Master's Thesis Proposal

Andra Alăzăroaie
Supervisor: Lilika Markatou
Daily Supervisor: Tjitske Koster

1 Introduction

Private Set Intersection (PSI) is a cryptographic primitive that allows two or more parties, each holding a private dataset, to compute the intersection of these datasets without revealing any other information about their non-intersecting elements. This enables privacy-preserving collaboration between mutually distrustful entities and has numerous applications in domains such as contact medical analysis, anomaly detection and social networks [1].

Many practical scenarios of PSI protocols require participation from multiple entities, motivating the study of *multi-party PSI* (MPSI). The MPSI problem generalizes PSI to n participants, ensuring that no party learns information beyond the intersection of all inputs. Constructions for PSI cannot be efficiently generalized, which makes the MPSI problem more challenging than its two-party counterpart [2].

A studied approach to PSI leverages Bloom filters, a probabilistic data structure used to test set membership efficiently. A Bloom filter begins as a bit array of size m (the number of "bins") initialized to zero. To insert an element, h different hash functions are used to map the element to h positions in the array, setting the bits at these positions to one. To check if an element is in the set, one checks if the bits at all its h hash positions are set to one. This representation enables efficient intersection computation using only bitwise `AND` operations. This design can result in false positives (a query incorrectly returns true), but no false negatives. We can express the false positive probability p as a function of the filter size m, the number of hash functions h and the number of inserted elements k.

Despite their efficiency, recent research has demonstrated that Bloom filter-based PSI protocols suffer from privacy vulnerabilities. The work by Vos et al. [3] shows that maliciously crafted inputs can reveal information about other parties' private elements, violating the privacy guarantees of PSI. This motivates the research question of this proposal:

Can we design an efficient and secure Bloom filter-based multi-party PSI (MPSI) protocol?

We also formulate the following subquestions:

1. What is the baseline performance (in terms of computation, communication, and scalability) of an existing, unmitigated Bloom filter MPSI protocol?
2. Under which conditions do Bloom filter-based MPSI protocols leak information and how does the attack by Vos et al. [3] exploit these conditions?

3. To what extent do existing mitigation techniques (such as adjusting the parameters m, k, and h, OPRF-based blinding, garbled Bloom filters and input validation using a judge) reduce or prevent this leakage? How do these mitigation techniques impact performance?

4. Can we design a new Bloom filter-based MPSI protocol that provides both improved privacy guarantees and improved performance?

5. How does the proposed protocol compare to existing mitigations in terms of computation, communication, and scalability across different network communication topologies (star, full-mesh, wheel etc.)?

To address these questions, the thesis will proceed in several stages, which are outlined in Section 3.

2 Related Work

Private Set Intersection (PSI) protocols and their multi-party variants are Secure Multi-Party Computation (SMPC) protocols that enable a group of n parties, each holding a private set X_i, to jointly compute the intersection $X_1 \cap \cdots \cap X_n$ without revealing any additional information about their inputs. A common formulation assumes a designated leader who learns the intersection. These protocols have significant practical applications across various domains, including coordinating available time slots across private calendars, executing joint marketing campaigns, identifying common security threats in networked systems, detecting botnet infections and discovering fraudulent activities [2].

Approximate PSI protocols trade exact correctness for efficiency, allowing false positives in the intersection result. A commonly used approach in this category is based on Bloom filters. In Bloom filter-based MPSI, each party encodes its set into a Bloom filter using h hash functions over m bins, and the filters are combined using a bitwise AND operation within a secure computation framework. This approach avoids false negatives, but may introduce false positives with probability p, making it suitable for applications where some inaccuracy is acceptable in exchange for lower computation and communication costs.

Some of the protocols that build on this design combine Bloom filters with homomorphic encryption. The protocol by Bay et al. [4] is a multi-party variant where clients use a shared public key to send encrypted inverted Bloom filters to a server. In an inverted Bloom filter, the bits of a standard Bloom filter are simply flipped. The server combines the filters and the clients jointly decrypt the result, allowing the server to learn the intersection.

