Lecture 15
Private Information Retrieval

Stefan Dziembowski

www.dziembowski.net
MIM UW

25.01.13

ver 1.0

Plan

1. Motivation and definition
2. Information-theoretic impossibility
3. A construction of Kushilevitz and Ostrovsky
4. Overview of some other related results

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**!

How can crypto help?

Note: this problem has nothing to do with secure communication!

Our settings

user **U**

database **D**

A new primitive:

Private Information Retrieval (PIR)

Plan

1. Definition of PIR
2. An ideal PIR doesn't exist
3. Construction of a computational PIR
4. Open problems

<u>Literature</u>:

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

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

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!

?

index $i = 1,\ldots,w$

database **B**:

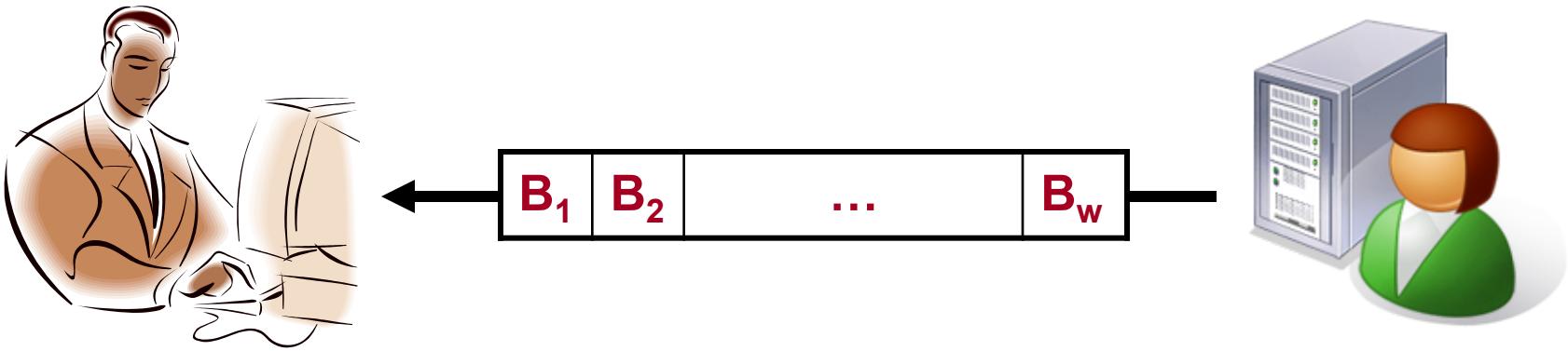
B_1	B_2		B_i		B_w

the user should learn B_i

each $B_i \in \{0,1\}$

(he may also learn other B_i's)

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 (PIR)

