Private Data Profiling

madhav@u.northwestern.edu

1 INTRODUCTION

With secure computation, mutually distrustful parties are able to compute arbitrary functions on their joint datasets, without revealing the underlying raw data. The parties share their encrypted datasets, and the only information revealed is the output of the functions. Recent work has built select-project-join query engines that can run a broad workload of SQL queries efficiently [1, 2, 15].

Existing work focuses on query processing and query release. This proposal considers data management problems that happen at the beginning of the data lifecycle: data profiling, data quality, data cleaning, and data integration, with two mutually distrustful parties. This setting creates unique set of challenges that are not addressed in previous non-privacy aware work. First, privacy and security goals are met through cryptographic primitives, changing the underlying computational model. This work uses two-party computation techniques [16] to provide privacy. In order for these systems to be efficient, new data structures and algorithms that cooperate with these techniques must be designed. Second, integrity and privacy are challenges not found in the cannonical settings. Release of data profiling metadata, or applications of data cleaning rules can lead to unknown privacy loss. This work aims to build an end-to-end system with provable privacy guarantees. This document considers data profiling, detailing the private functional dependency discovery problem.

2 PRIVATE DATA PROFILING

To start, an example data quality scenario that two parties may face: The CollegeBoard has data containing AP Computer Science scores, zip codes, school districts, student names, and teacher names. Code.org has data that details which school districts that received training, student exercise information, and a list of teachers who received training. Neither the CollegeBoard nor Code.org can release the student data and district data. However, they would like to answer the question: "Do the Code.org interventions lead to better learning outcomes, as measured by AP test scores?".

In our hypothetical scenario, the CollegeBoard and Code.org reach an agreement to release specific tables to each other (in plaintext). Upon receipt of the joint dataset, both parties realize that the quality of the joint dataset is poor, and that even with a shared join key across tables, their primary question cannot be answered. The parties need a tool to explore profiling properties, metadata, and quality information without revealing their inputs in plaintext. These profiling tools should be faster than running the encrypted queries on the datasets, and there should be opportunity to clean and modify the datasets. Application of the tools should result in bounded and well understood privacy loss for both parties. Our goal is to build upon and add to the existing work on private data cleaning [10], private record linkage [7], synthetic data generation [6], and functional dependency discovery [5] . As an example, we consider the secure two-party functional dependency discovery problem.

2.1 Secure Two-Party Functional Dependency Discovery

Functional dependencies (FDs) appear to be a first step in solving the more challenging problems in data profiling, quality, and cleaning. Functional dependency discovery is valuable in it's own right as well - FDs are a useful for data cleaning/repair [14], query optimization [9, 12], and data integration. Previous work on Secure Functional Dependency Discovery [5] focused on the multiparty setting. Our protocol can generalize to the multiparty setting, while the protocol by Ge et. al. [5] is not secure when there are only two parties. We give a problem statement with a prototype approach overview.

Definition 2.1 (Secure Two-Party Horizontally Partitioned FD Discovery). Two parties $\mathcal{P} = \{P_{ALICE}, P_{BOB}\}$, each with private datasets $\mathcal{D} = \{D_a, D_b\}$, and with shared schema $\mathbb{S}$. The goal is to discover a set of functional dependencies $\mathcal{F} = \{f_i\}$ that hold over the union of the datasets $D_a \bigcup D_b$. Alice and Bob are only willing to share their data to compute functional dependencies but not for any other purposes. Alice and Bob will remain honest-but-curious: they will faithfully follow protocol, but try and gain private information through side channels. The parties allow for the functional dependencies to be released through an agreed upon data release mechanism once the protocol is complete.

Prototype Approach:

(1) *Challenge: Naive solutions implemented with general purpose secure two-party computation require expensive oblivious operations.* Prototype Approach: Design specialized protocols and data structures to perform the required tasks for FD discovery. Current benchmarks of our naive protocol implementation already demonstrate an order of magnitude performance improvement. The private set cardinality estimation, and private sketching problems have similar structure to the FDD problem. Recent work [3, 8] on these problems construct specialized data structures to improve performance and utility. We believe new techniques inspired by this work will unlock significant performance gains for secure FDD.

(2) *Challenge: Releasing exact functional dependencies after discovery potentially reveals private information. For example, it can reveal the existence of tuple combinations that act as counterexamples for an FD.* Prototype Approach: Provide a differentially private *approximate* functional dependency definition, as well as an efficient differentially private release protocol. Our simple mechanism applies the Laplace mechanism in secure computation after an approximate functional dependency is discovered. This solution likely does not provide an efficient mechanism, as the Laplace noise is applied once per FD. We note that the secure approximate functional dependency discovery problem requires a

unique protocol design to that of exact functional dependency discovery. To the best of our knowledge, the secure approximate FDD has not been studied before.

2.2 Beyond Functional Dependencies

Functional dependency discovery are a warmup for the broad set of data integration, profiling, and cleaning problems that need to be reconsidered in the private setting. One goal of the FD discovery work is to define abstractions and data structures that are applicable beyond the FD use-case. One can imagine applying these techniques to denial constraints [4, 11], and through that being able to construct private data repair/cleaning protocols [13]. The data structures can also be extended to the streaming setting, where two parties stream in their data into a joint, encrypted dataset - while still adhering to agreed upon constraints/quality requirements upon ingest.

