Lecture 13a

Private Information Retrieval

Stefan Dziembowski

www.crypto.edu.pl/Dziembowski

University of Warsaw

11.01.17

version 1.0

Plan

1. Introduction
2. Construction

Private Information Retrieval (PIR)

In a nutshell:

a protocol that allows to access a database without revealing what is accessed.

Main difference with the secure two-party computations:

1. secrecy of only one party is protected,
2. **on the other hand**: there is a restriction on **communication complexity**.

PIR was introduced in:

B. Chor, E. Kushilevitz, O. Goldreich and M. Sudan,
Private Information Retrieval, Journal of ACM, 1998

Motivation: **AOL search data scandal** (2006)

#4417749:

- `clothes for age 60`
- `60 single men`
- `best retirement city`
- `jarrett arnold`
- `jack t. arnold`
- `jaylene and jarrett arnold`
- `gwinnett county yellow pages`
- `rescue of older dogs`
- `movies for dogs`
- `sinus infection`

→ Thelma Arnold
62-year-old widow
Lilburn, Georgia

Observation

The owners of databases know a lot about the users!

This poses a risk to users' privacy.

E.g. consider database with stock prices…

Can we do something about it?

problematic

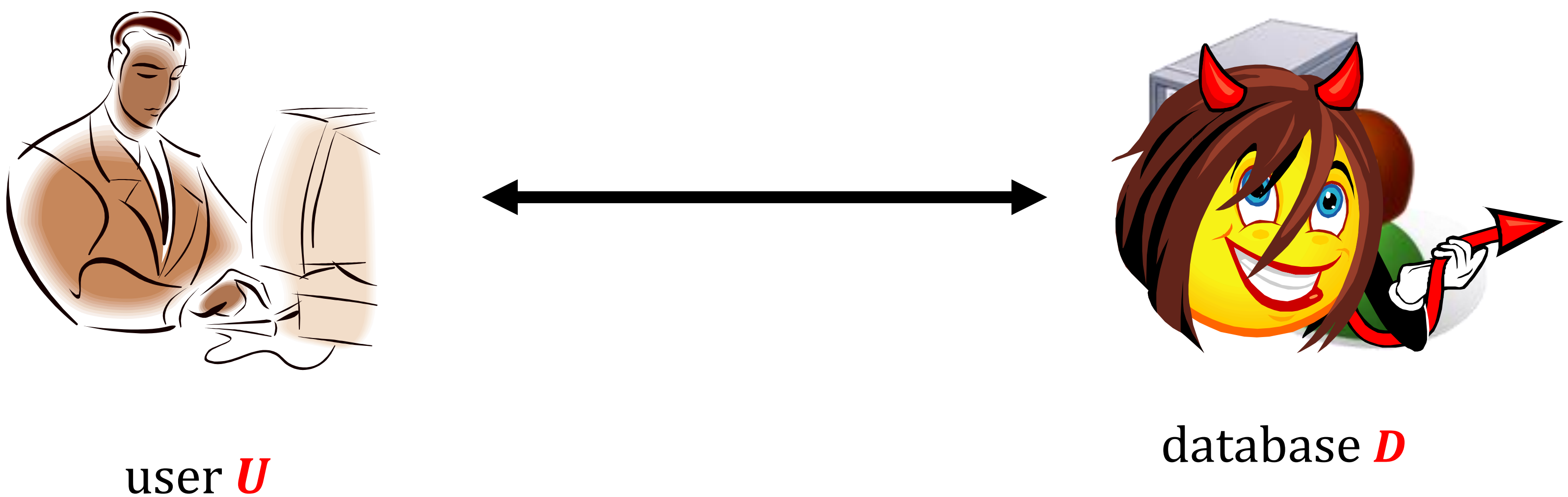
We can:

- **trust** them that they will protect our secrecy,

or

- use **cryptography**!

Our settings



Question

How to protect privacy of queries?