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

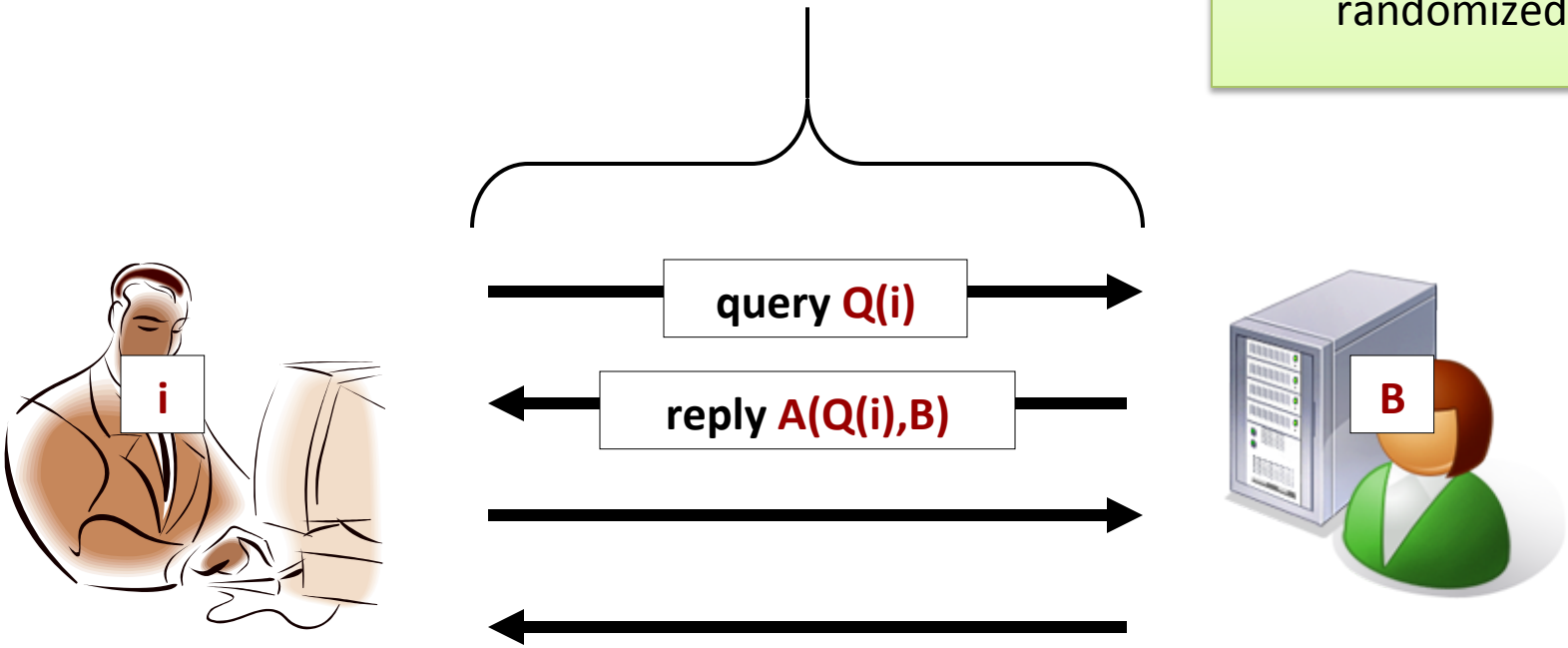
- at the end the user learns B_i ← **correctness**
- the database does not learn i ← **secrecy (of the user)**
- the total communication is $< w$ ← **non-triviality**

Note: secrecy of the database is not required

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

For fixed **i** and **B**
T(i,B)
is a **random variable**
(since the parties are randomized)

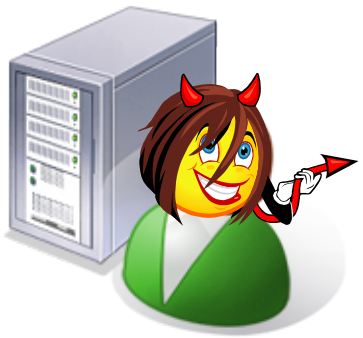
<u>Def</u>. **T(i,B)** – **transcript** of the conversation.



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

<u>Secrecy of the user</u>: for every **i,j ϵ {0,1}**

?



<u>single-round case</u>:

it is impossible to distinguish between **Q(i)** and **Q(j)**

<u>multi-round case</u>:

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

even if the adversary is malicious

For simplicity say that for any **i** and **j** the distributions of **T(i,B)** and **T(j,B)** have to be identical

Plan

1. Motivation and definition
2. Information-theoretic impossibility
3. A construction of Kushilevitz and Ostrovsky
4. Overview of some other related results

