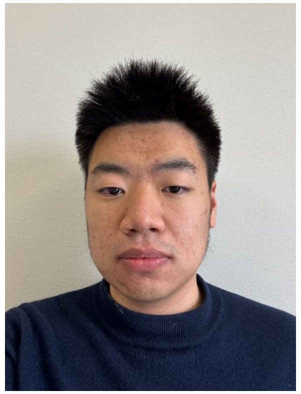
# Succinct Non-Interactive Arguments of Proximity

Liyan Chen

Tsinghua University → MIT

Zhengzhong Jin

Northeastern

Daniel Wichs

Northeastern and NTT Research

# Proving Properties of HUGE Objects

Claim: <5% fake accounts

How to **prove** it?

# **S**uccinct **N**on-interactive **ARG**uments (SNARGs)

“$x \in L$”

CRS

CRS

$\pi$

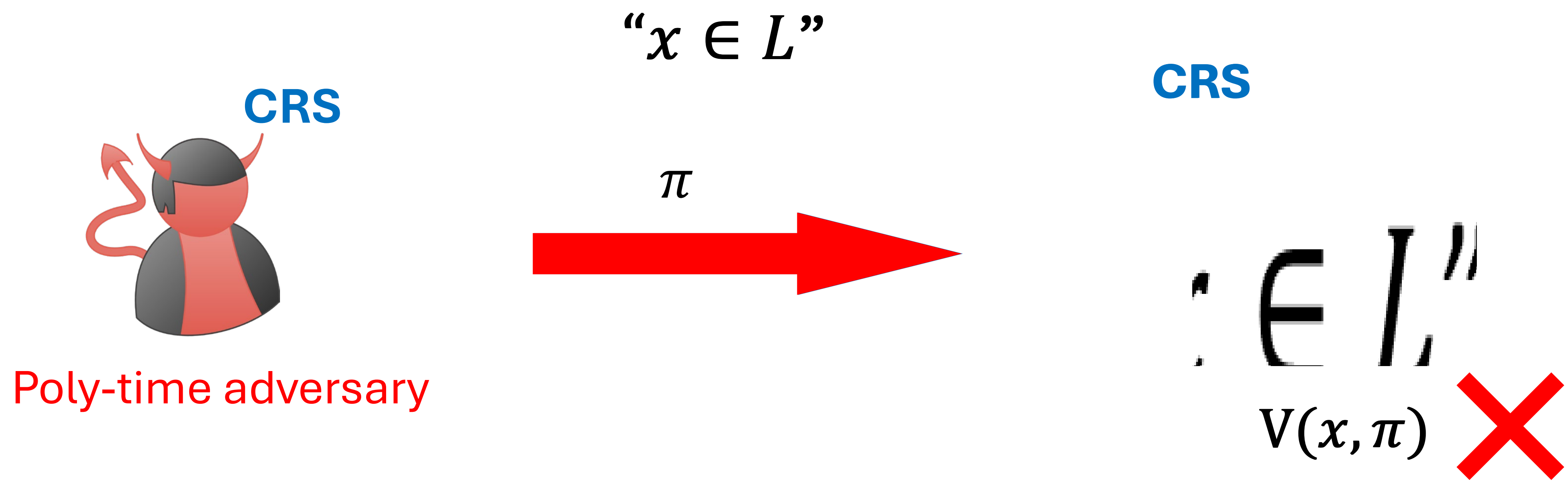
$\in L$”

$\mathrm{P}(x, w)$

$\mathrm{V}(x, \pi)$

- **Completeness:** $\forall x \in L$, the honestly generated proof is accepted.

# <u>**S**</u>uccinct <u>**N**</u>on-interactive <u>**ARG**</u>uments (SNARGs)

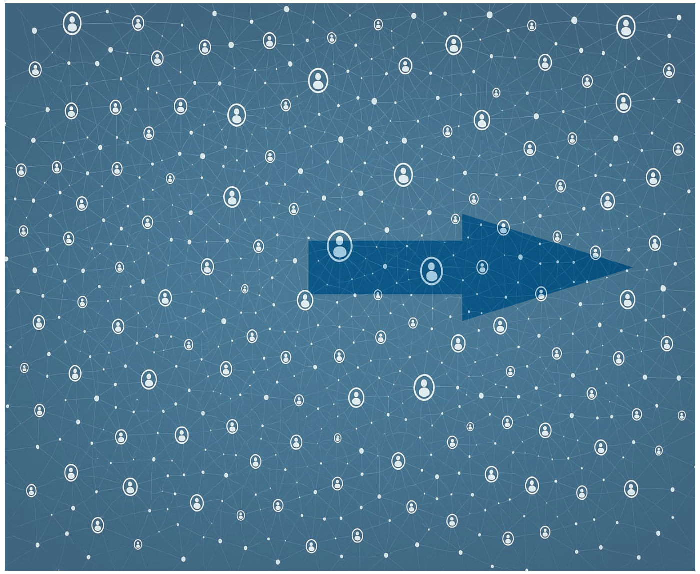



- **Completeness:** $\forall x \in L$, the honestly generated proof is accepted.
- **Soundness:** efficient adversary cannot produce a valid proof $\pi$ for $x \notin L$.
  Soundness can be *selective* or *adaptive*.
- **Succinct:** proof is short: ideally polylog(|x|), verifier efficient: ideally O(|x|)

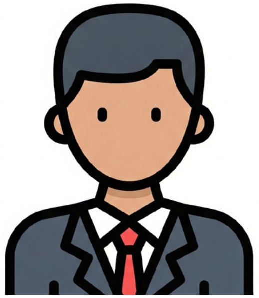
# Can we apply SNARGs?

