Author:
Lauren de Hoop
L.H.deHoop@student.tudelft.nl

Supervisors:
Dr. Z. Erkin, Dr. K. Atasu, L. Touwen
EEMCS, Delft University of Technology

Collaborative Detection of Malicious Clients for Financial Institutions

CSE3000 Research Project

Trustworthy Financial Crime Analysis
22/06/2025

01. Introduction

Financial crime has become increasingly sophisticated, with criminals exploiting gaps between institutions to launder money and commit fraud. Meanwhile, data protection regulations like GDPR limit the ability of financial institutions to share client information directly, even when criminal activity is suspected. Existing systems such as Pifi [1] allow for manual information sharing through a central database, but are expensive, inefficient, and reactive rather than proactive. This lack of efficient, privacy-preserving tools restrict the ability of institutions to collaboratively detect malicious clients.

This work explores privacy-preserving protocols, specifically Multi-Party Computation (MPC) and Private Set Intersection (PSI) to address these challenges. We present FT-MPSI, a Flagged Threshold PSI protocol that builds on T-MPSI [2], enabling institutions to jointly identify clients who appear in at least threshold *T* institutions and have been flagged by at least one as suspicious. The protocol was implemented in C++ and tested with up to 50 parties, showing practical run-times while preserving client privacy under the semi-honest model.

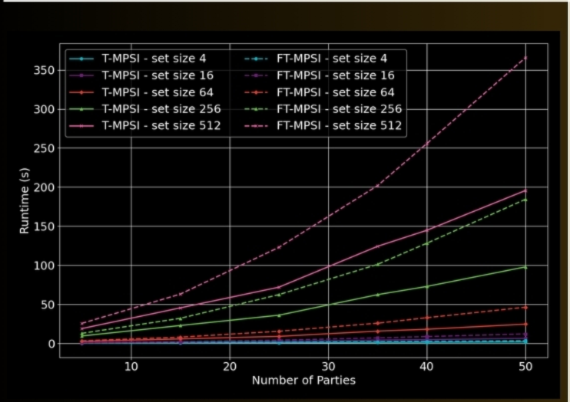

Figure 2: Run-time comparison of the T-MPSI and FT-MPSI protocols averaged over 10 runs

02. Preliminaries

Secret Sharing: You have a secret *S*, which you split into shares *{S*$_1$*, ..., S*$_n$*}*. In a *k*-out-of-*n* secret sharing algorithm, where *k* ≤ *n*, you need at least *k* out of the *n* shares to reconstruct *S*. Less than *k* shares will reveal nothing about *S*.

Bloom Filter: A space efficient probabilistic data structure which can be used to check whether an element is part of a set in a privacy perserving manner. The data structure is a list of bits, all 0 at first. For each item in a set, several hash functions are calculated and the bits at the positions corresponding to the hash are set to 1. The Bloom Filter is shared, and the other party can then calculate the intersection by hashing their items.

Homomorphic Encryption: A form of encryption that allows computations to be made directly on the encrypted data, without having to decrypt it first. Additive Homomorphic Encryption allows two values to be added to each other while encrypted. Decryption will then reveal the addition.

Private Set Intersection: Allowing multiple parties to intersect their private input without leaking information on items not in the intersection [3].

Threshold PSI: Multiple parties intersect their dataset, instead of an intersection between all parties, an intersection between a threshold amount T or more parties is identified.

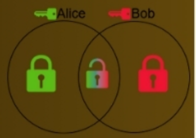

Figure 1: Private Set Intersection Diagram

03. Existing Data Sharing Solutions

Currently the following protocols exist:

1. **Bank Data Retrieval Portal** (VB) [4,5]:
   a. Authorised Institutions can request customer data;
   b. Authorised Institution: Dutch Investigative Services and the Tax Authorities;
   c. Automates obligatory legal proceedings for requesting customer data.
2. **Protocol Incident Warning System for Financial Institutions** (Pifi) [1,5]
   a. Collaborative warning system;
   b. Financial institutions register fraudulent users;
   c. Other financial institutions can request data on their (potential) clients.

