Private Information Retrieval

Gihyuk Ko

Carnegie Mellon University

original slides from Amos Beimel

Private Information Retrieval (PIR) [CGKS95]

- **Goal**

  allow user to query database while hiding the identity of the data-items she is after.

- **Note**

  hides identity of data-items; not existence of interaction with the user.

- **Motivation**

  patent databases; stock quotes; web access; many more....

- **Paradox(?)**

  imagine buying in a store without the seller knowing what you buy.

(Encrypting requests is useful against third parties; not against owner of data.)

Modeling

- **Server:** holds n-bit string x

  n should be thought of as very large

- **User:** wishes

  (to retrieve x_i), while (keeping i private)

- **Remark:** most basic version;

  building block for involved retrieval.

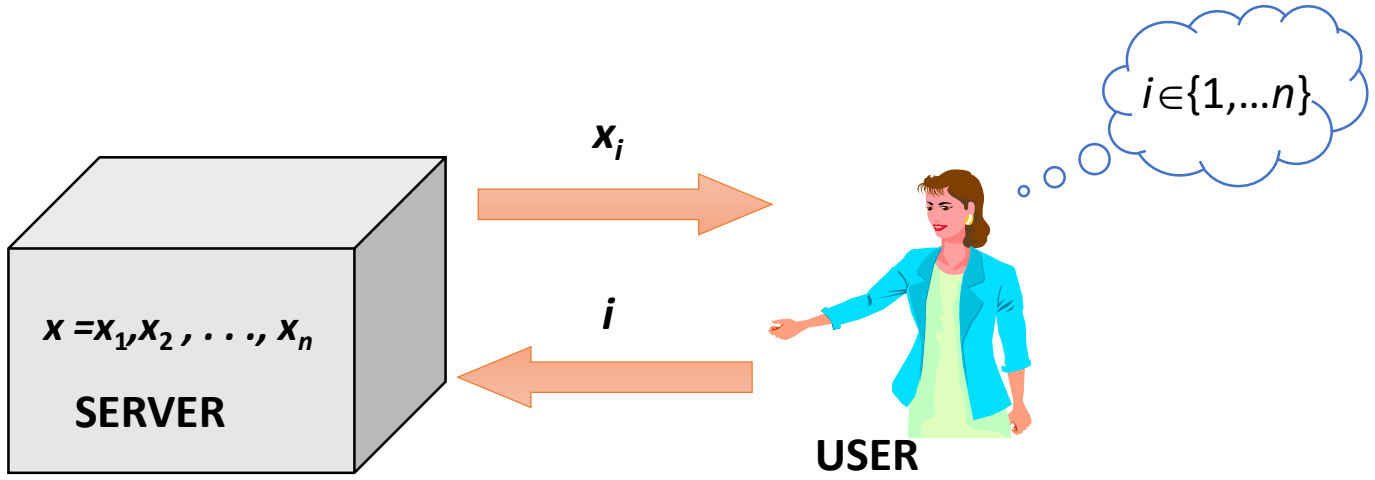
Private Information Retrieval (PIR) [CGKS95]