Similarly, Vos et al. [5] propose an MPSI protocol for large universes using ElGamal encryption. In this protocol, parties compute the secure OR of their inverted Bloom filters, which is equivalent to securely computing the AND of the standard Bloom filters.

Other protocols focus on specific MPSI scenarios. Ruan et al. [6] present a protocol for the unbalanced settings, where a server with a large set interacts with smaller clients. It uses ElGamal encryption and a trusted third party for key generation, allowing the server to homomorphically aggregate client filters. Ruan and Ai [7] adapt this for balanced scenarios, where all parties, including the server, encrypt their filters and engage in a decryption process.

The protocol by Miyaji and Nishida [8] takes an outsourced approach, using a non-participating dealer to reduce computation for the clients. Clients send encrypted filters to the dealer, who homomorphically adds them, with the final result being jointly decrypted by the clients.

However, recent work has demonstrated that this approximation can also introduce security vulnerabilities. The paper “On the Insecurity of Bloom Filter-Based Private Set Intersections” [3] shows that the actual false positive rate of the Bloom filter depends on the distribution of inputs, and this dependency can leak information when adversaries are allowed to choose their inputs. A party can exploit false positives to infer information about other parties’ private sets. The authors further show that making the false positive rate negligible by tuning the parameters, which would be required to close this security gap, results in a severe impact on the computational efficiency of Bloom filter-based MPSI. They also discuss other mitigation approaches, such as using OPRFs instead of hash functions or letting a third party authorize the input sets before proceeding with the protocol.

An approach to securing the filter’s contents, which avoids the issues related to observable false positives, is the use of Garbled Bloom Filters (GBF) [9, 10], which add some randomization when inserting elements in the Bloom Filter. A party provides a garbled data structure that allows another party to obliviously query for their elements. The evaluating party only learns the final output for their specific queries and learns nothing about the filter’s internal structure, preventing the known leakage.

These studies indicate that Bloom filter-based MPSI protocols entail a trade-off between efficiency and privacy. The proposed research builds on this observation, aiming to investigate mitigation strategies and to design a new Bloom filter-based MPSI protocol that preserves efficiency while addressing the privacy vulnerabilities.

3 Methodology

We will begin by implementing a Bloom filter-based MPSI protocol from the literature (e.g., [4], [5], [6], [7] or [8]). This implementation will serve both as a performance baseline and as a foundation for subsequent experimentation. We will then analyze the attack presented in [3] to determine the precise conditions under which privacy leakage occurs.

To address the vulnerability, we will experiment with several mitigation techniques:

- **Adjusting Bloom filter parameters to reduce collision-based leakage:** The most direct mitigation is to make the false positive rate p negligible ($p \leq 2^{-30}$). We will integrate this by adjusting the Bloom filter parameters: significantly increasing the filter size (m) and recalculating the number of hash functions (h). The expected result is a reduction in leakage at the cost of a large increase in computation and communication.

- **Integrating OPRFs to blind inputs prior to insertion into the Bloom filter:** The attack in [3] relies on the adversary’s ability to perform many offline queries to the public hash functions to find malicious inputs. To mitigate this, we will replace the standard hash functions with Oblivious Pseudorandom Functions (OPRFs) [9], which use a secret seed. In this integration, parties no longer have direct access to the hash function. Instead, to get the hash positions for an element, a party must engage in an OPRF protocol. This interaction can be limited, for example, to k total calls for their k input elements.

- **Using Garbled Bloom Filters (GBF) to mitigate leakage:** Garbled Bloom Filters introduce additional randomness during the element insertion process compared to Bloom Filters-based protocols. We will integrate a GBF-based protocol (e.g., [9, 10]) where parties construct a garbled data structure representing their filter. This structure is sent to the other parties, who can then obliviously test their own elements against it. This approach might make the attack from [3] more difficult, as the adversary cannot as easily observe and exploit the standard false positive behavior. [3] proposes studying whether the attack can be extended to protocols based on GBF, which will be explored during the thesis.

