Private Information Retrieval with Sublinear Online Time

Henry Corrigan-Gibbs
EPFL & MIT

Dmitry Kogan
Stanford

Eurocrypt 2020

This work

PIR schemes with

- linear-time offline phase,
- sublinear-time online lookups,
- no additional storage on the server.

Results preview

Two servers: $\sqrt{n}$ communication & online time from PRG

Single server: $n^{2/3}$ communication & online time from DCR

Talk outline

Background

The offline/online model

Our results

- 2-server scheme
- From two servers to one

Conclusion & open problems

Private information retrieval [CGKS95]

Goal

Read a record from a DB without

DB learning which record you read.

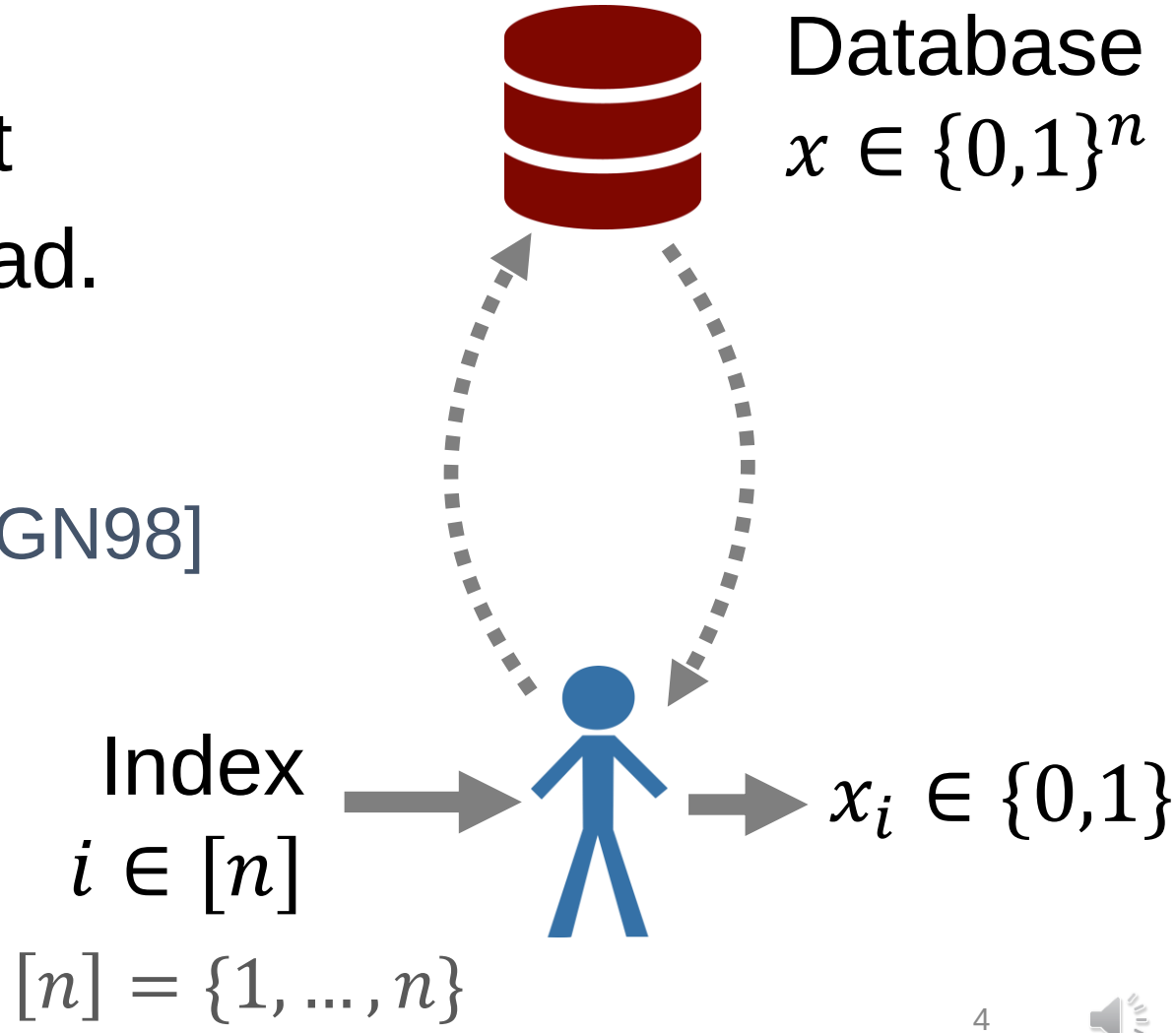
Extensions

larger records, key-value DBs [CGN98]

Applications

medical encyclopedia, stocks

private messaging, search, DNS



PIR requirements

Correctness

Client learns its bit of interest (with overwhelming prob.)

