cbs

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 :**

$$\mathit{Decrypt}(\mathbf{Function}(\mathit{Encrypt}(\mathbf{x}))) = \mathbf{Function}(\mathbf{x})$$

(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 (**ciphertext** · **ciphertext**)
  - Additive (**ciphertext** + **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: 173673460192093840927923765987234612387100209387877742341

Plaintext: 3

Ciphertext: 992837464510293746281238476009237498762346627747848822164

PSI using HE

Concept – key aspects

- **Set membership** of a private set can be expressed **numerically (1 = in** my set, **0 = 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**

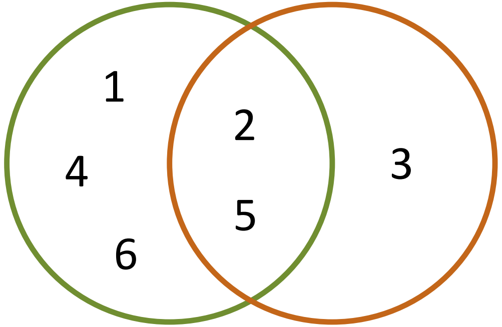
Example

P1

1. **Create** keypair

Public Key

Private Key



P2

ID	P1	P1P2
1	1	1
2	1	1
3	0	0
4	1	1
5	1	1
6	1	1

2. **Create** encrypted initial table

3. **Send** table + public key

Plaintext: 1 or 0

Ciphertext: ‘Random’ **13625374890387672524103874 6…**

Encrypted columns!

4. **Link records**: matching ID’s

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

ID	P1	P1P2	P2
1	1	~~1~~ 0	0
2	1	1	1
3	0	0	1
4	1	~~1~~ 0	0
5	1	1	1
6	1	~~1~~ 0	0
Σ	5	2	3

6. **Sum** columns

Encrypted columns!

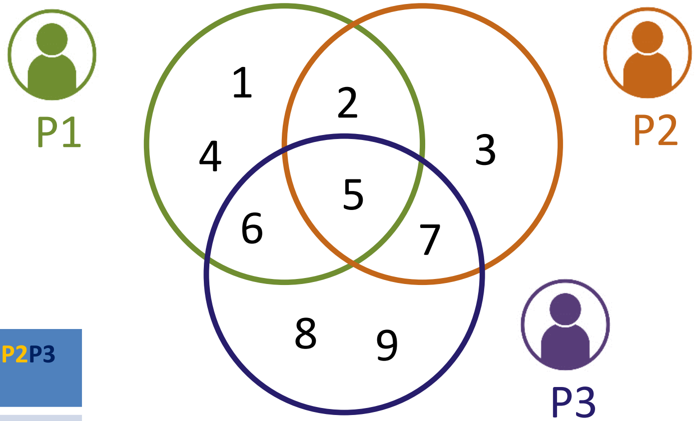
7. **Return** encrypted counts (bottom row)

	P1	P1P2	P2
Σ	5	2	3

6. **Decrypt** and reveal counts

Extending concept: more parties

ID	P1	P1P2	P1P3	P1P2P3
1	1	1	1	1
2	1	1	1	1
3	0	0	0	0
4	1	1	1	1
5	1	1	1	1
6	1	1	1	1
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0



ID	P1	P1P2	P1P3	P1P2P3	P2	P2P3
1	1	~~1~~ 0	1	~~1~~ 0	0	0
2	1	1	1	1	1	1
3	0	0	0	0	1	1
4	1	~~1~~ 0	1	~~1~~ 0	0	0
5	1	1	1	1	1	1
6	1	~~1~~ 0	1	~~1~~ 0	0	0
7	0	0	0	0	1	1
8	0	~~0~~ 0	0	~~0~~ 0	0	0
9	0	~~0~~ 0	0	~~0~~ 0	0	0

