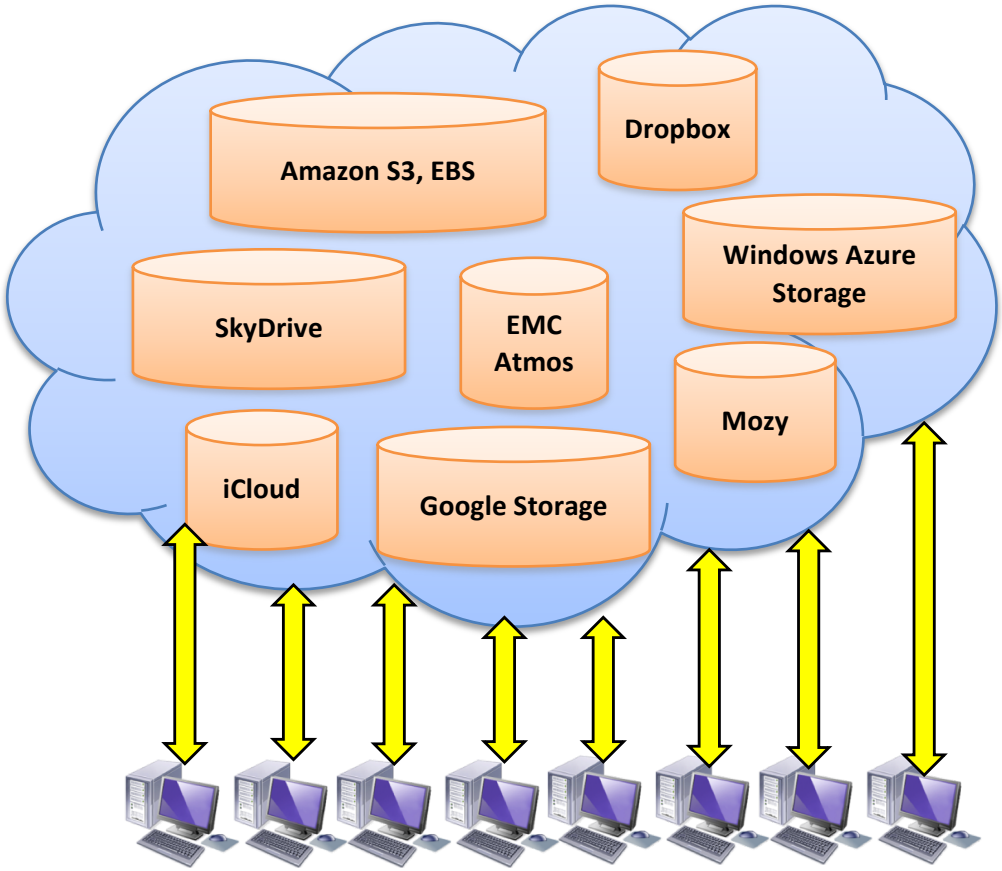
COM-402
Hiding Patterns from the Database Owner

- Motivation

- Private Information Retrieval (PIR)
  - IT-PIR
  - cPIR

Cloud Storage



Can we **TRUST** the cloud?

Data Privacy

- ***Data privacy*** is a growing concern.
  - Large attack surface (possibly hundreds of servers)
  - Infrastructure bugs
  - Malware
  - Disgruntled employees
  - Big brother