Security

(Malicious) server “learns nothing” about client’s desired bit

For all databases $x \in \{0,1\}^n$, for all $i, j \in [n]$,

$$\left\{\begin{array}{c}\text{View of server when}\\ \text{client reads bit } i\end{array}\right\} \approx_c \left\{\begin{array}{c}\text{View of server when}\\ \text{client reads bit } j\end{array}\right\}$$

PIR requirements

Correctness

Client learns its bit of interest (with overwhelming prob.)

Security

(Malicious) server “learns nothing” about client’s desired bit

Minimize communication

Modern PIR requires little communication

Multi-server PIR [CGKS95]

- Replicate DB on non-colluding servers
- State of the art (following [Amb97,CG97,BIO,BIKR02,Yek08,Efr12,…]):
  - Information-theoretic security: $n^{o(1)}$ communication [DG16]
  - Computational security: $O(\log n)$ communication [GI14, BGI15]

Single-server PIR [KO97]

- Requires cryptographic assumptions
- State of the art:
  - $\mathrm{polylog}(n)$ communication [CMS99, Lip05,…]

Modern PIR requires lots of **computation**

Server linearly scans the entire DB to respond to a query

⇒a barrier to deployment

Server **must do $\Omega(n)$ work** to respond to a query [BIM04]

- Intuition: If server doesn't touch bit i, client isn't reading bit i
- Holds even if you have **many non-colluding servers**
- Holds irrespective of **cryptographic assumptions**

Reducing computation in PIR

- Encode the DB: PIR with preprocessing [BIM04]
  - Advantage: significant decrease in server time
  - Disadvantage: significant increase in server storage
  - 1-server: DEPIR [BIPW17, CHR17], PANDA [HOWW18]
- Amortize cost: Batch PIR [IKOS04, IKOS06, LG15, Hen16, ACLS18]
- Reduce individual server's work: PIR with sharded DB [DHS14]
- Relax the privacy guarantee: PIR with differential privacy [TDG16]
- Move public-key operations to an offline phase:
  Private Stateful Information Retrieval [PPY18]

Talk outline

Background

The offline/online model

Our results

- 2-server scheme
- From two servers to one

Conclusion & open problems

The model
Step 1: Offline phase

$x \in \{0,1\}^n$

$x \in \{0,1\}^n$

$O(n)$ time

Hint $\approx \sqrt{n}$ bits

$o(n)$ bits

- The left server runs in linear time.
- But work happens before client decides which bit to read.

