MIT 6.875

Foundations of Cryptography
Lecture 21

TODAY: Oblivious Transfer and Private Information Retrieval

Basic Problem: Database Access

Database D

0	x_0
1	x_1
2	x_2
3	x_3
4	x_4
5	x_5
6	x_6
7	x_7

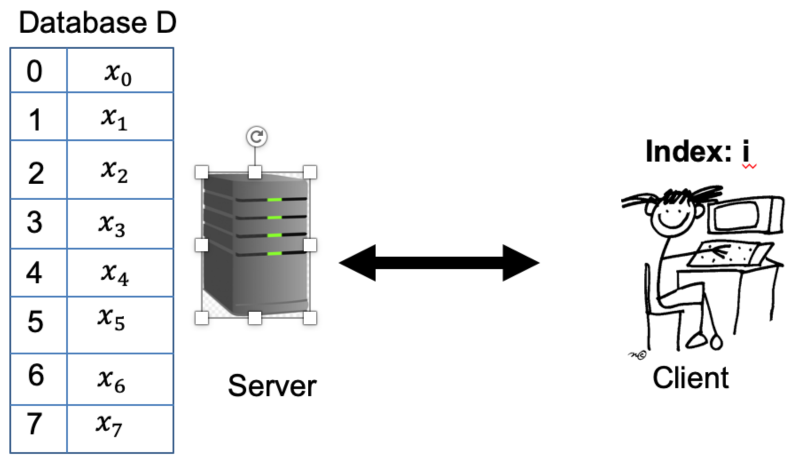
Server

Index: i

Client

Correctness: Client gets $D[i]$.

Privacy (for client): Server gets no information about i.



Here is a "solution": The server sends the entire DB to the client.

Two ways to overcome this triviality

Oblivious Transfer (OT)

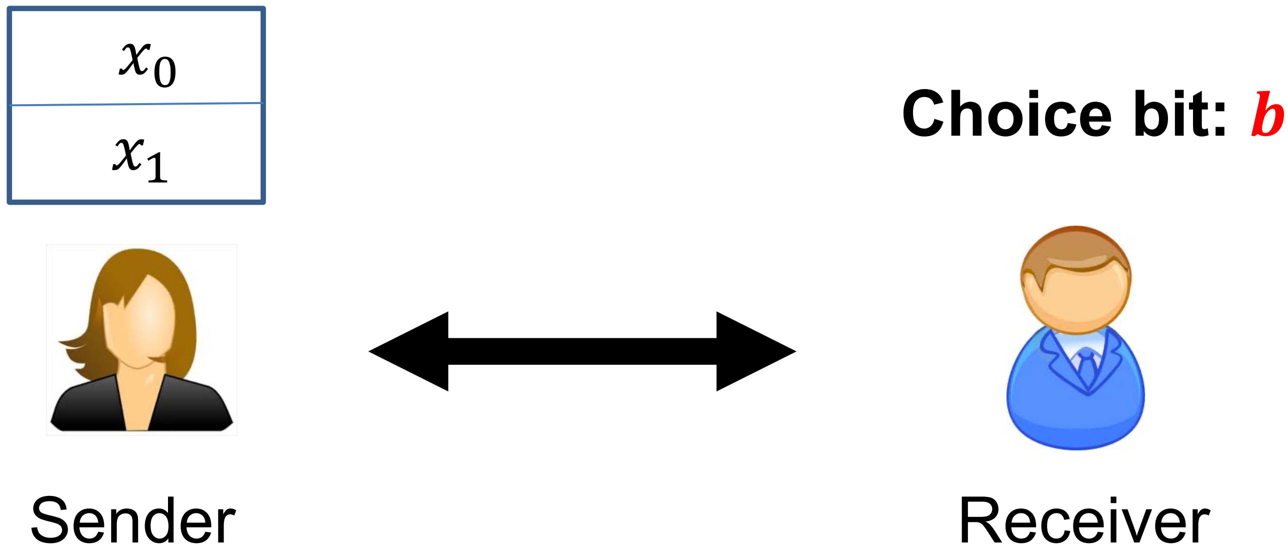
Add'l property: server privacy

Symmetric PIR = Succinctness + Server privacy

Private Information Retrieval (PIR)

Add'l property: succinctness

Oblivious Transfer (OT)



- Sender holds two bits x_0 and x_1.
- Receiver holds a choice bit b.
- Receiver should learn x_b, sender should learn nothing.