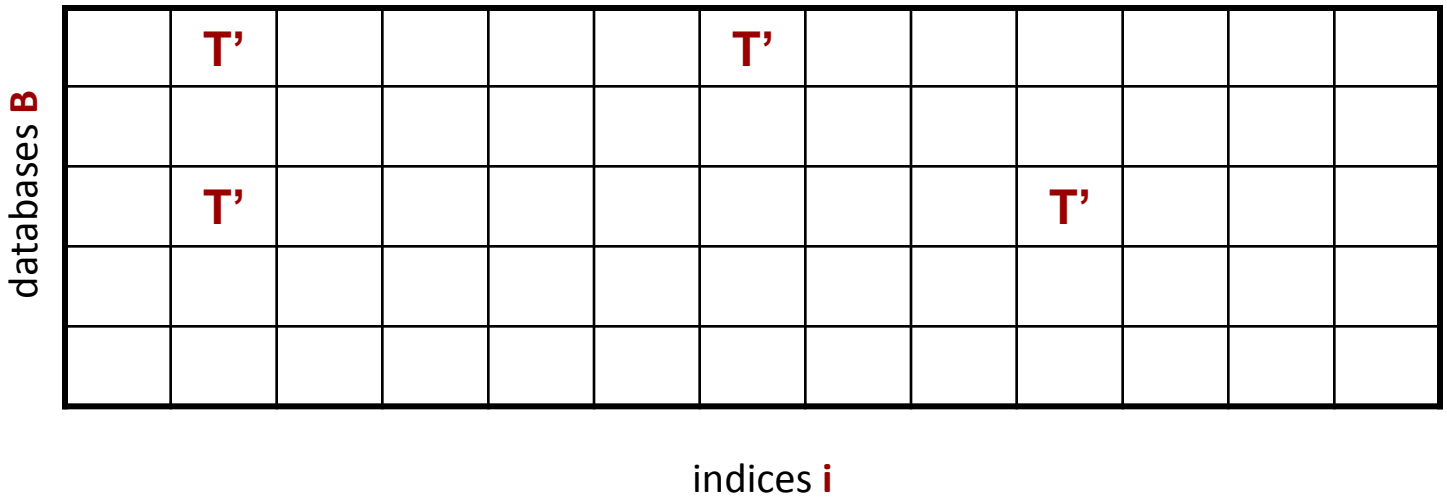
PIR doesn't exists [1/4]

We now show that **correctness**, **non-triviality** and **secrecy** **cannot** be satisfied simultaneously.

Def: A transcript **T** is **possible** for **(i,B)** **if**

$$P(T(i,B) = T) > 0$$

Take some **T'**, and look where it is **possible**:



PIR doesn't exists [2/4]

Observation:

secrecy → if

T' is possible for some **B** and **i**

then

it is possible for **B** and all the other **i**'s

databases **B**

T'	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**
T'	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**

indices **i**

PIR doesn't exists [3/4]

non-triviality → length(transcript) < length(database)

↓

 # transcripts < #databases

↓

there has to exist T' that is possible for two databases B_0 and B_1

T'	T'	T'	T'	T'	T'	T'	T'	T'	T'	T'	T'	T'	← B_0
T'	T'	T'	T'	T'	T'	T'	T'	T'	T'	T'	T'	T'	← B_1

databases **B**

indices **i**

PIR doesn't exists [4/4]

B_0 and **B_1** differ on at least one index **i'**
So, if **i'** is the input of the user then

correctness → contradiction

i'
↓

T'	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	← **B_0**
T'	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	**T'**	← **B_1**

databases **B**

indices **i**

So PIR doesn't exist!

- How to bypass the impossibility result?
- **Two ideas:**
  - limit the computing power of a cheating database

  - use a larger number of "independent" databases

Plan

1. Motivation and definition
2. Information-theoretic impossibility
3. A construction of Kushilevitz and Ostrovsky
4. Overview of some other related results

Computationally-secure PIR

computational-secrecy:

For every **i,j ϵ {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:

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

should be **negligible.**

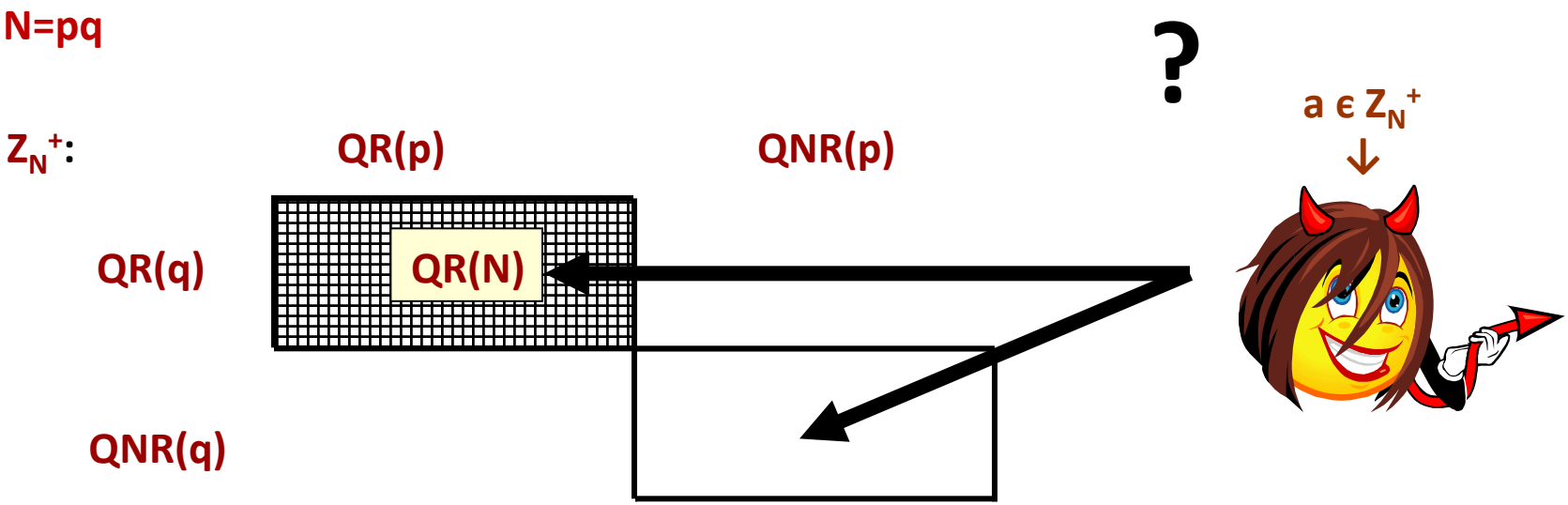
Hardness assumptions?

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

construct PIR based on the

Quadratic Residuosity Assumption

Quadratic Residuosity Assumption (QRA)



<u>Quadratic Residuosity Assumption (QRA)</u>:

For a random **a ϵ Z_N^+** it is computationally hard to determine if a **ϵ QR(N).**

<u>Formally</u>: for every **polynomial-time** probabilistic algorithm **G** the value:

$$|P(G(a) = Q(a)) - 0.5|$$

(where **a** is random) is **negligible.**

Homomorphism of **QR(pq)**

$$Q(N,a) := \begin{cases} 1 & \text{if } a \in QR(N) \\ 0 & \text{otherwise} \end{cases}$$

Homomorphism: for all $a,b \in Z_N^+$

$$Q(N,ab) = Q(N,a) \text{ xor } Q(N,b)$$

We are ready to construct PIR!

Our PIR will work in the group $\mathbf{Z_N^+}$, where **N=pq**.

What's so good about this group?:

- testing membership in $\mathbf{Z_N^+}$ is easy,
- testing membership in **QR(N)** is hard for random elements on $\mathbf{Z_N^+}$,
  unless one knows **p** and **q**.
- homomorphism of **Q**!

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

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

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

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

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

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!

<u>communication</u>:
user → database: $|B| \cdot |Z^*_n|$
database → user: $|Z^*_n|$

Call it:
$(|B|, 1)$ - PIR

How to fix it?