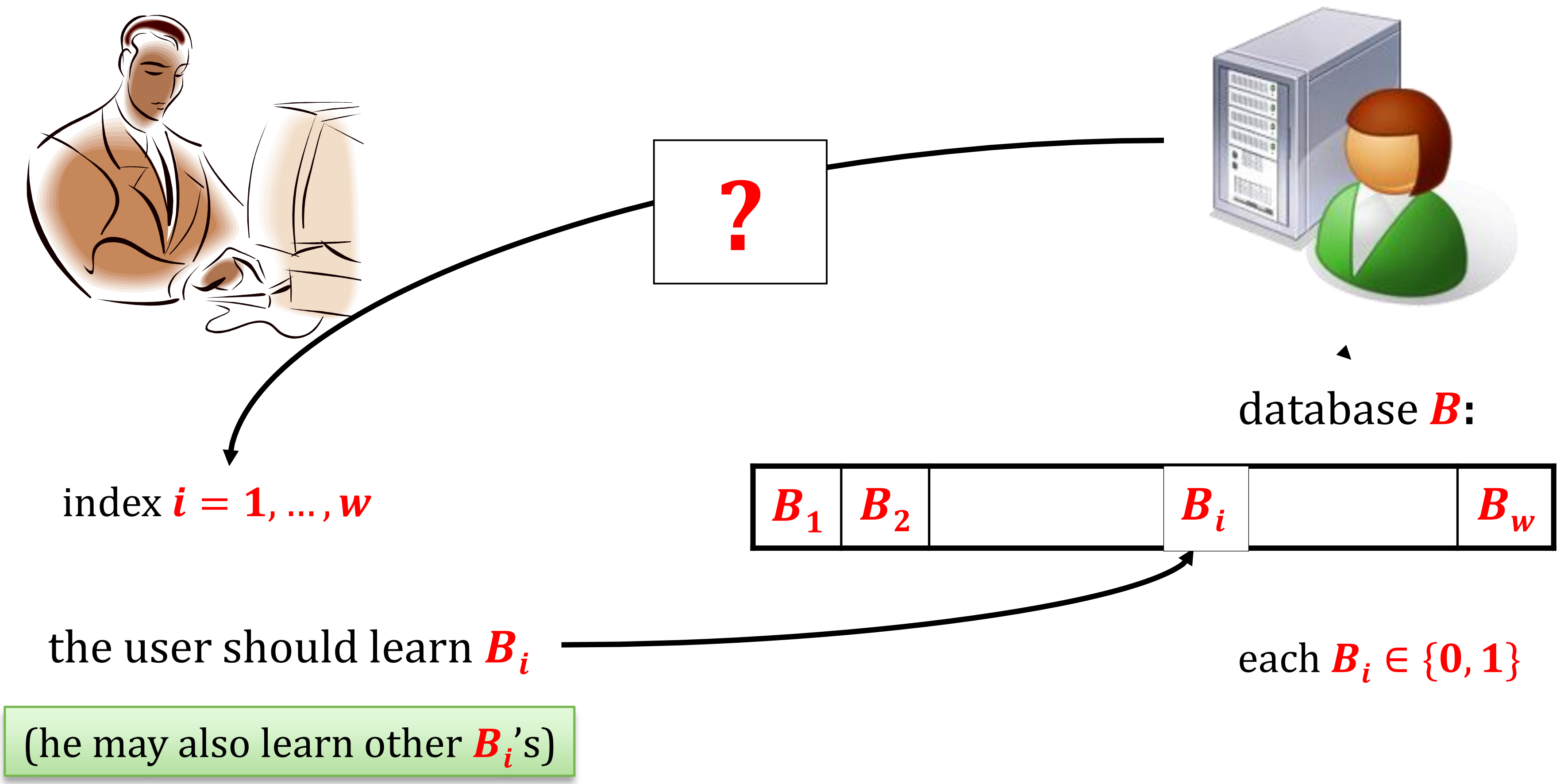
user ***U***

database ***D***

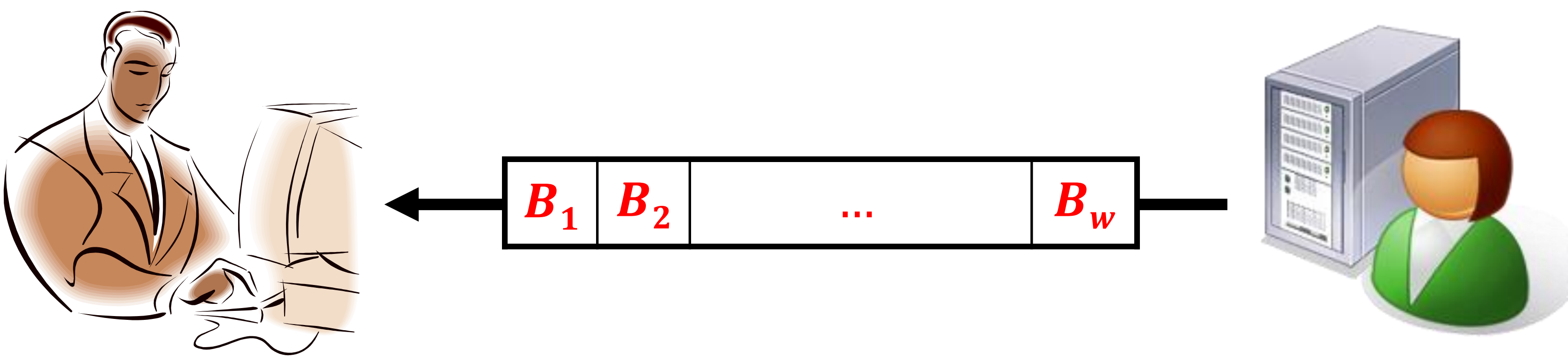
wants to retrieve some data from ***D***

shouldn't learn what ***U*** retrieved

Let's make things simple!



Trivial solution



The database simply sends everything to the user!

Non-triviality

The previous solution has a drawback:

the communication complexity is huge!

Therefore we introduce the following requirement:

"**Non-triviality**":