The model
Step 1: Offline phase

$x \in \{0,1\}^n$

$x \in \{0,1\}^n$

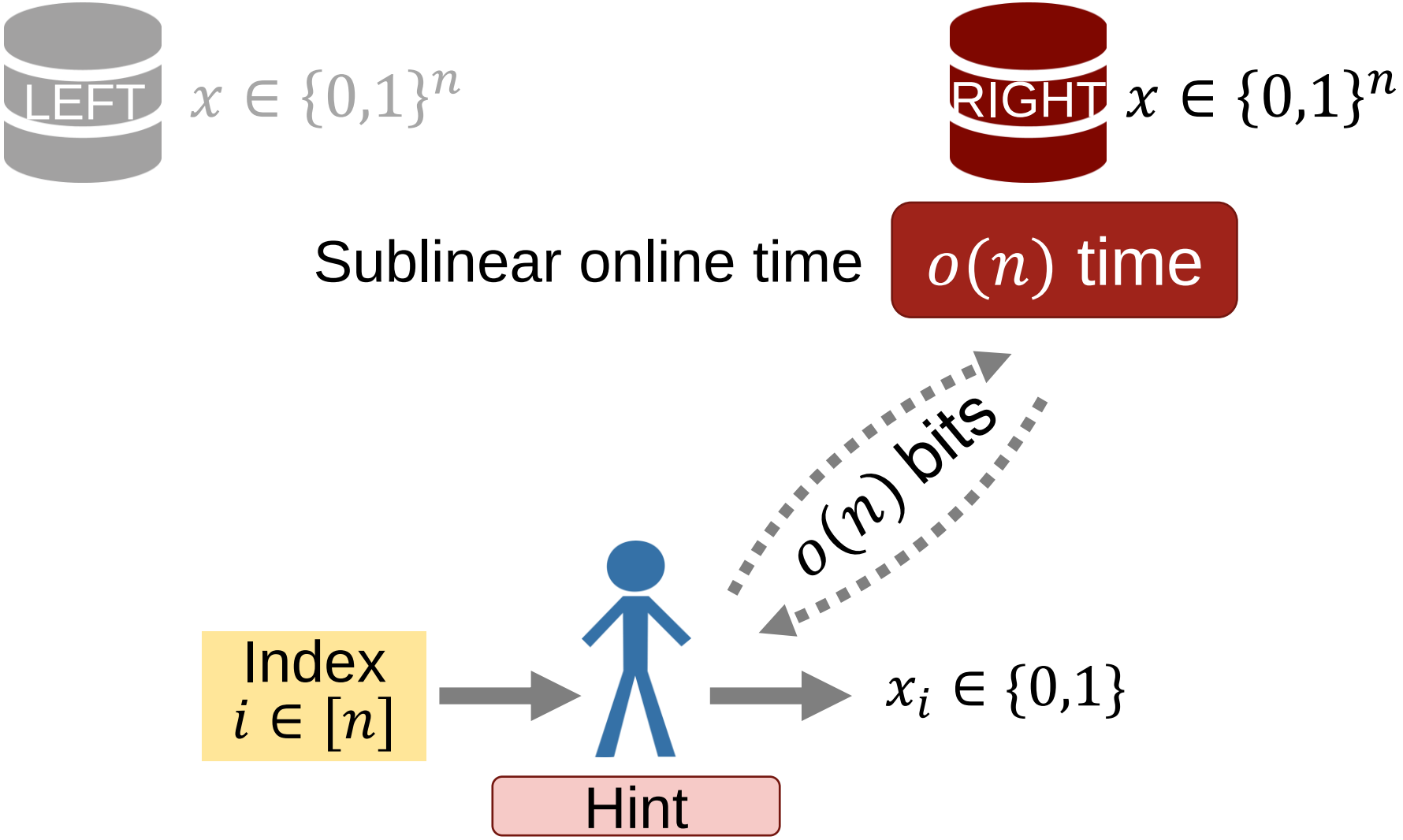
Client stores hint

Hint

The model

[DIO01, BIM04, BLW17, PPY18]

Step 2: Online phase – reading x_i



Our results

up to $\mathrm{poly}(\lambda, \log n)$ factors for length-n DB and sec. parameter λ

Two-server scheme

- $\sqrt{n}$ communication and online time (from any PRG)
- Can reuse a single offline interaction for many online queries

Single-server scheme

- $n^{2/3}$ communication and online time (from DDH, DCR,…)
  - $\sqrt{n}$ from FHE
- No public-key operations in the online phase

Our $\sqrt{n}$ schemes achieve optimal comm–online time tradeoff

Lower bound

- For offline/online schemes that store DB in its original form
- Communication C and online time T must be $C \cdot T \geq n$