<u>Idea</u>:

Given: $(|\mathrm{B}|, 1)$-**PIR**.

construct $\left(\sqrt{|\mathrm{B}|}, \sqrt{|\mathrm{B}|}\right)$-**PIR**.

<u>Suppose</u> that **|B| = v²** and present **B** as a **v×v**-matrix:

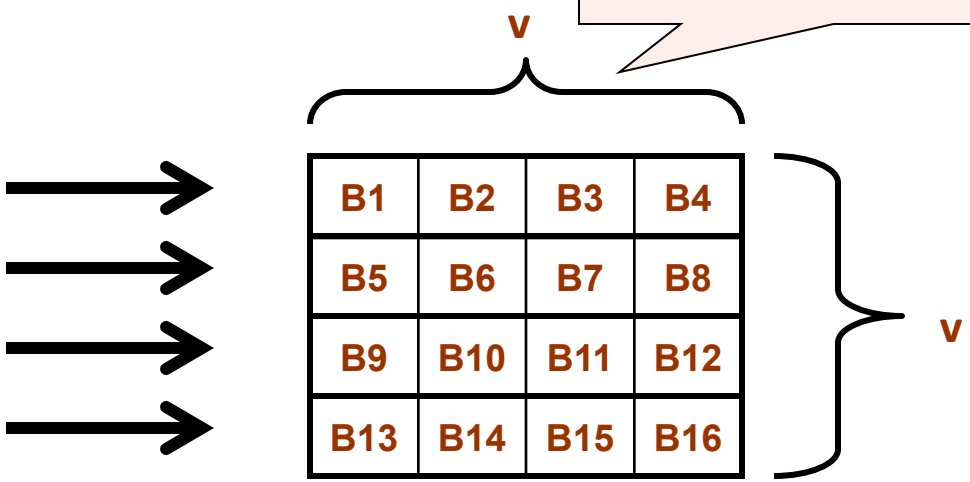
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16

consider each row as a separate database

An improved idea

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

execute **v**
(v,1) - PIRs
in parallel



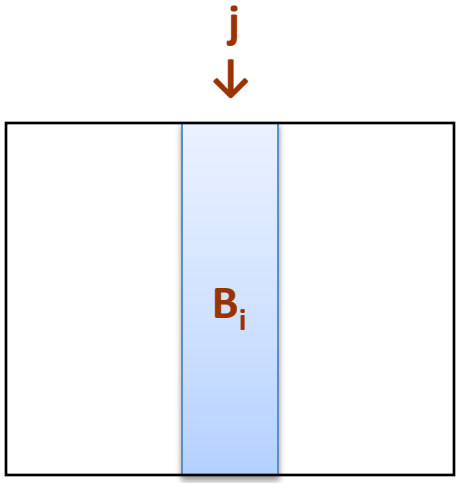
The method

Let **j** be the column where B_i is.

In every "row" the user asks for the **j**th element

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

Observe: in this way the user learns all the elements in the **j**th column!



Putting things together

jth column

kth row

B_1	…	B_{j-1}	B_j	B_{j+1}	…	B_v
			B_i			
	…				…	B_{vv}

here the same row is copied **v** times:

X_1	…	X_{j-1}	X_j	X_{j+1}	…	X_v
X_1	…	X_{j-1}	X_j	X_{j+1}	…	X_v

QR X_1	…	QR X_{j-1}	NQR X_j	QR X_{j+1}	…	QR X_v

for every $j = 1,\ldots,v$ set

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

only this counts

M_1
M_k
M_v

$B_j=0$ iff M_k is **QR**

multiply elements in each row

M_1

…

M_v

Y_1	…	Y_{j-1}	Y_j	Y_{j+1}	…	Y_v
					…	Y_{vv}

So we are done!