05. Experimental Setup

The protocol is implemented in C++ and depends on GMP and NTL libraries and the MurmurHash implementation. The experiment compares the FT-MSPI protocol against the original T-MSPI protocol with regards to their run-time performance.

Both protocols were tested with varying numbers of sets and set size, benchmarked over the average of 10 runs. All benchmarks were executed on a 64-bit Linux system with an AMD Ryzen Threadripper 7970X CPU at 32 × 1.5 - 4.0 GHz and 270 GB RAM.

06. Analysis

Security Analysis: T-MPSI has been proven to be secure in the semi-honest model, i.e. a subset of parties is considered corrupt and may communicate with each other to gain additional information that is not revealed by the protocol, but these parties still adhere to the protocol correctly. FT-MSPI is specifically designed to adhere to the exact same security protocols as T-MPSI, as such, FT-MPSI is also secure in the semi-honest model.

Complexity Analysis:

Table 1: Communication and Computation Complexities of T-MPSI and FT-MPSI.

	Communication		Computation	
Protocol	**Client**	**Server**	**Client**	**Server**
T-MPSI	$\mathcal{O}(\max(\lambda, t)n)$	$\mathcal{O}(nt\ell)$	$\mathcal{O}(\max(\lambda, t)n)$	$\mathcal{O}(nt)$
FT-MPSI	$\mathcal{O}(\max(\lambda, t)n)$	$\mathcal{O}(nt\ell)$	$\mathcal{O}(\max(\lambda, t)n)$	$\mathcal{O}(nt)$

Performance Analysis: Table 2 reports the mean run-time, in seconds, of the T-MSPI protocol and the standard deviation calculated over 10 runs. Table 3 reports the matching results for the FT-MSPI protocol. Figure 2 creates a linear representation of all run-times of both T-MPSI and FT-MPSI. Each colour corresponds to a set size, where the dotted lines represent the FT-MPSI protocol and the solid lines the T-MPSI protocol.

Table 2: T-MPSI: Run-time performance mean and standard deviation in seconds by number of parties and set size averaged over 10 runs.

	$\mathbf{n = 2^2}$	$\mathbf{n = 2^4}$	$\mathbf{n = 2^6}$	$\mathbf{n = 2^8}$	$\mathbf{n = 2^9}$
$t = 5$	0.26 ± 0.02	0.63 ± 0.03	2.35 ± 0.02	9.39 ± 0.05	18.86 ± 0.09
$t = 15$	0.51 ± 0.04	1.55 ± 0.03	5.68 ± 0.04	22.82 ± 0.26	45.40 ± 0.42
$t = 25$	0.72 ± 0.03	2.37 ± 0.03	8.93 ± 0.05	36.01 ± 0.36	71.896 ± 0.61
$t = 35$	1.14 ± 0.01	4.11 ± 0.04	15.63 ± 0.07	62.30 ± 0.30	124.18 ± 0.80
$t = 40$	1.31 ± 0.01	4.70 ± 0.03	18.05 ± 0.15	72.82 ± 0.38	144.71 ± 0.59
$t = 50$	1.77 ± 0.03	6.34 ± 0.03	24.54 ± 0.08	97.84 ± 0.38	195.63 ± 1.98

Table 3: FT-MPSI: Run-time performance mean and standard deviation in seconds by number of parties and set size averaged over 10 runs.

	$\mathbf{n = 2^2}$	$\mathbf{n = 2^4}$	$\mathbf{n = 2^6}$	$\mathbf{n = 2^8}$	$\mathbf{n = 2^9}$
$t = 5$	0.37 ± 0.02	0.87 ± 0.03	3.22 ± 0.03	12.76 ± 0.05	25.51 ± 0.12
$t = 15$	0.68 ± 0.05	2.13 ± 0.12	7.94 ± 0.08	32.03 ± 0.32	63.05 ± 0.54
$t = 25$	1.22 ± 0.03	4.03 ± 0.07	15.44 ± 0.18	62.38 ± 0.78	122.90 ± 0.67
$t = 35$	1.99 ± 0.02	6.92 ± 0.09	25.79 ± 0.16	101.10 ± 0.64	201.81 ± 1.18
$t = 40$	2.49 ± 0.04	8.68 ± 0.10	32.80 ± 0.38	128.15 ± 0.54	255.76 ± 1.18
$t = 50$	3.34 ± 0.04	12.03 ± 0.11	46.24 ± 0.29	184.37 ± 1.49	365.67 ± 2.05

04. Proposed Solution

The implementation is an adaptation of the Threshold Multi-Party Private Set intersection (T-MPSI) [2]. This adaptation is called Flagged Threshold Private Set Intersection, and is designed to enable financial institutions to collaboratively detect potentially malicious clients while preserving data privacy.

