Navigating the Performance-Security Trade-Off in Future Analytics on Shared Data

Zsolt István
Systems Group, TU Darmstadt, Germany
zsolt.istvan@tu-darmstadt.de

Securing analytics on shared data is important but expensive. Analyzing datasets from multiple data owners can yield valuable insights [1, 2, 3, 4, 5] but poses significant security risks. Even within enterprises – our primary focus – precautions are necessary when handling data across subsidiaries and geographic regions [6, 7]. Existing security solutions based on Trusted Execution Environments (TEEs) [8, 9], fully homomorphic encryption [10], and structured encryption [11] offer strong protections, albeit in a physically centralized manner. For more decentralization, there are exciting approaches based on Secure Multi-Party Computation (MPC) [12] that do not need a trusted third party nor merging datasets at a central location. Recent projects [6, 13, 14, 15] show that MPC can reduce the risk of leaks for analytics on shared data under stronger security guarantees. However, MPC queries are often impractically slow, requiring orders of magnitude more computation and communication than plain-text or TEE-based query execution.

Adding security measures is a balancing act in the enterprise. Conventional wisdom dictates not to compromise on security between distrusting parties at all – no matter the performance impact. In the context of in-house analytics at large enterprises, however, even if only parts of a query are run with improved security, there is already a benefit for the enterprise [6]. Adding protection through the use of TEEs and MPC to the existing DBMS-level ones is useful if performance does not plummet, and future databases should be able to decide, given a performance target, what level of security can be actually fulfilled.

Analytics on shared data need security-aware query planning. We are working on a platform that modularizes secure query execution and allows for different strategies for trading off performance and security at the operator and query level. One point in the trade-off space is protecting computation using TEEs: we are exploring how to run OLAP queries in TEEs without performance overhead [8]. Another solution is using MPC and we are investigating how to precisely control information leakage about data passing between operators in exchange for faster MPC query execution. In the future, the query planner will need to be able to combine local and distributed operators executing in plain-text, in TEEs, using MPC, etc., and under different adversarial models. For completeness, in addition to the systems-level challenges, it will be also necessary to define security levels that are tailored to DBMS use-cases.

Case study: Trading off intermediate result size protection for better performance. As a concrete example of trading off security for performance, consider how intermediate results are passed between operators in an MPC query. The execution of MPC operators is oblivious to the content of their input: an oblivious filter, for instance, produces an output equal in size to its input but with a secret column indicating which row is actually selected. Similarly, an oblivious join has an output size equal to the Cartesian product of its inputs. This results in data sizes snowballing as the query execution proceeds, especially for analytical queries with many joins [6, 14, 15, 16]. Related work explores the relaxation of intermediate result size protection in different ways, e.g., adding non-deterministic noise to the true intermediate result size [14] or entirely foregoing adding noise to it [6]. One common decision, however, is to combine the implementation of the intermediate result size protection with the actual operator logic.

In Reflex [17] we decouple the protection mechanism from the operator logic, achieving flexibility while retaining execution efficiency thanks to a highly parallel implementation. The benefit of implementing intermediate result size protection as a separate step after each oblivious operator is that we can define custom strategies for hiding the size of the intermediate results and, through this, offer different security/performance tradeoffs. These strategies could be based on related work, using, e.g., differentially private noise [14], or entirely new ones. Reflex approaches secure shared analytics differently from most related work: instead of prescribing a specific set of security guarantees, we build the mechanisms necessary for the query planner to pick the adequate protections for each query based on performance and security/privacy criteria.

REFERENCES

[1] D. W. Archer, D. Bogdanov, Y. Lindell, L. Kamm, K. Nielsen, J. I. Pagter, N. P. Smart, and R. N. Wright, "From keys to databases—real-world applications of secure multi-party computation," *The Computer Journal*, vol. 61, no. 12, pp. 1749–1771, 2018.