CRS

$x$



CRS



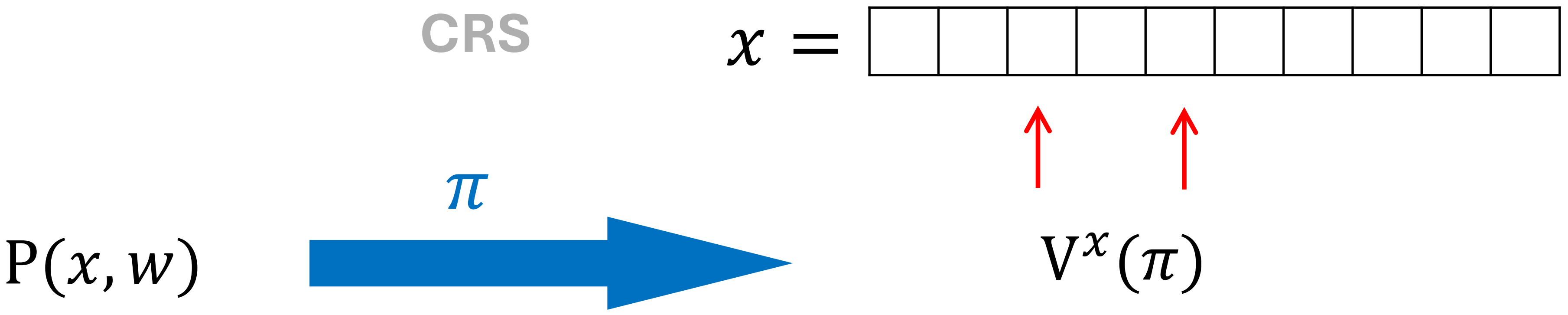
$\mathrm{P}(x, w)$

$\mathrm{V}(x, \pi)$

**Challenge:** The statement $x$ is too large (e.g., a social network graph).
Verifier needs to read the entire statement!

Can we define *succinct*, *non-interactive arguments* with **verification time sublinear** in the instance length?

# Succinct Non-interactive Arguments of Proximity (SNAP)




**Approximate Soundness:**

An efficient adversary cannot produce a valid proof for $x$ that is **$\epsilon$-fraction far** in Hamming distance from any instances in $L$.

- **Verifier efficiency** sublinear in |x|. Bounds proof size, queries.
- **Applications:** Verify social network properties / big data in healthcare / encoded data…
- *Fundamental* on its own: analog of property testing

# Prior Work [Kalai-Rothblum'15]

- Constructed designated-verifier SNAPs for P with selective soundness and verifier efficiency $O(n^{1-\gamma})$ for some $\gamma > 0$.
  - From sub-exp FHE
- Black-Box barrier for proving adaptive soundness of SNAPs for P with verifier efficiency = $o(\sqrt{n})$.
  - Similar to GW11 black-box provability barrier for SNARGs for NP.

# SNAPs 10 years later….

For what parameters and under what assumptions
can we build SNAPs for P or NP
with selective or adaptive soundness?

**Result 1:** Lower Bound on Adaptive SNAP for P

Adaptive SNAPs for P must have verifier time $\Omega(\sqrt{n})$

**Result 2:** Constructing Adaptive SNAPs for P

Adaptive SNAP for P with $\tilde{O}(\sqrt{n})$ verifier time from LWE / DLIN/ QR+DDH /...

*How about NP?*

## **Result 3:** Constructing Adaptive SNAPs for NP

Adaptive **SNAP** for P + Adaptive **SNARG** for NP $\Rightarrow$ Adaptive **SNAP** for NP

Get adaptive SNAP for NP with $\tilde{O}(\sqrt{n})$ verification time from

iO + (LWE/ QR+DDH / DLIN/...)

*Can we build non-adaptive SNAPs for P or NP with better than $\tilde{O}(\sqrt{n})$ verification time?*

**Result 4:** Constructing Non-adaptive SNAPs for NP

Non-adaptive SNAP for NP with polylog verification time

from sub-exp iO + sub-exp OWF + LWE.

*Can we do it under better assumptions for P?*

**Result 5:** Lower bound on Non-adaptive SNAPs for P

Any non-adaptive SNAP for P with verification time = $o(\sqrt{n})$

implies a (non-trivial) non-adaptive SNARG for NP.

# Summary of Our Results

| | Adaptive | Non-adaptive |
|---|---|---|
| P | SNAPs with $O(\sqrt{n})$-efficiency without iO<br><br>Unconditional $\Omega(\sqrt{n})$ lower bound | Breaking $O(\sqrt{n})$-bound implies SNARGs for NP |
| NP | SNAPs with $O(\sqrt{n})$-efficiency from iO | Fully succinct SNAPs from iO. |

