Bloom Filter-Based MPSI

Weekly Progress Meeting 11 Dec 2025

Andra Alăzăroaie

Supervisor: Lilika Markatou

Daily Supervisor: Tjitske Koster

11-12-2025

Bibliography

[1] Jelle Vos, Jorrit van Assen, Tjitske Koster, Evangelia Anna Markatou, and Zekeriya Erkin. On the insecurity of bloom filter-based private set intersections. Cryptology ePrint Archive, 2024.

[2] Aslı Bay, Zekeriya Erkin, Jaap-Henk Hoepman, Simona Samardjiska, and Jelle Vos. Practical multi-party private set intersection protocols. IEEE Transactions on Information Forensics and Security, 17:1–15, 2021.

[3] Ou Ruan, Changwang Yan, Jing Zhou, and Chaohao Ai. A practical multiparty private set intersection protocol based on bloom filters for unbalanced scenarios. Applied Sciences, 13(24):13215, 2023.

[4] Ou Ruan and Chaohao Ai. An efficient multi-party private set intersection protocols based on bloom filter. In Second International Conference on Algorithms, Microchips, and Network Applications (AMNA 2023), volume 12635, pages 282–287. SPIE, 2023.

[5] Jelle Vos, MauroConti, and Zekeriya Erkin. Fast multi-party private set operations in the star topology from secure ands and ors. Cryptology ePrint Archive, 2022.

Progress

From the plan last week:

- Continue working on the code for [4].
  - When I get to benchmarking, I want to also research better ways of measuring time in C++ and of calculating communication costs (take inspiration from the APSI paper). Next week
- Read [5] and inspect its implementation.
- Ongoing study: the attack and proofs in [1].

Progress

- Read [5] and went through its code.
- Finished the implementation of [4], tested it.
  - Still have to do benchmarking

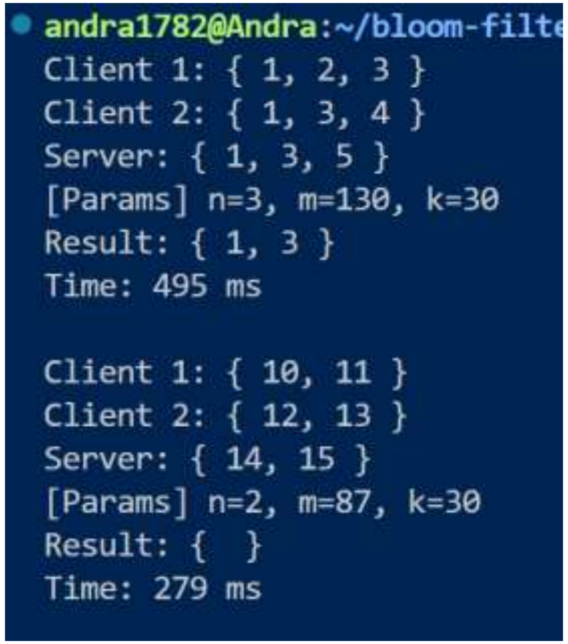
Progress

Description of the protocol of [4]:

- Clients create and encrypt their BFs using ElGamal (for the “0” bins, they encrypt a random group element)
- Server combines them (multiplies them), obtains the combined encrypted BF
- Clients compute decryption shares using the combined encrypted BF and their secret keys
- Server combines them (multiplies them), obtains the combined BF
- Server checks its own elements against the combined BF and thus computes the intersection

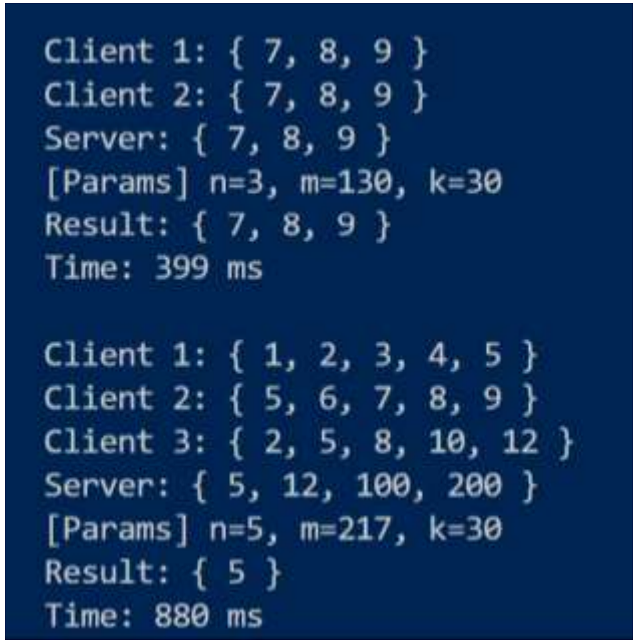
Progress

Tests for [4]:



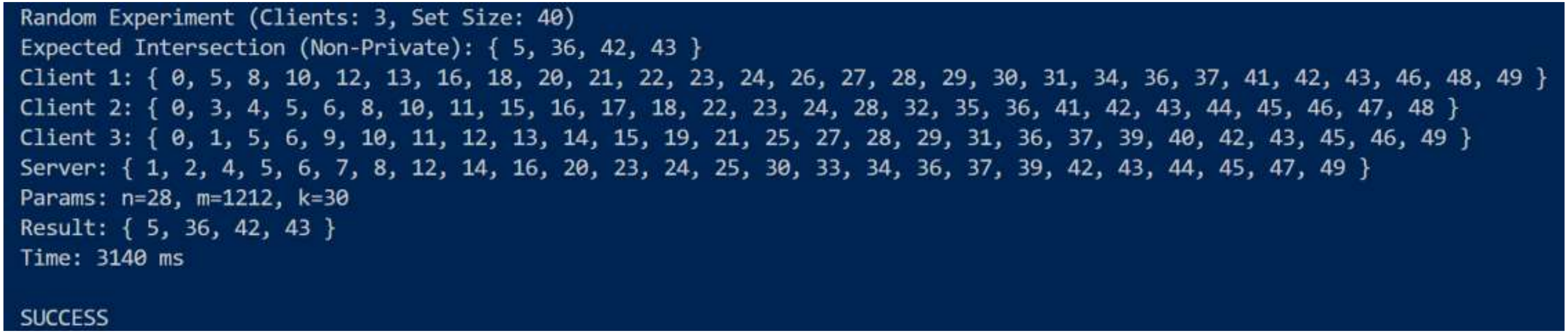
```
andra1782@Andra:~/bloom-filte
Client 1: { 1, 2, 3 }
Client 2: { 1, 3, 4 }
Server: { 1, 3, 5 }
[Params] n=3, m=130, k=30
Result: { 1, 3 }
Time: 495 ms

Client 1: { 10, 11 }
Client 2: { 12, 13 }
Server: { 14, 15 }
[Params] n=2, m=87, k=30
Result: {  }
Time: 279 ms
```



```
Client 1: { 7, 8, 9 }
Client 2: { 7, 8, 9 }
Server: { 7, 8, 9 }
[Params] n=3, m=130, k=30
Result: { 7, 8, 9 }
Time: 399 ms

Client 1: { 1, 2, 3, 4, 5 }
Client 2: { 5, 6, 7, 8, 9 }
Client 3: { 2, 5, 8, 10, 12 }
Server: { 5, 12, 100, 200 }
[Params] n=5, m=217, k=30
Result: { 5 }
Time: 880 ms
```



```
Random Experiment (Clients: 3, Set Size: 40)
Expected Intersection (Non-Private): { 5, 36, 42, 43 }
Client 1: { 0, 5, 8, 10, 12, 13, 16, 18, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 34, 36, 37, 41, 42, 43, 46, 48, 49 }
Client 2: { 0, 3, 4, 5, 6, 8, 10, 11, 15, 16, 17, 18, 22, 23, 24, 28, 32, 35, 36, 41, 42, 43, 44, 45, 46, 47, 48 }
Client 3: { 0, 1, 5, 6, 9, 10, 11, 12, 13, 14, 15, 19, 21, 25, 27, 28, 29, 31, 36, 37, 39, 40, 42, 43, 45, 46, 49 }
Server: { 1, 2, 4, 5, 6, 7, 8, 12, 14, 16, 20, 23, 24, 25, 30, 33, 34, 36, 37, 39, 42, 43, 44, 45, 47, 49 }
Params: n=28, m=1212, k=30
Result: { 5, 36, 42, 43 }
Time: 3140 ms

SUCCESS
```

Challenges

- Finishing the implementation details

Next Week

- Implement benchmarking for [4].
- Adapt the implementation to [3]?
- Ongoing study: the attack and proofs in [1].
- Revision of [2], [3], [5].

TU Delft

Thank you!

11-12-2025