- **Introducing a judge to verify parties' inputs before the protocol begins (Authorized PSI):** This mitigation outsources trust to a semi-honest third party [11]. To prevent offline attacks, the server's elements are first encrypted (e.g., raised to a secret power e) before being added to its filter. The client must send its set to the judge, who applies the same secret transformation to the client's elements, builds their Bloom filter, and signs it. This helps because the adversary cannot find exploitable collisions against the blinded server filter.

The performance of each mitigation strategy will be evaluated relative to the baseline protocol.

Next, we will design a new Bloom filter-based MPSI protocol that achieves an improved balance between efficiency and security. We will develop formal security arguments. We will also design experiments to compare the computation and communication costs of the new protocol against the previously implemented mitigations. These experiments will further assess the influence of different communication topologies among the parties. In a *star topology*, a designated leader coordinates communication; in a *full-mesh topology*, all parties communicate directly; and in a *wheel topology*, communication is restricted to neighboring parties.

All implementations will be carried out in C++ using standard cryptographic libraries. Experiments will be automated and reproducible.

4 Timeline

The projected timeline of the project is November 2025 - June 2026, structured as follows:

November - January: Familiarize with the literature on MPSI and Bloom filter-based approaches. Implement an existing Bloom filter MPSI protocol. Study the attack described in [3]. Analyze vulnerabilities and document the sources of information leakage in the implemented protocol.

January - February: Implement different mitigation for the discovered flaw. These could be based on larger Bloom filters, OPRFs [9], garbled Bloom filters [9, 10] and possibly the introduction of a judge.

February - May: Design and implement a new secure and efficient Bloom filter-based MPSI protocol. Conduct experiments to compare the speed of the new protocol with the implemented mitigations and see the effects of different topologies.

May - June: Finalize the writing of the thesis. The writing process will be continuous, with each stage's work documented as it completes. The introduction and related work sections will begin in the first stage. This period is reserved for final modifications.

Throughout the project, progress will be reported during weekly meetings. As preparation, a short slide deck will summarize progress, encountered challenges, and plans for the upcoming week.

References

[1] Daniel Morales, Isaac Agudo, and Javier Lopez. Private set intersection: A systematic literature review. *Computer Science Review*, 49:100567, 2023.

[2] Jelle Vos, Mauro Conti, and Zekeriya Erkin. Sok: Collusion-resistant multi-party private set intersections in the semi-honest model. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 465–483. IEEE, 2024.

[3] 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.

[4] 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.

[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] 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.

[7] 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.

[8] Atsuko Miyaji and Shohei Nishida. A scalable multiparty private set intersection. In *International conference on network and system security*, pages 376–385. Springer, 2015.

[9] 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.

[10] 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.

[11] Florian Kerschbaum. Outsourced private set intersection using homomorphic encryption. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pages 85–86, 2012.

Bloom Filter-Based Multi-Party Private Set Intersection (MPSI)