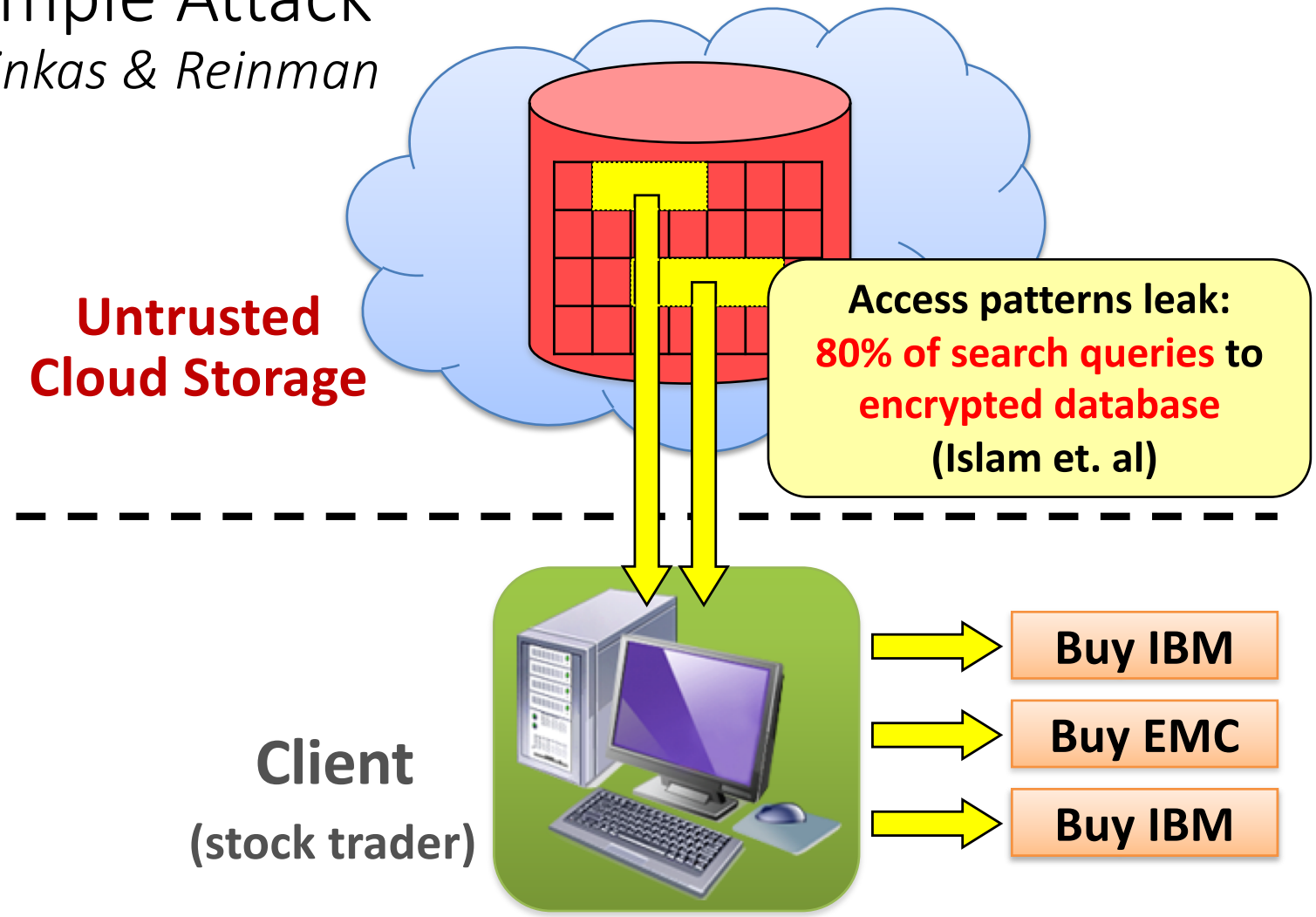
- So, many organizations ***encrypt*** their data.

But, encryption is not always enough.

Access patterns
can leak sensitive information.

Example Attack

by Pinkas & Reinman



If a sequence of data access requests is always followed by a stock exchange operation, the server can gain sensitive information even when the data is encrypted

Security for Outsourced Storage

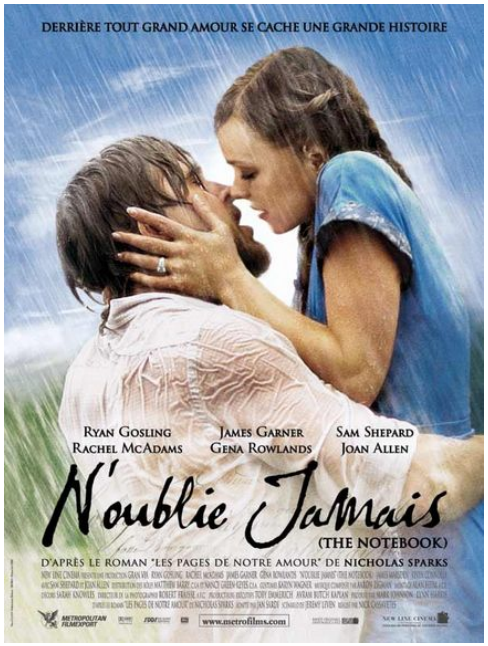
- **Confidentiality**
  - Encrypt
- **Integrity**
  - MAC / Sign
  - Merkle tree
- **Reliability**
  - Redundancy
  - Proofs of retrievability (PoR)
- **Access privacy?**
  - **Private Information Retrieval (PIR)**
  - **Oblivious RAM (ORAM)**