ID	P1	P1P2	P1P3	P1P2P3	P2	P2P3	P3
1	1	0	~~1~~ 0	~~0~~ 0	0	~~0~~ 0	0
2	1	1	~~1~~ 0	~~1~~ 0	1	~~1~~ 0	0
3	0	0	~~0~~ 0	~~0~~ 0	1	~~1~~ 0	0
4	1	0	~~1~~ 0	~~0~~ 0	0	~~0~~ 0	0
5	1	1	1	1	1	1	1
6	1	0	1	0	0	0	1
7	0	0	0	0	1	1	1
8	0	0	0	0	0	0	1
9	0	0	0	0	0	0	1
Σ	5	2	2	1	4	2	5

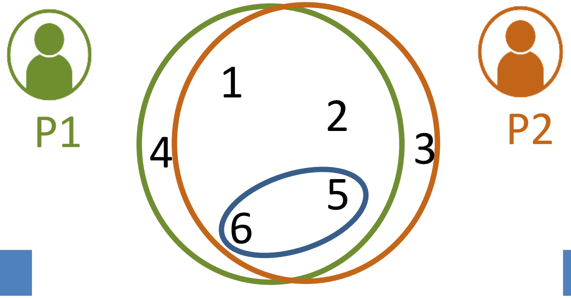
P1 encrypt→ P2 replace → P3 replace and sum →
P1 decrypt aggregates + broadcast

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*

ID	P1: Income
1	1700
2	2300
3	0
4	1500
5	5200
6	6100

- 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**

ID	P1: Income	P2: Mobile Roaming Costs	P2: *Filter (count)*
1	~~1700~~ 0	50	0
2	~~2300~~ 0	40	0
3	~~0~~ 0	160	0
4	~~1500~~ 0	0	0
5	5200	300	1
6	6100	250	1
Σ	11300		2

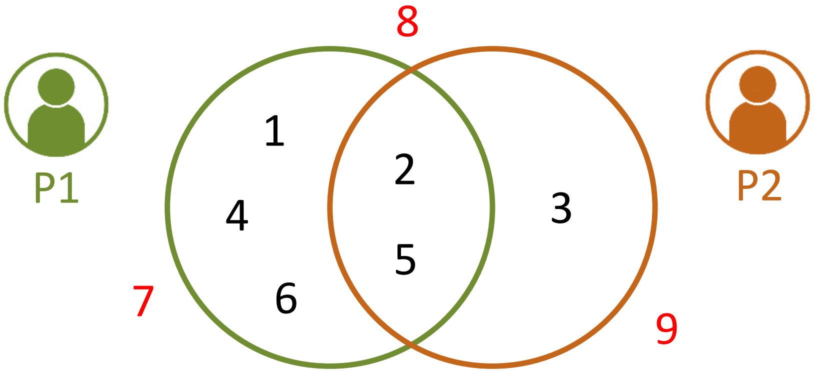
Encrypted column!

	Income sum	People count
Σ	11300	2

Avg = Decrypt(11300) / 2

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 → lower performance (ciphertext expansion & data exchange, more computations etc.)
- What if P1 and P2 sensitive and superset not viable?? → **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**

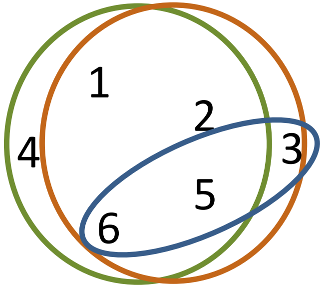
Example

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

ID	P1: Income
1	1700
2	2300
4	1500
5	5200
6	6100

PID	P1: Income
@a3	1700
5%u	2300
22>	1500
!ab	5200
?o9	6100

P1



P2

ID	PID	P2: Mobile Roaming Costs
1	@a3	50
2	5%u	40
3	gd2	210
5	!ab	300
6	?o9	250

PID	P2: Mobile Roaming Costs
gd2	210
!ab	300
?o9	250

	P1: income	Matches
Σ	11300	2

Avg: 11300 / 2

Send to P1 & P2

Helper

Match ID's and aggregate

PID	P1: Income	P2: Mobile Roaming Costs
@a3	1700	-
gd2	-	210
5%u	2300	-
22>	1500	-
!ab	5200	300
?o9	6100	250
Σ	11300	

Thank you!

Questions?

r.schreijen@cbs.nl

Facts that matter