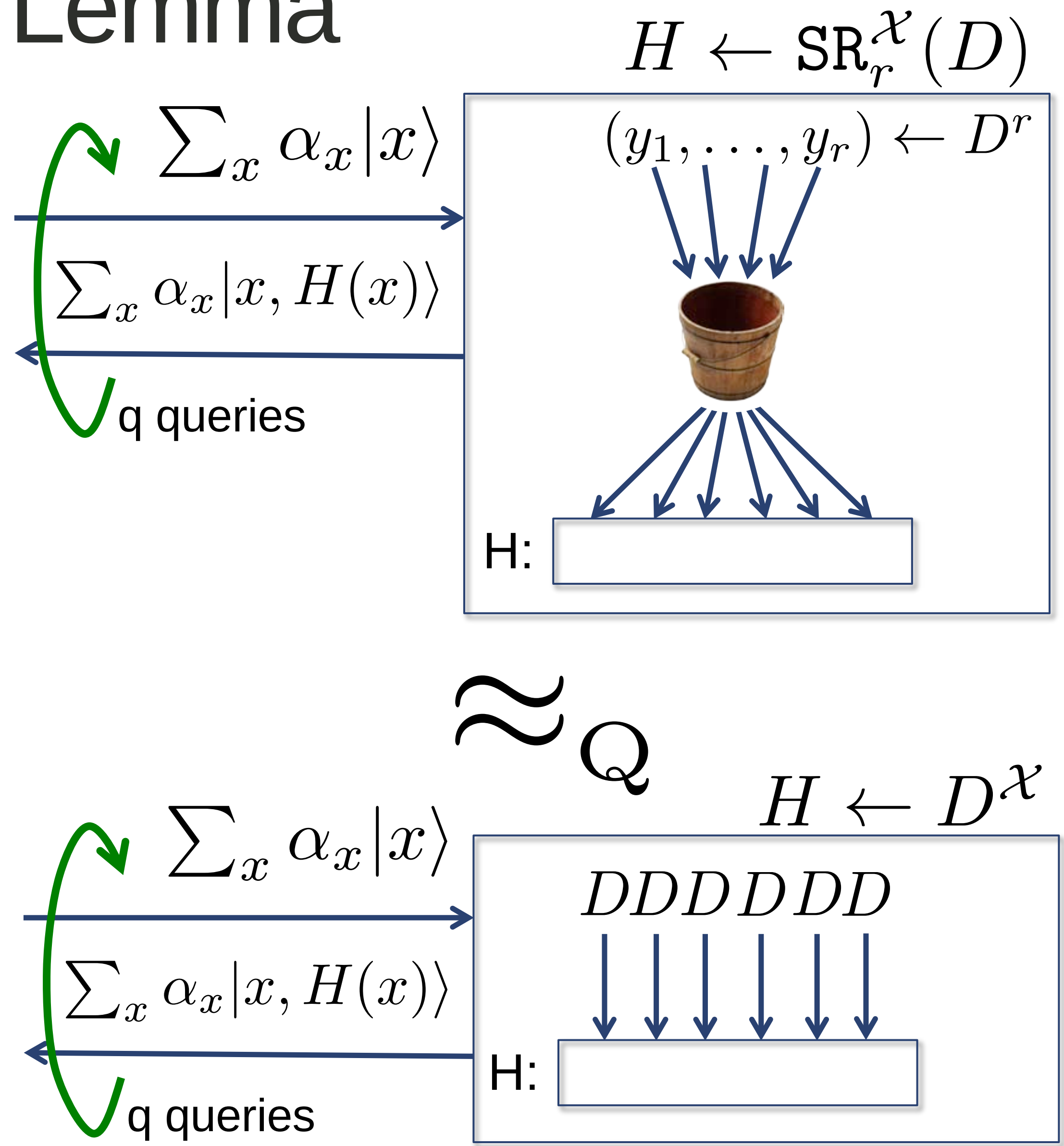
Lemma: $SR_r^X(D)$ is indistinguishable from D^X by any q-query quantum algorithm, except with probability $O(q^3/r)$



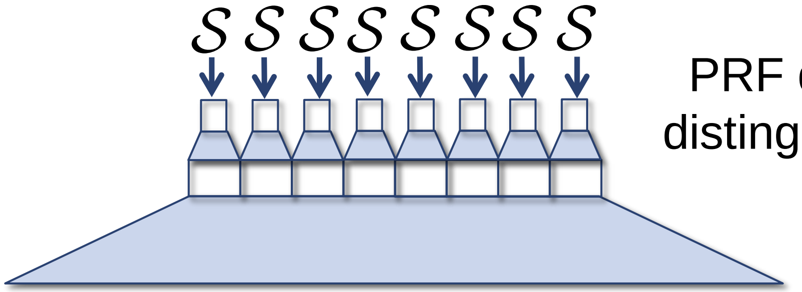
Main Technical Lemma

Lemma: $SR_r^X(D)$ is indistinguishable from D^X by any q-query quantum algorithm, except with probability $O(q^3/r)$

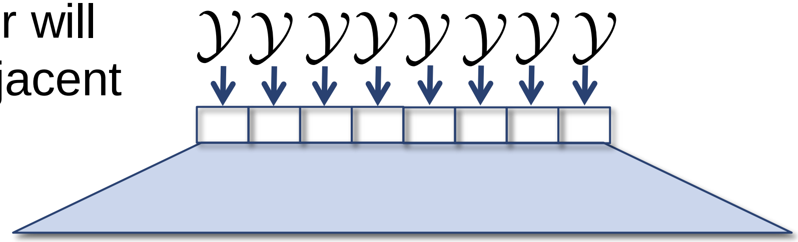
A lot of work!