Andra Alăzăroaie

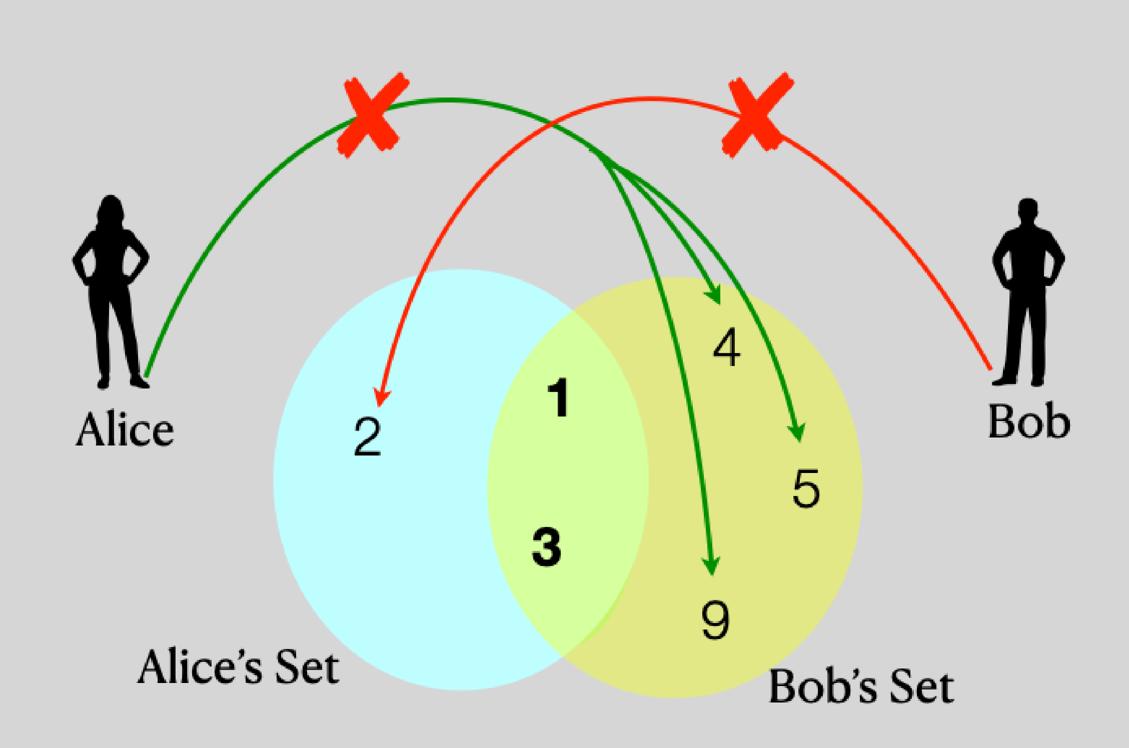
Supervisor: Lilika Markatou

Daily Supervisor: Tjitske Koster

14-11-2025

Introduction

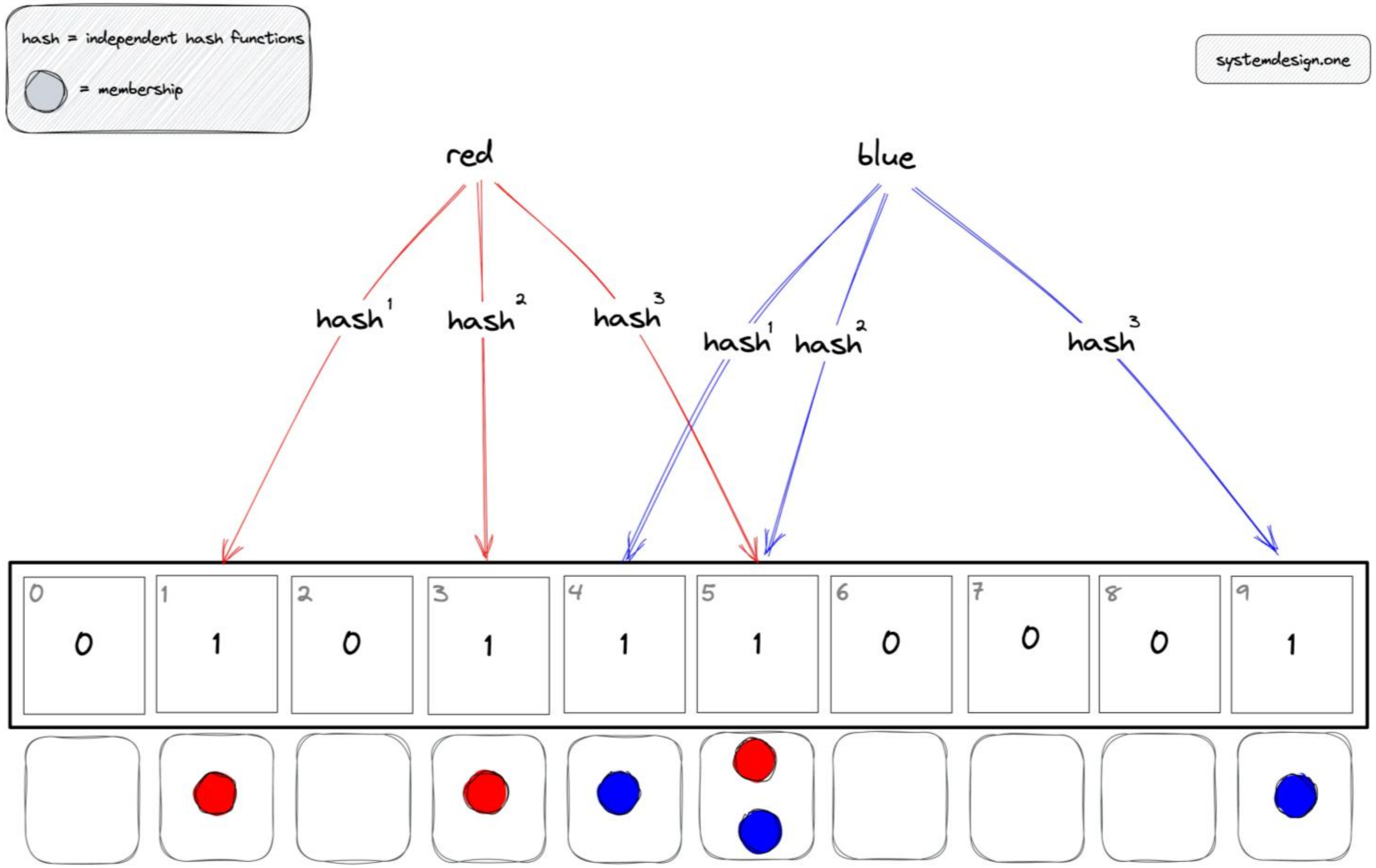
- **Private Set Intersection (PSI):** A cryptographic tool that allows two or more parties to find the items they have in common.
- Parties compute this intersection *without revealing any of their other private data* to each other.
- **Multi-Party PSI (MPSI):** This generalizes PSI to n participants, who want to find the items *all* of them hold in common. Unfortunately, simple two-party PSI protocols cannot be efficiently generalized to multiple parties [1].
- **Applications:** Contact medical analysis, anomaly detection, social networks, and joint marketing campaigns [2].



