MONDAY: SEE KEVIN'S NOTES

WEDNESDAY:

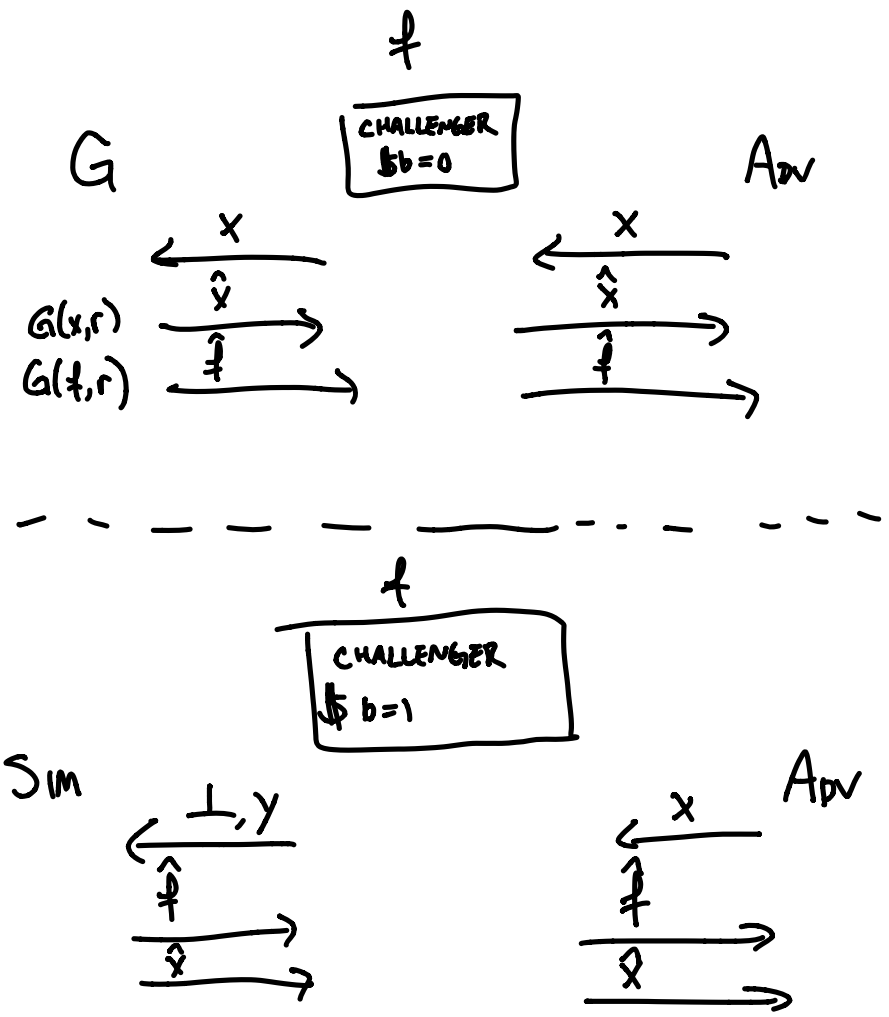
GARBLED CIRCUITS REMINDER:

G	E
$G(f,r) \to \hat{f}$	$\tilde{f}(\hat{x}) = f(x) = y$
$G(x,r) \to \hat{x}$	

$\hat{f}, \hat{x}$ REVEAL NOTHING EXCEPT $f(x) = y$

HOW DO WE FORMALIZE THIS? SIMULATION!