  (We will consider **honest-but-curious** adversaries; formal definition in a little bit…)

Why OT? The Dating Problem

$\alpha \in \{0,1\}$

Alice and Bob want to compute the AND $\alpha \wedge \beta$.

$\beta \in \{0,1\}$

Why OT? The Dating Problem

$\alpha \in \{0,1\}$

Alice and Bob want to compute the AND $\alpha \wedge \beta$.

$\beta \in \{0,1\}$

$x_0 = 0$

$x_1 = \alpha$

Run an OT protocol

Choice bit $b = \beta$

Bob gets α if β=1, and 0 if β=0

Here is a way to write the OT selection function: $\boldsymbol{x_1 b + x_0(1-b)}$

which, in this case is $= \alpha\beta$.

The Billionaires’ Problem

Who is richer?

The Billionaires' Problem

$f(X, Y) = 1$
if and only if $X > Y$

X

Y

...	0	1	0	0	...

...	0	1	1	1	1	1	1

Unit Vector u_X = 1 in the X^{th} location and 0 elsewhere

Vector v_Y = 1 from the $(Y+1)^{th}$ location onwards

$$\boldsymbol{f(X, Y) = \langle u_X, v_Y \rangle = \sum_{i=1}^{U} u_X[i] \wedge v_Y[i]}$$

~~**Compute each AND individually and sum it up?**~~

Detour: OT $\Rightarrow$ Secret-Shared-AND

Alice gets random γ, Bob gets random δ s.t. $\gamma \oplus \delta = \alpha\beta$.

$\alpha \in \{0,1\}$

Output: γ

$\beta \in \{0,1\}$

Output: δ

$x_0 = \gamma$
$x_1 = \alpha \oplus \gamma$

Run an OT protocol

Choice bit $b = \beta$

Alice outputs γ.

Bob gets $\boldsymbol{x_1 b} + \boldsymbol{x_0(1 \oplus b)} = (\boldsymbol{x_1} \oplus \boldsymbol{x_0})\boldsymbol{b} + \boldsymbol{x_0} = \alpha\beta \oplus \gamma := \delta$

The Billionaires' Problem

$f(X,Y) = 1$
if and only if $X > Y$

…	0	1	0	0	…

Unit Vector u_X

…	0	1	1	1	1	1	1

Vector v_Y

$$\boldsymbol{f(X,Y) = \langle u_X, v_Y \rangle = \sum_{i=1}^{U} u_X[i] \wedge v_Y[i]}$$

1. Alice and Bob run many OTs to get (γ_i, δ_i) s.t.

$$\gamma_i \oplus \delta_i = \boldsymbol{u_X[i] \wedge v_Y[i]}$$

2. Alice computes $\gamma = \oplus_i \gamma_i$ and Bob computes $\delta = \oplus_i \delta_i$.
3. Alice reveals γ and Bob reveals δ.

Check (correctness): $\gamma \oplus \delta = \boldsymbol{\langle u_X, v_Y \rangle = f(X,Y)}$.

The Billionaires' Problem

$f(X,Y) = 1$
if and only if $X > Y$

...	0	1	0	0	...

Unit Vector u_X

...	0	1	1	1	1	1	1

Vector v_Y

$$\boldsymbol{f(X,Y) = \langle u_X, v_Y \rangle = \sum_{i=1}^{U} u_X[i] \wedge v_Y[i]}$$

