PLONK SNARK

W.T.S. $\exists w : C(x, w) = 0$

based on some upper bound for $|C|$

Universal Setup

Run $pp \leftarrow \text{Setup}(d, \lambda)$ from polynomial commitment scheme

Circuit-specific Setup

Precompute f_{sel}, f_{rot}

for checking circuit wiring, see lecture notes

$O(|C|)$

Protocol

circuit outputs 0

① Prove $f_{trace}(\omega^{3|C|+1}) = 0$

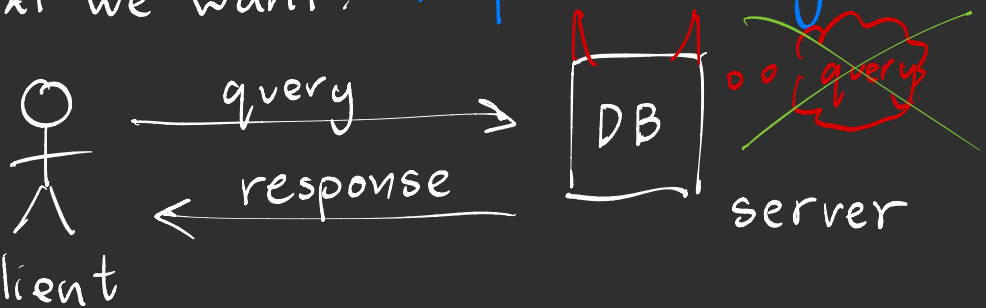
② ZeroTest$(f_{trace} - f_{in})$ — x is correctly fed as input

③ ZeroTest(f_{gates}) — circuit is evaluated correctly

④ ZeroTest$(f_{trace} - f_{trace} \circ f_{rot})$ — circuit is wired correctly

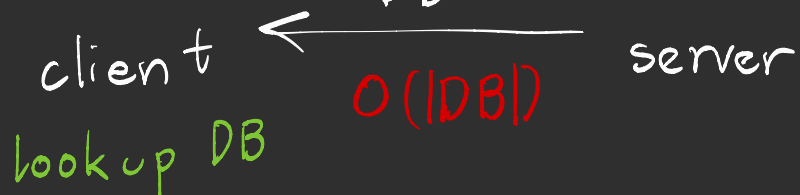
Private Information Retrieval (PIR)

What we want: "A private Google"



query DB without server learning the query

Idea 1 (Trivial PIR)



Want: $o(|DB|)$ communication (sublinear search)

How can we achieve this:

- non-colluding servers
- linearly homomorphic encryption

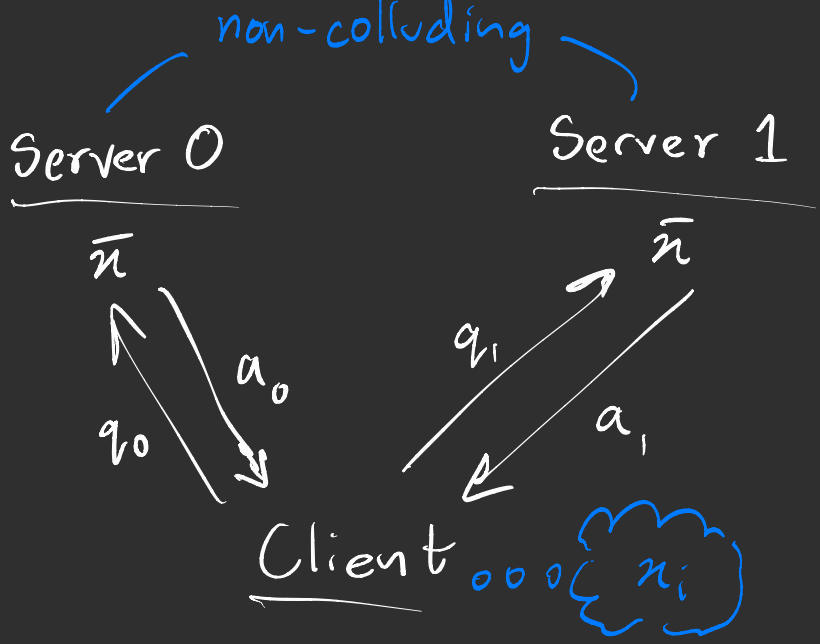
(same assumptions needed for public key crypto)