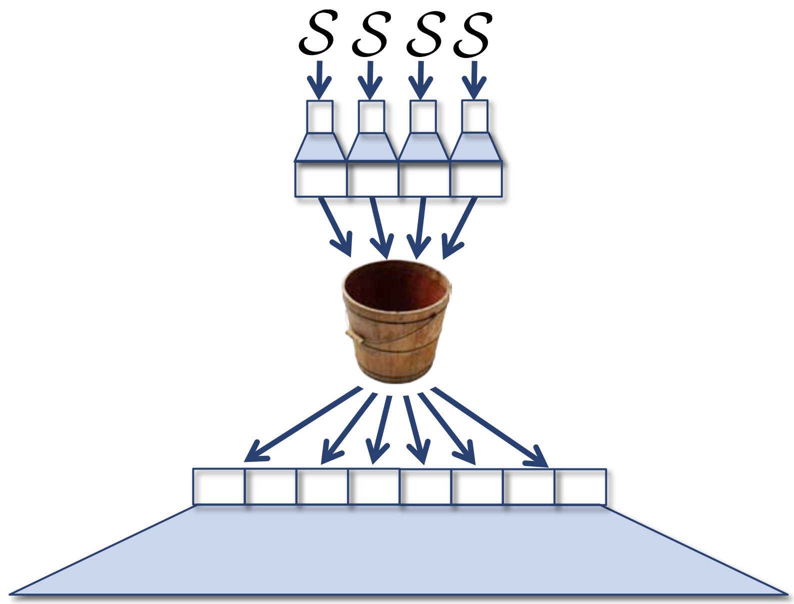
Fixing the GGM Proof

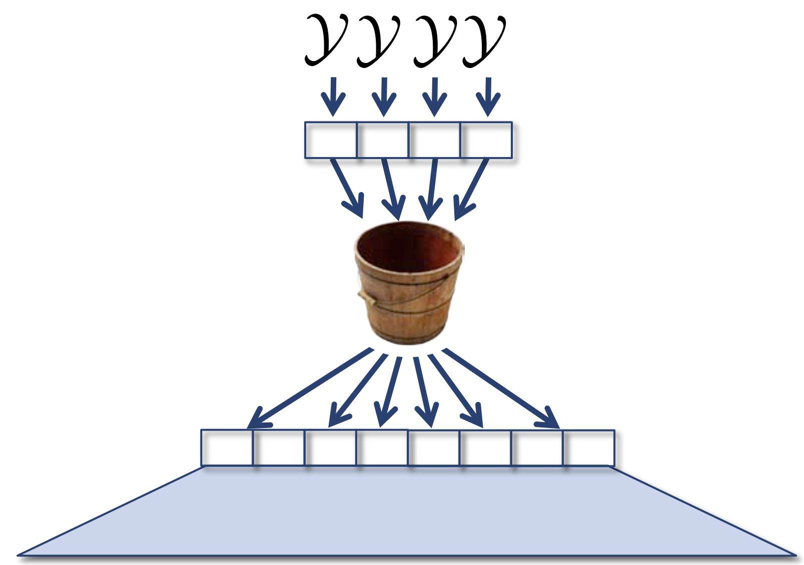


PRF distinguisher will distinguish two adjacent hybrids

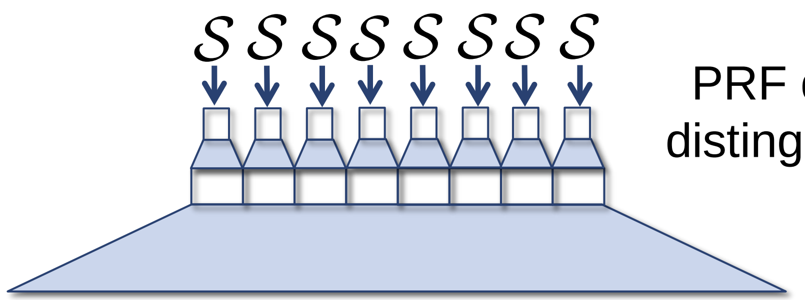


Fixing the GGM Proof

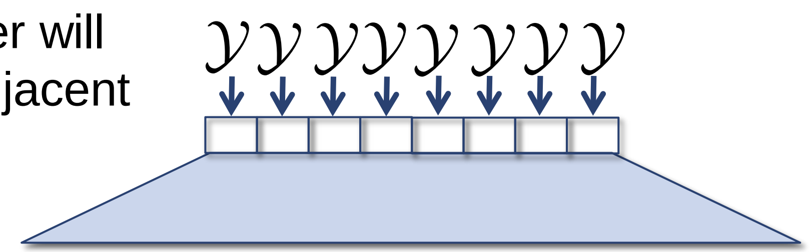




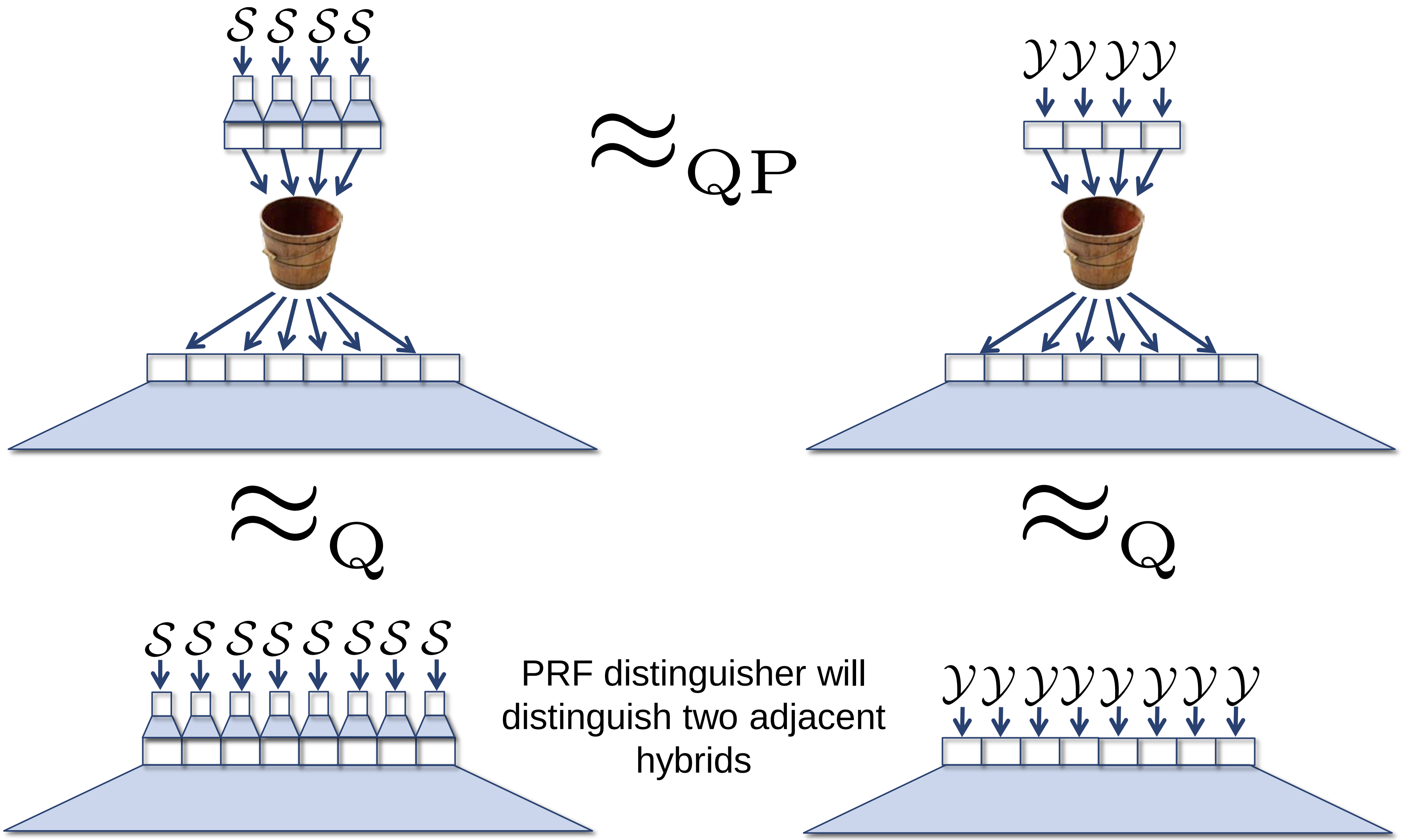
$\approx_Q$ $\approx_Q$



PRF distinguisher will distinguish two adjacent hybrids



Fixing the GGM Proof



Quantum Security Proof

Step 1: Hybridize over levels of tree ✓

Step 2: Simulate hybrids approximately using polynomially-many samples ✓

Step 3: Quantum pseudorandomness of one sample implies quantum pseudorandomness of polynomially-many samples ✓

Summary

Separation: PRFs ≠ QPRFs

We prove security for some classical PRF constructions:

- From quantum-secure pseudorandom generators [GGM'84]
- From quantum-secure pseudorandom synthesizers [NR'95]
- Directly from lattices [BPR'11]

Future Work

Quantum secure PRPs

Other crypto primitives:

- Signatures and MACs under quantum chosen message attacks
- Encryption secure under quantum chosen ciphertext attacks

Future Work

Quantum secure PRPs

Other crypto primitives:

- Signatures and MACs under quantum chosen message attacks
- Encryption secure under quantum chosen ciphertext attacks

Thank you!