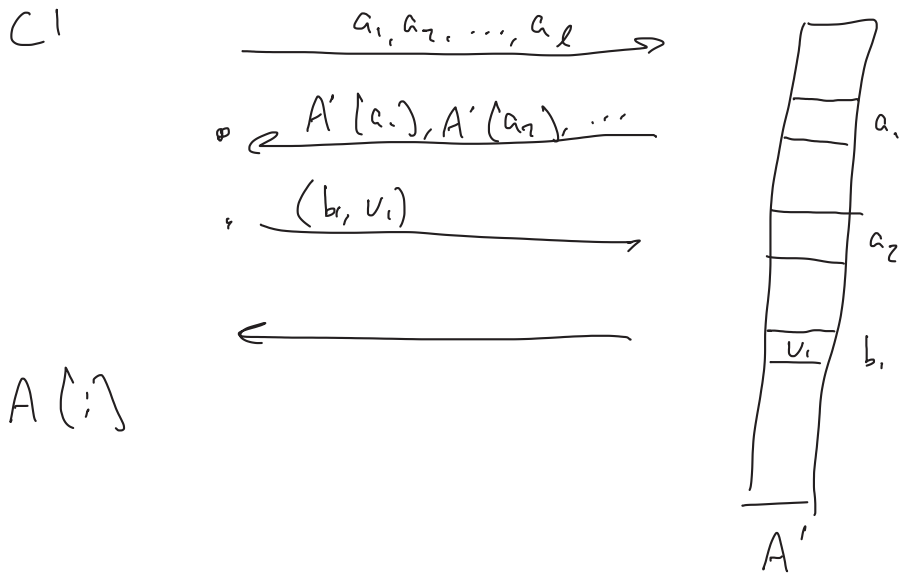
- Scribes?
- lecture recording
- final exam

Client (i)

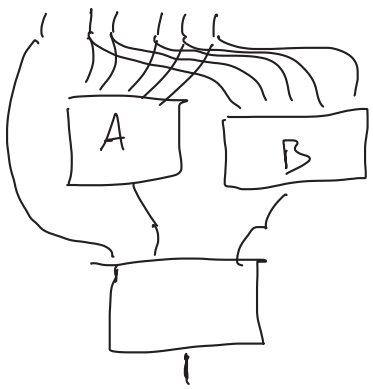
Server



Previous protocols all based on a circuit model of computation

```
if (x1 == 0)
    A
else
    B
```

running time max{A, B}



running time A+B

```
for (i=1 to N)
    A
```

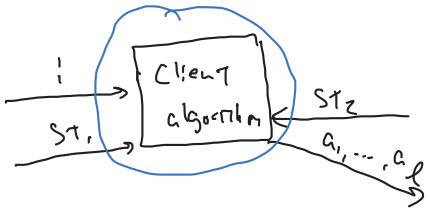
binary search on array of length N $\quad F(A, i) = A[i]$

RAM model : $O(\log N)$

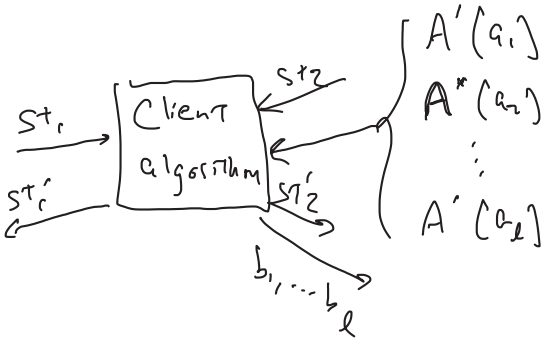
circuit model : $O(N)$

$P_1(i)$ $\quad P_2(A')$ $\quad A'$ is the initial ORAM memory corresponding to a sorted array A