1. Alice and Bob run many OTs to get (γ_i, δ_i) s.t.

$$\gamma_i \oplus \delta_i = \boldsymbol{u_X[i] \wedge v_Y[i]}$$

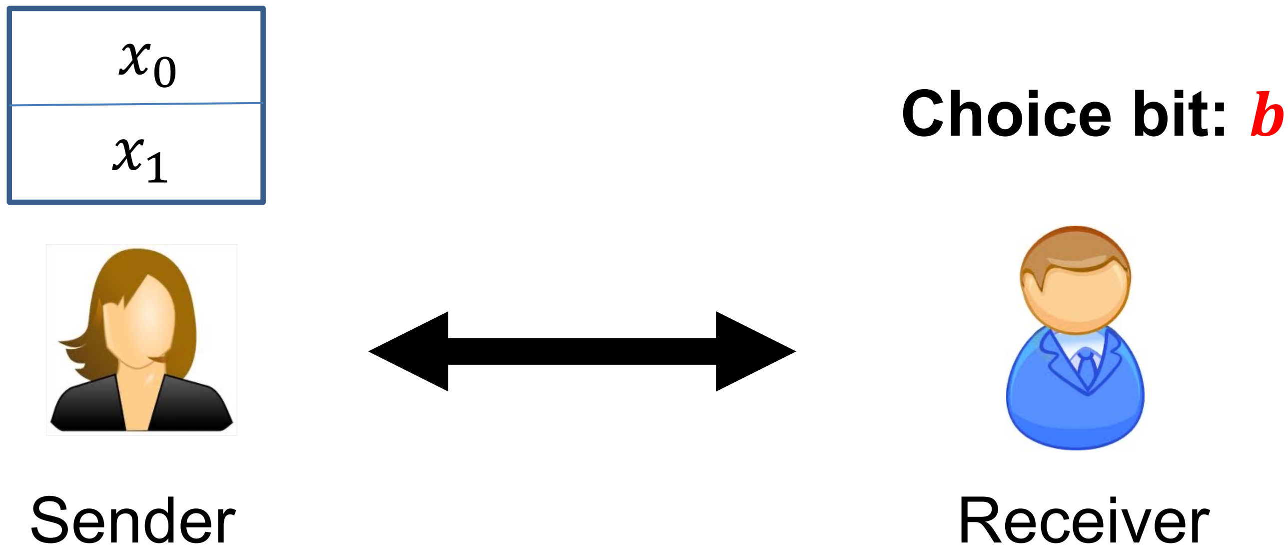
2. Alice computes $\gamma = \oplus_i \gamma_i$ and Bob computes $\delta = \oplus_i \delta_i$.

Check (privacy): Alice & Bob get a bunch of random bits.

“OT is Complete”

Theorem *(lec22-24):* OT can solve not just love and money, but ***any*** two-party (and multi-party) problem.

OT Definition

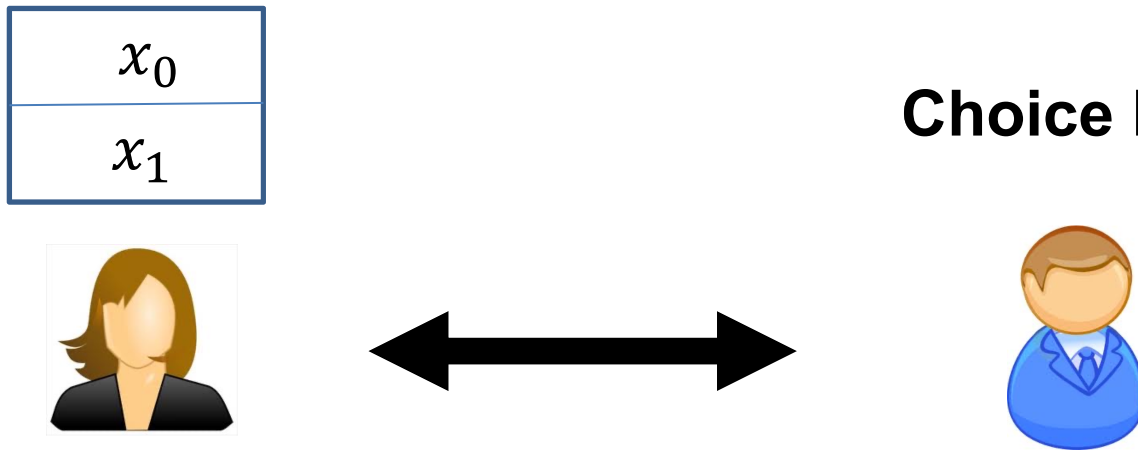


Receiver Security: Sender should not learn b.

Define Sender's view $View_S(x_0, x_1, b)$ = her random coins and the protocol messages.

OT Definition

Choice bit: *b*



Sender

Receiver

Receiver Security: Sender should not learn b.