# Key Difficulty of Adaptive SNAPs

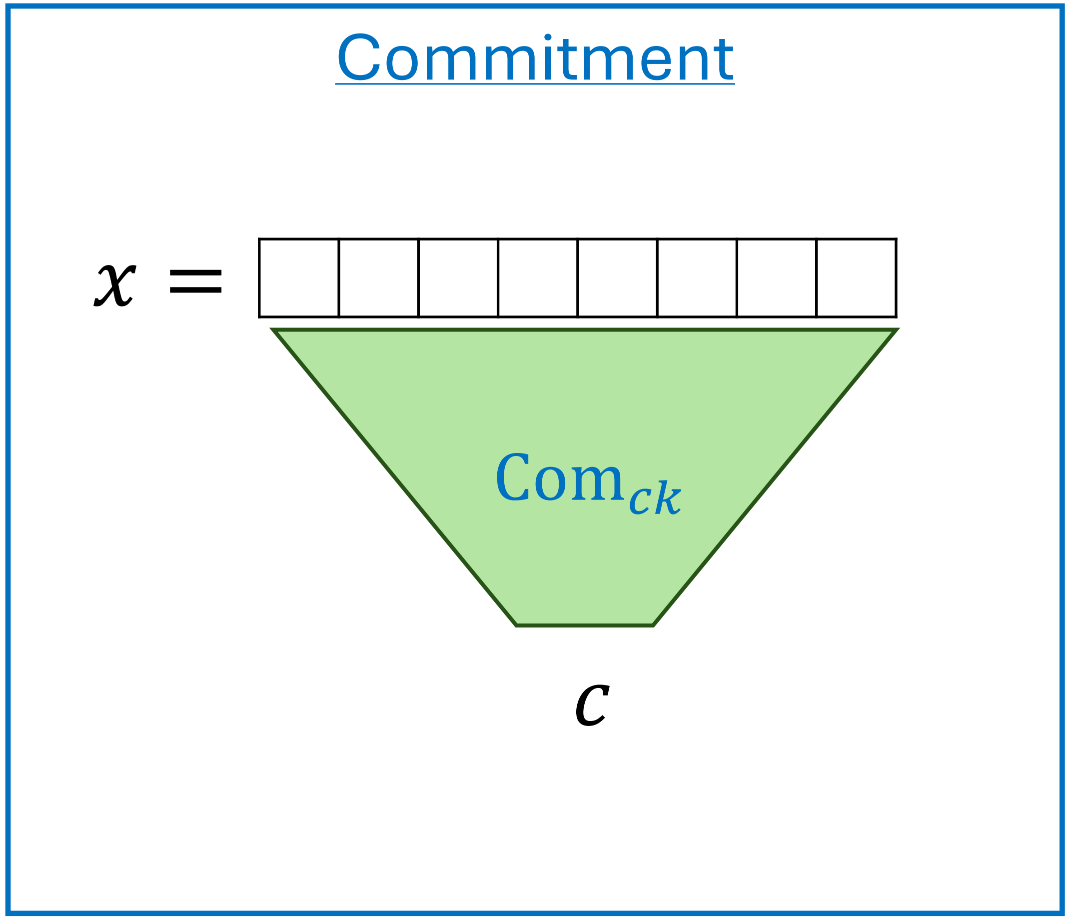
**Generic Attack:**

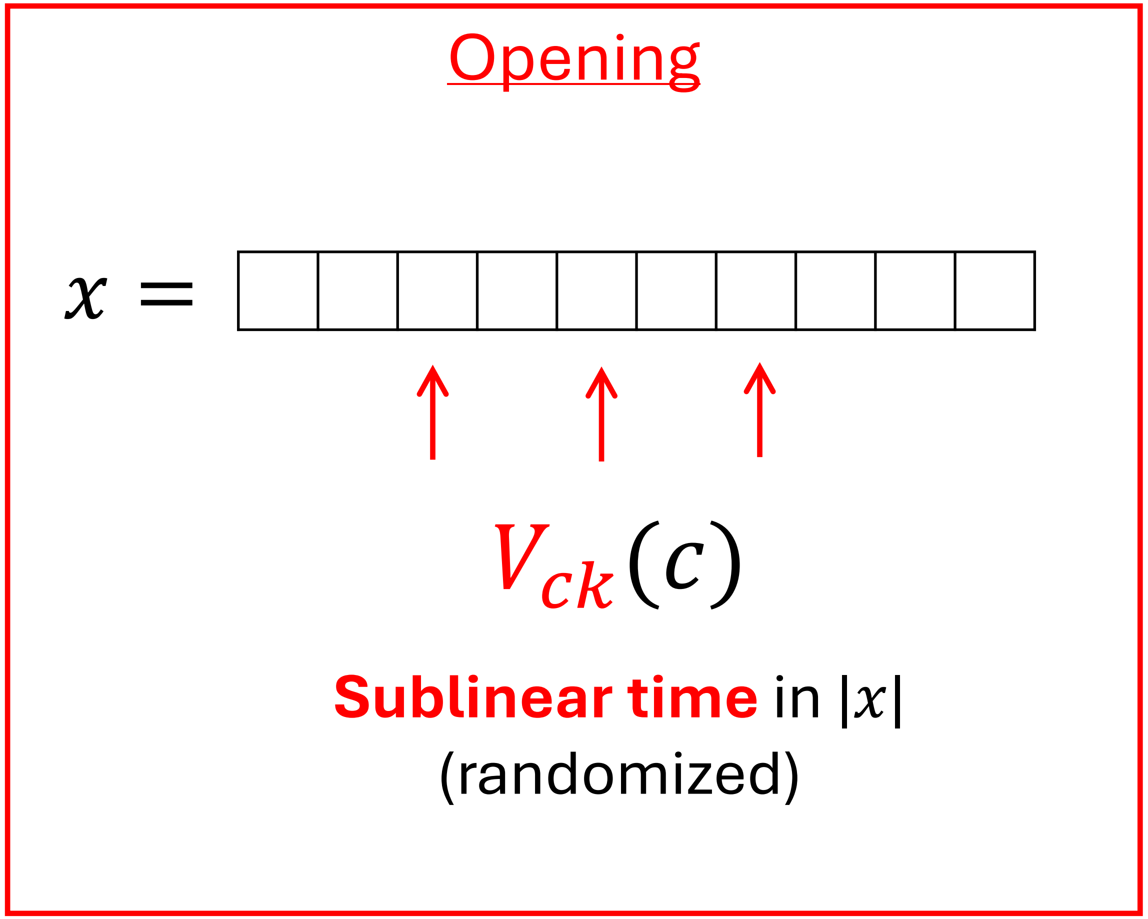
- Generate honest proof $\pi$ for true statement $x^*$.
- See which positions of $x^*$ are queried by $Ver^{x^*}(\pi)$.
- Change $x^*$ to a false $x$ by modifying any other position.
- Ensures that $Ver^{x}(\pi)$= $Ver^{x^*}(\pi)$= *accept*.