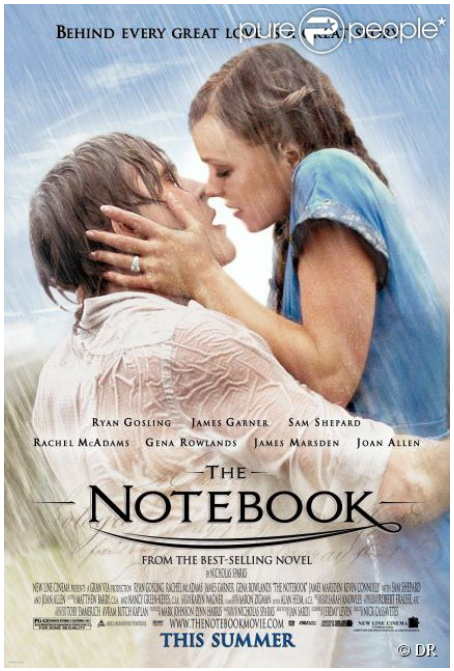
Private Information Retrieval (PIR)

A Real-World Example

Suppose there is a movie database and I want to find information on the movie *The Notebook*.

I don't want **the database operator** to know about my interest in this movie.

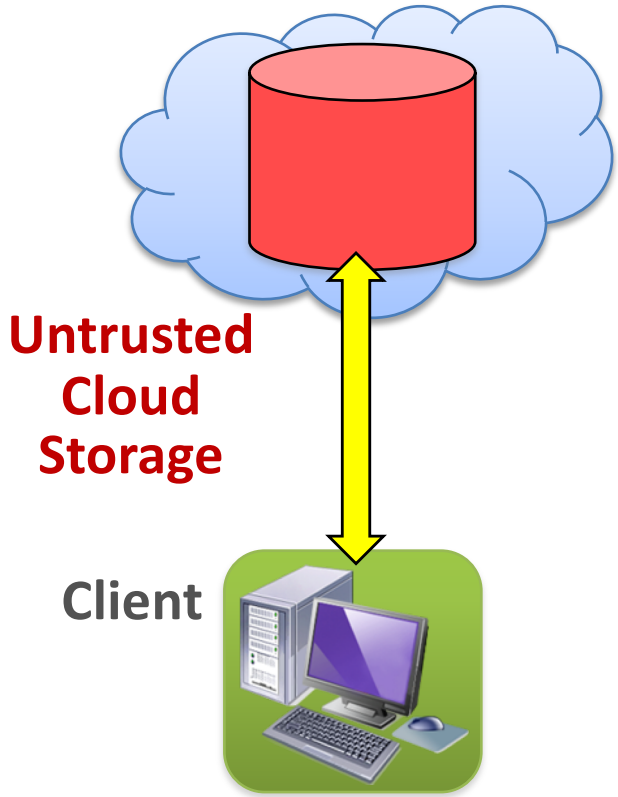




Private Information Retrieval (PIR)

- **Goal:** Protect privacy of **user's queries**.

- The database does not learn the query terms or responses.