There exists a PPT simulator SIM_S such that for any x_0,x_1 and b:

$$SIM_S(x_0, x_1) \cong Views_S(x_0, x_1, b)$$

OT Definition

x_0
x_1

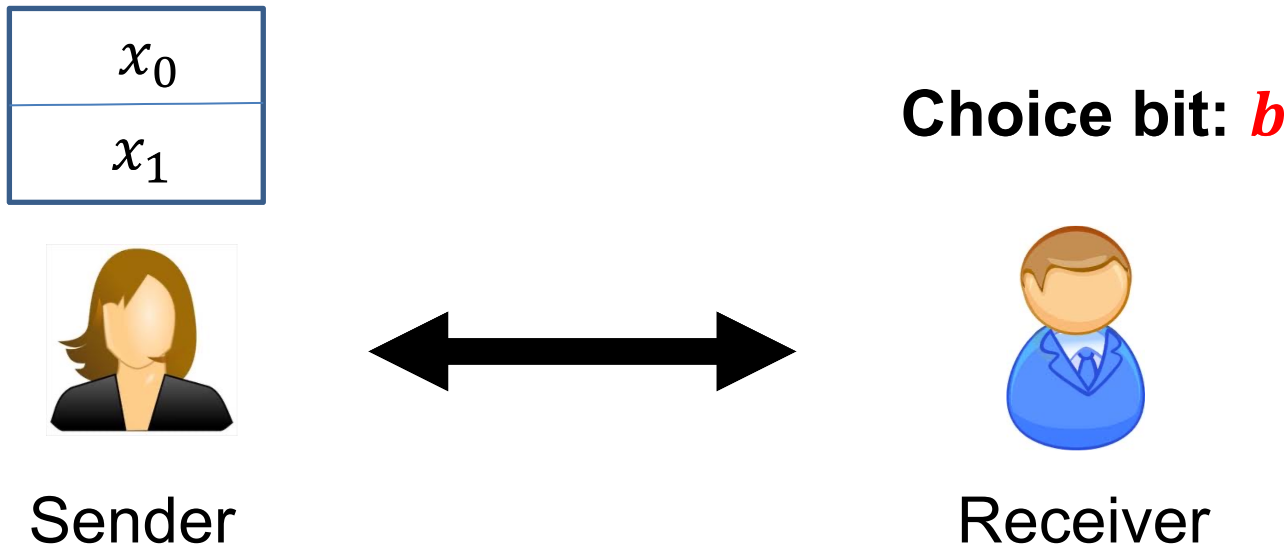
Choice bit: $\boldsymbol{b}$

Sender Receiver

Sender Security: Receiver should not learn x_{1-b}**.**

Define Receiver's view $View_R(x_0, x_1, b)$ = his random coins and the protocol messages.

OT Definition



Sender Security: Receiver should not learn x_{1-b}.

There exists a PPT simulator SIM_R such that for any x_0,x_1 and b:

$$SIM_R(b, x_b) \cong View_R(x_0, x_1, b)$$

OT Protocol 1: Trapdoor Permutations

For concreteness, let's use the RSA trapdoor permutation.

Input bits: (x_0, x_1)

Choice bit: b

Pick $N = PQ$ and RSA exponent e.

$$\xrightarrow{N, e}$$

Choose random r_b and set $s_b = r_b^e \bmod N$

Choose random s_{1-b}

$$\xleftarrow{s_0, s_1}$$

Compute r_0, r_1 and one-time pad x_0, x_1 using hardcore bits

$$\xrightarrow[x_1 \oplus HCB(r_1)]{x_0 \oplus HCB(r_0)}$$

Bob can recover x_b but not x_{1-b}

OT Protocol 1: Trapdoor Permutations

Input bits: (x_0, x_1)

N, e $\longrightarrow$

s_0, s_1 $\longleftarrow$

$x_0 \oplus HCB(r_0)$, $x_1 \oplus HCB(r_1)$ $\longrightarrow$

Choice bit: b

How about Bob's security

(a.k.a. Why does Alice not learn Bob's choice bit)?

Alice's view is s_0, s_1 one of which is chosen randomly from Z_N^* and the other by raising a random number to the e-th power. They look exactly the same!