the number of bits communicated between *U* and *D* has to be smaller than *w*.

Private Information Retrieval

polynomial time randomized interactive algorithms

input:
index $i = 1, \ldots, w$

input: | B_1 | B_2 | … | B_w |

This property needs to be defined more formally

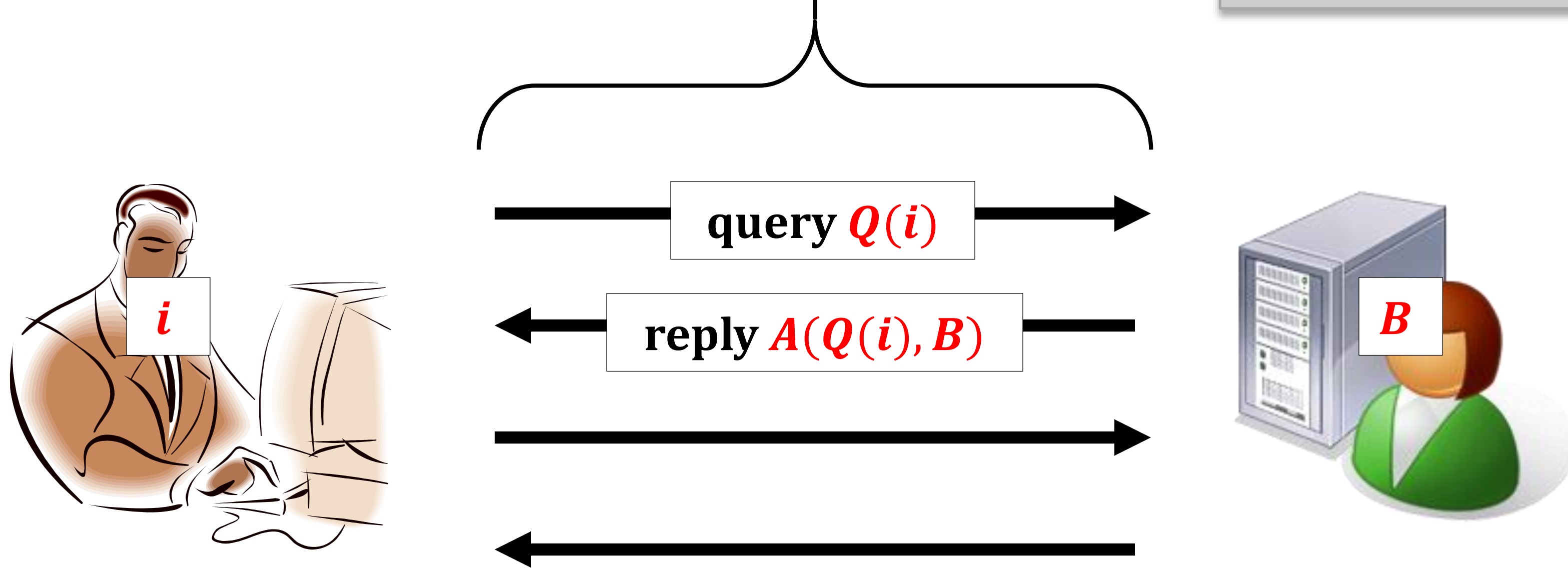
- at the end the user learns B_i ← **correctness**
- the database does not learn i ← **secrecy (of the user)**
- the total communication is $< w$ ← **non-triviality**