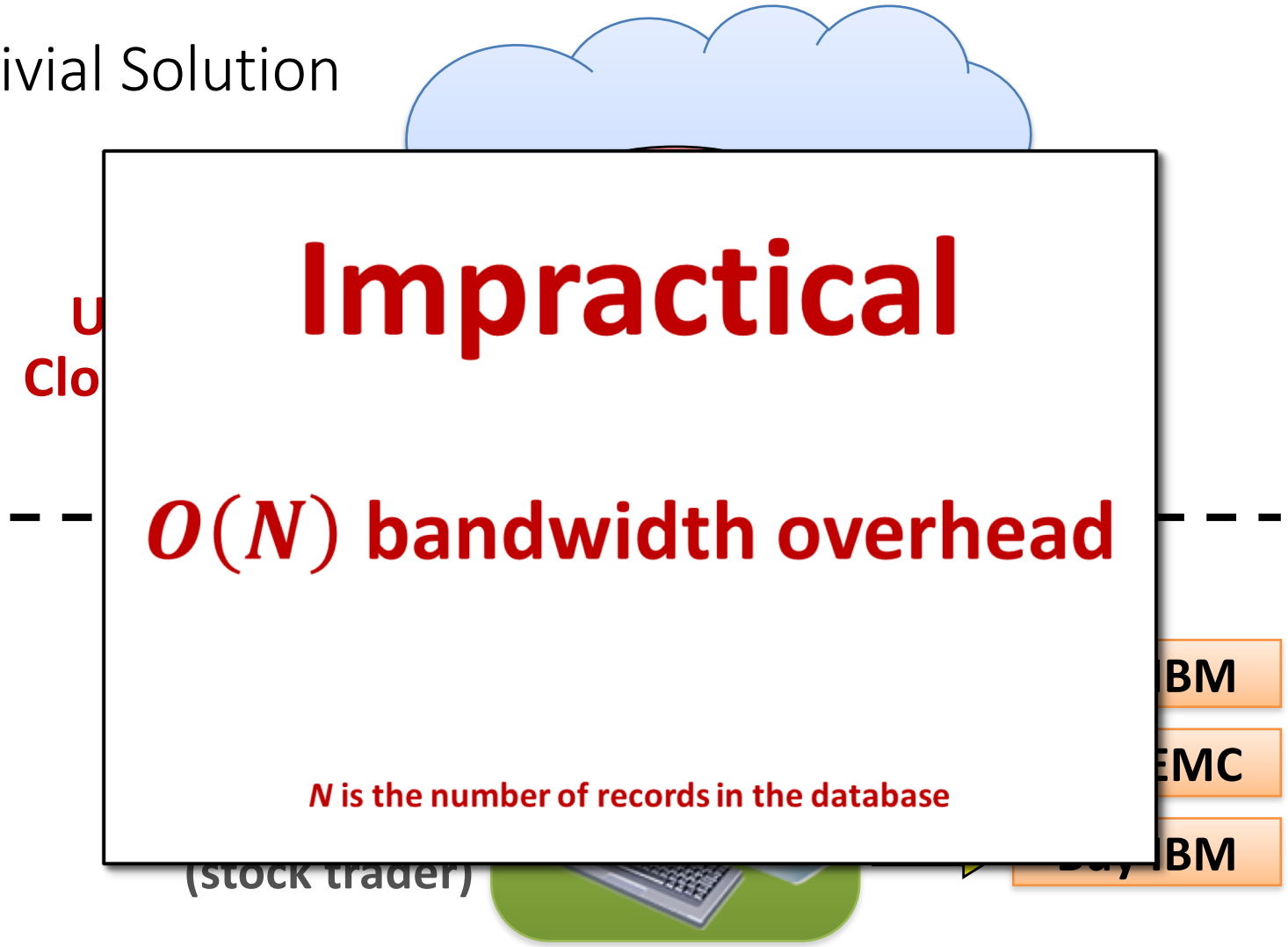


But…

How to do this?

Download the entire Database?

Trivial Solution



IT-PIR vs. cPIR

- **Information Theoretic PIR (IT-PIR)**
  - Non-colluding L servers.
  - Each server holds a copy of the database.
  - **Perfectly secure** if some number of these servers are not colluding.

- **Computational PIR (cPIR)**
  - Single database-server.
  - Uses cryptographic techniques to encrypt the user's query.
  - The security of cPIR relies on the security of the underling encryption.
  - Privacy is ensured **only against computationally-bounded attackers.**

IT-PIR: the Goal

- Suppose there is a database with blocks $D_1,\ldots,D_r$
- A client wants to retrieve block D_β from the database in such a way that the database operator learns **nothing** about β.
- Do this without downloading the entire database.