PIR from the previous slide:

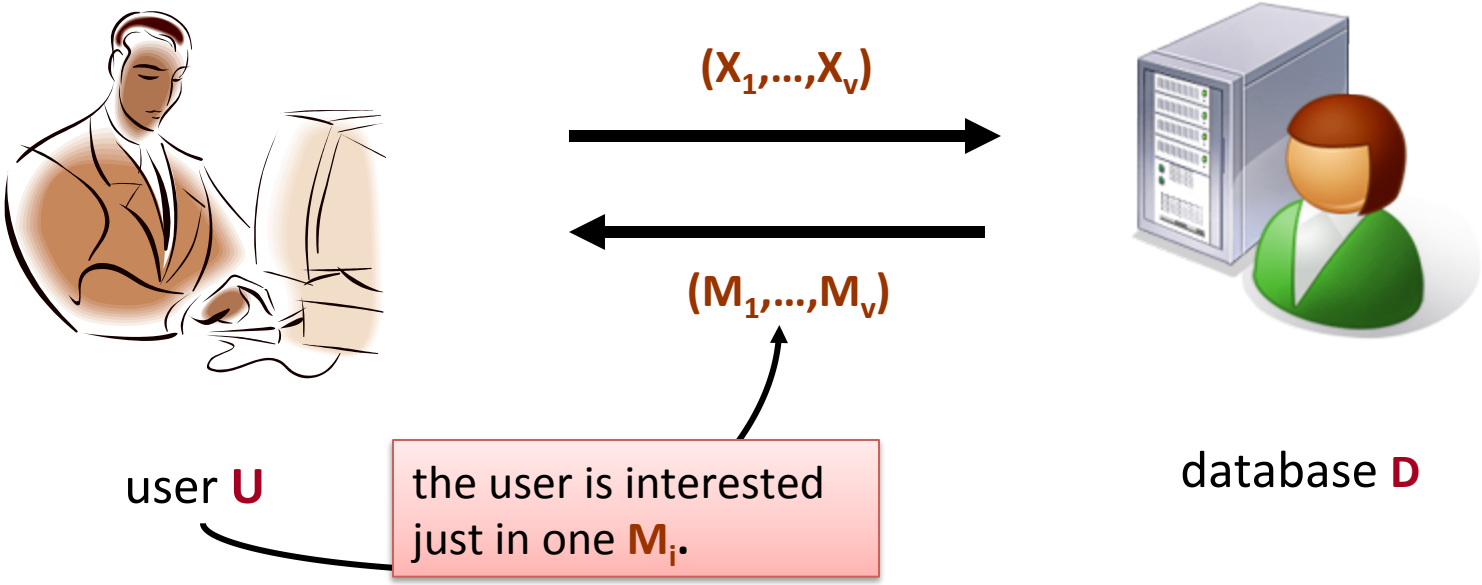
- **correctness** √
- **non-triviality:**
  communication complexity = **$2\sqrt{|B|} \cdot |Z_n|$** √
- **security**?

  The to learn i the database would need to distinguish NQR from QR.

  <u>Formally</u>:

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

Improvements



<u>Idea</u>: apply PIR recursively!

Plan

1. Motivation and definition
2. Information-theoretic impossibility
3. A construction of Kushilevitz and Ostrovsky
4. Overview of some other related results

Complexity of PIRs – overview of the results

Communication:

- "recursive" PIR of [KO97]:
  for every **c**: **$O(|B|^c)$**
- [**Cachin, Micali, Stadler**, 1999]:
  poly-logarithmic in **|B|**
- [**Lipmaa**, 2005]:
  $O(\log^2|B|)$

For practical analysis see:

- [**Sion, Carbunar**]
  On the Computational Practicality of Private Information Retrieval.

their conclusion:

It is the time-complexity that matters

In **real-life**:

it is still more practical
to **transmit the entire database.**

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**]

©2013 by Stefan Dziembowski. Permission to make digital or hard copies of part or all of this material is currently granted without fee *provided that copies are made only for personal or classroom use, are not distributed for profit or commercial advantage, and that new copies bear this notice and the full citation*.