Multi-User Encrypted Data Processing

Amani ABOU RIDA\textsuperscript{1}, Maxime BROS\textsuperscript{2}, Olivier GIMENEZ\textsuperscript{3}, Héber HWANG ARCOLEZI\textsuperscript{4}, Quentin YANG\textsuperscript{5}

CIRM, Marseille, France

\textbf{25-29 October 2021}

Sébastien Canard \hspace{4cm} Jérémy Chotard

\textsuperscript{1}University of Strasbourg, CNRS, ICube, \textsuperscript{2}University of Limoges, XLIM, \textsuperscript{3}IRISA Laboratory, Rennes and Orange Labs, \textsuperscript{4}University of Bourgogne-Franche-Comté, CNRS, \textsuperscript{5}University of Lorraine, Inria Nancy-Grand Est, LORIA.
Our Team

Amani **ABOU RIDA**

Maxime **BROS**

Olivier **GIMENEZ**

Héber **HWANG ARCOLEZI**

Quentin **YANG**
Multi-User Encrypted Data

Users: \( \{ x_i \} \)

Orange: Heavy computations

Pollster: Get statistics

Goals:
- Efficiency
- Privacy
- User-friendly
- Few actions from pollster
Data can be sensitive...

**WARNING: This website contains explicit adult material.**

You may only enter this Website if you are at least 18 years of age, or at least the age of majority in the jurisdiction where you reside or from which you access this Website. If you do not meet these requirements, then you do not have permission to use the Website.

[Enter] [Leave]

Warning message of [www.youporn.com](http://www.youporn.com)
Attacker Models

• Full Trust
• Semi-Honest
• Malicious with corruption of a reasonable amount of users
• A Coalition of malicious with corruption
Considered Privacy

- **Partial Privacy**: Orange can compute $f$ over a subset of users
- **Ideal Privacy**: Orange can only compute $f$ over all users
Functions

- Compute the Mean
- Compute the Weighted Mean
- Compute the Linear Regression
- Compute Any Given Function
Imprecision of Partial Privacy

Let $f$ be the mean function:

$$f(X) = \frac{1}{|X|} \sum_{x_i \in X} x_i$$

Evaluate $f$ over any subsets $X$ of $\{x_i\}_i$.

Partial Privacy $\Rightarrow$ Ideal Privacy
Imprecision of Partial Privacy

Let $f$ be the mean function:

$$f(X)$$

Evaluate $f$ over any subsets $X$ of $\{x_i\}_i$

$$\text{Enc}_{pk}(x_i) \rightarrow \text{Evalute } f \text{ over any subsets } X \text{ of } \{x_i\}_i \rightarrow \text{Enc}_{pk}(f(X))$$

Problem

$$\forall i, \ X = \{x_i\}, \ f(X) = x_i.$$ 

Partial Privacy $\Rightarrow$ Ideal Privacy
Outline

1. Introduction

2. Our Schemes
   - Building Blocks
     - Dummy Scheme
     - No Trust Scheme
     - No Drop Scheme
     - No Copy Scheme

3. Linear Regression

4. Coalition between Orange and the Pollster

5. Statistics Verifying Differential Privacy
Homomorphic Encryption

**Definition (Homomorphic property)**

\[
\text{Dec}_{sk} (\text{Enc}_{pk}(x) \cdot \text{Enc}_{pk}(y)) = x \cdot y.
\]

**Example: Exponential ElGamal (additive)**

\[
\begin{align*}
pk &= (g, h = g^s) \\
\text{Enc}_{pk}(x) &= (g^r, g^x h^r) \\
\text{Dec}_{sk}((u, v)) &= vu^{-s} \\
\text{Dec}_{sk}(\text{Enc}_{pk}(x)\text{Enc}_{pk}(y)) &= g^{x+y}
\end{align*}
\]
Zero Knowledge Proof (ZKP)

Used to prove correct computation.