SIMULATOR S KNOWING y SHOULD BE ABLE TO SIMULATE PROVER IN SECURITY GAME

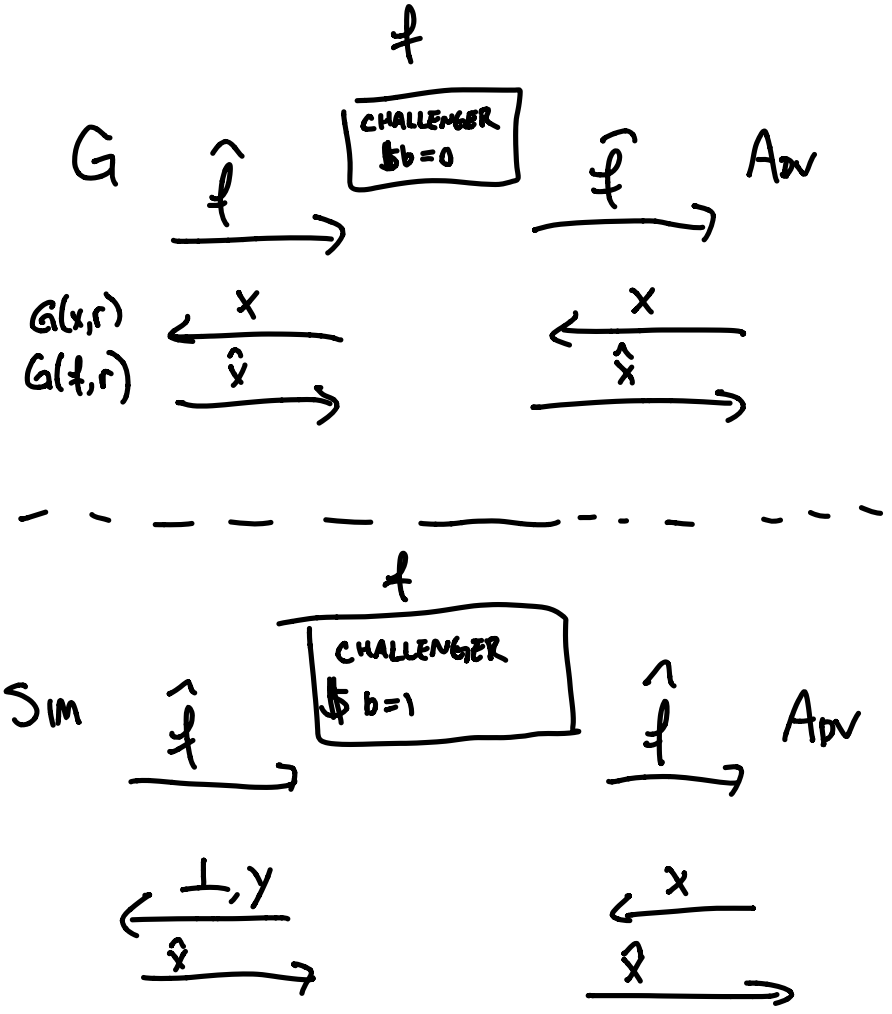


GARBLING IS SECURE IF NO ADV$_{EPT}$ CAN PREDICT b WITH PROB $\frac{1}{2}+\varepsilon$, ε NON-NEGLIGIBLE.

Adaptive vs. Non-adaptive GC:

Non-adaptive: Definition Above

Adaptive:

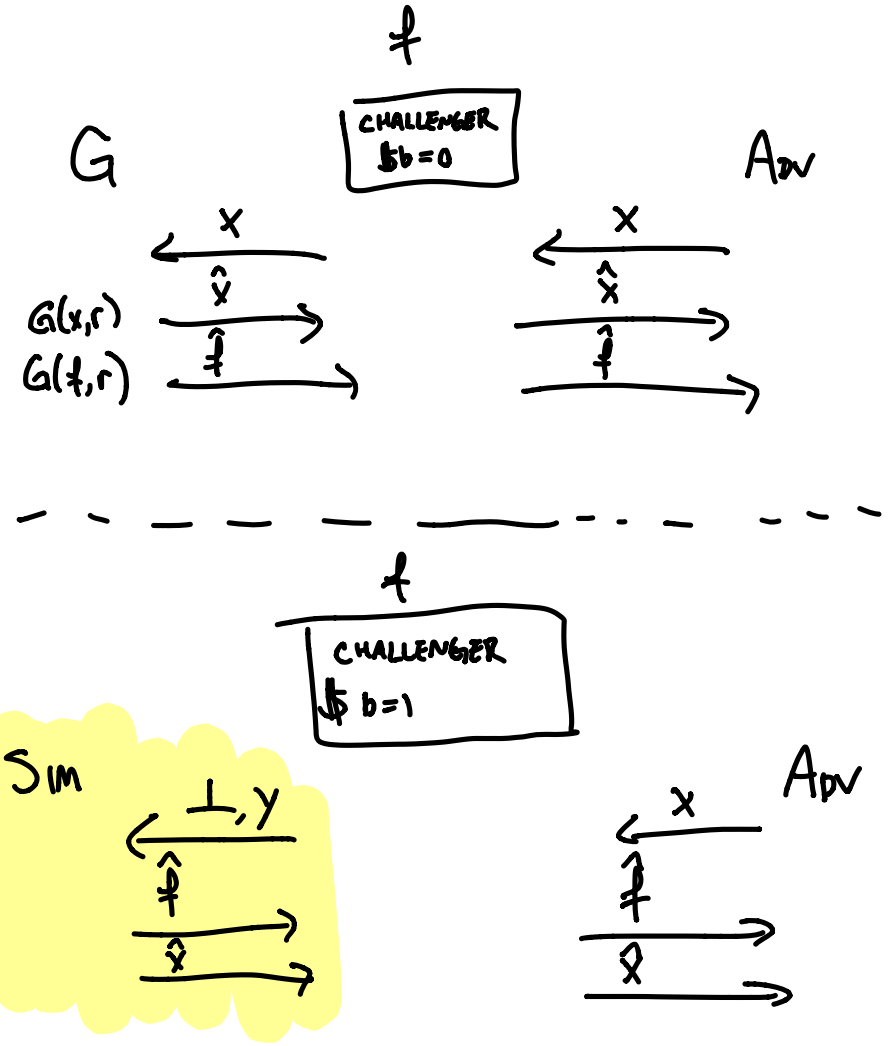


X COULD BE A FUNCTION OF $\hat{f}$!

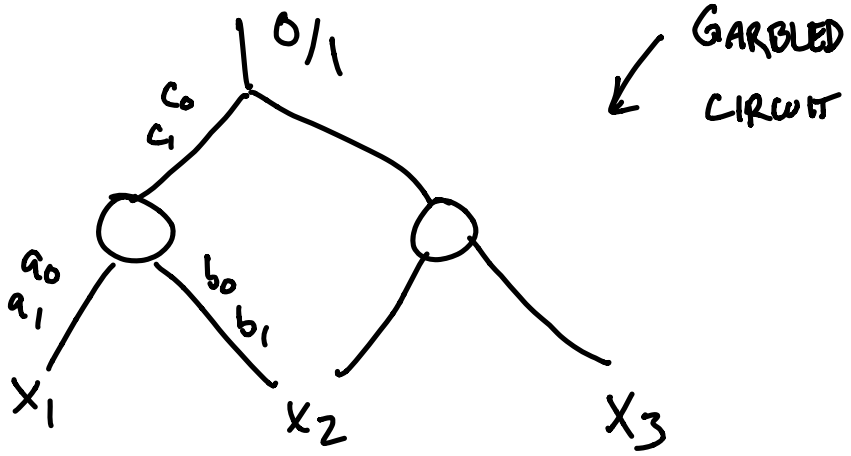
SECURITY PROOF IS MUCH HARDER: A_{DV} HAS FANCY TRICKS. BUT NO TIME IN THIS COURSE.

LINDELL-PINKAS PROOF OF NON-ADAPTIVE SECURITY FOR YGC:

GOAL: CONSTRUCT Sim_{PPT} SATISFYING NON-ADAPTIVE SECURITY GAME.



STRATEGY: HYBRID ARGUMENT BETWEEN REAL GARBLING + SIMULATED GARBLING



IN EACH HYBRID, BREAK AN ADDITIONAL GATE SO OUTPUT DOES NOT DEPEND ON INPUT.

$$E_{a_1}(E_{b_0}(c_0))$$
$$E_{a_0}(E_{b_1}(c_0))$$
$$E_{a_0}(E_{b_0}(c_0))$$
$$E_{a_1}(E_{b_1}(c_0))$$

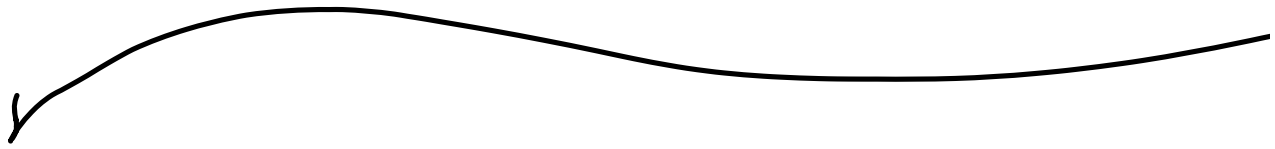
NOW, OUTPUT OF GATE IS THE SAME, C_0, BUT NOW TABLE DOES NOT DEPEND ON INPUT KEYS. EVALUATOR CAN'T TELL DIFFERENCE BECAUSE OUTPUT IS THE SAME, & CHANGED VALUES ARE ENCRYPTED UNDER UNKNOWN KEYS.