2-Server PIR

$n := |DB|$

DB: $\bar{x} = [x_1, \ldots, x_n] \in \mathbb{Z}_2^n$

Client wants to learn $x_i \in \mathbb{Z}_2$ for some $i \in [n]$



$q_0, q_1 \leftarrow \text{Query}(i)$

$a_b \leftarrow \text{Answer}(\bar{x}, q_b)$ $\quad b \in \{0,1\}$

$x_i \leftarrow \text{Recons}(a_0, a_1)$

Correctness

$$\Pr\left[\mathsf{Recons}(a_0, a_1) = x_i \;\middle|\; \begin{array}{l} q_0, q_1 \leftarrow \mathsf{Query}(i) \\ a_0 \leftarrow \mathsf{Answer}(\vec{x}, q_0) \\ a_1 \leftarrow \text{''} \quad (\text{''}, q_1) \end{array}\right] = 1$$

Security (Privacy)

$$\forall n \in \mathbb{N} : \forall i, j \in [n] : \forall b \in \{0,1\} :$$

$$\{q_b : q_0, q_1 \leftarrow \mathsf{Query}(i)\} \stackrel{c}{\approx} \{q_b : q_0, q_1 \leftarrow \mathsf{Query}(j)\}$$

q_b: view of server b

Sublinear Communication

$$\forall n \in \mathbb{N} : \forall i \in [n] : \forall \vec{x} \in \mathbb{Z}_2^n :$$

$$|\mathsf{Query}(i)| + |\mathsf{Answer}(\vec{x}, q_0)| + |\mathsf{Answer}(\vec{x}, q_1)| \in o(n)$$

$|q_b| + |a_b| \in o(n)$

Idea 2 (Strawman)

i^{th} pos

$e_i = [0, \ldots, 1, \ldots, 0]$ so $\bar{n} \cdot e_i = n_i$

Query

let $q_0 \oplus q_1 := e_i$

$q_0 \xleftarrow{\$} \mathbb{Z}_2^n$

$q_1 := e_i \oplus q_0$

$a_0 := \bar{n} \cdot q_0$ ← Answer

$a_1 := \bar{n} \cdot q_1$

$$\therefore a_0 \oplus a_1 = \bar{n} \cdot q_0 \oplus \bar{n} \cdot q_1 = \bar{n}(q_0 \oplus q_1) = \bar{n} \cdot e_i = n_i$$

Recons

correctness

Security: $\{q_b \mid \text{Query}(i)\} \equiv \{q_b \mid \text{Query}(j)\}$

$q_b \sim U[\mathbb{Z}_2^n]$

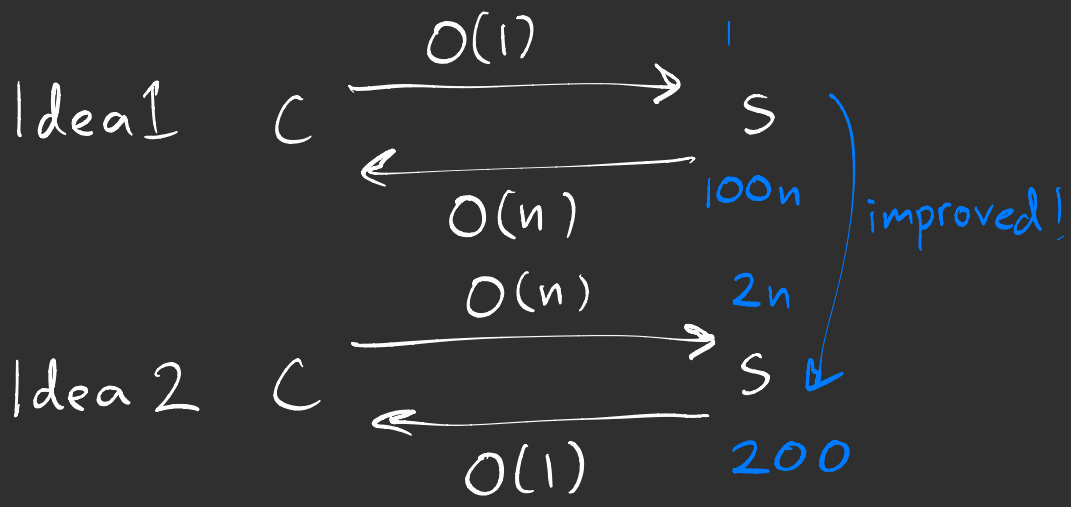
(one-time pad)

