Comparing Secure Multi-party Computation Protocols

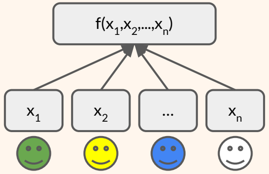
Pedro Gomes Moreira | pgomesmoreira@student.tudelft.nl

Supervisor: Lilika Markatou

TU Delft

1. Background

In Multiparty Computation (**MPC**), parties want to compute a function using their inputs, while preserving these input private to other parties. MPC protocols aim to achieve this, and can be classified in different categories.



- **Generic** protocols can compute any function; **Specialized** protocols can only compute some functions
- Protocols have different adversarial models. A **semi-honest** adversary tries to read more information, but follows the protocol. A **malicious** adversary does not follow the protocol and tries to change the computation.

2. Research Question

How do secure Multi-Party Computation protocols compare with each other and with other techniques for computation with encrypted data?

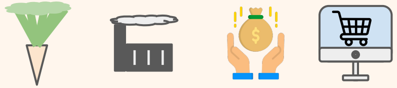
- How much do specialized protocols over perform generic ones and in which cases?
- What is the impact of the adversarial model on efficiency and security?
- How to convert semi-honest protocols to malicious-secure ones?
- What are possible optimizations to the protocol?
- How does MPC compare to other techniques for computing on encrypted data: fully homomorphic encryption (FHE), oblivious RAM (ORAM), structured encryption (StE), and trusted execution environments (TEEs)?

3. Methodology

- A literature survey was conducted on the field of MPC
- 56 papers were used in the survey.
- Snowball sampling: from two works that gave an overview of MPC, collected references on relevant topics. For each paper, read the relevant sections taking notes. Then, found references to broaden or deepen on the topics. Repeated this process for each paper.
- Used Scopus to find more up-to-date literature on state-of-the-art solutions.
- Papers on other techniques were provided by peers.

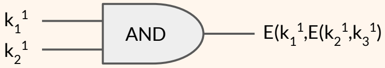
4. Real world applications

- Danish sugar beet auction (2009) [1]: first large-scale practical application.
- Deployed to analyze economic situation of industrial sector without revealing confidential information (2011) [2]: first usage over WAN with real data.
- Used to investigate wage inequality without revealing sensitive information (2018) [3].
- Private Set Intersection Sum with Cardinality developed by Google researchers to calculate ad conversion rate (2020) [4].



5. Protocols

- Main idea: convert function to a Boolean or arithmetic circuit
- Then, either garble (encrypt) the wire values…

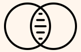


or secretly share them among parties A and B.

- The first approach is done by Yao's GC [5] and BMR [6]. The second approach is done by GMW [7] and BGW [8].
- BMR can be seen as an extension of Yao's GC for multiple parties.

- The protocols are semi-honest secure by default. However, techniques such as cut-and-choose and zero-knowledge proofs (e.g. GMW compiler) can make them malicious-secure, but with some costs.
- Some protocols can be optimized using point-and-permute, free XOR, garbled row reduction, pre-processing phase with multiplication triples, etc.

- Private Set Intersection (PSI),: two parties want to compute the intersection of their private values without revealing values that are not in the intersection.



- A specialized PSI protocol can be implemented using oblivious pseudorandom functions (OPRF) [9] and performs better than circuit-based (i.e. generic) protocols, though the former cannot be easily adapted and is only semi-honest secure.

Table 1: Comparison of MPC protocols. Checks indicate affirmative, crosses indicate negative, and minus signs indicate that malicious security is not present by default, but can be added. The headings are number of parties, maliciously secure, dishonest majority, information-theoretical security, round complexity, circuit type, practical usage, generic or specialized.

	Parties	Mal. secure	Dis. maj.	IT. sec	Round cplx.	Circuit type	Usage	Generic	Specialized
Yao's GC [5]	2	➖	✅	❌	Constant	Boolean	Large	✅	❌
GMW [7]	2 or more	➖	❌	❌	Linear	Bool./arith.	Medium	✅	❌
BGW [8]	2 or more	➖	❌	✅	Linear	Arithmetic	Medium	✅	❌
CCD [10]	2 or more	➖	❌	✅	Linear	Arithmetic	Little	✅	❌
BMR [6]	2 or more	➖	✅	❌	Constant	Boolean	Medium	✅	❌
PSI [9]	2	❌	✅	❌	Constant	N/A	Medium	❌	✅

Table 2: Comparison of techniques for computing on encrypted data. Overhead for MPC refers to round complexity

	Threat model	Leakage	Overhead	Usage	Computation	Parties
FHE	IND-CCA2	Nothing	Polynomial	Little	Any	Client(s)-server
MPC	SH / Mal.	Nothing	Const. / Lin.	Medium	Any	Any two or more
ORAM	Semi-honest	Side-channel attacks	Logarithmic	Medium	Any data access	Client(s)-server
StE	Semi-honest	Access patterns	Sublinear	Medium	Specific data access	Client(s)-server
TEEs	Malicious	Access patterns	Constant	Large	Any	Client-server + attestation dev.