OT Protocol 1: Trapdoor Permutations

Input bits: (x_0, x_1)

$\xrightarrow{N, e}$

$\xleftarrow{s_0, s_1}$

$\xrightarrow[x_1 \oplus HCB(r_1)]{x_0 \oplus HCB(r_0)}$

Choice bit: b

How about Bob's security
(a.k.a. Why does Alice not learn Bob's choice bit)?

Exercise: Show how to construct the simulator.

OT Protocol 1: Trapdoor Permutations

Input bits: (x_0, x_1)

$$\xrightarrow{N, e}$$

$$\xleftarrow{s_0, s_1}$$

$$\xrightarrow[x_1 \oplus HCB(r_1)]{x_0 \oplus HCB(r_0)}$$

Choice bit: b

How about Alice's security
(a.k.a. Why does Bob not learn both of Alice's bits)?

Assuming Bob is semi-honest, he chose s_{1-b} uniformly at random, so the hardcore bit of $s_{1-b} = r_{1-b}^{d}$ is computationally hidden from him.

OT from Trapdoor Permutations

Input bits: (x_0, x_1)

$\xrightarrow{N, e}$

$\xleftarrow{s_0, s_1}$

$\xrightarrow[x_1 \oplus HCB(r_1)]{x_0 \oplus HCB(r_0)}$

Choice bit: b

How about Alice's security
(a.k.a. Why does Bob not learn both of Alice's bits)?

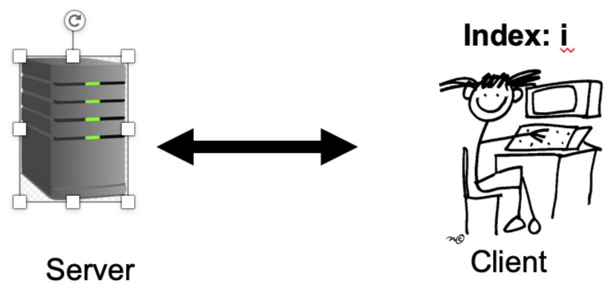
Exercise: Show how to construct the simulator.

Many More Constructions of OT

***Theorem**:* OT protocols can be constructed based on the hardness of the Diffie-Hellman problem, factoring, quadratic residuosity, LWE, elliptic curve isogeny problem etc. etc.

Database D

0	x_0
1	x_1
2	x_2
3	x_3
4	x_4
5	x_5
6	x_6
7	x_7



Two ways to overcome the triviality

Oblivious Transfer (OT)

Add’l property: server privacy

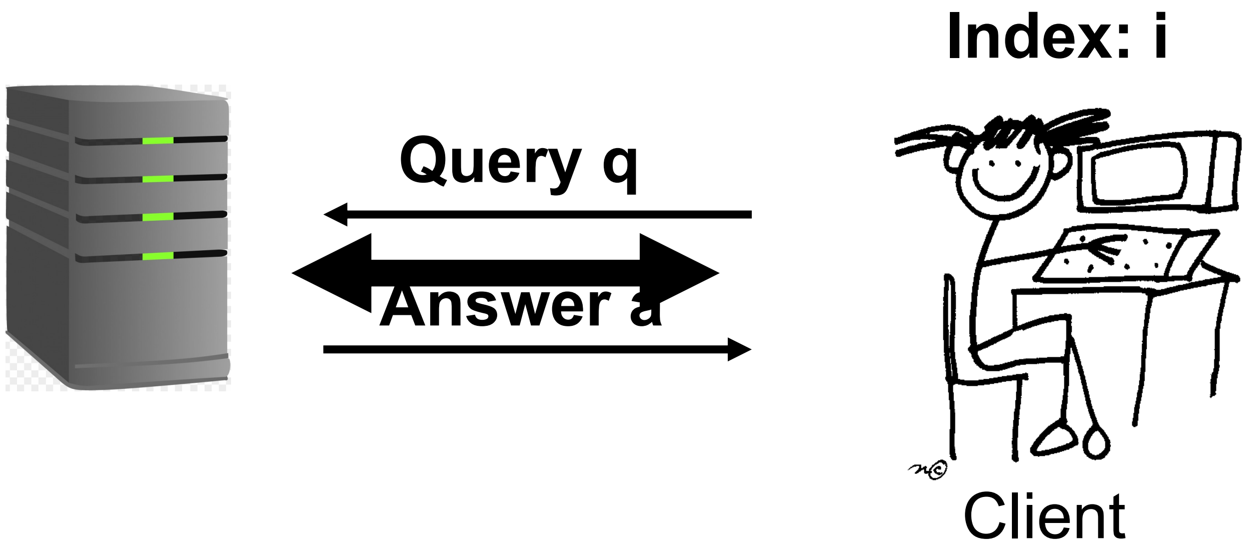
Symmetric PIR = Succinctness + Server privacy