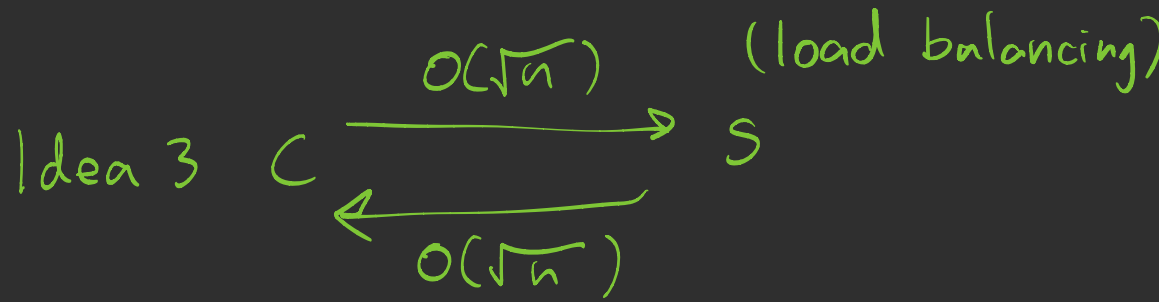
Not sublinear: $q_b \in \mathbb{Z}_2^n$

$|q_b| = n \notin o(n)$

Computational Costs

$|n_i| = 100$ bits





$$DB: X = \begin{bmatrix} n_1 & \cdots & n_{\sqrt{n}} \\ \vdots & \ddots & \vdots \\ n_{\sqrt{n}\cdot 1} & \cdots & n_{\sqrt{n}\cdot\sqrt{n}} \end{bmatrix} \in \mathbb{Z}_2^{\sqrt{n} \times \sqrt{n}}$$

n entries

Idea 3 (Load Balancing)

① Use Idea 2 to learn a column of X (one of $\sqrt{n}$)

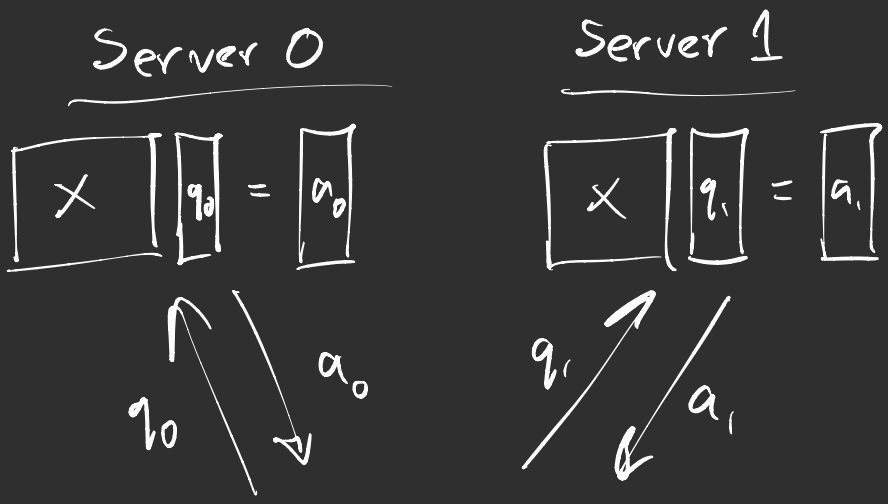
② Apply Idea 1 to that column

$i, j \in [\sqrt{n}]$ $\quad q_0 \oplus q_1 = e_j$

Query(i,j): $q_0 \xleftarrow{\$} \mathbb{Z}_2^{\sqrt{n}}$, $q_1 := e_j \oplus q_0$

Answer$(\vec{x}, q_b)$: $a_b := X \cdot q_b$

Recons(a_0, a_1): $X_{ij} := (a_0 \oplus a_1)_i$



Correctness

$$\text{Recons}(a_0, a_1) = (a_0 \oplus a_1)_i$$
$$= (X_{q_0} \oplus X_{q_1})_i$$
$$= (X(q_0 \oplus q_1))_i$$
$$= (Xe_j)_i = X_{ij}$$

Security: $q_0 \sim U[\mathbb{Z}_2^{\sqrt{n}}]$

(one-time pad)

Communication

$$\text{upload} = |q_0| + |q_1| = 2\sqrt{n}$$

$$\text{download} = |a_0| + |a_1| = 2\sqrt{n}$$

$$\text{total} = 4\sqrt{n} \in o(n)$$

next time

1. single server PIR
2. $O(n^{1/k})$ comm.