**Preventing the Attack:**

- Allow randomized verification!
- Queried locations are independent of the proof. Useful?

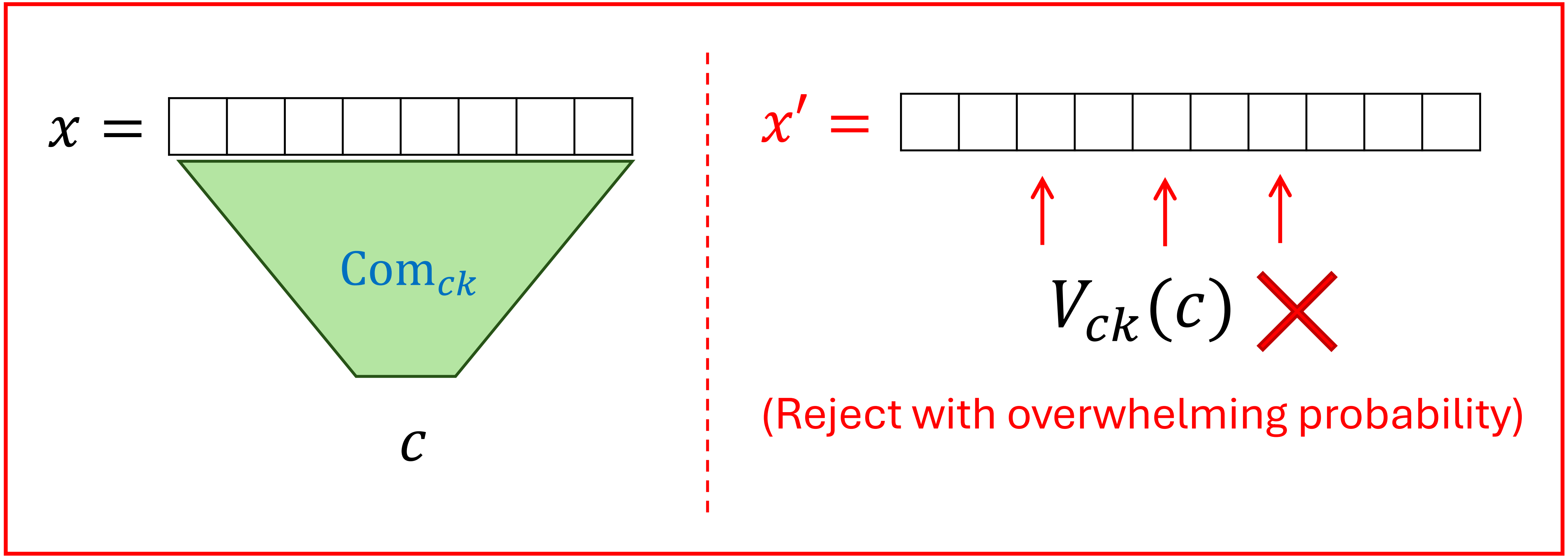
# Binding? Impossible due to Sublinear Verification



# Binding of Proximity

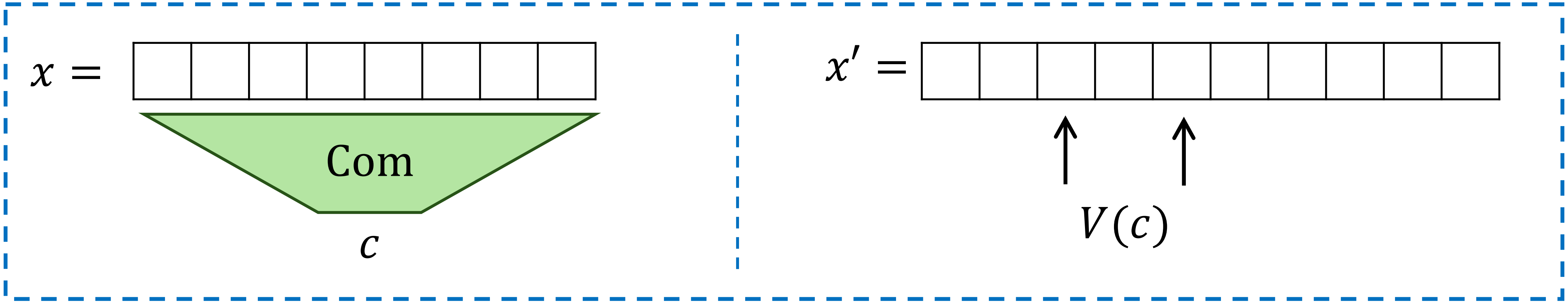
For any  $\rightarrow (x, x')$ with $\Delta(x, x') \geq k$ ($k$: a parameter)