Note: secrecy of the database is not required

How to define secrecy of the user [1/2]?

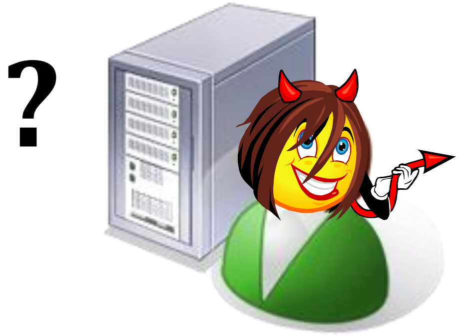
For fixed $\boldsymbol{i}$ and $\boldsymbol{B}$
$\boldsymbol{T(i,B)}$
is a **random variable**
(since the parties are randomized)

<u>Def</u>. $\boldsymbol{T(i,B)}$ – **transcript** of the conversation.



How to define secrecy of the user [2/2]?

Secrecy of the user: for every $\boldsymbol{i}, \boldsymbol{j} \in \{0, 1\}$



single-round case:

it is impossible to distinguish between $\boldsymbol{Q}(\boldsymbol{i})$ and $\boldsymbol{Q}(\boldsymbol{j})$

multi-round case:

it is impossible to distinguish between $\boldsymbol{T}(\boldsymbol{i}, \boldsymbol{B})$ and $\boldsymbol{T}(\boldsymbol{j}, \boldsymbol{B})$

even if the adversary is malicious

depending on the settings: **computational** or **unconditional indistinguishability**

Computationally-secure PIR – formally

computational-secrecy:

For every $i, j \in \{0, 1\}$

it is impossible to distinguish
efficiently
between
$T(i, B)$ and $T(j, B)$

Formally: for every **polynomial-time** probabilistic algorithm A the value:

$$\left| P(A(T(i,B)) = 0) - P(A(T(j,B)) = 0) \right|$$

should be **negligible.**

What it possible?

Fact

Information-theoretically secure single-server **PIR** does not exist **[exercise]**.

What can be constructed is the following:

- **computationally-secure PIR** (we show it now)
- **information-theoretically secure multi-server PIR [exercise]**

PIR vs OT

PIR looks similar to the 1**-out-of-*w* OT**

Differences:

- **advantage of PIR**: **low communication complexity**
- **advantage of OT**: **privacy of the database is protected**

Can we combine both?

Yes! It's called "**symmetric PIR**".

