Privacy Preserving Analytics

PSI + Analytics using Homomorphic Encryption

R. Schreijen – IT Solution Architect
Nov. 17th, 2021
Topics

• Purpose
• Homomorphic Encryption (short introduction)
• PSI using HE
• Analytics using HE
• Analytics using HE + helper
• Q&A
Purpose
High level purpose

**Private Set Intersection:**
- Determine **set intersection** of datasets from **multiple owners** while preserving input privacy
  
  *e.g. ‘How many people are customers of both company A and B without revealing specific customers to each other?’*

**Privacy Preserving Analytics:**
- Perform **statistical analysis** on datasets from **multiple owners** while preserving input privacy

  *e.g. ‘What’s the average spending of customers of company A who are also customer of company B?’*
Preserving input privacy

Several privacy preserving technologies:

• Trusted Execution Environment
• Garbled circuits
• Secret Sharing
• Homomorphic Encryption
• ...
• Combination(s) of above technologies
Homomorphic Encryption
Homomorphic Encryption

• Computations on encrypted data possible without decrypting first

• Result after decrypting equals equivalent computation on unencrypted cleartext:

\[ \text{Decrypt}(\text{Function}(\text{Encrypt}(x))) = \text{Function}(x) \]

(actual function in the encrypted domain is not identical to function in unencrypted domain)

• Asymmetric: different keys for encrypting and decrypting

→ Enables ‘outsourcing’ of computations on your sensitive data to others
2 Types of HE

- **Partial**, only **single type of operation** possible, e.g.:
  - Multiplicative \((\text{ciphertext} \cdot \text{ciphertext})\)
  - Additive \((\text{ciphertext} + \text{ciphertext})\)

- **Fully**, both **additive and multiplicative**
  - Severe performance drop
  - Very large ciphertexts and keys
  - Limited arithmetic circuit depth
  - Added complexity
Homomorphic Encryption: Important aspects

- Ciphertexts are:
  - large, random-looking numbers
  - *re-randomizable* (multiply by encrypted 1 or add encrypted 0...)
  - indistinguishable!

Plaintext: 3
Ciphertext: 1736734601920938409279237659872346123871002093878777742341

Plaintext: 3
Ciphertext: 9928374645102937462812384760092374987623466277478488222164
PSI using HE
Concept – key aspects

- **Set membership** of a private set can be expressed **numerically** ($1 = \text{in my set}$, $0 = \text{not in my set}$)
- HE encrypted 1’s and 0’s are **indistinguishable**
- HE encrypted **set membership** can be **added numerically** (counting, using ‘simple’ additive HE Scheme)
- Each party **replaces** set entries for entities **not in their set** by encrypted 0’s
- **Summing** encrypted 1’s and 0’s **creates intersection count**
Example

1. Create keypair

Public Key

Private Key

2. Create encrypted initial table

<table>
<thead>
<tr>
<th>ID</th>
<th>P1</th>
<th>P1P2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Plaintext: 1 or 0

Ciphertext: ‘Random’ 136253748903876725241038746...

3. Send table + public key

4. Link records: matching ID’s

5. Create intersection: zeroing cells for non-matching ID’s

<table>
<thead>
<tr>
<th>ID</th>
<th>P1</th>
<th>P1P2</th>
<th>P2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Σ 5 2 3

6. Sum columns

7. Return encrypted counts (bottom row)

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P1P2</th>
<th>P2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Σ</td>
<td>5</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

6. Decrypt and reveal counts

Encrypted columns!
Extending concept: more parties

P1 encrypt → P2 replace → P3 replace and sum → P1 decrypt aggregates + broadcast
PPA using HE
Concept – key aspects

• Builds on PSI example
• All numbers are indistinguishable (not only 0 and 1)
• Enables passing encrypted fact data to other parties
• Parties filter / select rows conditionally based on own facts / data
• Other parties can manipulate facts ‘blinded’ under HE (e.g. replacing by a specific number or adding/multiplying etc.)
• Last party aggregates under HE
Example

Calculate **average income** for people with mobile roaming costs > 200

<table>
<thead>
<tr>
<th>ID</th>
<th>P1: Income</th>
<th>P2: Mobile Roaming Costs</th>
<th>P2: Filter (count)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1700</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2300</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>160</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1500</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>5200</td>
<td>300</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>6100</td>
<td>250</td>
<td>1</td>
</tr>
<tr>
<td>Σ</td>
<td>11300</td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

**Income sum**

<table>
<thead>
<tr>
<th>ID</th>
<th>Income sum</th>
<th>People count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Σ</td>
<td>11300</td>
<td>2</td>
</tr>
</tbody>
</table>

Avg = Decrypt(11300) / 2

*Can also be extended to parties > 2*

*Any party can act as filter or aggregatable party*

*Complex analytics require FHE scheme and/or multiple communication rounds*
PPA using HE + helper party
Limitations

- Some population disclosure inevitable...
- Initial population should not be sensitive
  - Union of P1 and P2 (if both not sensitive)
  - P1 or P2 (if only P2 or P1 is sensitive)
  - Superset of P1 and P2 (if P1 and P2 sensitive: e.g. ‘all people in country’)
- But: larger population $\rightarrow$ lower performance (ciphertext expansion & data exchange, more computations etc.)
- What if P1 and P2 sensitive and superset not viable?? $\rightarrow$ Helper party
Concept – key aspects

• **Data parties**
  - Jointly create shared keypair
  - Filter and encrypt own data locally
  - Pseudonimize ID’s

• **Helper party**
  - Performs intersection + aggregate calculations
  - Sends encrypted aggregates back to data parties for decryption

• **No party learns other population, only sizes**

• **Data parties should not collude with helper party**
Calculate average income for people with mobile roaming costs > 200

### Example

<table>
<thead>
<tr>
<th>ID</th>
<th>P1: Income</th>
<th>PID</th>
<th>P1: Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1700</td>
<td>@a3</td>
<td>1700</td>
</tr>
<tr>
<td>2</td>
<td>2300</td>
<td>5%u</td>
<td>2300</td>
</tr>
<tr>
<td>4</td>
<td>1500</td>
<td>22&gt;</td>
<td>1500</td>
</tr>
<tr>
<td>5</td>
<td>5200</td>
<td>lab</td>
<td>5200</td>
</tr>
<tr>
<td>6</td>
<td>6100</td>
<td>?o9</td>
<td>6100</td>
</tr>
</tbody>
</table>

Helper

<table>
<thead>
<tr>
<th>PID</th>
<th>P2: Mobile Roaming Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>@a3</td>
<td>50</td>
</tr>
<tr>
<td>5%u</td>
<td>40</td>
</tr>
<tr>
<td>22&gt;</td>
<td>210</td>
</tr>
<tr>
<td>lab</td>
<td>300</td>
</tr>
<tr>
<td>?o9</td>
<td>250</td>
</tr>
</tbody>
</table>

Avg: 11300 / 2

Send to P1 & P2
Thank you!

Questions?

r.schreijen@cbs.nl
Facts that matter