THUS, BY HYBRID ARGUMENT, ORIGINAL CIRCUIT IS INDISTINGUISHABLE FROM "BROKEN" CIRCUIT WHERE ALL OUTPUTS ARE CONSTANT.

SINCE KEYS ARE INDISTINGUISHABLE FROM RANDOM, SIM OUTPUTS TABLES OF THIS "BROKEN" CIRCUIT

- MODIFIES OUTPUT TABLE SUCH THAT OUTPUT IS y ON THESE CONSTANT VALUES.

THIS IS INDISTINGUISHABLE FROM $\hat{f}$ BUT OUTPUTS y. ☒

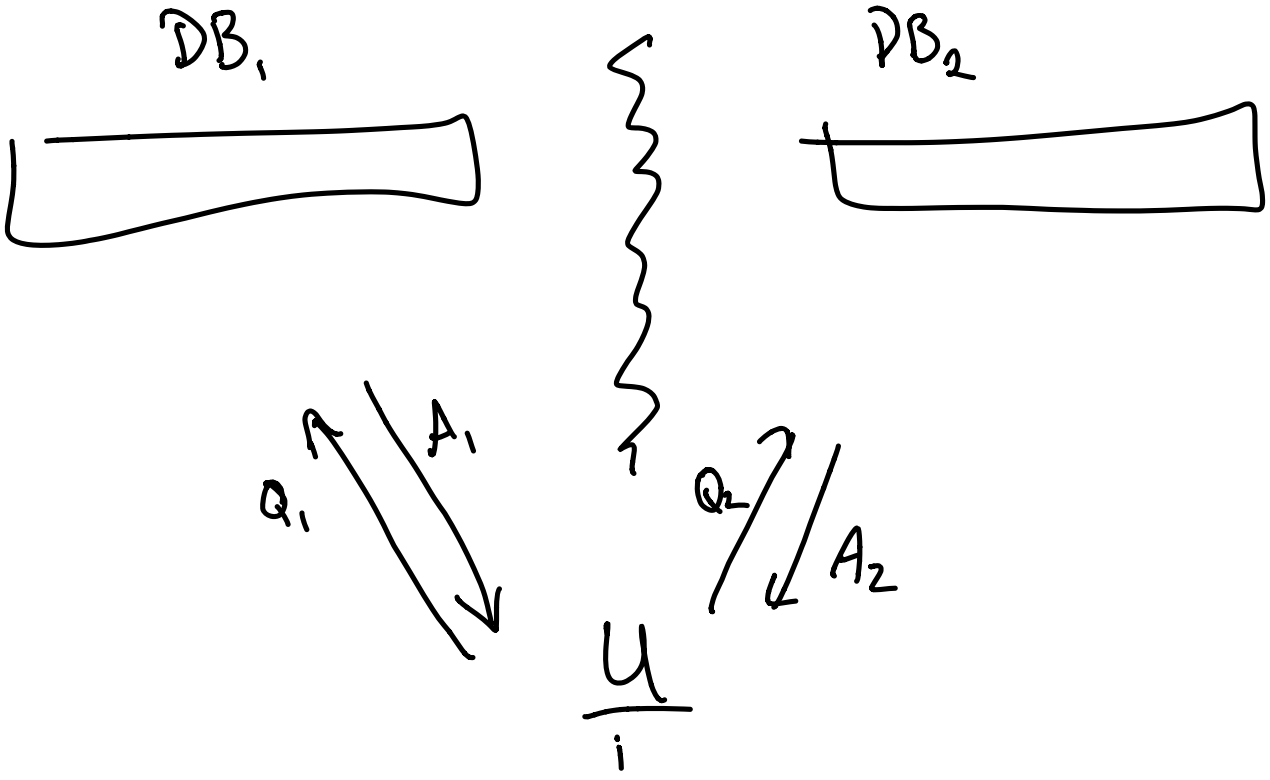


Information-Theoretic 2-party PIR

Multi-Server PIR:

DB_1, DB_2 copies of same database

No communication



(1) CORRECTNESS: U LEARNS X_i

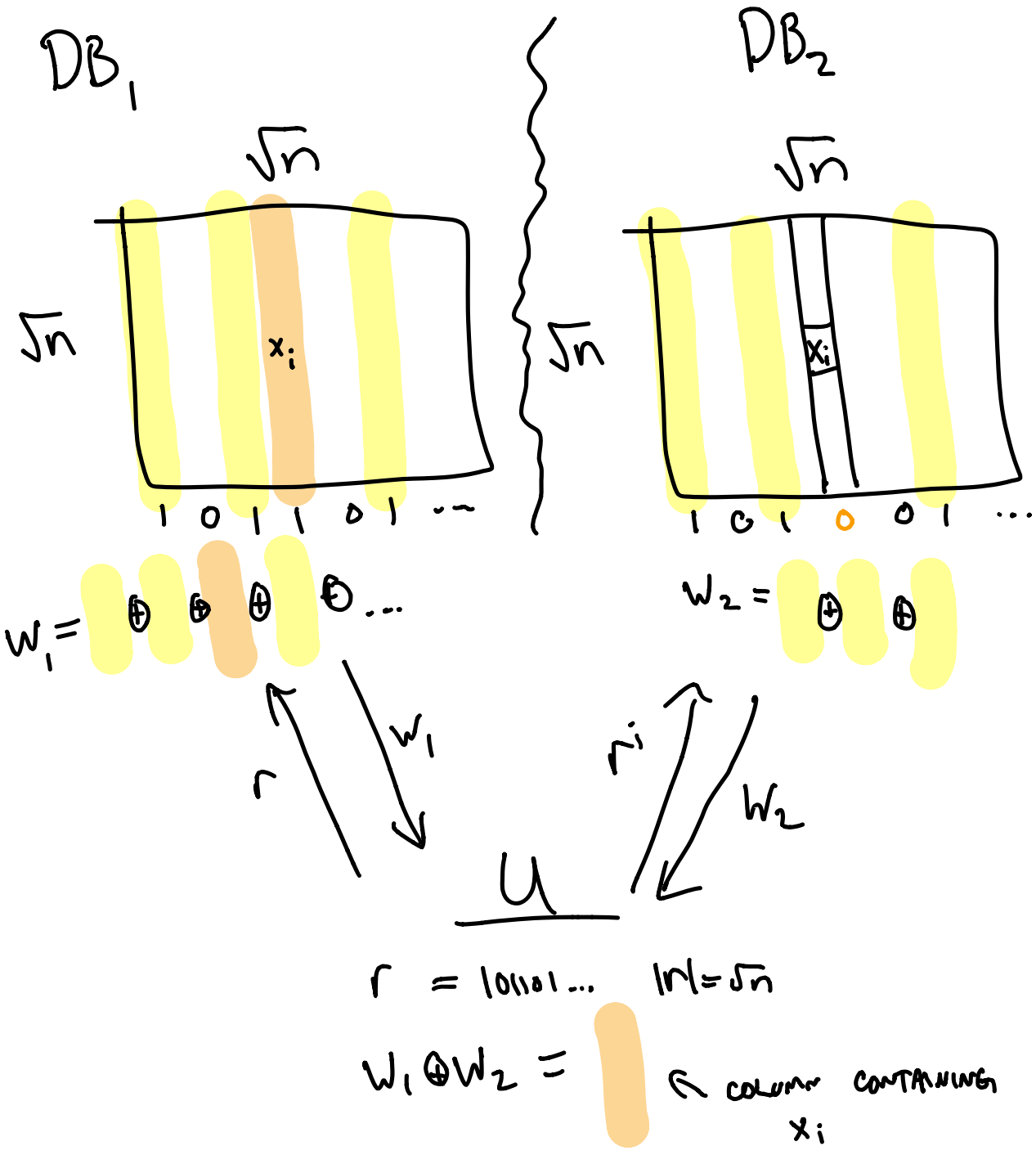
(2) PRIVACY: DB_1 + DB_2 LEARN NOTHING ABOUT i

(3) COMMUNICATION COMPLEXITY $|Q_1| + |Q_2| + |A_1| + |A_2| \ll n$

AS LONG AS DBs DON'T COMMUNICATE,

CAN CONSTRUCT THIS WITH NO CRYPTOGRAPHIC ASSUMPTIONS

$\sqrt{n}$ SOLUTION:



COMMUNICATION: $Q_1 = Q_2 = \sqrt{n}$

$A_1 = A_2 = \sqrt{n}$

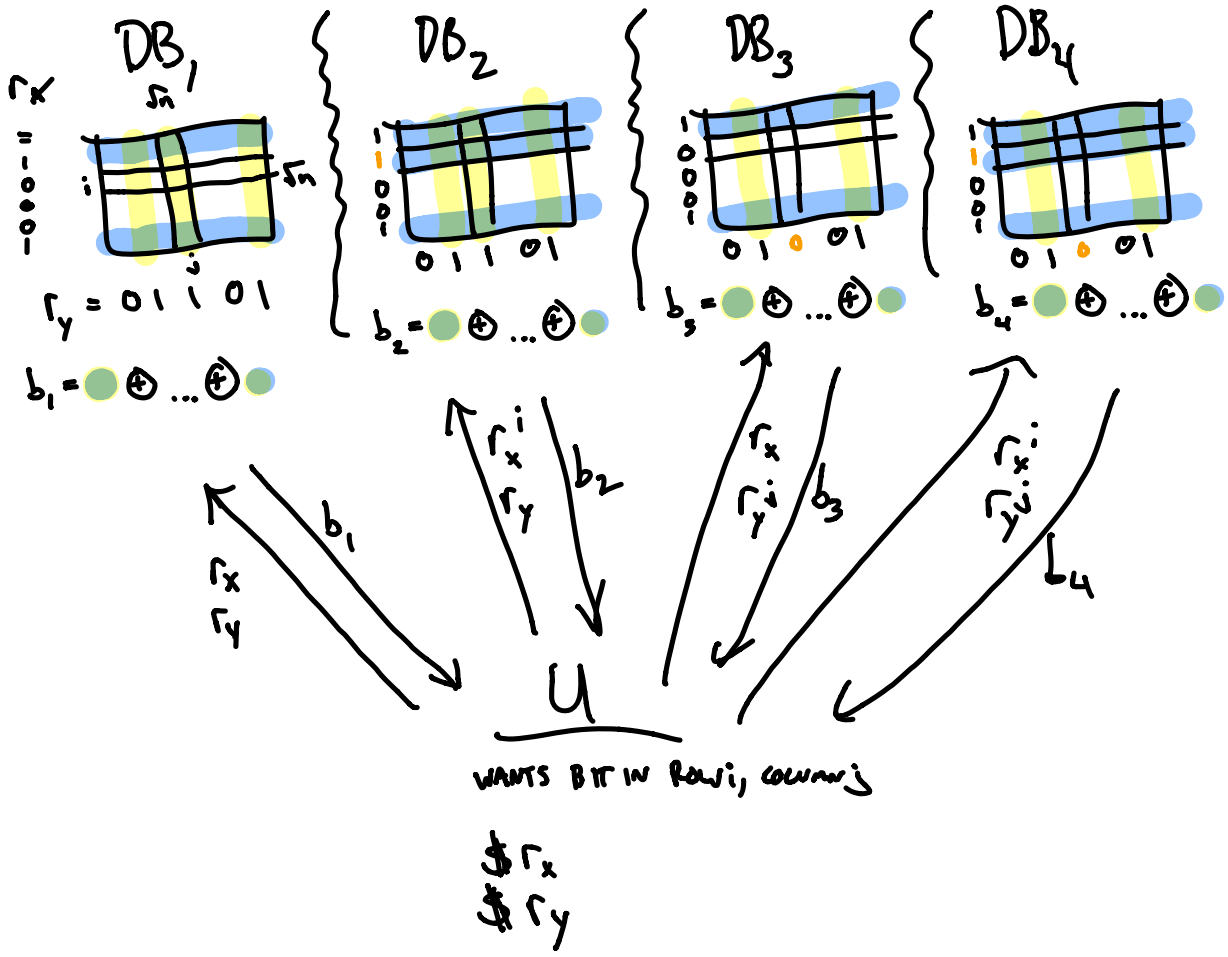
$|Q| + |A| = 4\sqrt{n}$ ✓

PRIVACY: Γ, Γ' ARE PAIRWISE INDEPENDENT.

LOOK RANDOM UNLESS DBS

TALK TO EACH OTHER.

