Towards Building Practical Secure Multi-Party Databases

Yuzhe Tang
Syracuse University, Syracuse, NY 13244
ytang100@syr.edu

Wenqing Zhuang
Syracuse University, Syracuse, NY 13244
wzhuang@syr.edu

Abstract. This work aims at building secure multi-party database systems, for emerging federation networks in healthcare, finance, and other marketplaces. The technical challenges come from the interface gap between existing distributed query processing (DQP) and multi-party computation software. We propose compositional MPC for modular DQP and MPC-aware query optimizations.

An information network is an emerging federated big-data platform where autonomous data owners store, exchange and process their data. Information networks have application domains ranging from health-care (e.g. Health-care Information Exchange Networks or HIE [2, 3]), finance technology (e.g. block-chain), to enterprise computing (e.g. Google's global database Spanner [12]).

On the one hand, security and privacy is a major concern in an information network. Data in the information network is typically privacy sensitive (e.g. EMR to patient) and the exchange of it (e.g. for distributed query processing) has to be privacy preserving for the compliance with HiPAA [1] alike data-protection laws. On the other hand, existing secure-computation techniques, notably Multi-Party Computation protocols (MPC), are prohibitively slow and impractical, despite the extensive MPC-optimization researches proposed recently [19, 9, 21, 17, 11, 7, 24, 28, 6, 4, 10, 22, 25].

We tackle the problem of *efficient and privacy-preserving processing of distributed queries in federated databases*. The proposed system to build is a multi-party query processing framework including query compilation/optimization and query execution. The optimized execution plan is expected to minimize the invocations of expensive MPCs. The query executor executes a given query plan among multiple parties against private data, and produces the (declassified) query result as output.

Distinction from existing research There has been a large body of researches on secure distributed databases. Solutions in the domain of distributed privacy-preserving data mining [26, 18, 9, 13], take an ad-hoc, domain-specific approach for security which renders the system built unable to extend (meaning the system has to be rebuilt from scratch to support a new query) and are inapplicable to supporting the full set of SQL queries. Our work takes a different system-building philosophy: Instead of composing multiple less expressive primitives, we base the entire database systems on one expressive primitive – more-than-three party computation (MPC) [11]. To strike a balance in the expressiveness-performance trade-off, our work tailors the *use of MPC* to specific query computations with minimal invocations.

One of relevant researches is the recent work [5] on building two-party federated database systems [5]; the work leverages the ObliVM compiler [20] and a runtime based on Yao's Garbled Circuit [27]. Our work considers more-than-three-party networks and addresses the scalability in data and network sizes. Note the problem of secure multi-party databases is different from secure outsourced databases (e.g. CryptDB [23]); the former considers multiple untrusted data owners performing joint computations, while the latter considers one single owner outsourcing the computation to an untrusted party. Other MPC optimization work [22, 25] optimizes the sizes of MPC circuit (or other intermediate representations) by leveraging certain semantics (e.g. hardware-circuit synthesis and Map-reduce programs); these optimizations are not specific to database applications.

The key challenge of building secure multi-party databases stems from the gap between secure multi-party computations (as cryptography protocols) and distributed query processing (as a part of the database systems) – These two areas are in different researches communities and their approaches have semantic gaps. Specifically, a distributed database optimizes the performance by localizing the distributed query processing to as few parties as possible, while the multi-party computation intentionally distributes "shares" of a secret to as many parties as possible [27, 15] for security reasons. In addition, database systems usually execute queries in a data-dependent fashion, while secure multi-party computations deliberately decouple the secret data from the execution flow to achieve "obliviousness" and resilience against side-channel attacks [16, 20, 8, 14].

These semantic gaps lead to challenges in designing a multi-party database system with both efficiency and strong security. We aim at developing the following techniques in bridging the gaps.

For MPC query execution, we propose *circuit localization and composition*. Most existing MPC protocols (for more than three parties) are based on a monolithic circuit that gets broadcast to and evaluated in the entire network. To bridge the gap between a monolithic circuit and module-based DQP, we propose a compositional MPC protocol where the MPC circuit for a target query can be decomposed into a series of smaller circuits, each for one query operator. Connecting different small circuits is a new secret-resharing and reshuffling process (based on the Boolean-value sharing). This modular execution model can help *localize* the expensive MPC computation to fewer but more relevant parties, improving efficiency and scalability.

For MPC query optimization, we consider a centralized optimizer which looks at non-sensitive metadata and produces the execution plan with the expected best runtime performance. We consider the non-sensitive metadata model because the metadata used in regular database optimizations (e.g. query, data location and MPC cost model) are public information and non-sensitive.

CIDR '17

1. REFERENCES

[1] Hipaa on data disclosure: http://www.gpo.gov/fdsys/pkg/CFR-2010-title45-vol1/pdf/CFR-2010-title45-vol1-sec164-502.pdf.

[2] Nwhin: http://www.hhs.gov/healthit/healthnetwork.

[3] Shin-ny: http://www.health.ny.gov/technology/projects/.