Talk outline

Background

The offline/online model

Our results

- **2-server scheme**
- From two servers to one

Conclusion & open problems

Our scheme
Step 1: Offline phase

$x \in \{0,1\}^n$

$x \in \{0,1\}^n$

Computes $h_1, \ldots, h_m \in \{0,1\}$

$$h_j = \sum_{\ell \in S_j} x_\ell \bmod 2$$

$h_1, \ldots, h_m$

$S_1, \ldots, S_m$

Random subsets $S_1, \ldots, S_m \subset [n]$
each of size $|S_j| = \sqrt{n}$

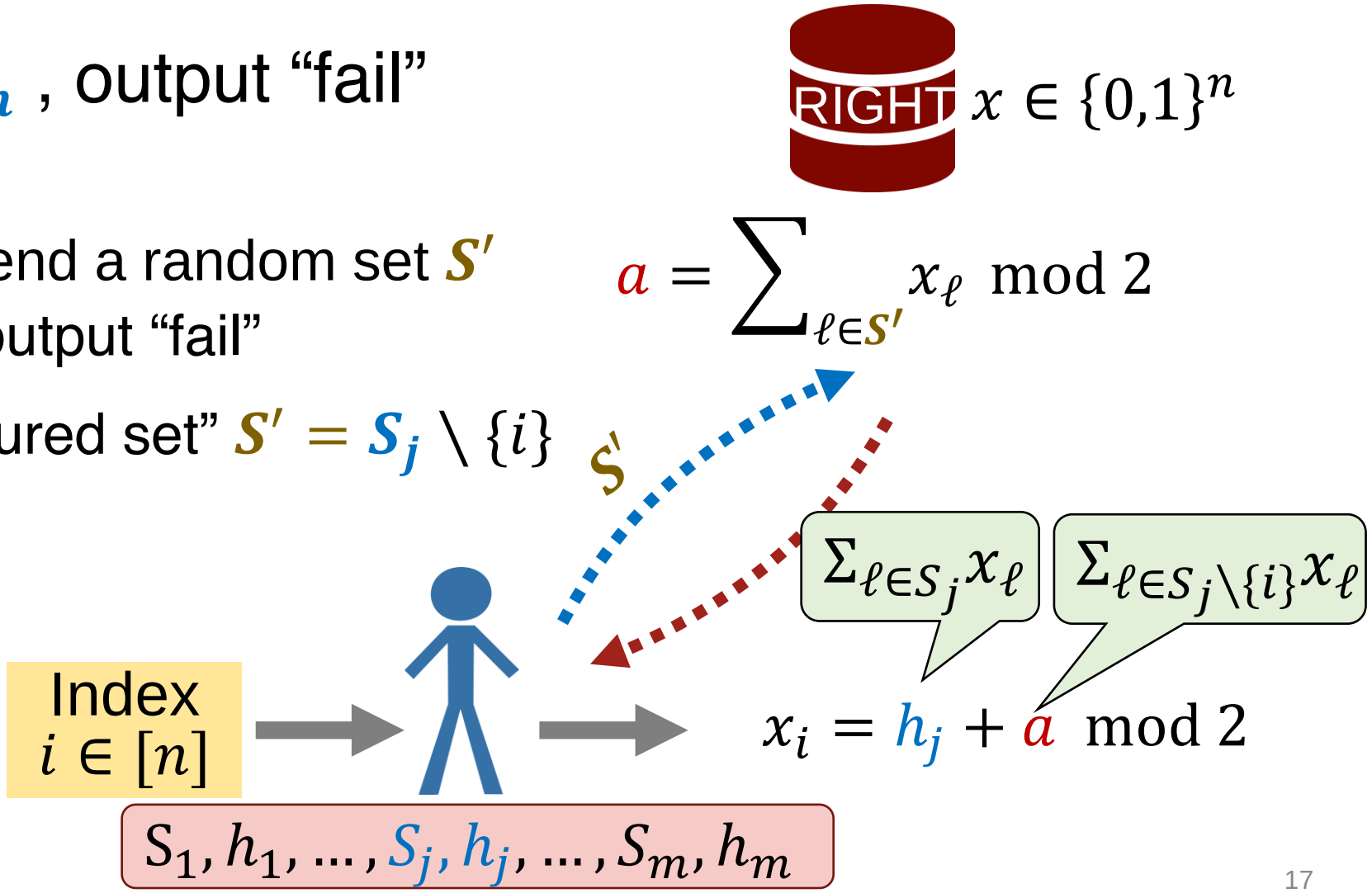
Use pseudorandomness to compress to $O_\lambda(1)$

$S_1, h_1, \ldots, S_m, h_m$

Our scheme
Step 2: Online phase – reading x_i

If $i \notin S_1 \cup \cdots \cup S_m$, output "fail"
Else, $i \in S_j$,

- With prob $\frac{\sqrt{n}-1}{n}$, send a random set S' containing i and output "fail"
- Else, send "punctured set" $S' = S_j \setminus \{i\}$



Our scheme
Correctness

If $i \notin S_1 \cup \cdots \cup S_m$, output "fail"

Else, $i \in S_j$,

- With prob $\frac{\sqrt{n}-1}{n}$, send a random set S' containing i and output "fail"
- Else, send $S' = S_j \setminus \{i\}$

Choose

$$m \approx \sqrt{n} \cdot \log n$$

Then:

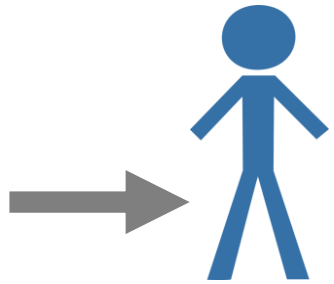
- $\Pr[\text{Fail}_1] \leq \text{negl}(n)$
  ($n \log^2 n$ balls into n bins)
- $\Pr[\text{Fail}_2] \leq 1/\sqrt{n}$

$a =$

S'

Repeat all λ times to drive down failure prob.