6. Other Techniques

- **FHE**: the encryption of the result of a function on inputs is equal to the function applied to ciphertexts of the inputs, for any function. FHE can be used for MPC, and it is very efficient communication-wise, but very expensive computation-wise. In practice, other forms of homomorphic encryption are preferred.
- **ORAM**: a client using an untrusted server for storage can hide access patterns on the data. The context is more restricted, being only client-server. In this specific context, ORAM can achieve sublinear complexity, whereas MPC is always at least linear.
- **StE**: a client stores structured encrypted data on a database in such a way that it can still be queried. The context is similar to ORAM, more restricted compared to MPC. StE aims at practical efficiency and as such is not as secure as MPC and ORAM (for example, some information leakage is allowed).
- **TEEs**: a client can securely outsource computation on an untrusted server. The confidential computation relies on a trusted computing base and trusted hardware vendor. The context is again more restricted than MPC; it achieves higher efficiency than MPC, but with different security assumptions, involving trust in other entities.

7. Conclusions

- Specialized protocols can outperform generic ones for certain tasks, but are less flexible to extend and adapt.
- Real-world applications usually use either asymmetric (client-server) or specialized protocols.
- A semi-honest adversarial model is weaker, but more efficient than a malicious one.
- Semi-honest protocols can be converted into malicious-secure by cut-and-choose or zero-knowledge proofs, but this results in costs.
- Some protocols can be optimized by techniques such as free XOR and garbled row reduction.
- Comparisons of protocols and techniques are shown in Tables 1 and 2.

References

[1] D. Beaver, S. Micali, and P. Rogaway. The round complexity of secure protocols. In Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing, STOC '90, pages 503–513, New York NY, USA, 1990. Association for Computing Machinery. event-place: Baltimore, Maryland, USA.

[2] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault- tolerant distributed computation. In Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, STOC '88, pages 1–10, New York, NY, USA, 1988. Association for Computing Machinery. event-place: Chicago, Illinois, USA.

[3] Dan Bogdanov, Riivo Talviste, and Jan Willemson. De- ploying secure multi-party computation for financial data analysis, 2011. Published: Cryptology ePrint Archive, Paper 2011/662.

[4] Peter Bogetoft, Dan Lund Christensen, Ivan Damg˚ard, Martin Geisler, Thomas Jakobsen, Mikkel Krøigaard, Janus Dam Nielsen, Jesper Buus Nielsen, Kurt Nielsen, Jakob Pagter, Michael Schwartzbach, and Tomas Toft. Secure Multiparty Computation Goes Live. In Roger Dingledine and Philippe Golle, editors, Financial Cryptography and Data Security, pages 325–343, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[5] David Chaum, Claude Crepeau, and Ivan Damgard. Multiparty unconditionally secure protocols. In Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, STOC '88, pages 11–19, New York, NY, USA, 1988. Association for Computing Machinery. event-place: Chicago, Illinois, USA.

[6] O. Goldreich, S. Micali, and A. Wigderson. How to play ANY mental game. In Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing, STOC '87, pages 218–229, New York, NY, USA, 1987. Association for Computing Machinery. event place: New York, New York, USA.

[7] Mihaela Ion, Ben Kreuter, Ahmet Erhan Nergiz, Sarvar Patel, Shobhit Saxena, Karn Seth, Mariana Raykova, David Shanahan, and Moti Yung. On Deploying Secure Computing: Private Intersection-Sum-with-Cardinality. In 2020 IEEE European Symposium on Security and Privacy (EuroS&P), pages 370–389, 2020.

[8] Andrei Lapets, Frederick Jansen, Kinan Dak Albab, Rawane Issa, Lucy Qin, Mayank Varia, and Azer Bestavros. Accessible Privacy-Preserving Web-Based Data Analysis for Assessing and Addressing Economic Inequalities. In Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies, COMPASS '18, New York, NY, USA, 2018. Association for Computing Machinery. event-place: Menlo Park and San Jose, CA, USA.

[9] Yehuda Lindell. Secure Multiparty Computation (MPC), 2020. Published: Cryptology ePrint Archive, Paper 2020/300.

[10] Andrew Chi-Chih Yao. How to generate and exchange secrets. In 27th Annual Symposium on Foundations of Computer Science (sfcs 1986), pages 162–167, 1986.