- Proof of knowledge
- Zero Knowledge
- Non-Interactive

**Generalization:** a NP language \( L \) with relation \( R \subseteq \mathcal{W} \times \mathcal{X} \).
Outline

1. Introduction

2. Our Schemes
   - Building Blocks
   - Dummy Scheme
   - No Trust Scheme
   - No Drop Scheme
   - No Copy Scheme

3. Linear Regression

4. Coalition between Orange and the Pollster

5. Statistics Verifying Differential Privacy
$c_i = \text{Enc}_{pk}(x_i)$

$M = \prod_i c_i^{w_i}$

$\mu = \text{Dec}_{sk}(M)$

result = $\mu$
Dummy Scheme

```
Ip

M = \prod_i c_{i}^{w_{i}} \rightarrow \mu = \text{Dec}_{sk}(M)
result = \mu
```

Full Trust, Ideal Privacy, No Attribute, (Weighted) Mean
Outline

1. Introduction

2. Our Schemes
   - Building Blocks
   - Dummy Scheme
   - No Trust Scheme
   - No Drop Scheme
   - No Copy Scheme

3. Linear Regression

4. Coalition between Orange and the Pollster

5. Statistics Verifying Differential Privacy
Secret Sharing (SS)

Orange

\[ s_1 \leftarrow \mathbb{Z}_p \]
\[ S_1 = g^{s_1} \]

\[ pk = S_1 S_2 \]

Pollster

\[ s_2 \leftarrow \mathbb{Z}_p \]
\[ S_2 = g^{s_2} \]

\[ pk = g^{s_1 + s_2} \]

\[ H(S_1) \]
\[ H(S_2) \]

\[ S_1 \]
\[ S_2 \]

Shared Public Key

\[ pk = S_1 S_2 \]

Shared Secret: \( s_1 + s_2 \)

Decrypting \( \text{Enc}_{pk}(m) \) requires 2 Partial Decryptions (PDec):

\[ \text{PDec}_{s_2}(\text{PDec}_{s_1}(\text{Enc}_{pk}(m))) = \text{PDec}_{s_1}(\text{PDec}_{s_2}(\text{Enc}_{pk}(m))) = m \]
Key Distribution

- PKI
  - Users
    - (sk_i, pk_i)
  - Orange
    - pk
  - Pollster
    - SS
    - s_1, pk
    - s_2, pk
No Trust Scheme (Step 1)

\[
\sigma_i = \operatorname{Sign}_{sk_i}(c_i) \quad (c_i, \sigma_i)
\]

\[
M = \prod c_i^{w_i} \quad (c_i, \sigma_i)_i, M, \tilde{m}
\]

\[
\tilde{m} = \operatorname{PDec}_{s_1}(M)
\]

\[
M \overset{?}{=} \prod c_i^{w_i} \quad \text{Verify}_{pk_i}(\sigma_i, c_i)
\]

\[
\mu = \operatorname{PDec}_{s_2}(\tilde{m})
\]
Attack

\[\begin{align*}
\text{Users} & : pk_i, sk_i \\
\text{Orange} & : pk, s_1 \\
\text{Pollster} & : pk, s_2
\end{align*}\]

\[\begin{align*}
(c_i, \sigma_i) & \rightarrow M = \prod c_i^{w_i} \\
\tilde{m} & = \text{PDec}_{s_1}(c_1) \\
(c_i, \sigma_i)_i, M, \tilde{m} & \rightarrow M = \prod c_i^{w_i} \\
\text{Verify}_{pk_i}(\sigma_i, c_i) & \\
x_1 & = \text{PDec}_{s_2}(\tilde{m})
\end{align*}\]
No Trust Scheme (Step 2)

Users
\( pk_i, sk_i \)

Orange
\( pk, s_1 \)

Pollster
\( pk, s_2 \)

\[
M = \prod c_i^{w_i} \\
\tilde{m} = \text{PDec}_{s_1}(M) \\
\pi = \text{ZKP}(\tilde{m} = \text{PDec}_{s_1}(M))
\]

\[
(c_i, \sigma_i)_i, M, \tilde{m}, \pi \xrightarrow{\text{Verify}_{pk_i}(\sigma_i, c_i)} M = \prod c_i^{w_i} \\
\tilde{m} = \text{PDec}_{s_1}(M) \text{ with } \pi \\
\mu = \text{PDec}_{s_2}(\tilde{m})
\]
No Trust Scheme (Step 2)

<table>
<thead>
<tr>
<th>Users</th>
<th>Orange</th>
<th>Pollster</th>
</tr>
</thead>
<tbody>
<tr>
<td>$pk_i, sk_i$</td>
<td>$pk, s_1$</td>
<td>$pk, s_2$</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
M &= \prod c_i^{w_i} \\
\tilde{m} &= \text{PDec}_{s_1}(M) \\
\pi &= \text{ZKP}(\tilde{m} = \text{PDec}_{s_1}(M)) \\
\end{align*}
\]

\[
\begin{align*}
\text{Verify}_{pk_i}(\sigma_i, c_i) \\
\tilde{m} &= \text{PDec}_{s_1}(M) \text{ with } \pi \\
\mu &= \text{PDec}_{s_2}(\tilde{m}) \\
\end{align*}
\]