## Near-Optimal Commitment of Proximity

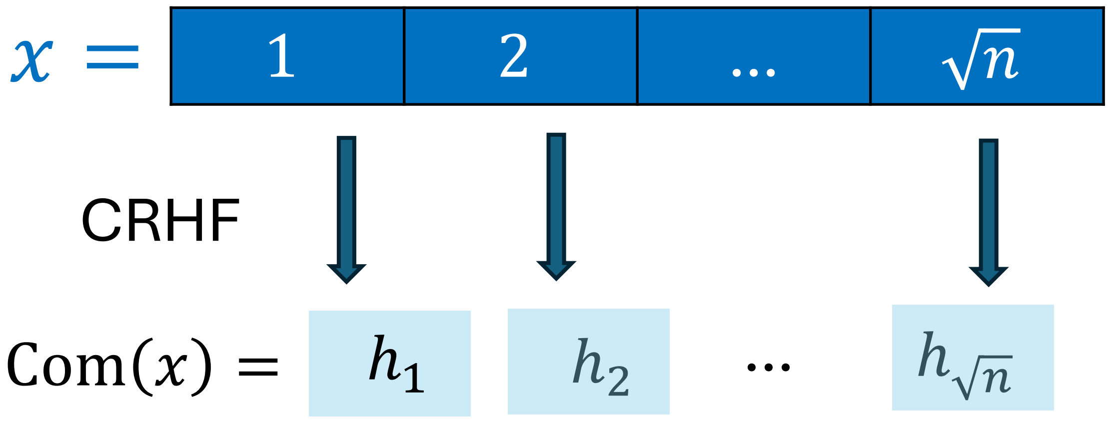
Assuming collision-resistant hash functions (CRHF), get commitment of proximity:

- Commitment size $\tilde{O}(\sqrt{n})$
- Verifier's query complexity $\tilde{O}(\sqrt{n})$
- Binding of Proximity: $\Delta(x, x') \leq \sqrt{n}$



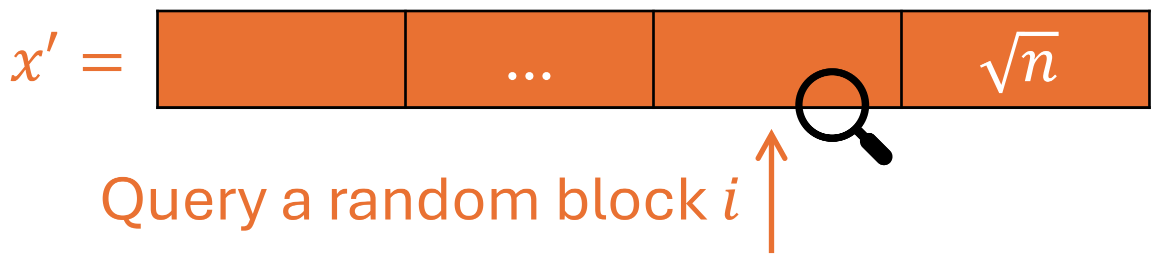
# A Naïve Construction: Divide-and-Hash

## Commitment




Length: $O(\sqrt{n})$

## Opening ($V$)




Recompute $i$-th hash, check if it $= h_i$

(repeat for $1/\epsilon$ times)

#queries: $\tilde{O}(\sqrt{n})$

**Binding of Proximity:** $V$ accepts $\Rightarrow$ # of different blocks $\leq \epsilon$-fraction

$\Rightarrow \Delta(x, x') \leq \epsilon \cdot n$

Improve?

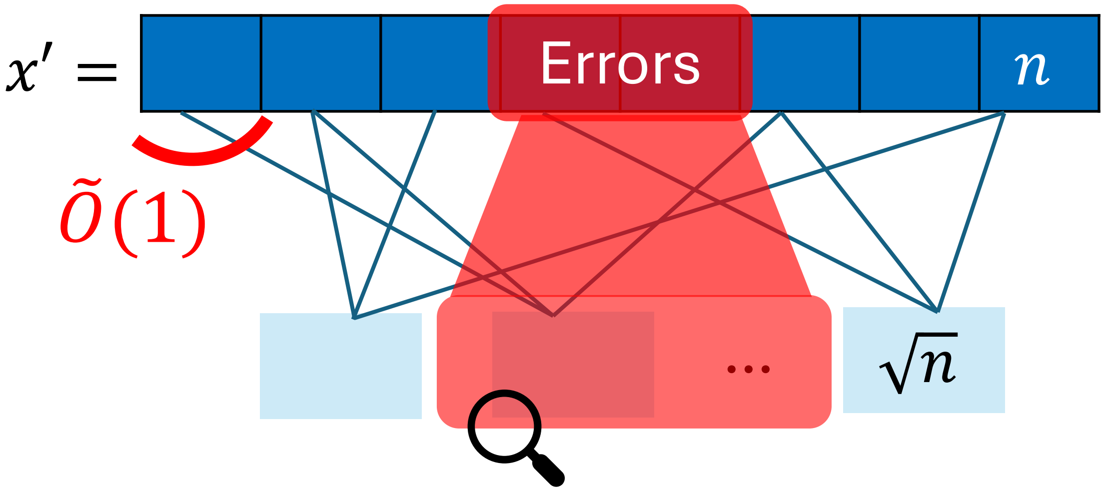
# Near-Optimality via Expander Graphs

## Commitment




Assignments are given by an expander of **O(1)-factor expansion**

## Opening ($V$)




If any neighbor of the errors is checked, then Verifier rejects.

