Lecture 13
Private Information Retrieval

Stefan Dziembowski
University of Rome
La Sapienza

BiSS 2009
Bertinoro International
Spring School
2-6 March 2009
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?

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

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

Literature:


Question

How to protect privacy of queries?

user $U$

wants to retrieve some data from $D$

database $D$

shouldn’t learn what $U$ retrieved
Let’s make things simple!

index $i = 1,...,w$

the user should learn $B_i$

(he may also learn other $B_i$’s)

database $B$: $B_1 B_2 B_i B_w$

each $B_i \in \{0,1\}$
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: \( B_1 \quad B_2 \quad \ldots \quad B_w \)

- at the end the user learns \( B_i \)
- the database does not learn \( i \)
- the total communication is \(< w\)

**Note:** secrecy of the database is not required
How to define secrecy of the user [1/2]?

Def. $T(i, B)$ – transcript of the conversation.

For fixed $i$ and $B$ 
$T(i, B)$ is a random variable (since the parties are randomized)
How to define secrecy of the user [2/2]?

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

**single-round case:**

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

**multi-round case:**

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 exist [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 exist [2/4]

**Observation:**

secrecy $\rightarrow$

if $T'$ is possible for some $B$ and $i$

then it is possible for $B$ and all the other $i$’s
PIR doesn’t exist \([3/4]\)

**non-triviality** → \(\text{length(transcript)} < \text{length(database)}\)

\[
\downarrow
\]


# transcripts < #databases

\[
\downarrow
\]

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

\[
\begin{array}{ccccccccccccc}
T' & T' & T' & T' & T' & T' & T' & T' & T' & T' & T' & T' & T' \\
\hline
T' & T' & T' & T' & T' & T' & T' & T' & T' & T' & T' & T' & T' \\
\hline
T' & T' & T' & T' & T' & T' & T' & T' & T' & T' & T' & T' & T' \\
\hline
\end{array}
\]

\(\leftarrow B_0\)

\(\leftarrow B_1\)

indices \(i\)
PIR doesn’t exist [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

\[
\begin{array}{cccccccccccc}
T' & T' & T' & T' & T' & T' & T' & T' & T' & T' & T' & T' \\
\end{array}
\]

\[
\begin{array}{cccccccccccc}
T' & T' & T' & T' & T' & T' & T' & T' & T' & T' & T' & T' \\
\end{array}
\]

\[
\begin{array}{cccccccccccc}
\end{array}
\]

\[
\begin{array}{cccccccccccc}
\end{array}
\]

\[
\begin{array}{cccccccccccc}
\end{array}
\]

\[
\begin{array}{cccccccccccc}
\end{array}
\]

\[
\begin{array}{cccccccccccc}
\end{array}
\]

\[
\begin{array}{cccccccccccc}
\end{array}
\]

\[
\begin{array}{cccccccccccc}
\end{array}
\]
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

we show this
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 \( i,j \in \{0,1\} \)

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

Formally: for every polynomial-time probabilistic algorithm \( A \) the value:

\[
|P(A(T(i,B)) = 0) - P(A(T(j,B))=0)|
\]

should be negligible.
Hardness assumptions?


construct PIR based on the **Quadratic Residuosity Assumption**
Quadratic Residuosity Assumption (QRA):

For a random \( a \in \mathbb{Z}_N^+ \) it is computationally hard to determine if \( a \in \mathbb{QR}(N) \).

Formally: for every polynomial-time probabilistic algorithm \( G \) the value:
\[
|P(G(a) = Q(a)) - 0.5|
\]
(where \( a \) is random) is negligible.
Homomorphism of \( \text{QR}(pq) \)

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

Homomorphism: for all \( a,b \in \mathbb{Z}_N^+ \)

\[
Q(N,ab) = Q(N,a) \oplus Q(N,b)
\]
We are ready to construct PIR!

Our PIR will work in the group $\mathbb{Z}_N^+$, where $N=pq$.