Index
$i \in [n]$



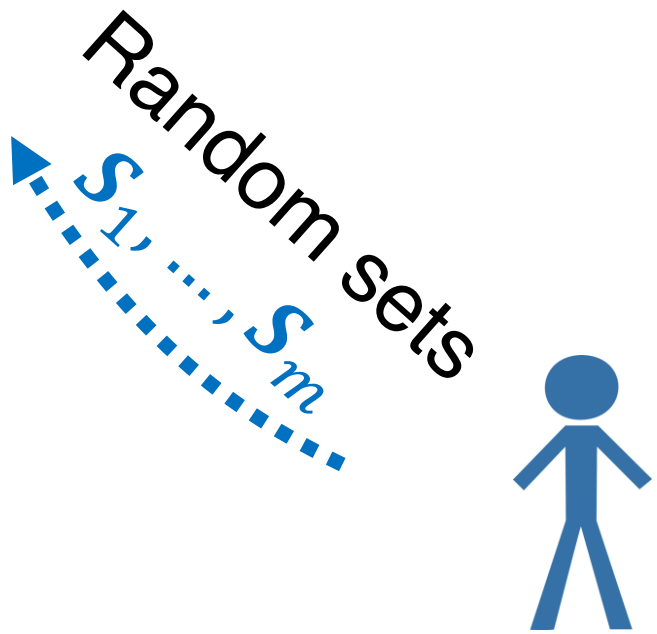
$x_i = h_j + a \mod 2$

$S_1, h_1, \ldots, S_j, h_j, \ldots, S_m, h_m$

Our scheme
Security – left server

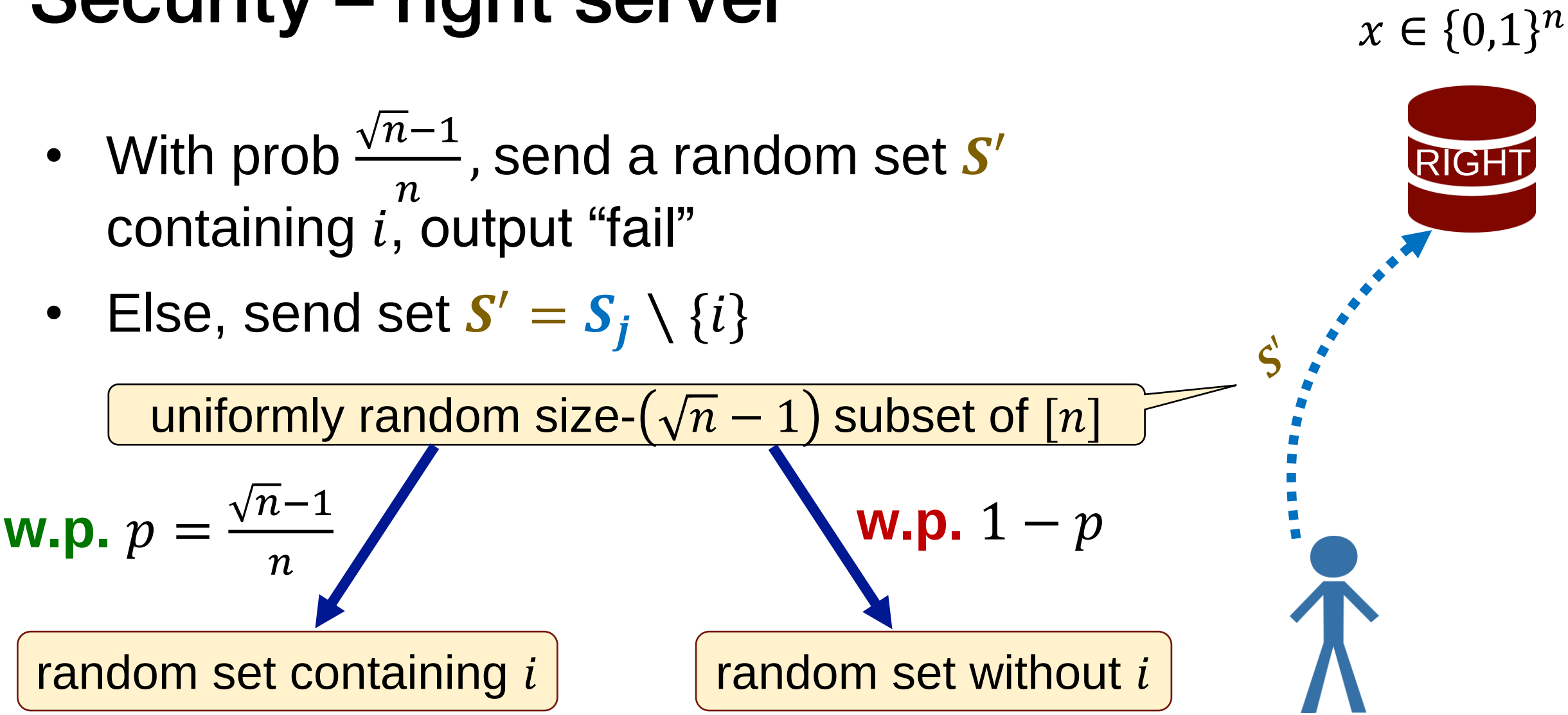
$x \in \{0,1\}^n$