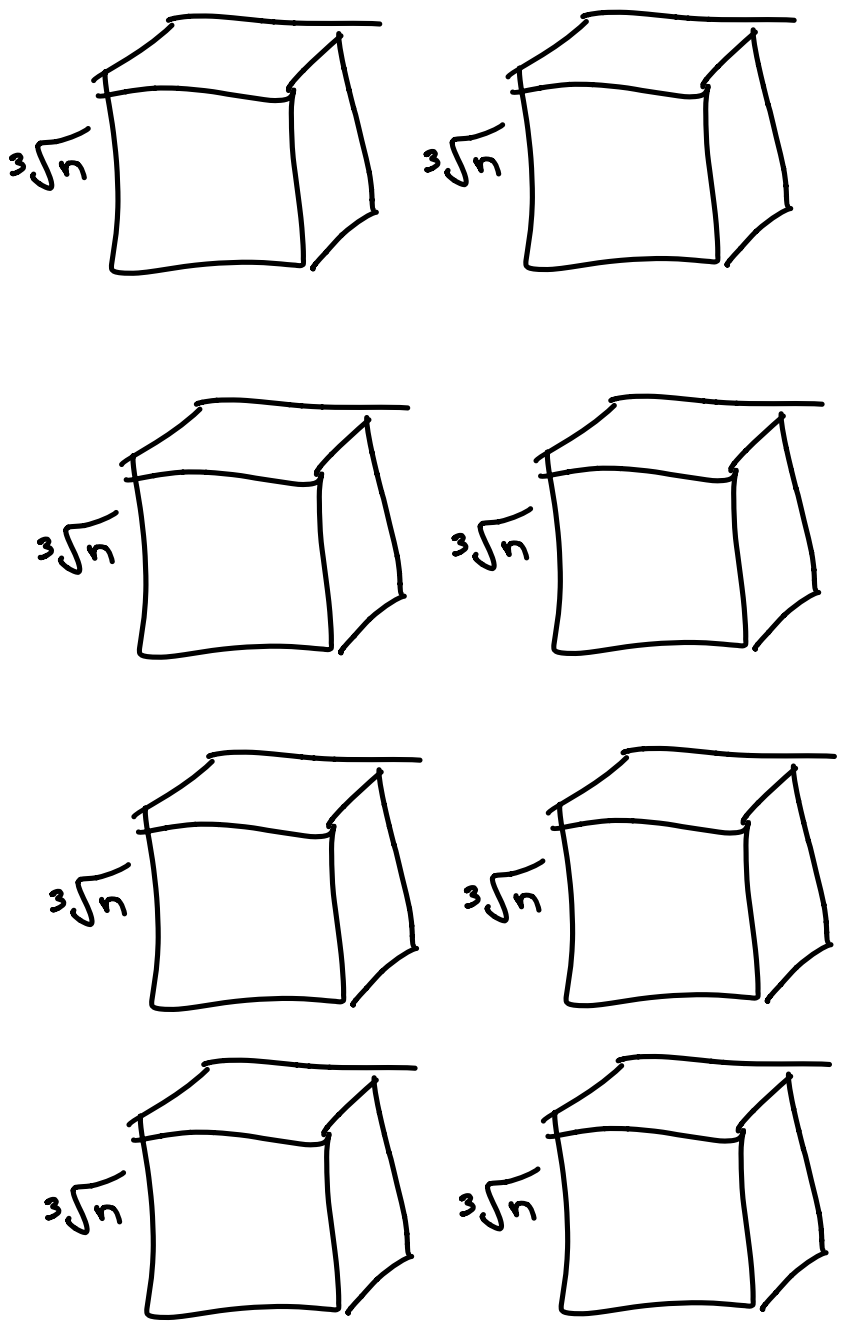
$3\sqrt{n}$ SOLUTION : SUPPOSE WE HAVE 4 DB COPIES (EASIER)



$b_1 \oplus b_2 \oplus b_3 \oplus b_4 =$ BIT IN (i,j)

COMMUNICATION: $8\sqrt{n} + 4$, STILL $\sqrt{n}$

TRY 3-DIMENSIONAL DATABASE!



USE SAME TRICK AS BEFORE.