IT-PIR: Goldberg's Scheme [Gol07]

A database of *r* blocks is represented as an *r x s* matrix *D* and by using linear algebra, one can get the β^{th} block (β^{th} row) of *D* by doing:

$$D_\beta = e_\beta \cdot D$$

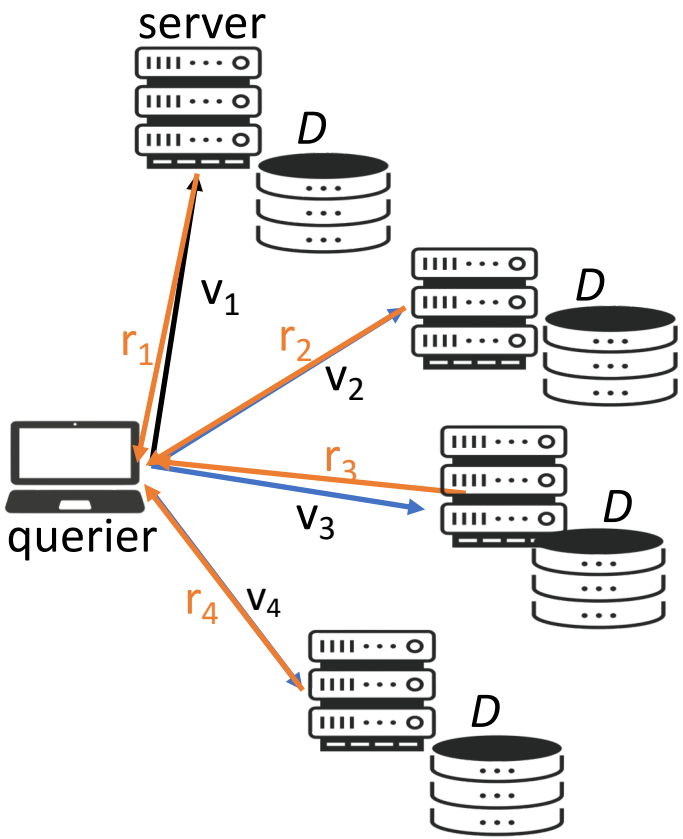
where e_β = [0 0 … 1 … 0] is a vector with all zeros, except a one for the β coordinate.

IT-PIR: Goldberg's Scheme (ct'd)

- To hide which information is retrieved:
  - The database is stored (duplicated) in *l* servers.
  - The querier creates secret shares

    $v_1, \ldots, v_l$

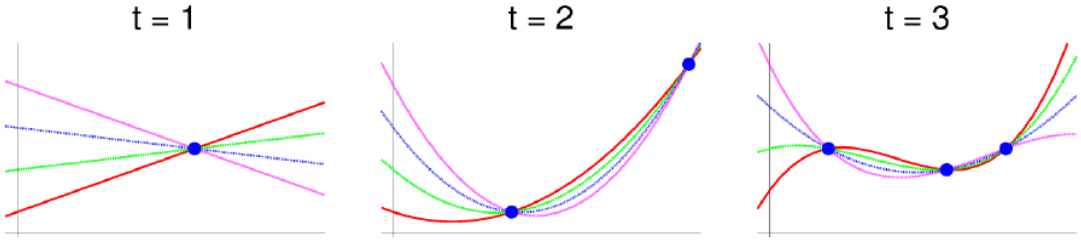
    of e_β and sends them such that each server only gets one.
  - Each server computes $r_i = vi \cdot D$ and sends it back.
  - The querier retrieves $e_\beta = \Sigma\, r_i$



IT-PIR: Sharing & Robustness

Sharing: The shares can be created using Shamir Secret Sharing [Sha79] :