Semi-Honest, Ideal Privacy, No Attribute, (Weighted) Mean
Outline

1 Introduction

2 Our Schemes
   - Building Blocks
   - Dummy Scheme
   - No Trust Scheme
   - No Drop Scheme
   - No Copy Scheme

3 Linear Regression

4 Coalition between Orange and the Pollster

5 Statistics Verifying Differential Privacy
No Drop Scheme

\[ M = \prod c_i^{w_i} \]
\[ \tilde{m} = \text{PDec}_{s_1}(M) \]
\[ \pi = \text{ZKP}(\tilde{m} = \text{PDec}_{s_1}(M)) \]

\[ M \overset{?}{=} \prod c_i^{w_i} \]
\[ \text{Verify}_{p_{k_i}}(\sigma_i, c_i) \]
\[ \tilde{m} \overset{?}{=} \text{PDec}_{s_1}(M) \text{ with } \pi \]
\[ \mu = \text{PDec}_{s_2}(\tilde{m}) \]
## Attack

### Public Board

### Users

<table>
<thead>
<tr>
<th>User</th>
<th>Certificate and Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>$c_1, \text{Sign}_{sk_1}(c_1)$</td>
</tr>
<tr>
<td>$u_2$</td>
<td>$c_2, \text{Sign}_{sk_2}(c_2)$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$u_{n-k}$</td>
<td>...</td>
</tr>
<tr>
<td>$u_n$</td>
<td>...</td>
</tr>
</tbody>
</table>
### Attack

#### Users

<table>
<thead>
<tr>
<th>$u_1$</th>
<th>$c_1, \text{Sign}_{sk_1}(c_1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_2$</td>
<td>$c_2, \text{Sign}_{sk_2}(c_2)$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$u_{n-k}$</td>
<td>$c_1, \text{Sign}<em>{sk</em>{n-k}}(c_1)$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$u_n$</td>
<td>$c_1, \text{Sign}_{sk_n}(c_1)$</td>
</tr>
</tbody>
</table>

#### Diagram

- **Public Board**
  - $c_1, \text{Sign}_{sk_1}(c_1)$
  - $c_1, \text{Sign}_{sk_{n-k}}(c_1)$
  - $c_1$

- **Orange**
  - $c_1$
  - $c_1$
  - $c_1$
  - $c_1, \text{Sign}_{sk_i}(c_1)$
### Attack

<table>
<thead>
<tr>
<th>Users</th>
<th>Public Board</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>$c_1, \text{Sign}_{sk_1}(c_1)$</td>
</tr>
<tr>
<td>$u_2$</td>
<td>$c_2, \text{Sign}_{sk_2}(c_2)$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$u_{n-k}$</td>
<td>$c_1, \text{Sign}<em>{sk</em>{n-k}}(c_1)$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$u_n$</td>
<td>$c_1, \text{Sign}_{sk_n}(c_1)$</td>
</tr>
</tbody>
</table>
No Copy Scheme

The public board waits for a **deadline** before publishing the results.

Malicious and Corruption, Ideal Privacy, No Attribute, (Weighted) Mean

A. Rida, Bros, Gimenez, H. Arcolezi, Yang

REDOCS 2021: Team Orange

24 / 46
1 Introduction

2 Our Schemes

3 Linear Regression
   • Building blocks for Linear Regression
   • Several leads for Linear Regression

4 Coalition between Orange and the Pollster

5 Statistics Verifying Differential Privacy

6 Conclusion
Functional Encryption

Definition (Multi-Input Functional Encryption)

\[ \text{Dec}_{sk}(\text{Enc}_{pk}(x_1), \ldots, \text{Enc}_{pk}(x_n)) = f(x_1, \ldots, x_n). \]

Example: AGT’s scheme

Agrawal, Goyal, Tomida, Multi-input Quadratic Functional Encryption from Pairings. CRYPTO 2021.
Fully Homomorphic Encryption