[2] D. Bogdanov, L. Kamm, B. Kubo, R. Rebane, V. Sokk, and R. Talviste, "Students and taxes: a privacy-preserving study using secure computation," *Proceedings on Privacy Enhancing Technologies*, 2016.

[3] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, 2017.

[4] A. Rajan, L. Qin, D. W. Archer, D. Boneh, T. Lepoint, and M. Varia, "Callisto: A cryptographic approach to detecting serial perpetrators of sexual misconduct," in *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, COMPASS '18, (New York, NY, USA), Association for Computing Machinery, 2018.

[5] N. Crooks, "Efficient data sharing across trust domains," *ACM SIGMOD Record*, vol. 52, no. 2, pp. 36–37, 2023.

[6] W. Fang, S. Cao, G. Hua, J. Ma, Y. Yu, Q. Huang, J. Feng, J. Tan, X. Zan, P. Duan, Y. Yang, L. Wang, K. Zhang, and L. Wang, "Secretflow-scql: A secure collaborative query platform," *Proc. VLDB Endow.*, vol. 17, p. 3987–4000, Nov. 2024.

[7] S. Becker, C. Bösch, B. Hettwer, T. Hoeren, M. Rombach, S. Trieflinger, and H. Yalame, "Multi-party computation in corporate data processing: Legal and technical insights." Cryptology ePrint Archive, Paper 2025/463, 2025.

[8] A. Lutsch, M. El-Hindi, M. Heinrich, D. Ritter, Z. István, and C. Binnig, "Benchmarking analytical query processing in intel sgxv2," in *Proceedings 28th International Conference on Extending Database Technology, EDBT 2025, Barcelona, Spain, March 25-28, 2025* (A. Simitsis, B. Kemme, A. Queralt, O. Romero, and P. Jovanovic, eds.), pp. 516–528, OpenProceedings.org, 2025.

[9] P. Antonopoulos, A. Arasu, K. D. Singh, K. Eguro, N. Gupta, R. Jain, R. Kaushik, H. Kodavalla, D. Kossmann, N. Ogg, *et al.*, "Azure sql database always encrypted," in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pp. 1511–1525, 2020.

[10] S. Bian, Z. Zhang, H. Pan, R. Mao, Z. Zhao, Y. Jin, and Z. Guan, "He3db: An efficient and elastic encrypted database via arithmetic-and-logic fully homomorphic encryption," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2930–2944, 2023.

[11] R. A. Popa, C. M. Redfield, N. Zeldovich, and H. Balakrishnan, "Cryptdb: Protecting confidentiality with encrypted query processing," in *Proceedings of the twenty-third ACM symposium on operating systems principles*, pp. 85–100, 2011.

[12] R. Cramer, I. B. Damgård, *et al.*, *Secure multiparty computation*. Cambridge University Press, 2015.

[13] J. Liagouris, V. Kalavri, M. Faisal, and M. Varia, "SECRECY: Secure collaborative analytics in untrusted clouds," in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pp. 1031–1056, 2023.

[14] J. Bater, X. He, W. Ehrich, A. Machanavajjhala, and J. Rogers, "Shrinkwrap: efficient sql query processing in differentially private data federations," *Proceedings of the VLDB Endowment*, vol. 12, no. 3, 2018.

[15] J. Bater, Y. Park, X. He, X. Wang, and J. Rogers, "Saqe: practical privacy-preserving approximate query processing for data federations," *Proceedings of the VLDB Endowment*, vol. 13, no. 12, pp. 2691–2705, 2020.

[16] Y. Zhang, J. Bater, K. Nayak, and A. Machanavajjhala, "Longshot: Indexing growing databases using mpc and differential privacy," *Proceedings of the VLDB Endowment*, vol. 16, no. 8, pp. 2005–2018, 2023.

[17] L. Gu, S. Zeitouni, C. Binnig, and Z. István, "Reflex: Speeding up smpc query execution through efficient and flexible intermediate result size trimming." `https://arxiv.org/abs/2503.20932`, 2025.