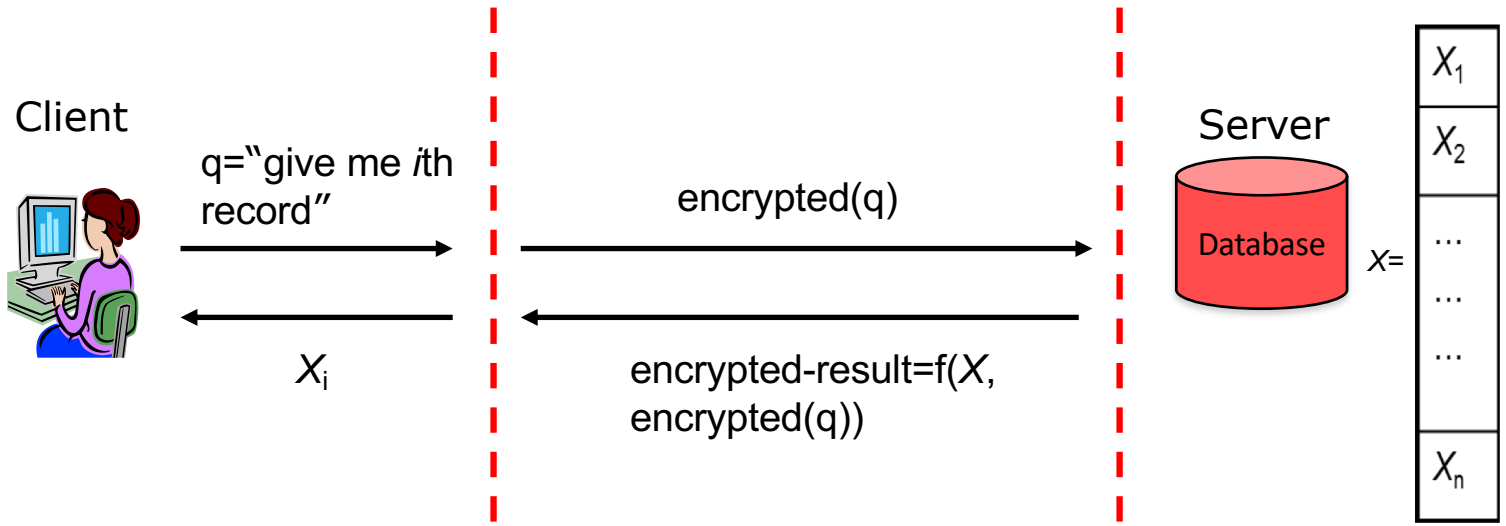
- Each party knows 1 point of a unique of order *t* going through the *t*+1 points.
- The intersection with the *y*-axis is the shared secret.



Robustness: Shares can be encoded using specific encodings (i.e., Reed-Solomon) that enables the final response to be constructed from any subset (larger than a pre-defined size) of the shares.

Computational PIR (cPIR)

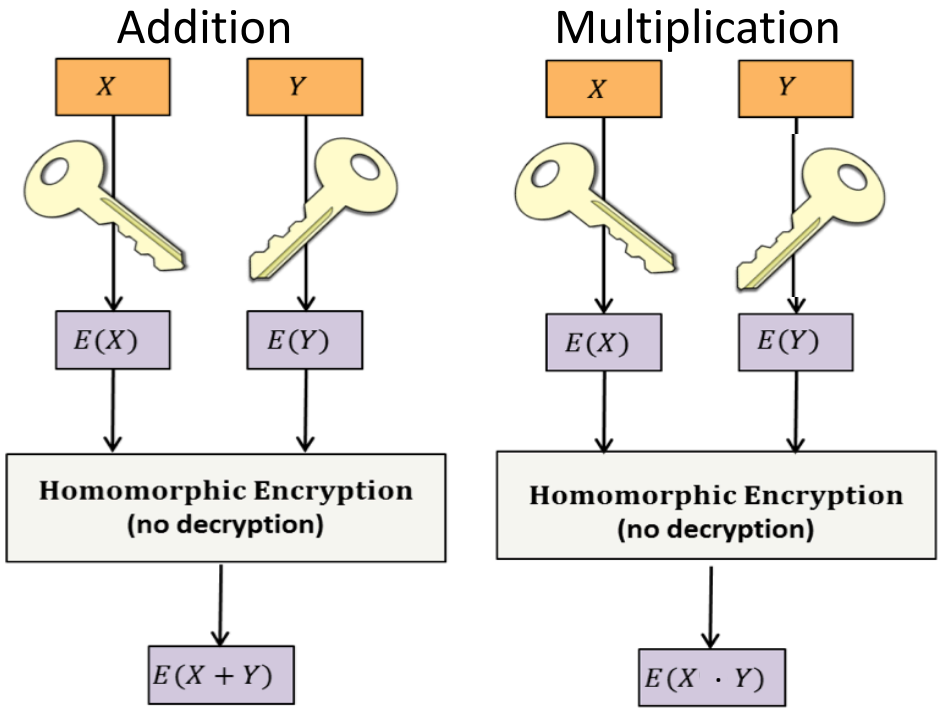
- User privacy is related to the (assumed) intractability of a mathematical problem.
- **Principle:** Achieve computationally complete privacy by applying cryptographic computations over the entire public data.



