

# Bloom Filter-Based MPSI

Weekly Progress Meeting 12 Feb 2026

Andra Alăzăroaie

Supervisor: Lilika Markatou

Daily Supervisor: Tjitske Koster

12-02-2026



# 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, Mauro Conti, and Zekeriya Erkin. Fast multi-party private set operations in the star topology from secure ands and ors. Cryptology ePrint Archive, 2022.
- [6] Vladimir Kolesnikov, Naor Matania, Benny Pinkas, Mike Rosulek, and Ni Trieu. Practical multi-party private set intersection from symmetric-key techniques. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pages 1257–1272, 2017.
- [7] Alireza Kavousi, Javad Mohajeri, and Mahmoud Salmasizadeh. Efficient scalable multi-party private set intersection using oblivious prf. In International Workshop on Security and Trust Management, pages 81–99. Springer, 2021.
- [8] Florian Kerschbaum. Outsourced private set intersection using homomorphic encryption. In Proceedings of the 7<sup>th</sup> ACM Symposium on Information, Computer and Communications Security, pages 85–86, 2012.

# Progress

From the plan last week:

- Prepare the presentation for the 1<sup>st</sup> stage evaluation
- Continue working on the mitigations

# Progress

OPRFs:

- Each client generates a key  $k_{OPRF}$
- Pre-processing stage:
  - Original protocol: each client hashes their element using the  $k$  hash functions to get the indices in the BF
  - Mitigation: each client computes the indices by using their  $k_{OPRF}$
- Online stage:
  - Original protocol: server computes the  $k$  hashes for each of its elements to get indices and collect the encrypted values at those indices from the clients' encrypted BFs
  - Mitigation: server has to engage in an OPRF protocol with each client in order to get the indices

# Progress

OPRFs:

- Use this OPRF

```
1: // Server S initiates a request:
2: Request( $\mathcal{M}$ )  $\rightarrow t, \mathcal{B}$ 
3:    $t \leftarrow \mathbb{Z}_q$ 
4:    $\mathcal{B} \leftarrow \{a^t\}_{a \in \mathcal{M}}$ 
5:   return  $t, \mathcal{B}$ 

6: // Client C applies PRF:
7: Eval( $\mathcal{B}, k$ )  $\rightarrow C$ 
8:   return  $\{b^k\}_{b \in \mathcal{B}}$ 

9: // Server S recovers elements:
10: Recover( $C, t$ )  $\rightarrow \mathcal{D}$ 
11:   return  $\{c^{\frac{1}{t}}\}_{c \in C}$ 
```

Figure 8: OPRF Protocol

# Next Week

- Implement the OPRF mitigation
- Start the Garbled BF mitigation ([6] and [7] use GBFs to implement OPRF)
- Read about the Authorized MPSI mitigation [8]

# Thank you!

12-02-2026