$x \in \{0,1\}^n$

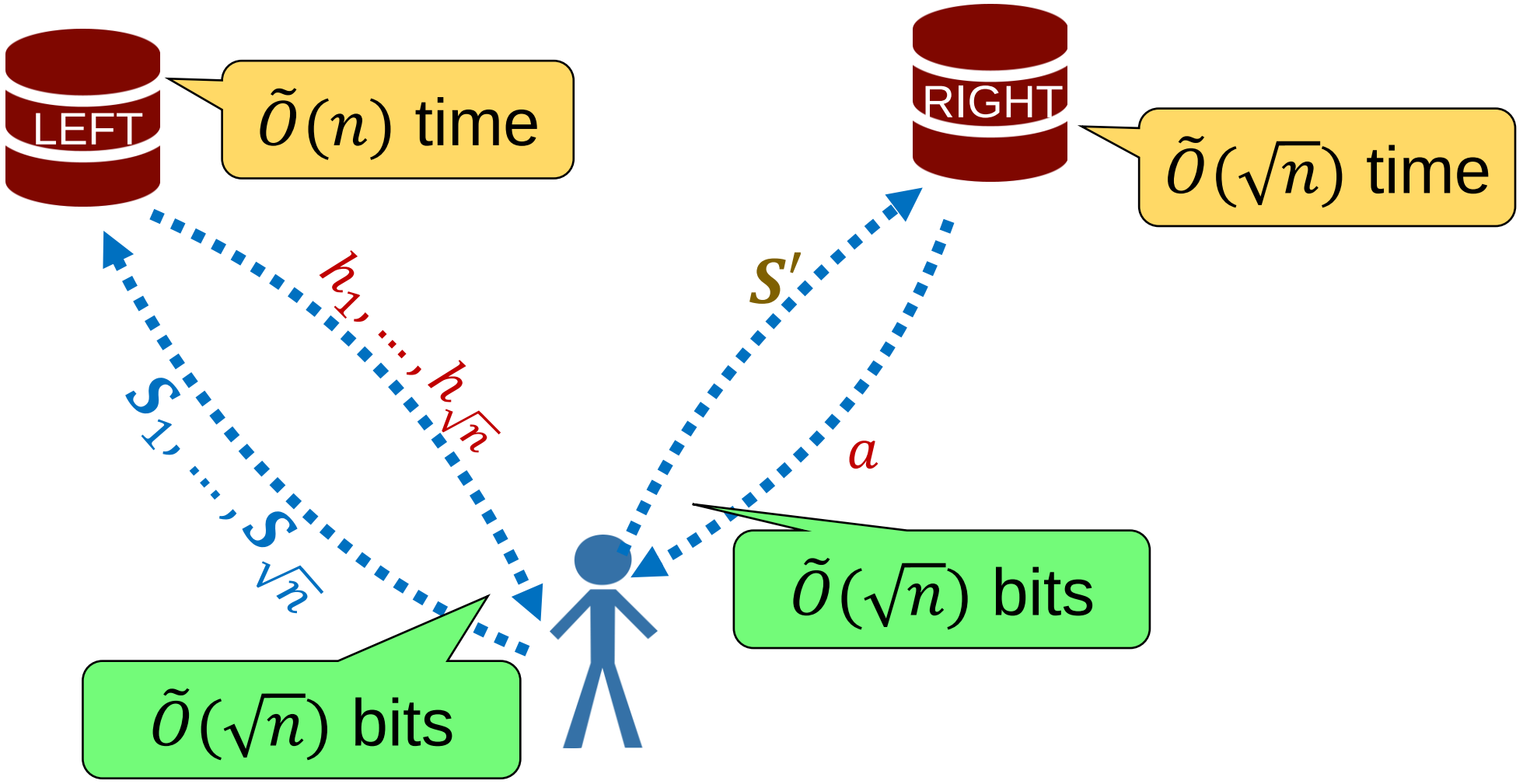


Our scheme
Security – right server

- With prob $\frac{\sqrt{n}-1}{n}$, send a random set S' containing i, output "fail"
- Else, send set $S' = S_j \setminus \{i\}$