$x=x_1,x_2,\ldots,x_n \in \{0,1\}^n$

SERVER

USER

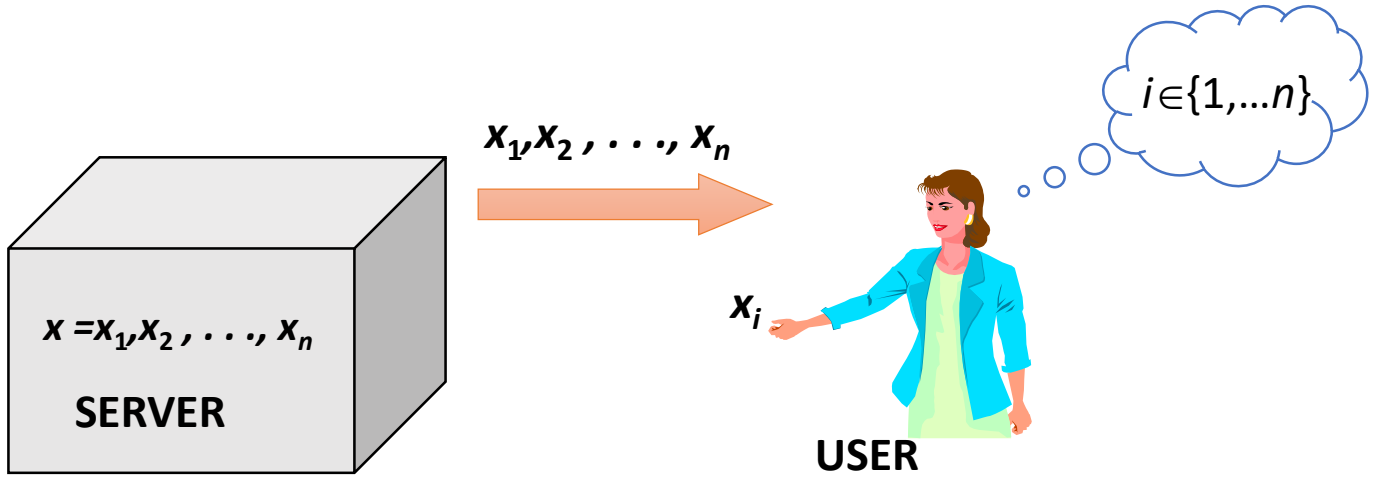
Non-Private Protocol



NO privacy!!!

Communication cost: **log *n***

Trivially Private Protocol



Server sends entire database ***x*** to User.

Information theoretic privacy

Communication cost: ***n***

…is this optimal?

Challenge

Theorem [CGKS]:

In any 1-server PIR with information-theoretic privacy,
the communication cost is at least n.

More “solutions”

- User asks for additional random indices.

  Drawback: reveals a lot of information

- Employ general crypto protocols to compute xi privately.

  Drawback: highly inefficient (polynomial in n).

- Anonymity (e.g., via Anonymizers).

  Note: different concern: hides identity of user, not x_i.

Two Approaches

Information-Theoretic PIR [CGKS95,Amb97,...]

Replicate database among k servers.

Unconditional privacy against t servers.

Default: t=1

Computational PIR [CG97,KO97,CMS99,...]

Computational privacy, based on cryptographic assumptions.

Known Computational Upper Bounds

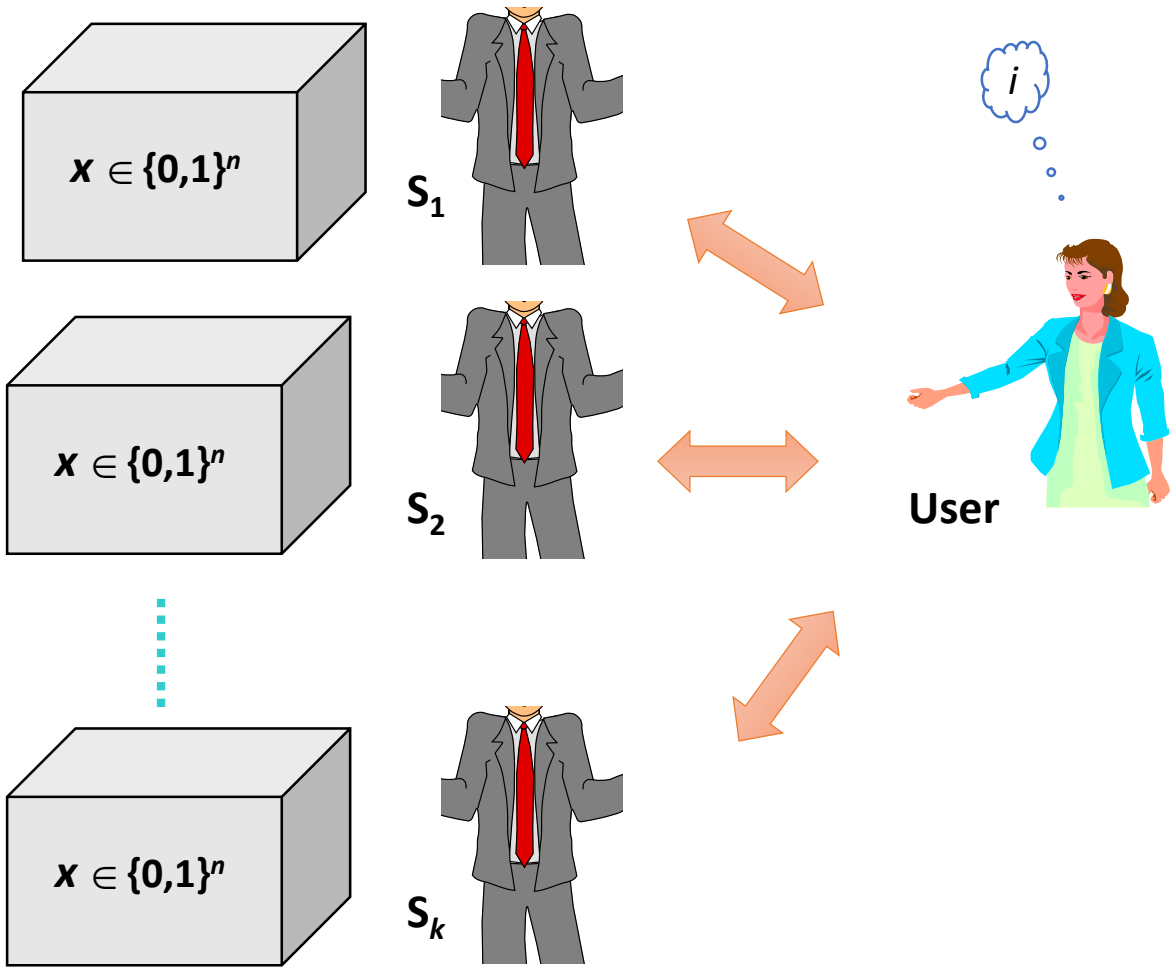
Multiple servers, information-theoretic PIR:

- 2 servers, comm. $n^{1/3}$ [CGKS95]
- k servers, comm. $n^{1/\Omega(k)}$ [CGKS95, Amb96,…,BIKR02]
- log n servers, comm. Poly(log(n)) [BF90, CGKS95]

Single server, computational PIR:

- Comm. Poly(log(n))
  - Under appropriate computational assumptions [KO97,CMS99]

Approach I: k-Server PIR



Correctness: User obtains x_i

Privacy: No single server gets information about i

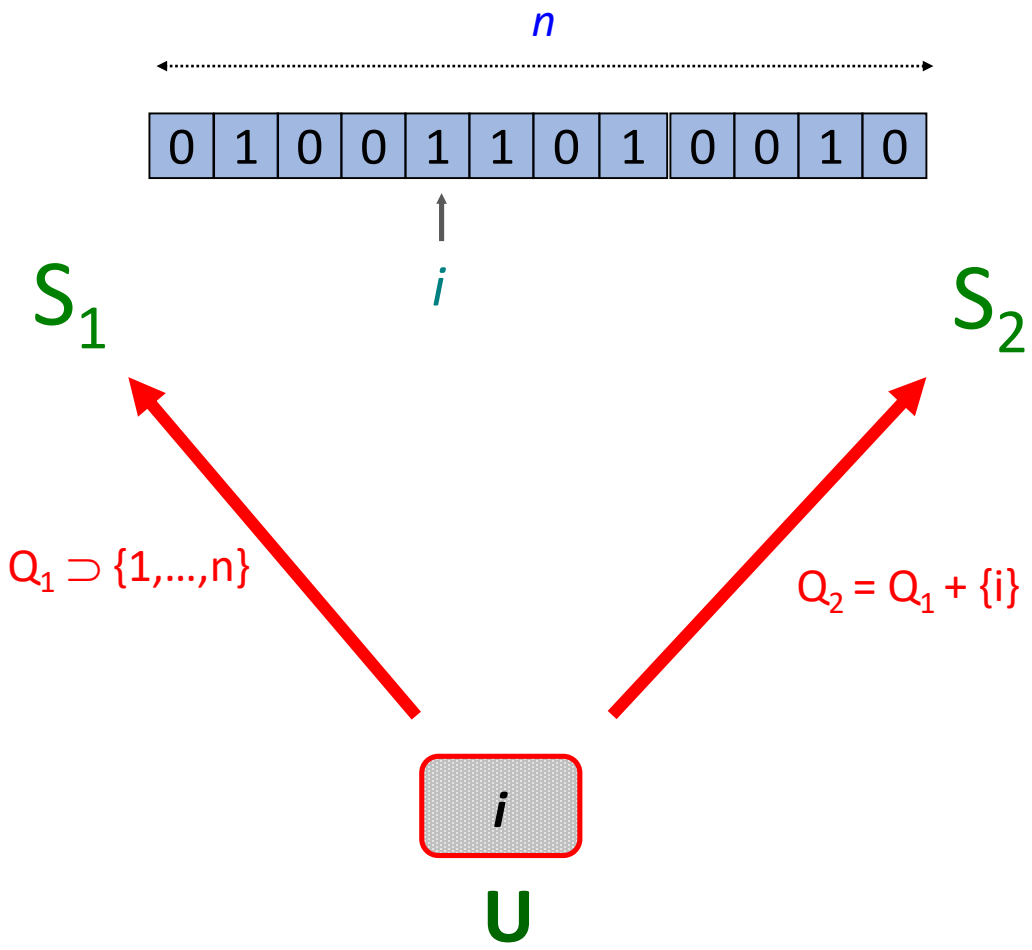
Information-Theoretic 2-Server PIR

- Best Known Protocol: communication cost $n^{1/3}$ [CGKS95]
- Open Question: Is this optimal?
- This Talk: comm. $n^{1/2}$