cPIR: Theoretical Background

cPIR can be implemented by using homomorphic encryption (more on this very soon !)

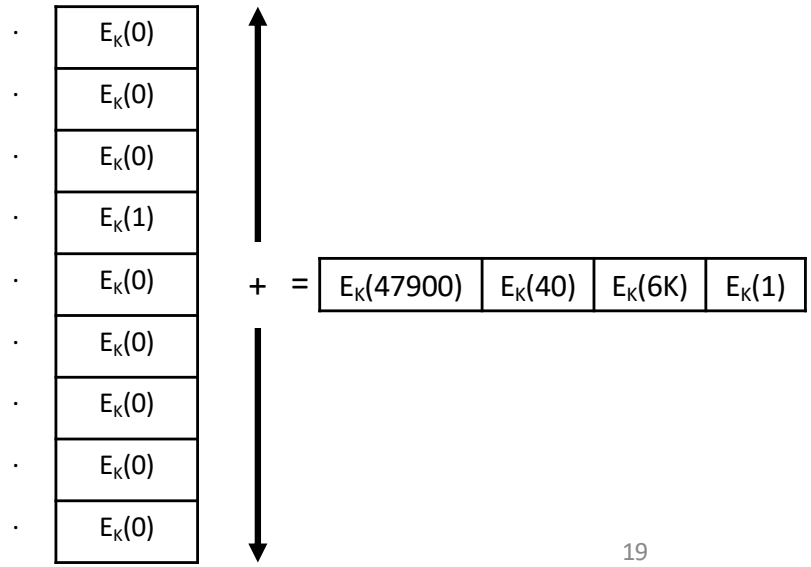
<u>(Fully) Homomorphic Encryption Intuition:</u>



cPIR: Example Solution

- The database is encrypted under Bob's key **K.**
- To privately retrieve data of *patient 4*, Bob encrypts a vector **V** and sends it to the database server.
- This vector multiplies all the database' lines and the result is summed up such that only *line 4* remains. Bob can then decrypt the result.

	ZIP Code	Age	Salary	Disease
1	$E_K(47600)$	$E_K(20)$	$E_K(3K)$	gastric ulcer ($E_K(0)$)
2	$E_K(47600)$	$E_K(20)$	$E_K(4K)$	gastritis ($E_K(1)$)
3	$E_K(47600)$	$E_K(20)$	$E_K(5K)$	stomach cancer ($E_K(2)$)
4	$E_K(47900)$	$E_K(40)$	$E_K(6K)$	gastritis ($E_K(1)$)
5	$E_K(47900)$	$E_K(40)$	$E_K(11K)$	flu ($E_K(3)$)
6	$E_K(47900)$	$E_K(40)$	$E_K(8K)$	bronchitis ($E_K(4)$)
7	$E_K(47600)$	$E_K(30)$	$E_K(7K)$	bronchitis ($E_K(4)$)
8	$E_K(47600)$	$E_K(30)$	$E_K(9K)$	pneumonia ($E_K(5)$)
9	$E_K(47600)$	$E_K(30)$	$E_K(10K)$	stomach cancer ($E_K(2)$)



Conclusion on PIR

- **PIR** does not allow write operations on the database.
- **PIR** scheme is single-rounded query-answer protocol (common communication pattern in context of databases).
- **PIR** allows multiple users to access the database.
- Data in **PIR** is not necessarily encrypted.

How is it used nowadays ?

PIR incurs high performance overhead and is difficult to use in practice.

Nevertheless:

PIR solutions become more and more practical as the performance of underlying cryptographic schemes are improving.