U WANTS (i,j,k). SENDS

$r_x, r_y, r_z \longrightarrow DB_1$

$r_x^i, r_y, r_z \longrightarrow DB_2$

$r_x, r_y^j, r_z \longrightarrow DB_3$

$r_x, r_y, r_z^k \longrightarrow DB_4$

$r_x^i, r_y^j, r_z \longrightarrow DB_5$

$r_x, r_y^j, r_z^k \longrightarrow DB_6$

$r_x^i, r_y, r_z^k \longrightarrow DB_7$

$r_x^i, r_y^j, r_z^k \longrightarrow DB_8$

GETS BACK $b_1, \ldots, b_8$.

$b_1 \oplus \ldots \oplus b_8$ IS PRECISELY LOCATION (i,j,k).

COMMUNICATION: $24\sqrt[3]{n} + 8$

BUT HOW DO WE DO IT W/

2 DATABASES?

EACH DB SIMULATES $\mathcal{U}$

DBS FROM PREVIOUS SOLUTION.

DB_1 GETS r_x, r_y, r_z

COMPUTES $r_x^i, r_y, r_z \quad \forall i \in [\sqrt{n}]$

COMPUTES $r_x, r_y^j, r_z \quad \forall j \in [\sqrt{n}]$

COMPUTES $r_x, r_y, r_z^k \quad \forall k \in [\sqrt{n}]$

RETURNS $b_1, \ldots, b_{\sqrt{n}}$ TO $\mathcal{U}$

DB_2 DOES SAME WITH

$$r_x^i, r_y^j, r_z$$

$$r_x, r_y^j, r_z^k$$

$$r_x^i, r_y, r_z^k$$

$$r_x^i, r_y^j, r_z^k$$

RETURNS $d_1, \ldots, d_{3\sqrt{n}}$

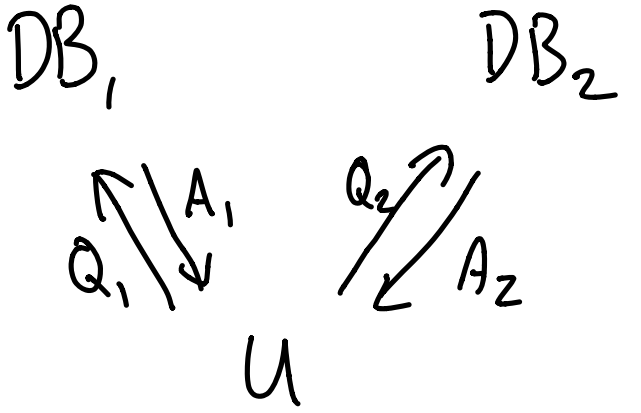
$\mathcal{U}$ COMPUTES $b_\ell \oplus d_\ell \; \forall \ell \in [3\sqrt{n}]$ TO GET COLUMN CONTAINING SPECIAL BIT.

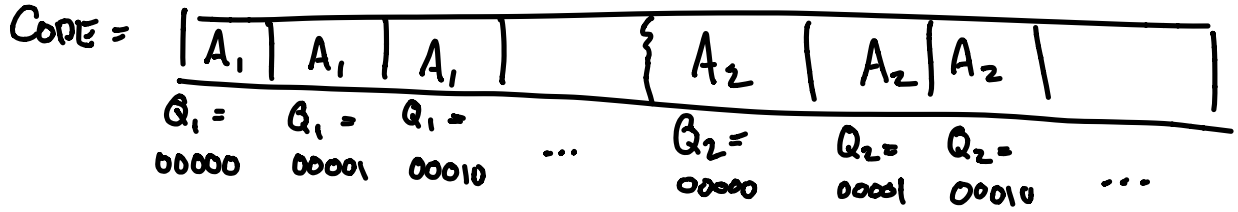
Connection to Locally Decodable Codes (LDC)

Goal: Encode message such that if some bits are corrupted, can recover lost information.

IDEA: QUESTIONS ARE RANDOM, BUT GIVE SAME ANSWER!

WANT TO ENCODE i^{TH} BIT OF DB





ANSWERS ARE ALWAYS VALID NO MATTER WHAT VALUE OF Q_1, Q_2 IS USED!

EVEN IF ADVERSARY WIPES OUT LARGE MAJORITY OF CODE, WE WILL HAVE SOME UNTOUCHED Q_1, Q_2 BLOCK WITH HIGH PROBABILITY, FROM WHICH WE CAN RECONSTRUCT i^{TH} BIT OF DB.