Expansion property $\Rightarrow$ $\sqrt{n}$ Hamming errors have $\tilde{O}(\sqrt{n})$ neighbors

$\Rightarrow \Delta(x, x') \leq \sqrt{n}$

# Commitment of Proximity ⇒ SNAP (1st attempt)

**Prover**

- Compute $c = \mathrm{Com}(x)$
- Compute $\pi_{SNARK}$ for:

$$\exists x \in L: \mathrm{Com}(x) = c$$

Proof: $c, \pi_{SNARK}$ →

**Verifier**

Verify $V^{x'}(c) = 1$ and $\pi_{SNARK}$

Can we replace **SNRAKs for NP** with standard assumptions (**SNARGs for P**)?

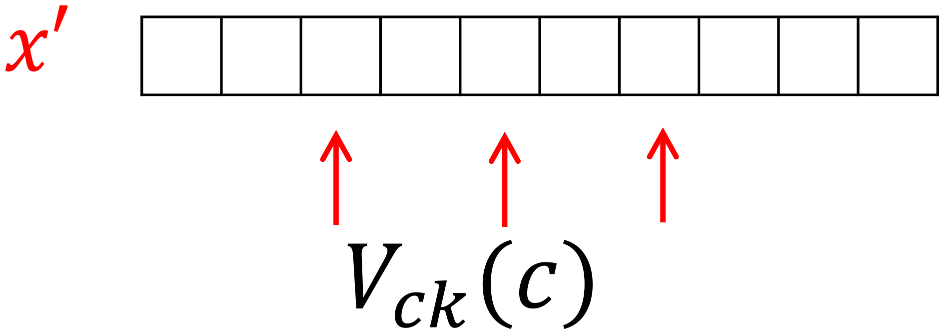
# Extractable Commitment of Proximity

Basic:

For any  → x,x' if $V_{ck}^{x'}(Com(x))$=1 then $\Delta(x, x') \leq \sqrt{n}$

Extractable:

For any  → c,x' if $V_{ck}^{x'}(c)$=1 then can extract $x$ s.t. $\Delta(x, x') \leq \sqrt{n}$ and $c = Com(x)$.



# Extractable Commitment of Proximity ⇒ SNAP

**Prover**

- Compute $c = \mathrm{Com}(x)$
- Compute $\pi_{SNARG}$ for:

$$\exists x \in L: \mathrm{Com}(x) = c$$

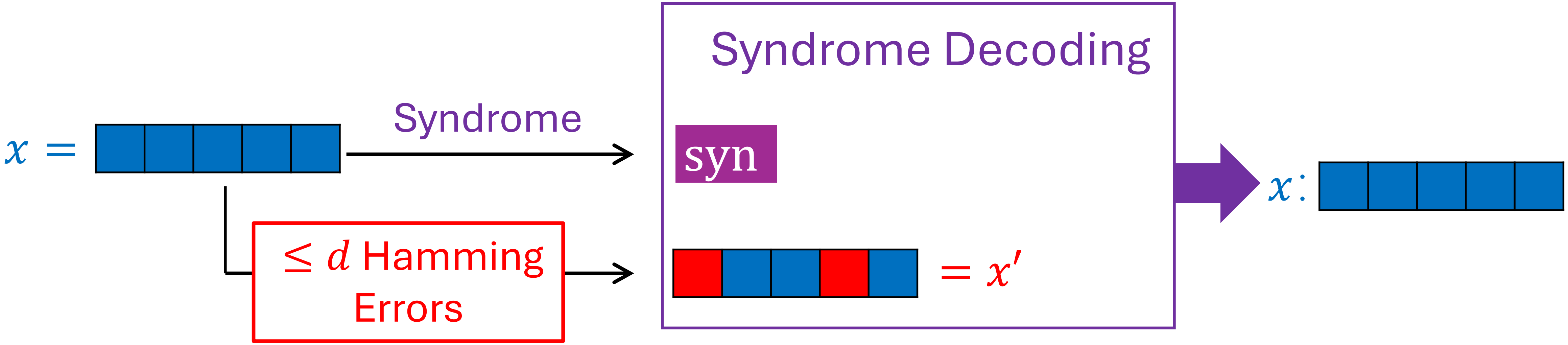
Proof: $c, \pi_{SNARG}$ →

**Verifier**

Verify $V^{x'}(c) = 1$ and $\pi_{SNARG}$

How to construct extractable commitment of proximity?

# Recall: Syndrome Decoding




- **Correctness**: $\mathrm{Decode}(\mathrm{syn}, x') = x$ as long as $\Delta(x, x') \leq d$
- **Succinctness**: $|\mathrm{syn}| \leq \tilde{O}(d)$.