Plan

1. Introduction
2. Construction

The construction

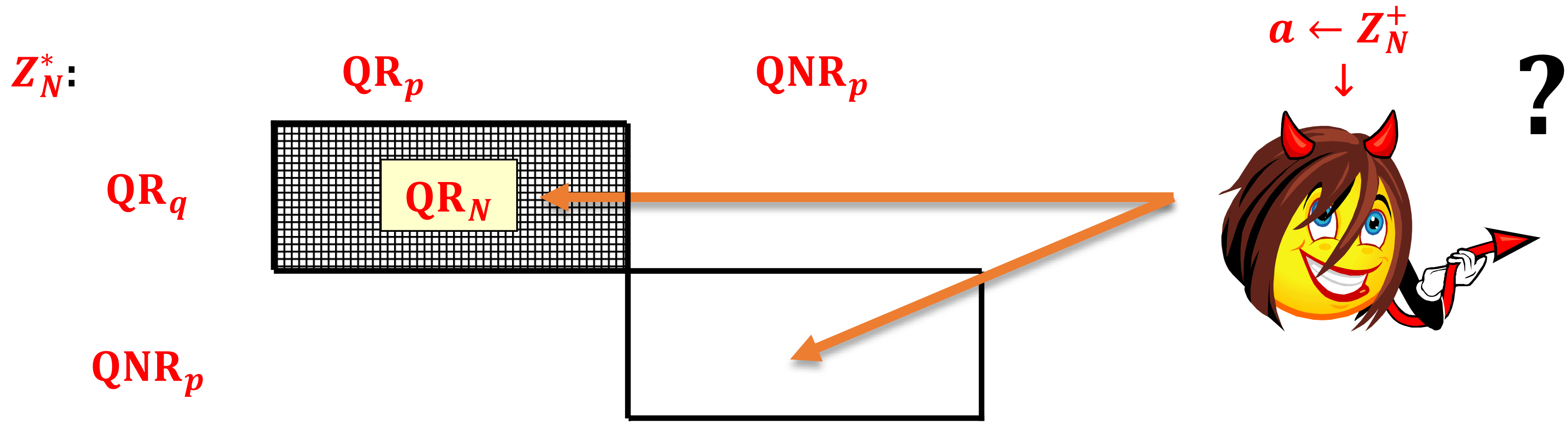
Kushilevitz and R. Ostrovsky **Replication Is NOT Needed: SINGLE Database, Computationally-Private Information Retrieval**, FOCS 1997

based on the **Quadratic Residuosity Assumption**.

Our presentation strategy:

1. we first present a **wrong** solution
2. then we **fix it**.

Quadratic Residuosity Assumption



Quadratic Residuosity Assumption (QRA):

For a random $a \leftarrow Z_N^+$ it is computationally hard to determine if $a \in QR_N$.

Formally: for every **polynomial-time** probabilistic algorithm D the value:

$$\left| P\left(D(N,a) = Q_N(a)\right) - \frac{1}{2} \right|$$

(where $a \leftarrow Z_N^+$) is **negligible**.

Where a predicate $Q_N : Z_N^+ \rightarrow \{0,1\}$ is defined as follows:

$Q_N(a) = 0$ if $a \in QR_N$

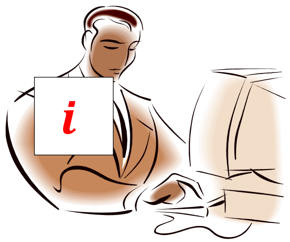
$Q_N(a) = 1$ otherwise

Homomorphism of $\boldsymbol{Q_N}$

For all $\boldsymbol{a, b \in Z_N^+}$

$$\boldsymbol{Q_N(ab) = Q_N(a) \oplus Q_N(b)}$$

First (wrong) idea



i ↓

B_1	B_2	…	B_{i-1}	B_i	B_{i+1}	…	B_{w-1}	B_w

QR X_1	QR X_2	…	QR X_{i-1}	NQR X_i	QR X_{i+1}	…	QR X_{w-1}	QR X_w

for every $j = 1, \dots, w$ the database sets

$$Y_j = \begin{cases} X_j^2 & \text{if } B_j = 0 \\ X_j & \text{otherwise} \end{cases}$$

QR Y_1	QR Y_2	…	QR Y_{i-1}	Y_i	QR Y_{i+1}	…	QR Y_{w-1}	QR Y_w

Y_i is a **QR** iff $B_i = 0$

M is a **QR** iff $B_i = 0$

Set $M = Y_1 \cdot Y_2 \cdot \dots \cdot Y_w$

M ←

the user checks if M is a **QR**

Problems!

PIR from the previous slide:

- **correctness** √
- **security**?

To learn i the database would need to distinguish **NQR** from **QR**. √

QR	QR	...	QR	NQR	QR	...	QR	QR
X_1	X_2		X_{i-1}	X_i	X_{i+1}		X_{w-1}	X_w

- **non-triviality**? doesn't hold!

communication:
user → database: $|B| \cdot |N|$
database → user: $|N|$

Call it:
$(|B|, 1)$–**PIR**

How to fix it?