https://en.wikipedia.org/wiki/Private_set_intersection

Background: Bloom Filters

- A common MPSI approach uses **Bloom Filters**, a probabilistic data structure for testing set membership.
- **Insert:** To add an item, hash it h times to get h positions in a bit array (size m). Set those bits to **1**.
- **Query:** To check if an item is in the set, hash it h times and check if all h bits are **1**.
- Finding the intersection is just a **bitwise AND** operation on all parties' filters.
- No *false negatives*, but it can have *false positives.*



https://systemdesign.one/bloom-filters-explained/

The Research Gap

- Bloom filter-based MPSI was long considered a good trade-off: **high efficiency** in exchange for a **small, acceptable error rate** (false positives).
- Recent research (Vos et al., 2024 [3]) demonstrates that these protocols are **fundamentally insecure**.
  - An adversary can use **maliciously crafted inputs** to exploit the filter's properties.
  - By observing the false positive rate which depends on the *distribution* of inputs, the attacker can **infer information about other parties' private data**.
  - This **violates the privacy guarantee** of PSI.
- The simple fix is to make the false positive rate negligible (e.g., $p \leq 2^{-30}$). However, doing this requires such a large filter size (m) that the protocol becomes **computationally inefficient**, defeating its entire purpose.

Main Research Question

This leads to the central question of this thesis:

Can we design an efficient and secure Bloom filter-based multi-party PSI (MPSI) protocol?

The goal is to find a new design that addresses the vulnerability from Vos et al. [3] *without* sacrificing the performance that made Bloom filters attractive in the first place.

Sub-Questions

To answer the main question, we will investigate five sub-questions:

1. What is the **baseline performance** (in terms of computation, communication, and scalability) of an existing, unmitigated Bloom filter MPSI protocol?

2. Under which conditions do Bloom filter-based MPSI protocols leak information and how does the attack by Vos et al. [3] exploit these conditions?