Definition (Fully Homomorphic property)

\[ \text{Dec}_{sk}(\text{Enc}_{pk}(x) + \text{Enc}_{pk}(y)) = x + y, \]
\[ \text{Dec}_{sk}(\text{Enc}_{pk}(x)\text{Enc}_{pk}(y)) = xy. \]

Example:

Boneh, Gennaro, Goldfeder, Jain, Kim, Rasmussen, Sahai, Threshold Cryptosystems from Threshold Fully Homomorphic Encryption, CRYPTO 2018.
Outline

1 Introduction

2 Our Schemes

3 Linear Regression
   • Building blocks for Linear Regression
   • Several leads for Linear Regression

4 Coalition between Orange and the Pollster

5 Statistics Verifying Differential Privacy

6 Conclusion
What about Linear Regression?

Data \((x_i, y_i)_i\) from users, \(\bar{x} = \text{mean}(\{x_i\}_i), \bar{y} = \text{mean}(\{y_i\}_i)\).

Find coefficients \(a, b\) s.t. \(y \approx ax + b\).

\[
a = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}, \quad b = \bar{y} - a\bar{x}.
\]
What about Linear Regression?

Idea 1: Use Functional Encryption

Wish List for Christmas: An Encryption scheme s.t.

\[ \text{Dec}_{sk}(\text{Enc}_{pk}(x_1), \text{Enc}_{pk}(y_1), \cdots, \text{Enc}_{pk}(x_n), \text{Enc}_{pk}(y_n)) = \left( \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}, \bar{y} - ax \right) \]

Problem:
We could not find Santa.

A. Rida, Bros, Gimenez, H. Arcolezi, Yang
REDOCS 2021: Team Orange
What about Linear Regression?

Idea 1: Use Functional Encryption

Wish List for Christmas: An Encryption scheme s.t.

\[
\text{Dec}_{sk}(\text{Enc}_{pk}(x_1), \text{Enc}_{pk}(y_1), \ldots, \text{Enc}_{pk}(x_n), \text{Enc}_{pk}(y_n)) = \left(\frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}, \bar{y} - ax \right)
\]

Problem:

We could not find Santa.
What about Linear Regression?

Recall: Slope of the Linear Regression \( = \frac{\sum_{i}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i}(x_i - \bar{x})^2} \)

Idea 2: Use Homomorphic Functional Encryption for inner product

Wish List for Christmas: A **homomorphic** Encryption scheme such that

\[
\text{Dec}_{sk}(\text{Enc}_{pk}(x_1), \text{Enc}_{pk}(y_1), \ldots, \text{Enc}_{pk}(x_n), \text{Enc}_{pk}(y_n)) = \sum_{i} x_i y_i
\]
What about Linear Regression?

Recall: Slope of the Linear Regression

\[
\frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}
\]

Idea 2: Use Homomorphic Functional Encryption for inner product

Wish List for Christmas: A homomorphic Encryption scheme such that

\[
\text{Dec}_{sk}(\text{Enc}_{pk}(x_1), \text{Enc}_{pk}(y_1), \ldots, \text{Enc}_{pk}(x_n), \text{Enc}_{pk}(y_n))
= \sum_i x_i y_i
\]

- \( \forall \ i, \:
  \text{Enc}_{pk}(x_i)\text{Enc}_{pk}(\bar{x})^{-1} = \text{Enc}_{pk}(x_i - \bar{x}), \)
  \( \text{Enc}_{pk}(y_i)\text{Enc}_{pk}(\bar{y})^{-1} = \text{Enc}_{pk}(y_i - \bar{y}), \)
- Compute \( N = \sum_i (x_i - \bar{x})(y_i - \bar{y}) \) using \( \text{Dec}_{sk} \),
- Compute \( D = \sum_i (x_i - \bar{x})^2 \) using \( \text{Dec}_{sk} \).
What about Linear Regression?

\[ \forall \ i, \]
\[ \text{Enc}_{pk}(x_i)\text{Enc}_{pk}(\bar{x})^{-1} = \text{Enc}_{pk}(x_i - \bar{x}), \]
\[ \text{Enc}_{pk}(y_i)\text{Enc}_{pk}(\bar{y})^{-1} = \text{Enc}_{pk}(y_i - \bar{y}), \]

- Compute \( N = \sum_i (x_i - \bar{x})(y_i - \bar{y}) \) using \( \text{Dec}_{sk} \),
- Compute \( D = \sum_i (x_i - \bar{x})^2 \) using \( \text{Dec}_{sk} \).