Idea

Given:

$$(|\boldsymbol{B}|, \mathbf{1})\text{–PIR}$$

construct

$$\left(\sqrt{|\boldsymbol{B}|}, \sqrt{|\boldsymbol{B}|}\right)\text{–PIR}$$

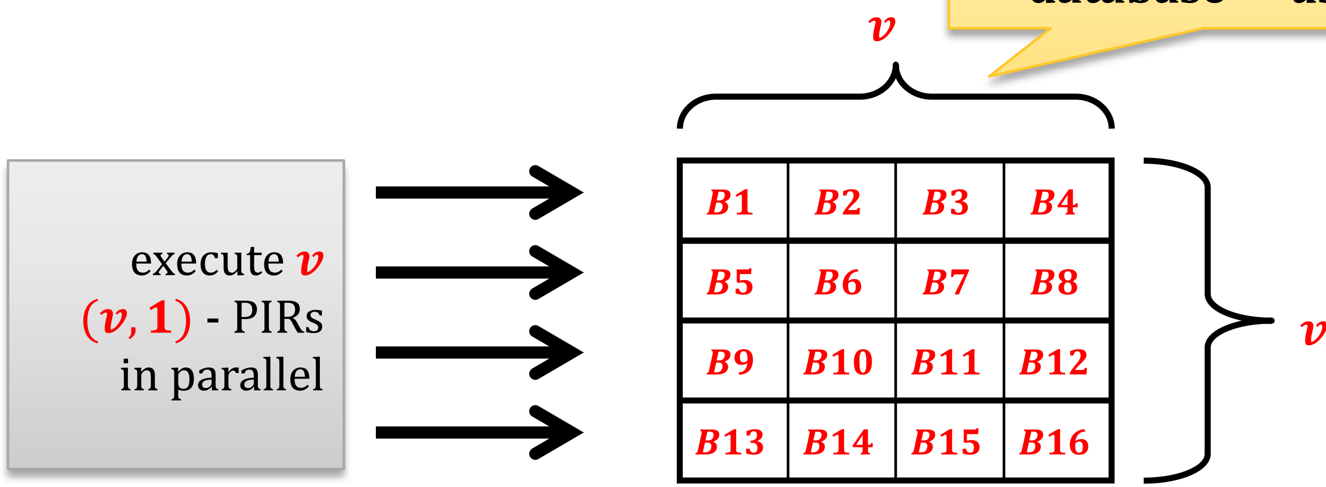
Suppose that $|\boldsymbol{B}| = \boldsymbol{v}^2$ and present $\boldsymbol{B}$ as a $\boldsymbol{v} \times \boldsymbol{v}$–matrix:

$B1$	$B2$	$B3$	$B4$	$B5$	$B6$	$B7$	$B8$	$B9$	$B10$	$B11$	$B12$	$B13$	$B14$	$B15$	$B16$

consider each row as a separate database

An improved idea

Looks even worse:
<u>communication</u>:
user → database: $v^2 \cdot |N|$
database → user: $v \cdot |N|$



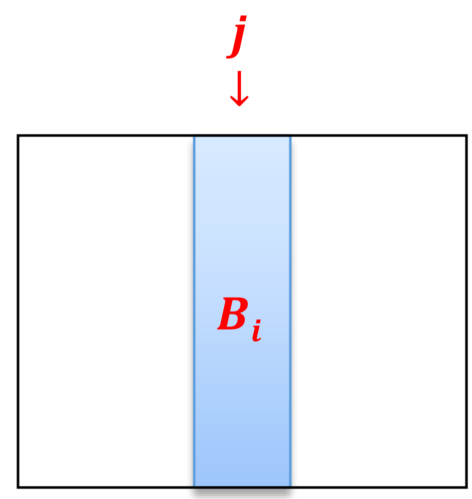
The method

Let j be the column where B_i is.