Private Information Retrieval (PIR)

Add’l property: succinctness

Private Information Retrieval

0	x_0
1	x_1
2	x_2
3	x_3
4	x_4
5	x_5
6	x_6
7	x_7



Privacy (for client): Server gets no information about i.

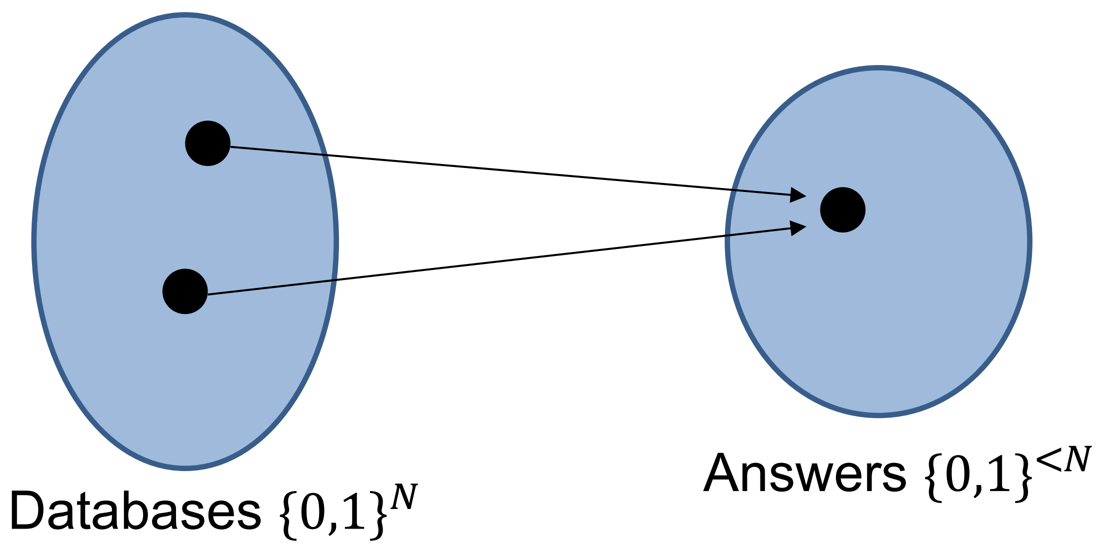
Succinctness: Total communication $< N$ bits, ideally $O(\log N)$.

Lower Bound

Theorem: Any PIR protocol that communicates $< N$ bits *cannot be information-theoretically* (client-)*private*.

Idea: Pigeon-hole principle.

Consider the function (parameterized by the query) that maps databases to answers.