Problem:
It leaks extra data.
What about Linear Regression?

\[ \forall \ i, \]
\[ \text{Enc}_{pk}(x_i)\text{Enc}_{pk}(x) \neq \text{Enc}_{pk}(x_i - x), \]
\[ \text{Enc}_{pk}(y_i)\text{Enc}_{pk}(y) \neq \text{Enc}_{pk}(y_i - y), \]

Compute \( N = \sum_i (x_i - \bar{x})(y_i - \bar{y}) \) using \( \text{Dec}_{sk} \),

Compute \( D = \sum_i (x_i - \bar{x})^2 \) using \( \text{Dec}_{sk} \).

Problem:

It leaks extra data.

A solution? Use random masking.

Reveal \( \alpha N \) and \( \alpha D \) for some random \( \alpha \).
What about Linear Regression?

Recall: \[ \frac{N}{D} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2} \]

from \[ X_i = \text{Enc}_{pk}(x_i - \bar{x}), \]
\[ Y_i = \text{Enc}_{pk}(y_i - \bar{y}). \]

- Choose random \( \alpha \).
- Compute \( X'_i = X_i^\alpha = \text{Enc}_{pk}(\alpha(x_i - \bar{x})) \),
- Evaluate \( \text{Dec}_{sk}((X'_i, Y_i)_{1 \leq i \leq n}) = \alpha N \).
- Evaluate \( \text{Dec}_{sk}((X'_i, X_i)_{1 \leq i \leq n}) = \alpha D \).
What about Linear Regression?

Summary of Idea 2
(Homomorphic Functional Encryption for inner product)

- $\text{Enc}_{pk}(x) \text{Enc}_{pk}(y) = \text{Enc}_{pk}(x + y)$,
- $\text{Dec}_{sk}((\text{Enc}_{pk}(x_i), \text{Enc}_{pk}(y_i))) = \sum_i x_i y_i$,
- Zero Knowledge Range Proof for $\alpha$,
- Zero Knowledge Proof for the $X_i'$.

Problems

- $\alpha$ cannot be too large.
- Verifying zero knowledge proofs (for $\alpha$ and $X_i'$) requires heavy computations.
- So far, no scheme matches our specific needs.
Let’s do some basic maths!

The sums are done over $n$ terms:

$$\frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2} = \frac{n^2 \sum_i (x_i - \bar{x})(y_i - \bar{y})}{n^2 \sum_i (x_i - \bar{x})^2} = \frac{\sum_i (nx_i - n\bar{x})(ny_i - n\bar{y})}{\sum_i (nx_i - n\bar{x})^2} = \frac{\sum_i (nx_i - \sum_i x_i)(ny_i - \sum_i y_i)}{\sum_i (nx_i - \sum_i x_i)^2}$$

**Our trick**

We got rid of the integer divisions in the numerator and in the denominator!
What about Linear Regression?

Idea 3: Use Fully Homomorphic Encryption

\[ \text{Dec}_{sk} \left( \text{Enc}_{pk}(x) \text{Enc}_{pk}(y) \right) = xy, \]
\[ \text{Dec}_{sk} \left( \text{Enc}_{pk}(x) + \text{Enc}_{pk}(y) \right) = x + y, \]
\[ \text{Zero Knowledge Range Proof for } \alpha, \]
\[ \text{Zero Knowledge Proof for the } X_i' \]

Problem

- Usually FHE not efficient at all... **but not with our trick!**
- The proofs are hard to verify.
Outline

1. Introduction
2. Our Schemes
3. Linear Regression
4. Coalition between Orange and the Pollster
5. Statistics Verifying Differential Privacy
6. Conclusion
Impact of a coalition

Users $\xleftarrow{\text{pk}}$ Orange $\overset{\text{Enc}_{pk}(x_i)}{\xrightarrow{\text{pk}, s_1}}$ SS $\xrightarrow{\text{pk}, s_2}$ Pollster

Can decrypt all users’ data!
Protecting users against a coalition

Any idea?

1. Encrypt with own secret key.
2. Users participate in the Distributed Key Generation.
Protecting users against a coalition

Any idea?

1. Encrypt with own secret key. Cannot aggregate data.
2. Users participate in the Distributed Key Generation. Not realistic.
Any idea?

1. Encrypt with own secret key. **Cannot aggregate data.**
2. Users participate in the Distributed Key Generation. **Not realistic.**
3. Trust an Auditor.