What’s so good about this group?:

- testing membership in $\mathbb{Z}_N^+$ is easy,
- testing membership in $\text{QR}(N)$ is hard for random elements on $\mathbb{Z}_N^+$, unless one knows $p$ and $q$.
- homomorphism of $\mathbb{Q}$!
First (wrong) idea

<table>
<thead>
<tr>
<th>QR</th>
<th>QR</th>
<th>...</th>
<th>QR</th>
<th>NQR</th>
<th>QR</th>
<th>QR</th>
<th>QR</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X_1)</td>
<td>(X_2)</td>
<td>...</td>
<td>(X_{i-1})</td>
<td>(X_i)</td>
<td>(X_{i+1})</td>
<td>...</td>
<td>(X_{w-1})</td>
</tr>
</tbody>
</table>

for every \(j = 1, \ldots, w\) 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 \(B_i = 0\)

the user checks if \(M\) is a QR

Set \(M = Y_1 \cdot Y_2 \cdot ... \cdot Y_w\)
Problems!

PIR from the previous slide:

- **correctness** √
- **security**?

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

<table>
<thead>
<tr>
<th>QR</th>
<th>QR</th>
<th>...</th>
<th>QR</th>
<th>NQR</th>
<th>QR</th>
<th>...</th>
<th>QR</th>
<th>QR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$x_2$</td>
<td>...</td>
<td>$x_{i-1}$</td>
<td>$x_i$</td>
<td>$x_{i+1}$</td>
<td>...</td>
<td>$x_{w-1}$</td>
<td>$x_w$</td>
</tr>
</tbody>
</table>

- **non-triviality**? doesn’t hold!

**communication:**

- **user** → **database**: $|B| \cdot |Z^*_n|$
- **database** → **user**: $|Z^*_n|$

Call it: $(|B|, 1) - PIR$
How to fix it?

Idea:
Given: $(|B|, 1)$-PIR.

construct $(\sqrt{|B|}, \sqrt{|B|})$-PIR.

Suppose that $|B| = v^2$ and present $B$ as a $v \times v$-matrix:

\[
\begin{bmatrix}
B_1 & B_2 & B_3 & B_4 & B_5 & B_6 & B_7 & B_8 & B_9 & B_{10} & B_{11} & B_{12} & B_{13} & B_{14} & B_{15} & B_{16}
\end{bmatrix}
\]

consider each row as a separate database
An improved idea

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

Looks even worse:
communication:
user \( \rightarrow \) database: \( v^2 \cdot |Z_n^*| \)
database \( \rightarrow \) user: \( v \cdot |Z_n^*| \)

<table>
<thead>
<tr>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>B4</th>
</tr>
</thead>
<tbody>
<tr>
<td>B5</td>
<td>B6</td>
<td>B7</td>
<td>B8</td>
</tr>
<tr>
<td>B9</td>
<td>B10</td>
<td>B11</td>
<td>B12</td>
</tr>
<tr>
<td>B13</td>
<td>B14</td>
<td>B15</td>
<td>B16</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>QR</th>
<th>...</th>
<th>QR</th>
<th>NQR</th>
<th>QR</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>...</td>
<td>$X_{j-1}$</td>
<td>$X_j$</td>
<td>$X_{j+1}$</td>
<td>...</td>
</tr>
</tbody>
</table>

**kth row**

Here the same row is copied $v$ times:

| $X_1$ | ... | $X_{j-1}$ | $X_j$ | $X_{j+1}$ | ... | $X_v$ |

| $Y_1$ | ... | $Y_{j-1}$ | $Y_j$ | $Y_{j+1}$ | ... | $Y_v$ |

**jth column**

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

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

$B_j = 0$ iff $M_k$ is QR

Multiply elements in each row

$B_1, \ldots, B_{j-1}, B_j, B_{j+1}, \ldots, B_v$
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.

**Formally:**

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

user $U$

the user is interested just in one $M_i$. 

database $D$

Idea: 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]