[4] G. Asharov, Y. Lindell, T. Schneider, and M. Zohner. More efficient oblivious transfer and extensions for faster secure computation. In *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, pages 535–548, 2013.

[5] J. Bater, G. Elliott, C. Eggen, S. Goel, A. N. Kho, and J. Duggan. SMCQL: secure query processing for private data networks. *CoRR*, abs/1606.06808, 2016.

[6] M. Bellare, V. T. Hoang, S. Keelveedhi, and P. Rogaway. Efficient garbling from a fixed-key blockcipher. In *2013 IEEE Symposium on Security and Privacy, SP 2013, Berkeley, CA, USA, May 19-22, 2013*, pages 478–492, 2013.

[7] A. Ben-David, N. Nisan, and B. Pinkas. Fairplaymp: a system for secure multi-party computation. In *ACM Conference on Computer and Communications Security*, pages 257–266, 2008.

[8] E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza. Succinct non-interactive zero knowledge for a von neumann architecture. In *Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014.*, pages 781–796, 2014.

[9] D. Bogdanov, S. Laur, and J. Willemson. Sharemind: A framework for fast privacy-preserving computations. In *Computer Security - ESORICS 2008, 13th European Symposium on Research in Computer Security, Málaga, Spain, October 6-8, 2008. Proceedings*, pages 192–206, 2008.

[10] M. Burkhart, M. Strasser, D. Many, and X. A. Dimitropoulos. SEPIA: privacy-preserving aggregation of multi-domain network events and statistics. In *19th USENIX Security Symposium, Washington, DC, USA, August 11-13, 2010, Proceedings*, pages 223–240, 2010.

[11] S. G. Choi, K. Hwang, J. Katz, T. Malkin, and D. Rubenstein. Secure multi-party computation of boolean circuits with applications to privacy in on-line marketplaces. In *Topics in Cryptology - CT-RSA 2012 - The Cryptographers' Track at the RSA Conference 2012*, pages 416–432, 2012.

[12] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, W. C. Hsieh, S. Kanthak, E. Kogan, H. Li, A. Lloyd, S. Melnik, D. Mwaura, D. Nagle, S. Quinlan, R. Rao, L. Rolig, Y. Saito, M. Szymaniak, C. Taylor, R. Wang, and D. Woodford. Spanner: Google's globally distributed database. *ACM Trans. Comput. Syst.*, 31(3):8, 2013.

[13] W. Du and M. J. Atallah. Protocols for secure remote database access with approximate matching. In *E-Commerce Security and Privacy*, pages 87–111. 2001.

[14] D. G. E. T. E. Ben-Sasson, A. Chiesa and M. Virza. Tinyram architecture specification, v0.991. http://www.sciprlab.org/system/files/tinyram-spec-0.991.pdf, 2013.

[15] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 218–229, 1987.

[16] O. Goldreich and R. Ostrovsky. Software protection and simulation on oblivious rams. *J. ACM*, 43(3):431–473, 1996.

[17] Y. Huang, D. Evans, J. Katz, and L. Malka. Faster secure two-party computation using garbled circuits. In *USENIX Security Symposium*, 2011.

[18] M. Kantarcioglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE Trans. Knowl. Data Eng.*, 16(9):1026–1037, 2004.

[19] B. Kreuter, A. Shelat, B. Mood, and K. R. B. Butler. PCF: A portable circuit format for scalable two-party secure computation. In *Proceedings of the 22th USENIX Security Symposium*, pages 321–336, 2013.

[20] C. Liu, X. S. Wang, K. Nayak, Y. Huang, and E. Shi. Oblivm: A programming framework for secure computation. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, pages 359–376, 2015.

[21] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella. Fairplay - secure two-party computation system. In *USENIX Security Symposium*, pages 287–302, 2004.

[22] K. Nayak, X. S. Wang, S. Ioannidis, U. Weinsberg, N. Taft, and E. Shi. Graphsc: Parallel secure computation made easy. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, pages 377–394, 2015.

[23] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan. Cryptdb: protecting confidentiality with encrypted query processing. In *Proceedings of the 23rd ACM Symposium on Operating Systems Principles 2011, SOSP 2011, Cascais, Portugal, October 23-26, 2011*, pages 85–100, 2011.

[24] A. Rastogi, M. A. Hammer, and M. Hicks. Wysteria: A programming language for generic, mixed-mode multiparty computations. In *2014 IEEE Symposium on Security and Privacy, SP 2014, Berkeley, CA, USA, May 18-21, 2014*, pages 655–670, 2014.

[25] E. M. Songhori, S. U. Hussain, A. Sadeghi, T. Schneider, and F. Koushanfar. Tinygarble: Highly compressed and scalable sequential garbled circuits. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, pages 411–428, 2015.

[26] J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, pages 639–644, 2002.

[27] A. C. Yao. How to generate and exchange secrets (extended abstract). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 162–167, 1986.

[28] Y. Zhang, A. Steele, and M. Blanton. PICCO: a general-purpose compiler for private distributed computation. In *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, pages 813–826, 2013.