### Diagram

```
  Users                      Orange                          SS                          Pollster
       |                      s1, pk                  s2, pk                s3, pk
  Enc_{pk}(x_i)   ->  SS  ->  Auditor  ->  Pollster
```

**Cannot decrypt without auditor**
## Protecting users against a coalition

### Any idea?

1. Encrypt with own secret key. **Cannot aggregate data.**
2. Users participate in the Distributed Key Generation. **Not realistic.**
3. Trust an Auditor. **Proof are hard to verify.**

![Diagram](image)

- Users
- Orange
- SS
- Pollster
- Auditor
- $\text{Enc}_{pk}(x_i)$
- Cannot decrypt without auditor
Outline

1. Introduction
2. Our Schemes
3. Linear Regression
4. Coalition between Orange and the Pollster
5. Statistics Verifying Differential Privacy
6. Conclusion
Some Flaws of Privacy-Preserving Publication of Statistics

"Anonymity Threshold" \( k = 10 \)

<table>
<thead>
<tr>
<th>Zip</th>
<th>Age</th>
<th>Gender</th>
<th>Nationality</th>
<th>Fav. App</th>
<th>Hours in Social Media</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>[21; 31]</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>17</td>
</tr>
<tr>
<td>*</td>
<td>[21; 31]</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>≥15</td>
</tr>
<tr>
<td>*</td>
<td>[21; 31]</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>18</td>
</tr>
<tr>
<td>*</td>
<td>[31; 41]</td>
<td>*</td>
<td>*</td>
<td>* (Porn)</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>[31; 41]</td>
<td>*</td>
<td>*</td>
<td>* (Porn)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>[31; 41]</td>
<td>*</td>
<td>*</td>
<td>* (Porn)</td>
<td></td>
</tr>
</tbody>
</table>

10 individuals Mean: \( ≥ 15 \)

10 individuals Mode: Porn

Obvious (among many) attack by "Homogeneity":
- Known 29-year-old app-friend \( \leadsto \) "high" spending time in social media.
- Known 35-year-old app-friend \( \leadsto \) porn (published frequency and/or mode).

Remark

- Attacker has background knowledge \( \leadsto \) privacy breaks down.
- Find a criterion for a database to be anonymized \( \leadsto \) always flawed.
- Privacy should apply to the anonymization mechanism, not to its output.
Anonymization mechanism $\mathcal{M}$ → not leak private information.

- Dalenius (1977): For all $1 \ X_1, X_2 \in \mathcal{X}$, $\mathcal{M}(X_1) = \mathcal{M}(X_2)$.
- Dwork (2006): Dalenius’ goal is fundamentally **impossible** to achieve.

Differential Privacy (DP) → $\mathcal{M}(X_1) \approx \mathcal{M}(X_2)$.

- You can quantify the privacy loss (parameter $\varepsilon$);
- You no longer need attack modeling.

---

$1$Neighboring databases (differ in only one record): $||X_1 - X_2||_1 = 1$
First DP Scheme: “Centralized”

Assuming no coalition of Orange and Pollster

- Assumption: Users trust Orange only.
- Orange computes $f$ over ciphertexts of users $C$;
- Orange computes the ciphered DP noise $C_N = \text{Enc}_{pk}(N)$;
- Orange reports $M(C, f(.), C_N) = f(C) + C_N$;

Remark

- Centralized DP is still subject to security breaches.
- Does not solve the coalition between Orange and Pollster.
Second DP scheme: “Decentralized”

Mitigating the coalition of Orange and Pollster

- Assumption: Users do not trust anyone else in the system;
- E.g., DP mean estimation.
- Each user \( i \in \mathcal{I} \) with input value \( x_i \) do:
  - Compute \( \hat{x}_i = x_i + N \)
  - Report: \( \hat{c}_i = \text{Enc}_{pk}(\hat{x}_i) \)
- Orange computes \( f \) over all \( \hat{c}_i \).

Remark

- Local DP algorithms are (usually) designed for a specific task!
- DP noise in each \( x_i \) may compromise data utility.
Conclusion

Goal: privacy-preserving scheme for a pollster service by Orange.

What we did:
- Several schemes for different levels of attacks
- Solutions with FE and FHE

Problem encountered:
- Currently no FE compatible with our needs
- Only leads for coalition attack

Further Work:
- Security proofs of our schemes
- Further check over Differential Privacy solutions
- Suitable FE scheme
Thanks!