Our scheme
Efficiency



Multiple queries

Goal: amortize cost of offline phase

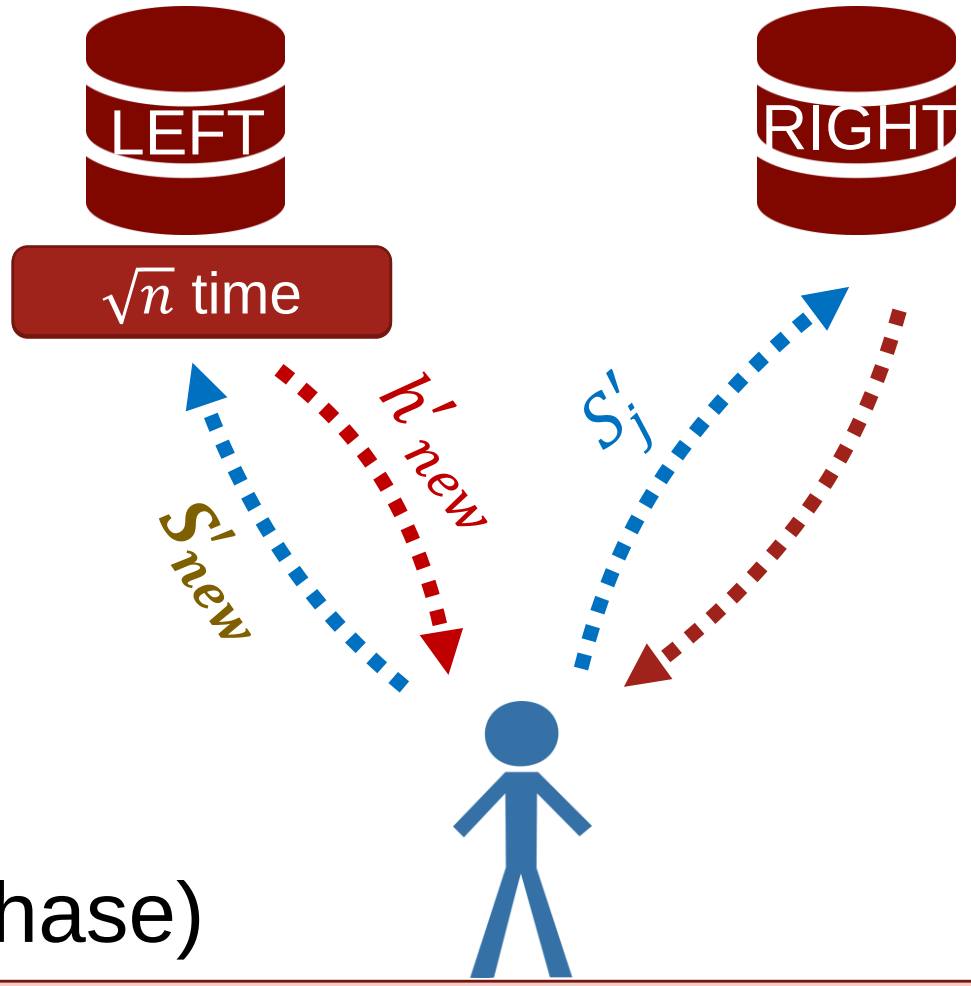
Problem: cannot reuse S_j

Given S_j^1 and S_j^2, server knows $i_1 = S_j^2 \setminus S_j^1$

Idea: sample replacement set S_{new}
fetch its parity h_{new} from **left** server

Preserving joint distribution of $\{S_j\}$ and privacy from left server requires care (see paper)

Runs in $\sqrt{n}$ time (vs. n to redo offline phase)



Two-server scheme summary

- $\sqrt{n}$ communication, online time, amortized total time per-query
- Client uses $\sqrt{n}$ time and storage
- Only need PRGs

Extensions (see paper)

- Trade-off communication for online time
- Statistical-security variant: $n^{2/3}$ communication and client time
- Reducing online communication to $\log n$
  - Using short description of ‘Puncturable sets’
  - Client storage and time increase to $n^{5/6}$