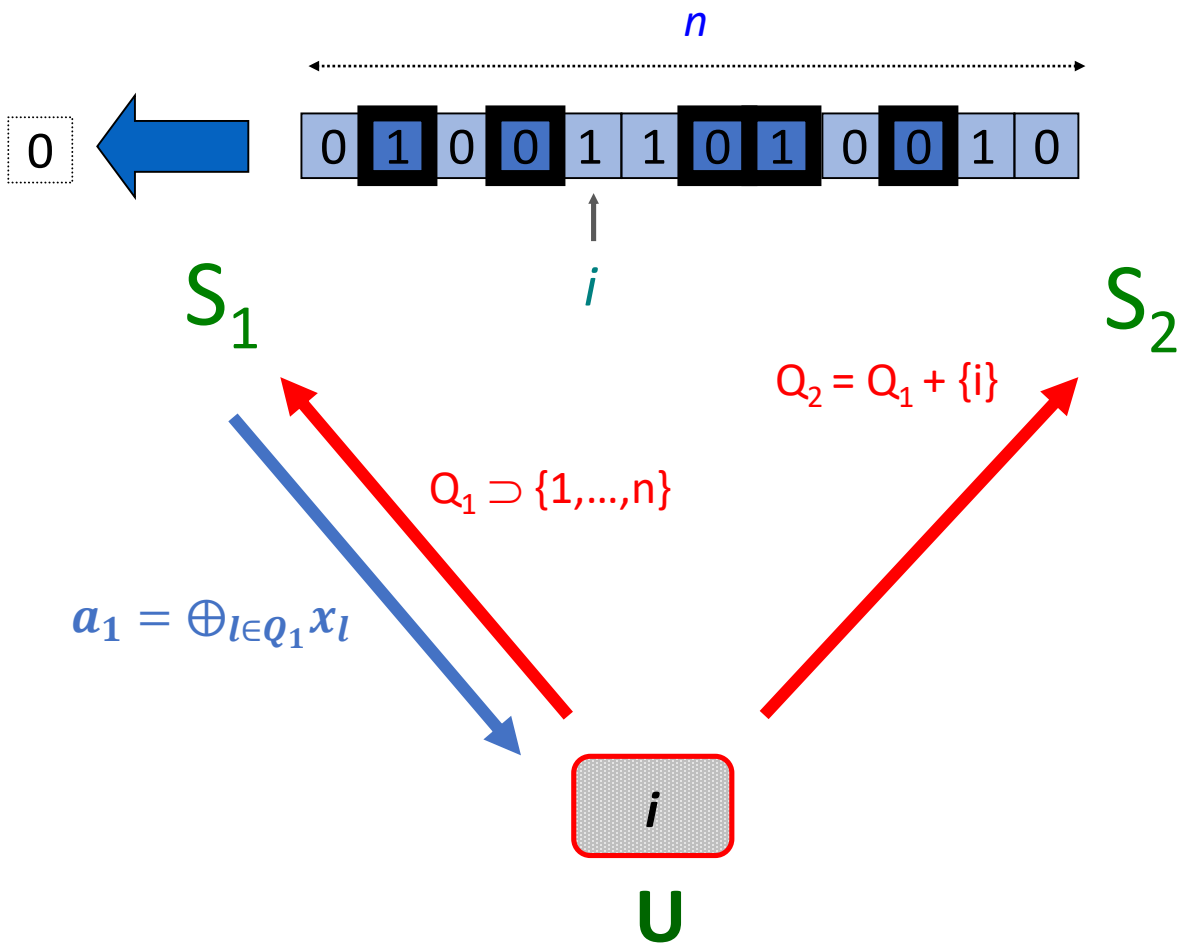
Two Stages:

1. Protocol I: n bit queries, 1 bit answers
2. Protocol II: $n^{1/2}$ bit queries, $n^{1/2}$ bit answers

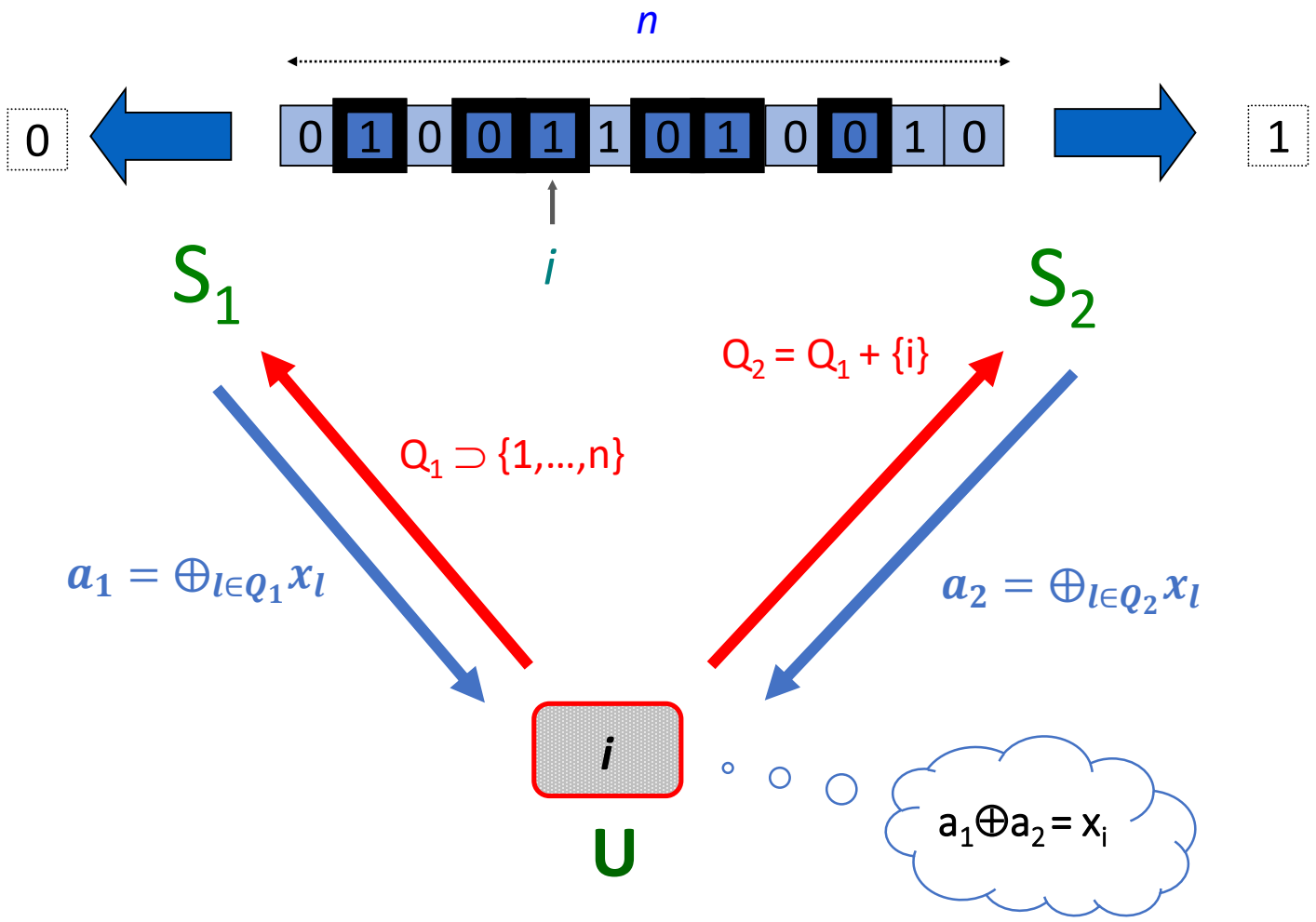
Protocol I: 2-Server PIR



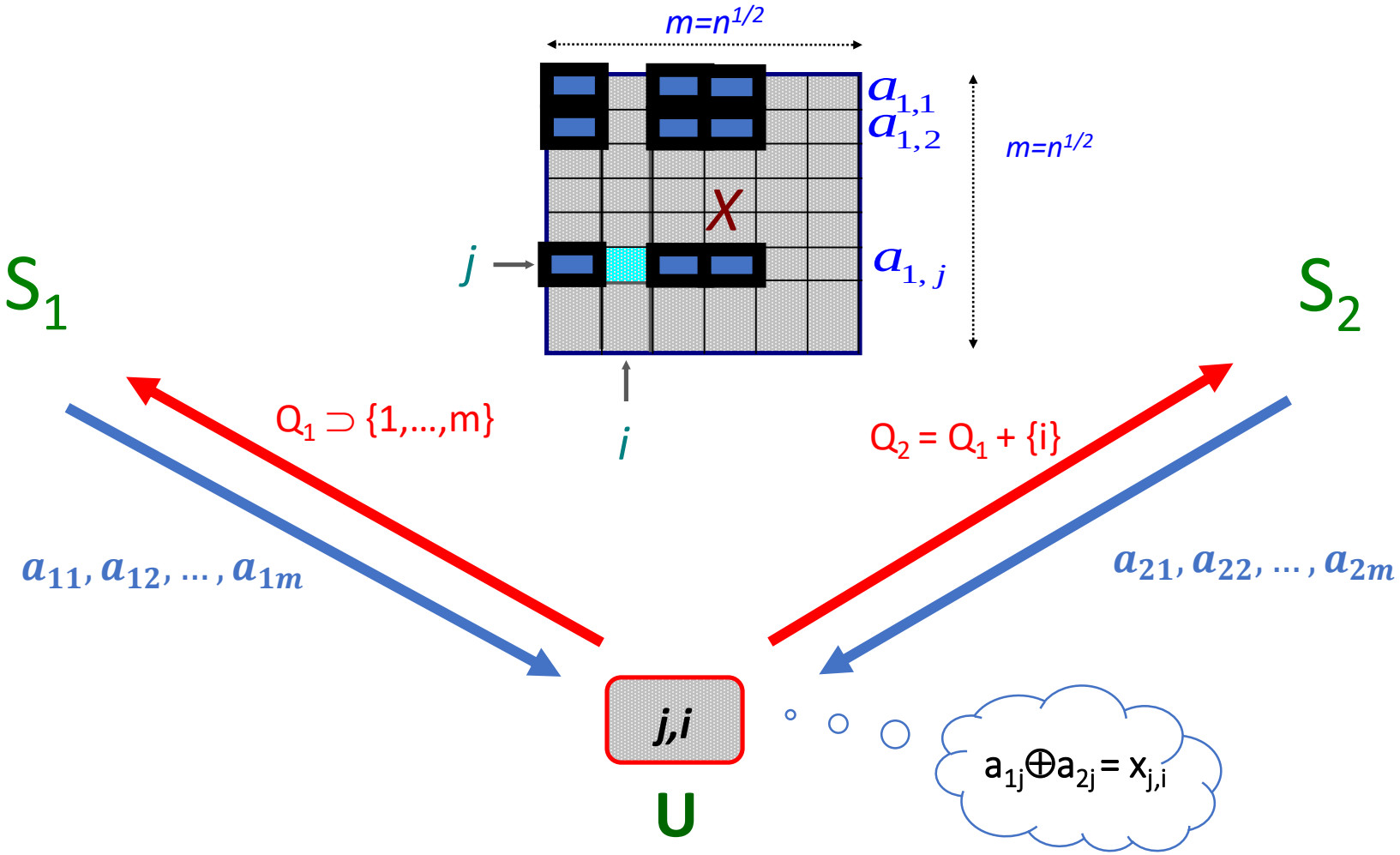
Protocol I: 2-Server PIR



Protocol I: 2-Server PIR



PIR with $O(n^{1/2})$ Communication



Comp. PIR with $O(n^{1/2})$ Communication

Tool: (randomized) homomorphic encryption

Example Quadratic Residue:

E(0) → QR_N

E(1) → NQR_N

QR ~~×~~ QR = QR

NQR ~~×~~ QR = NQR

PIR – Related Work

- Extensions:
  - Symmetric PIR [GIKM,NP]
  - t-privacy [CGKS,IK,BI]
  - Robust PIR [BS]

- More settings [OS,GGM,DIO,BIM,...]
- PIR as a building-block [NN,FIMNSW,...]

Future Directions

Focus so far: communication complexity

Obstacle: time complexity

- All existing protocols require high computation by the servers (linear computation per query).

Theorem [BIM]:

Expected computation of the servers is $\Omega(n)$

Major research goal: Improving time complexity via preprocessing / amortization / off-line computation (Preliminary results in [BIM,IKOS])

Thanks!