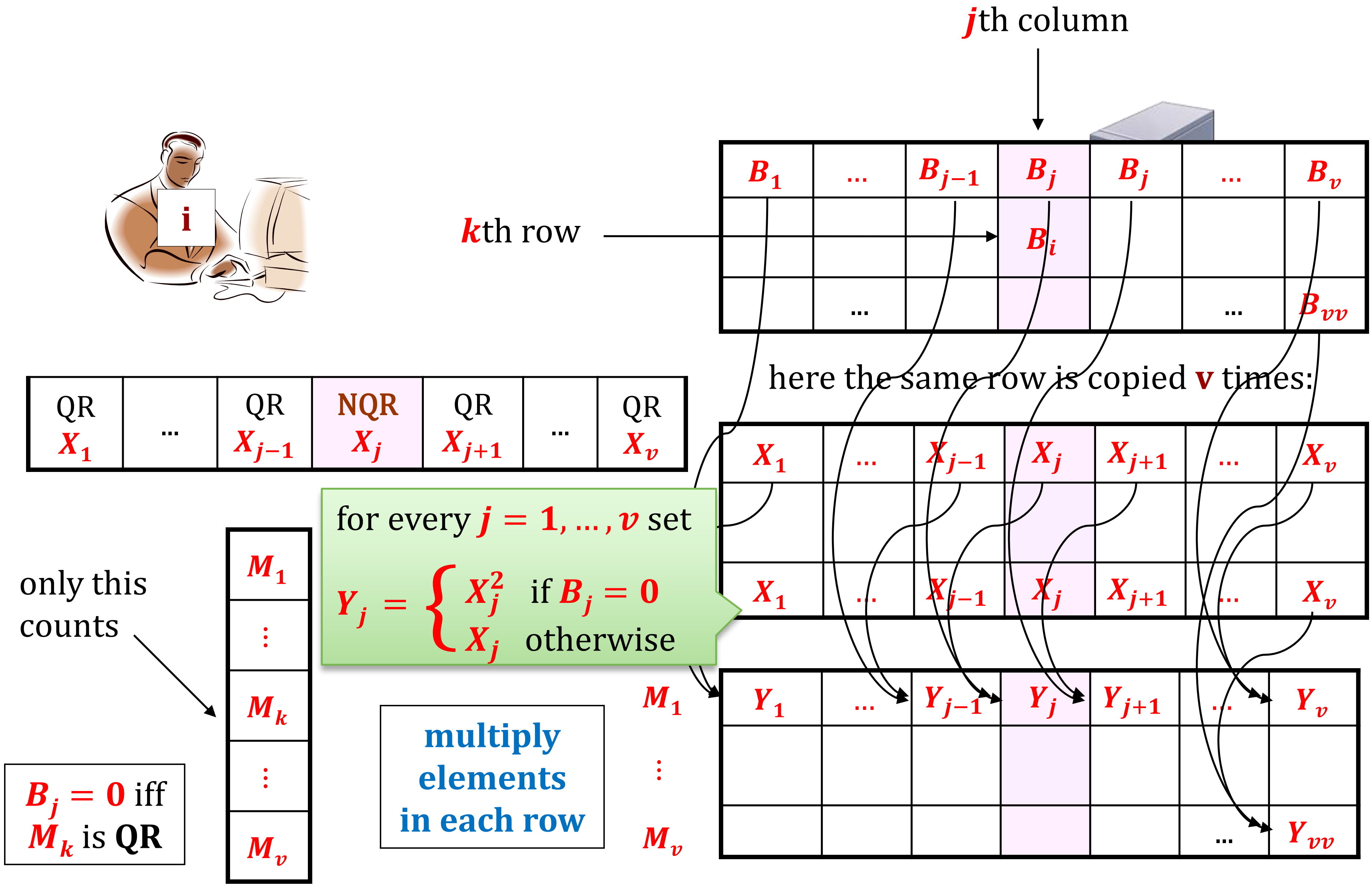
In every "row" the user asks for the jth element

So, instead of sending v queries the user can send one!

Observe: in this way the user learns
all the elements in the jth column!



Putting things together



So we are done!

PIR from the previous slide:

- **correctness** √
- **non-triviality:**
  communication complexity = $2\sqrt{|B| \cdot |N|}$ √
- **security**?
  To learn i the database would need to distinguish **NQR** from **QR**.

Formally:

from
any adversary that **breaks our scheme**
we can construct
an algorithm that **breaks QRA**

Improvements

$(X_1, \dots, Xv)$

$(M_1, \dots, M_v)$

user U

the user is interested just in one M_i.

database D

Idea: apply **PIR** recursively!

Extensions

- Symmetric PIR (also protect privacy of the database).

  [**Gertner, Ishai, Kushilevitz, Malkin**. 1998]
- Searching by key-words

  [**Chor, Gilboa, Naor**, 1997]
- Public-key encryption with key-word search

  [**Boneh, Di Crescenzo, Ostrovsky, Persiano**]