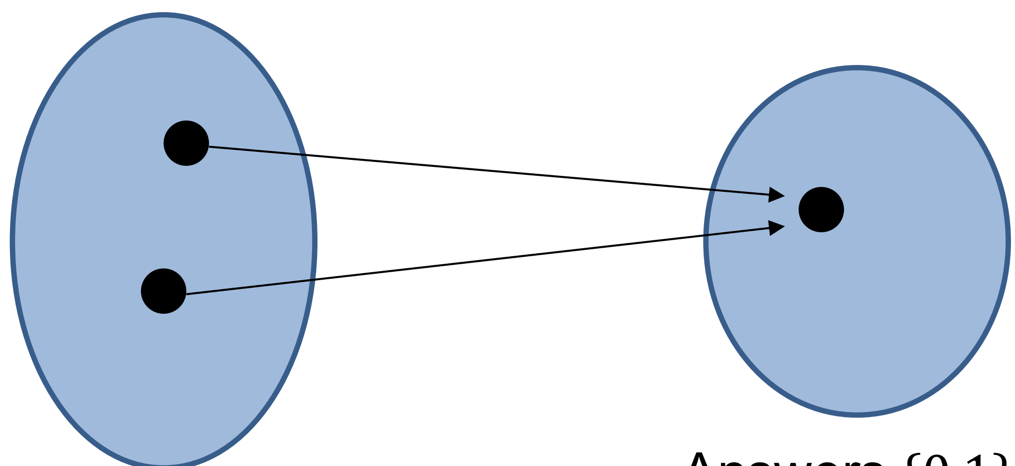


Lower Bound

Theorem: Any PIR protocol that communicates $< N$ bits *cannot be information-theoretically* (client-)*private*.

The two databases differ in at least one index, say $\boldsymbol{i^*}$.

By correctness, the queried index could not have been $\boldsymbol{i^*}$. This reveals some information about the query. QED.



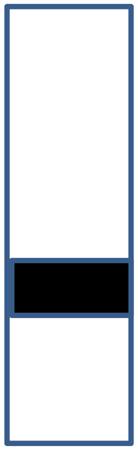
Databases $\{0,1\}^N$

Answers $\{0,1\}^{<N}$

Construction 0: Using Additive HE

Database D

N



Pretty long! $O(N\lambda)$ bits.

$Enc(\mathrm{sk}, \overrightarrow{u_i})$

Client wants to retrieve index i

Homomorphically compute inner product with the database

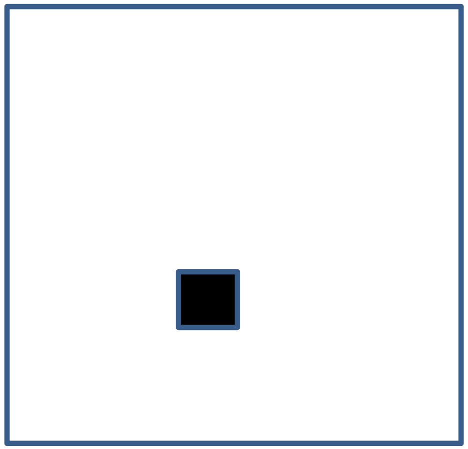
$Enc(\mathrm{sk}, \overrightarrow{u_i} \cdot D)$ $= Enc(\mathrm{sk}, D_i)$

Pretty short! $O(\lambda)$ bits, where λ is the security parameter.

Constr. 1: Using Additive HE (better)

Database D

$\sqrt{N}$



$\sqrt{N}$

Database D = $\sqrt{N}$ by $\sqrt{N}$ matrix

$O(\sqrt{N}\lambda)$ bits.

$Enc(\text{sk}, \overrightarrow{u_i})$

Client wants to retrieve index (i, j)

Homomorphically compute inner product with each column

$$
\begin{aligned}
Enc(\text{sk}, \overrightarrow{u_i} \cdot D_1) &= Enc(\text{sk}, D_{i,1}) \\
Enc(\text{sk}, \overrightarrow{u_i} \cdot D_2) &= Enc(\text{sk}, D_{i,2}) \\
\ldots \quad &= Enc(\text{sk}, D_{i,j}) \\
Enc(\text{sk}, \overrightarrow{u_i} \cdot D_{\sqrt{N}}) &= Enc(\text{sk}, D_{i,\sqrt{N}})
\end{aligned}
$$

$O(\sqrt{N}\lambda)$ bits.

Construction 2 (The "Ultimate" PIR)

Write the database access function:

$$F_D(x_1 x_2 \ldots x_n) = \sum_{i=i_1 i_2 \ldots i_n} D_i \cdot (x =_? i)$$

$$= \sum_{i=i_1 i_2 \ldots i_n} D_i \cdot \underbrace{\prod_{j=1}^{n} (x_j = i_j)}$$

This is 1 if and only if $x = i$.

$O(\log N \cdot \lambda)$ bits.