3. To what extent do existing **mitigation techniques** (such as adjusting the parameters m, k, and h, OPRF-based blinding, garbled Bloom filters and input validation using a judge) reduce or prevent this leakage? How do these mitigation techniques impact performance?

4. Can we design a **new Bloom filter-based MPSI protocol** that provides both improved **privacy guarantees** and improved **performance**?

5. How does the proposed protocol compare to existing mitigations in terms of computation, communication, and scalability across different network communication topologies (star, full-mesh, wheel etc.)?

Methodology: Stage 1 – Familiarization

- **Implement Baseline Protocol:**
  - We will implement an existing Bloom filter MPSI protocol from the literature (e.g., Bay et al. [4], Vos et al. [5], Ruan et al. [6] or Ruan and Ai [7]).
  - This implementation will serve as our performance baseline for future comparisons.

- **Analyze the Attack:**
  - We will study the attack presented by Vos et al. [3].
  - We will identify the precise conditions that allow information leakage in our baseline implementation.

Methodology: Stage 2 - Mitigations

Next, we will implement and evaluate four known mitigation strategies:

1. **Parameter Tuning:**
   - Make the false positive rate negligible ($p \leq 2^{-30}$) by significantly increasing the filter size m and recalculating the number of hash functions h.
   - **Expected result:** Reduction in leakage, but increase in computation and communication costs.

2. **OPRFs (Oblivious Pseudorandom Functions):**
   - Replace public hash functions with a secret-seeded OPRF. Parties must engage in a protocol to get hash positions.
   - **Expected result:** Prevents the attacker from performing many offline queries, as they cannot compute hashes freely.

Methodology: Stage 2 - Mitigations

3. **Garbled Bloom Filters (GBF):**
   - GBFs introduce additional randomness during element insertion. Parties construct a garbled data structure representing their filter, which is sent to other parties who can obliviously test their own elements against it.
   - **Expected result:** Prevents the adversary from observing and exploiting the false positive behavior.

4. **Authorized PSI (Judge):**
   - We outsource trust to a semi-honest third party (judge) to pre-process and encrypt inputs *before* they are inserted into the filter. The judge signs the Bloom filter.
   - **Expected result:** Prevents the adversary from constructing malicious inputs.

Methodology: Stage 3 – New Protocol

- **Design New Protocol:**
  - Using insights from Stages 1 and 2, we will design a **new Bloom filter-based MPSI protocol,** while the primary goal is to achieve an improved balance: **security** against the [1] attack while maintaining **efficiency**.
  - We will develop formal security arguments for the new protocol.
- **Evaluation:**
  - We will experimentally compare our new protocol against the baseline and mitigation strategies.
  - **Metrics:** Computation and communication cost, scalability.
  - **Network Topologies:** We will test how performance is affected by different communication topologies (Star, Full-Mesh, and Wheel).
  - **Tools:** All implementations will be in C++ using standard crypto libraries.

Timeline (Nov 2025 – Jun 2026)

1. **Baseline Implementation (November – January)**
   - Implement an existing Bloom filter based MPSI protocol
   - Analyze why the implementation is vulnerable based on [1]
2. **Mitigations (January – February)**
   - Implement and evaluate mitigations (larger Bloom Filters, OPRF, GBF, Authorized PSI)
3. **New Bloom Filter-based protocol (February – May)**
   - Design and implement a new, secure, Bloom-filter based MPSI protocol
   - Conduct experiments to compare the new protocol with the baseline and mitigations
4. **Writing (May – June)**
   - Finalize thesis writing and prepare presentation

Bibliography

[1] Jelle Vos, Mauro Conti, and Zekeriya Erkin. Sok: Collusion-resistant multi-party private set intersections in the semi-honest model. In 2024 IEEE Symposium on Security and Privacy (SP), pages 465–483. IEEE, 2024.

[2] Daniel Morales, Isaac Agudo, and Javier Lopez. Private set intersection: A systematic literature review. Computer Science Review, 49:100567, 2023.

[3] 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.

[4] 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.

[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] 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.

[7] 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.

TU Delft

Thank you!

14-11-2025