Talk outline

Background

The offline/online model

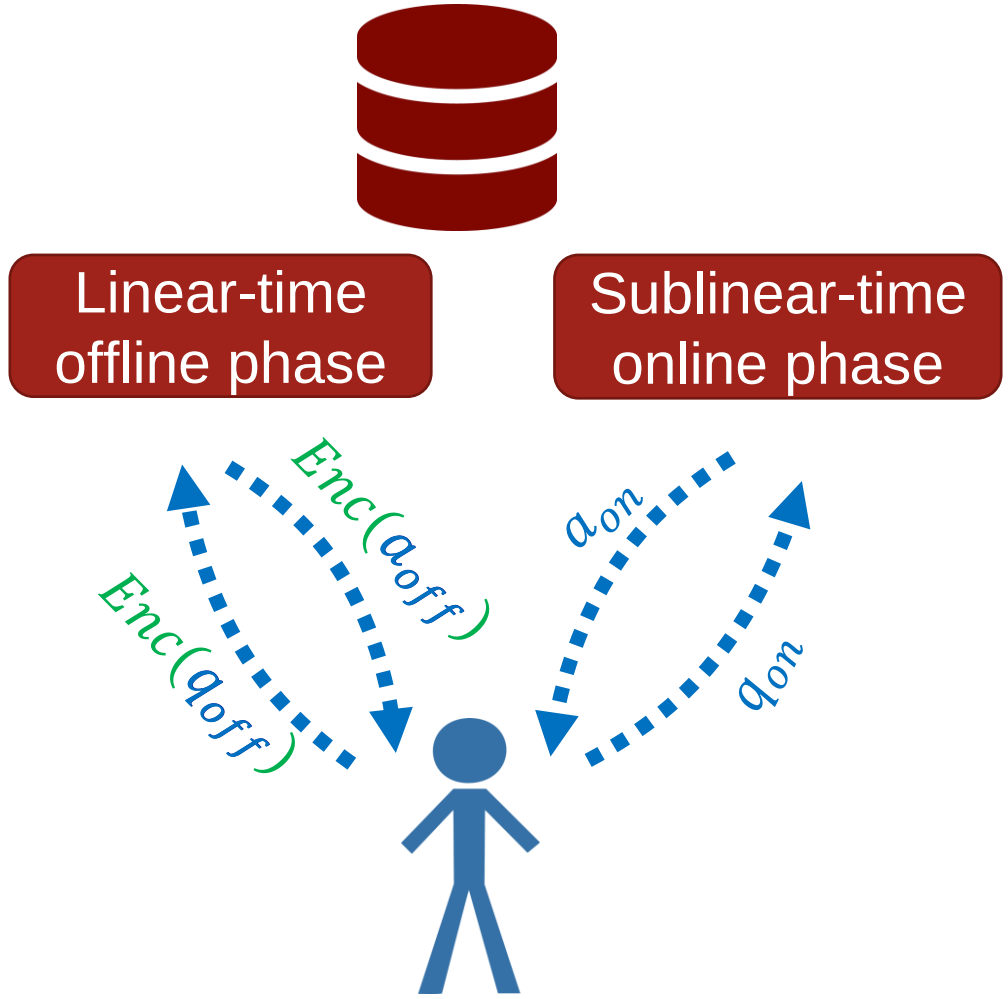
Our results

- 2-server scheme
- **From two servers to one**

Conclusion & open problems

From 2 servers to 1

Security only holds if server does not see both offline and online queries



Run both offline and online phases with the same server

Single server homomorphically evaluates offline query

- Option 1: Fully HE
  - $\sqrt{n}$ communication and online time
- Option 2: Additively HE
  - $n^{2/3}$ communication and online time

Summary

PIR with sublinear online time and no additional server storage

2-server: Open problem: amortize between clients

- Offline: $\tilde{O}_\lambda(\sqrt{n})$ communication, linear time
- Online: $O_\lambda(\log n)$ communication, $\tilde{O}_\lambda(\sqrt{n})$ server time, $\tilde{O}_\lambda(n^{5/6})$ client time

Open problem: reduce client work

1-server: $\tilde{O}_\lambda(n^{2/3})$ communication & online time ($\tilde{O}_\lambda(\sqrt{n})$ with FHE)

Matching communication-online time **lower bound** (see paper)

- Reduction from Yao's box problem

Open problems

Thank you!

dkogan@cs.stanford.edu
henrycg@csail.mit.edu
eprint 2019/1075