FT-MPSI protocol:
Given *t* parties *P*$_i$ and *k* random hash functions. The last party *P*$_t$ is denoted as the server, the rest of the parties are the clients.

1. Each party computes an encrypted Bloom Filter (EBF) of their set and a separate encrypted Bloom Filter for their flagged items (FEBF) and sends both to the server.
2. For each item in the set of the server, the server then:
   a. Calculates all the hashes, for each client it then gets the bits at the positions in the EBF and sums them.
   b. This sum is will then reveal how many client sets hold the item. The sum is compared to *t* - 1, if the sum is ≥ *t* - 1, proceed to the next step. Otherwise discard the item.
3. The server then repeats the above steps for with the FEBF of the clients. For each item in the set of the server:
   a. Calculates all the hashes, for each client it then gets the bits at the positions in the FEBF and sums them.
   b. If this sum is ≥ 1, or the server set has flagged the client, the item is added to the intersection set, otherwise discard the item.
4. The server then reveals the intersection set to the other parties.

07. Conclusion

FT-MPSI Performance:

- FT-MPSI run-time increases almost linearly with the number of parties, and set sizes.
- This matches the complexity of *O(max(λ, t)n)*, where *λ* is a security parameter, *t* the number of parties and *n* the size of the sets.
- The standard deviation across the experiments are low, indicating a stable performance

T-MPSI vs FT-MPSI:

- T-MPSI consistently performs faster than FT-MPSI, taking half the run-time.
- FT-MPSI's additional cost comes from computing two Bloom Filters, one for all items in a party and one with only flagged items.
- FT-MPSI does scale similarly to T-MPSI and remains practical to implement.

Assumptions:

- All experiments were run on a single machine with no simulated delays.
- Parties were emulated but not physically separate.
- Protocol assumes all parties are honest and do not misuse the flags in the protocol to gain additional knowledge.

Future work:

- FT-MPSI uses a Secure Comparison Protocol, since its implementation, better versions of this protocol have been created, which could reduce the run-time of FT-MPSI significantly.
- The flagged intersection comparison could be improved by using a Private Set Union instead of the Threshold Set Intersection.
- Future implementations should test the protocol in a distributed environment.
- Legal compliance with the GDPR should be evaluated.

References

[1] 'Protocol Incidenten-waarschuwingssysteem Financiële Instellingen (Pifi)'. Accessed: May 08, 2025. [Online]. Available: https://www.nvb.nl/publicaties/protocollen-regelingen-richtlijnen/protocol-incidenten-waarschuwingssysteem-financiele-instellingen-pifi/

[2] A. Bay, Z. Erkin, J.-H. Hoepman, S. Samardjiska, and J. Vos, 'Practical Multi-Party Private Set Intersection Protocols', IEEE Transactions on Information Forensics and Security, vol. 17, pp. 1–15, 2022, doi: 10.1109/TIFS.2021.3118879.

[3] D. Morales, I. Agudo, and J. Lopez, 'Private set intersection: A systematic literature review', Computer Science Review, vol. 49, p. 100567, Aug. 2023.

[4] M. van J. en Veiligheid, 'Verwijzingsportaal Bankgegevens (VB) - Producten- en diensten catalogus - Justitiële Informatiedienst'. Accessed: May 08, 2025. [Online]. Available: https://www.justid.nl/producten-en-dienstencatalogus/digitaal-uitwisselen/routeren-informatie/verwijzingsportaal-bankgegevens-vb

[5] 'Data delen tegen criminaliteit'. Accessed: May 08, 2025. [Online]. Available: https://www.nvb.nl/themas/digitaal-van-dienst/uw-persoonsgegevens/data-delen-tegen-criminaliteit/