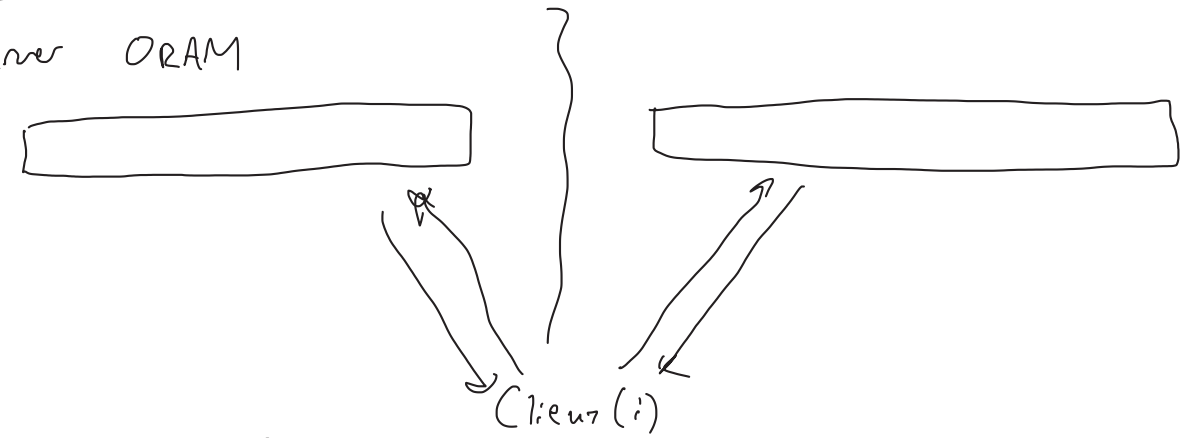
st_1

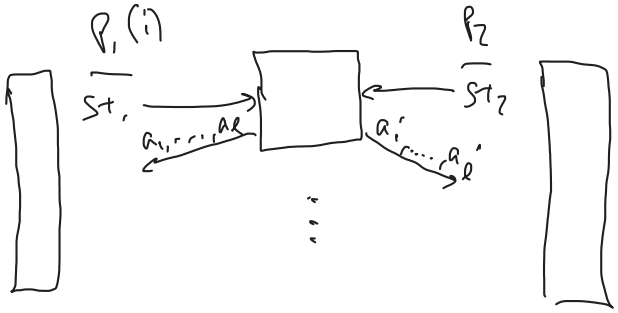


$st_1 \oplus st_2$ = initial state for the client in the ORAM



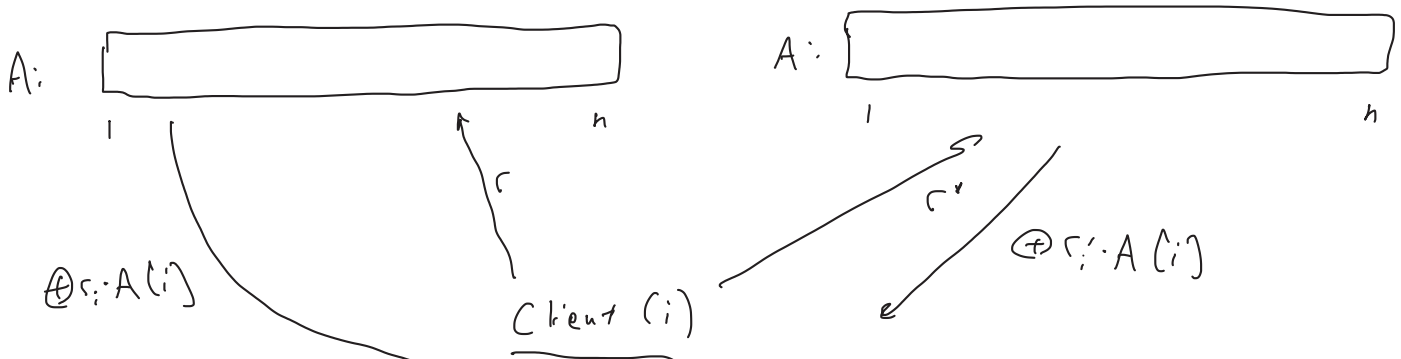
2-server ORAM





Private information retrieval (PIR)

2-server PIR



generate $r \leftarrow \{0,1\}^n$

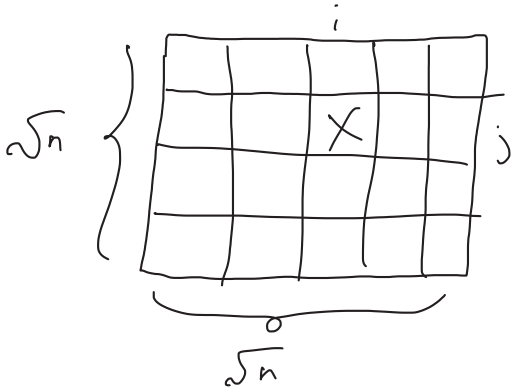
$r' = r$ w/ i^{th} bit flipped

// $r \oplus r' = 0 \cdots 0\,1\,0 \cdots 0$

client outputs $(\oplus r_i \cdot A[i]) \oplus (\oplus r'_i \cdot A[i])$

$= A[i]$

4-server PIR



Client is interested in learning the value of the array at some position (i,j)

Client chooses $r, s \leftarrow \{0,1\}^{\sqrt{n}}$

$r' = r$ w/ i^{th} bit Flipped

$s' = s$ w/ j^{th} Flipped

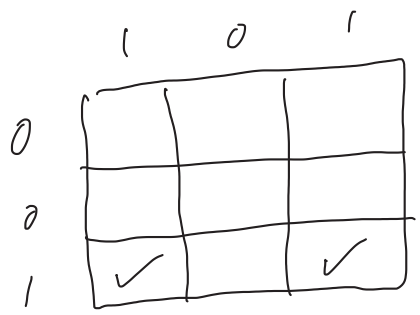
send (r, s) to server 1

(r', s) to server 2

(r, s') to server 3

(r', s') to server 4

Each server sends $\oplus r_i s_j \cdot A[i][j]$, client XORs the results $\Rightarrow A[i][j]$



$A[3][1] \oplus A[3][3]$

$r = 101$

$s = 001$

2^d - sever scheme using $O(n^{1/d})$ communication

e.g., $d=3$, 8 servers

client has index (i,j,k) $\quad 1 \le i,j,k \le \sqrt[3]{n}$

$r, s, t \leftarrow \{0,1\}^{\sqrt[3]{n}}$

$r' = r$ w/ i^{th} bit flipped

$s' = s$ w/ j^{th} bit flipped

$t' = t$ w/ k^{th} bit flipped

- (r, s, t) to server 1
- (r', s, t) to server 2
- (r, s', t) to server 3
- (r, s, t') to server 4
- (r', s', t) to server 5
- (r', s, t') to server 6
- (r, s', t') to server 7
- (r', s', t') to server 8

$\Rightarrow$ 2-Sever Scheme w/ Comm. $O(n^{1/3})$

Server A will emulate servers 1–4

- Send to server A (r, s, t)
- Compute server 1 response
- Compute server 2 response For every possible r'
- Compute server 3/4 response For every possible s'/t'