# Basic ⇒ Extractable CoP (1st attempt)

$x' =$ [ | | | | | | ]

Verifier $V^{x'}(c)$

Committer

$c := \mathrm{Com}(x)$, s := syn(x)

$c, s, \pi$

$\pi$: RAM SNARG proof for

$\exists x{:}\ \mathrm{Com}(x) = c \wedge syn(x) = s$

Circularity!

- **Extraction:** $x = \mathrm{Decode}(s, x')$
- If commitment to x is honestly generated then extractor will output x.
- How to extract from general commitment?

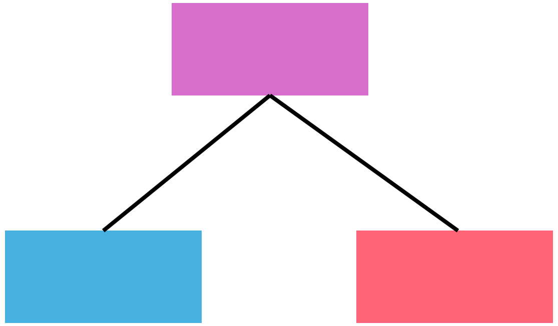
# Basic ⇒ Extractable CoP (SNARG gymnastics)

- Rely on syndromes + somewhere extractable hashing + RAM SNARGs.

- Extraction in parts:
  - Make hash extractable on *different parts* $x_i$ of $x$ in different hybrids.
  - Use RAM SNARGs to argue that basic CoP + syndrome computed correctly for $x_i$.
  - Extract $x_i$ from verifier's input $x'$ and the syndrome. Extraction has to remain correct in subsequent hybrids.

# Recall: Somewhere Extractable Hash

Hash:



- **Key Indistinguishability:**

  $k(\text{'}L\text{'}) \approx_c k(\text{'}R\text{'})$

- **Extractable:**

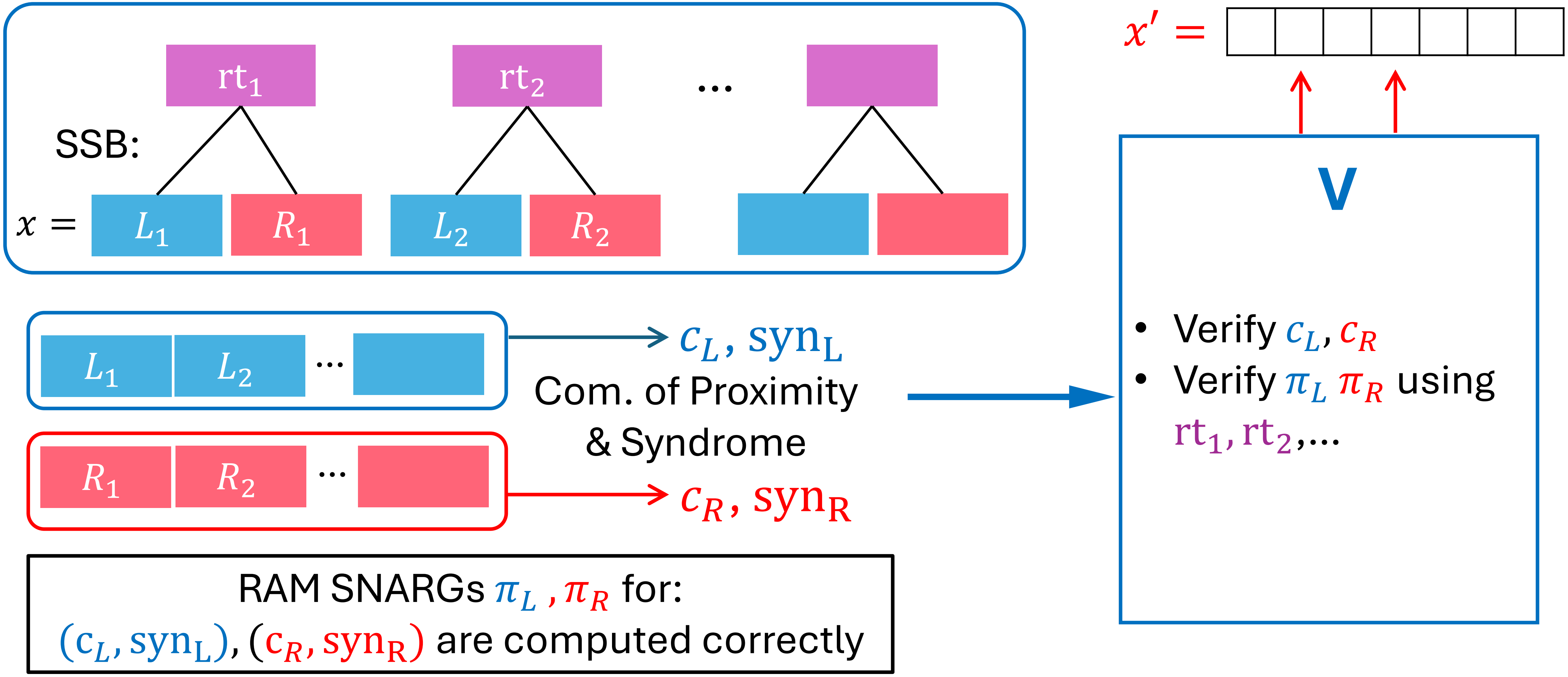
  Extract(td,  ) → 

  Under key $k(\text{'}R\text{'})$

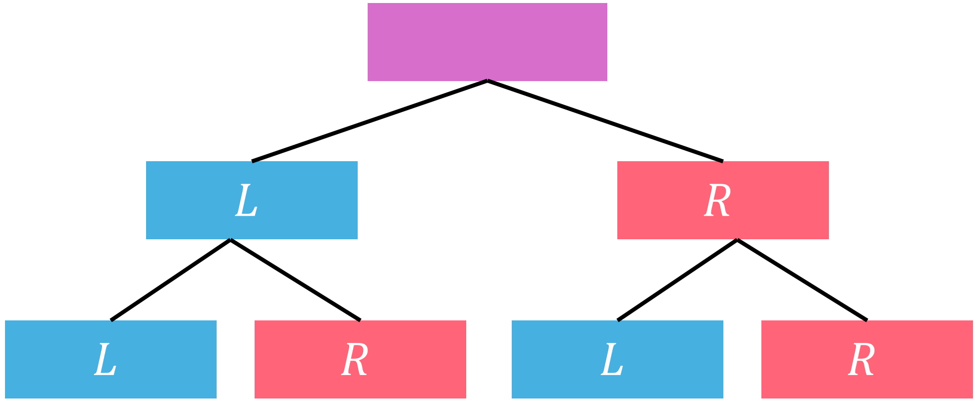
- **Rate-1:**

  |Hash value| ≈ |one child|

# Non-trivial Extractable CoP via 2 Layer Merkle Tree



# Generalize to Full Merkle Tree




Apply Commitment of Proximity & Syndrome to each layer
(left children & right children separately)

**V**

For each layer:

- Verify Com
- Verify RAM SNARG

RAM SNARG proof at each layer:
"syndromes are computed correctly"
(Leaf layer: prove $x \in L$)

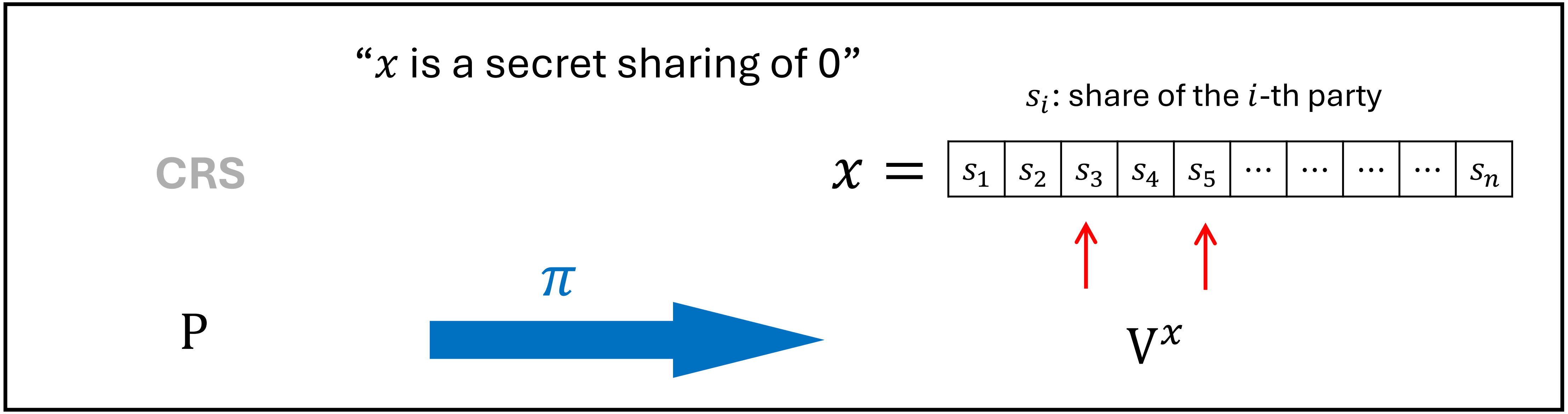
*Soundness Proof:*
Recursively extract layer-by-layer

# Summary of Our Results

| | Adaptive | Selective |
|---|---|---|
| P | SNAPs with $O(\sqrt{n})$ verification from LWE/DDH/... <br> Unconditional lower bound | Breaking $O(\sqrt{n})$-bound implies SNARGs for NP |
| NP | SNAPs with $O(\sqrt{n})$-proof size & query complexity from iO + LWE/DDH/... | Fully succinct SNAPs from iO + LWE. |

# Hard Language of SNAPs

Secret Sharing!




Attack: sample $x$ as a secret sharing of 1

Issue: How to handle $\pi$?

# Attack Strategy

| | |
|---|---|
| Property Testing: | $\{x \leftarrow SS_0\} \approx \{x^* \leftarrow SS_1\}$ |
| SNAPs: | $\{(x,\pi): x \leftarrow SS_0, \pi \leftarrow P\} \approx \{(x^*,\pi^*): x^* \in SS_1\}$ |

...against ***query-bounded*** adversary

Strategy: Choose $(x,\pi)$. Set $\pi^* = \pi$.
Flip some bits of $x$ to get $x^*$.

Analyzed using *Bit-Fixing* techniques
in **A**uxiliary-**I**nput **R**andom **O**racle **M**odel

# Future Directions

- Other metric: $\ell_2$ or $\ell_1$ distance? Edit distance?
- Circumvent $\sqrt{n}$ -lower bound for interesting special cases?
- Other Applications of SNAPs or Commitment of Proximity?

Thank you!

Q & A