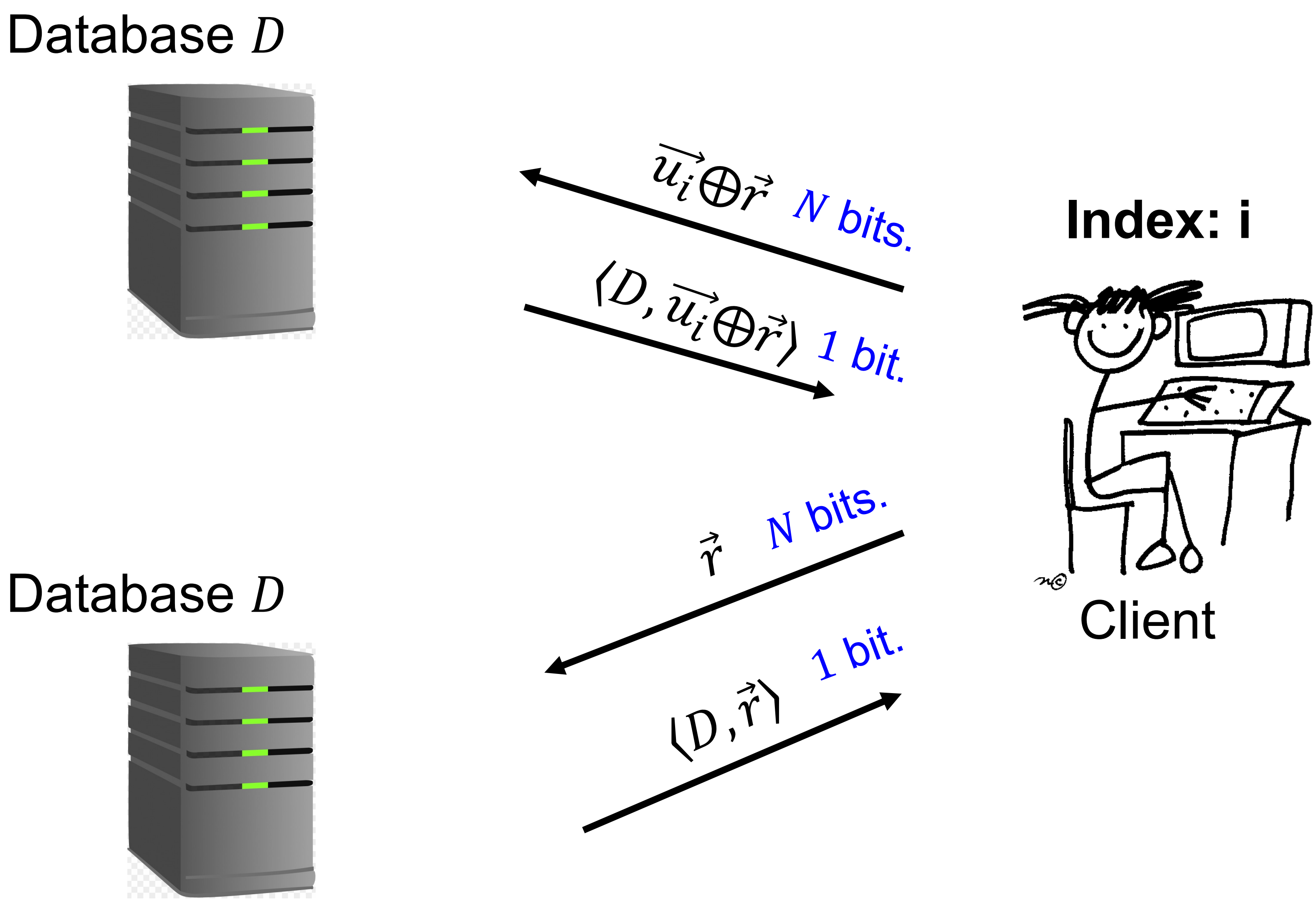
Client encrypts x. Server homomorphically evaluates F_D.

Can we Achieve *Unconditionally* Secure PIR?

I thought you proved this is impossible?!

Change the model: two or more non-communicating servers!
(also see PS 6)

Two-Server PIR



$$\langle D, \overrightarrow{u_i} \oplus \vec{r} \rangle \oplus \langle D, \vec{r} \rangle = \langle D, \overrightarrow{u_i} \rangle = D_i$$

WE SAW: Oblivious Transfer and Private Information Retrieval

L22-24: How to solve ***any*** two-party (and multi-party) problem.