With the differentially private definitions of functional dependencies as a base, we can define private data quality metrics for two parties that are trying to securely share data. Consider the example of the CollegeBoard and Code.org: the CollegeBoard can measure multiple data quality metrics on their dataset combined with Code.org's dataset, with bounded privacy loss, and without having to share the data in plaintext. With a better understanding of their combined data quality, the CollegeBoard and Code.org can decide if combining their datasets together offers the desired utility.

REFERENCES

[1] Johes Bater, Gregory Elliott, Craig Eggen, Satyender Goel, Abel Kho, and Jennie Rogers. 2016. SMCQL: Secure Querying for Federated Databases. *VLDB* 10, 6 (2016), 673–684. https://doi.org/10.14778/3055330.3055334

[2] Johes Bater, Xi He, William Ehrich, Ashwin Machanavajjhala, and Jennie Rogers. 2019. Shrinkwrap: Differentially-Private Query Processing in Private Data Federations. *VLDB* 12, 3 (2019), 307–320.

[3] Seung Geol Choi, Dana Dachman-Soled, Mukul Kulkarni, and Arkady Yerukhimovich. 2020. Differentially-Private Multi-Party Sketching for Large-Scale Statistics. Cryptology ePrint Archive, Report 2020/029. https://ia.cr/2020/029.

[4] Xu Chu, Ihab F. Ilyas, and Paolo Papotti. 2013. Discovering Denial Constraints. *Proc. VLDB Endow.* 6, 13 (aug 2013), 1498–1509. https://doi.org/10.14778/2536258.2536262

[5] Chang Ge, Ihab F. Ilyas, and Florian Kerschbaum. 2019. Secure Multi-Party Functional Dependency Discovery. *Proc. VLDB Endow.* 13, 2 (Oct. 2019), 184–196. https://doi.org/10.14778/3364324.3364332

[6] Chang Ge, Shubhankar Mohapatra, Xi He, and Ihab F. Ilyas. 2021. Kamino: Constraint-Aware Differentially Private Data Synthesis. *Proc. VLDB Endow.* 14, 10 (jun 2021), 1886–1899. https://doi.org/10.14778/3467861.3467876

[7] Xi He, Ashwin Machanavajjhala, Cheryl Flynn, and Divesh Srivastava. 2017. Composing Differential Privacy and Secure Computation: A case study on scaling private record linkage. In *CCS*. ACM, 1389–1406.

[8] Changhui Hu, Jin Li, Zheli Liu, Xiaojie Guo, Yu Wei, Xuan Guang, Grigorios Loukides, and Changyu Dong. 2020. How to Make Private Distributed Cardinality Estimation Practical, and Get Differential Privacy for Free. Cryptology ePrint Archive, Report 2020/1576. https://ia.cr/2020/1576.

[9] Jan Kossmann and Felix Naumann. 2021. Data dependencies for query optimization: a survey. *The VLDB Journal* (06 2021). https://doi.org/10.1007/s00778-021-00676-3

[10] Sanjay Krishnan, Jiannan Wang, Michael J. Franklin, Ken Goldberg, and Tim Kraska. 2016. PrivateClean: Data Cleaning and Differential Privacy. In *Proceedings of the 2016 International Conference on Management of Data* (San Francisco, California, USA) *(SIGMOD '16)*. Association for Computing Machinery, New York, NY, USA, 937–951. https://doi.org/10.1145/2882903.2915248

[11] Ester Livshits, Alireza Heidari, Ihab F. Ilyas, and Benny Kimelfeld. 2020. Approximate Denial Constraints. *Proc. VLDB Endow.* 13, 10 (jun 2020), 1682–1695. https://doi.org/10.14778/3401960.3401966

[12] Thorsten Papenbrock, Jens Ehrlich, Jannik Marten, Tommy Neubert, Jan-Peer Rudolph, Martin Schönberg, Jakob Zwiener, and Felix Naumann. 2015. Functional Dependency Discovery: An Experimental Evaluation of Seven Algorithms. *Proc. VLDB Endow.* 8, 10 (June 2015), 1082–1093. https://doi.org/10.14778/2794367.2794377

[13] Theodoros Rekatsinas, Xu Chu, Ihab F. Ilyas, and Christopher Ré. 2017. HoloClean: Holistic Data Repairs with Probabilistic Inference. *Proc. VLDB Endow.* 10, 11 (aug 2017), 1190–1201. https://doi.org/10.14778/3137628.3137631

[14] El Kindi Rezig, Mourad Ouzzani, Walid G. Aref, Ahmed K. Elmagarmid, Ahmed R. Mahmood, and Michael Stonebraker. 2021. Horizon: Scalable Dependency-Driven Data Cleaning. *Proc. VLDB Endow.* 14, 11 (jul 2021), 2546–2554. https://doi.org/10.14778/3476249.3476301

[15] Nikolaj Volgushev, Malte Schwarzkopf, Ben Getchell, Andrei Lapets, Mayank Varia, and Azer Bestavros. 2018. Conclave Worflow Manager for MPC. https://github.com/multiparty/conclave

[16] Andrew Chi-Chih Yao. 1986. How to Generate and Exchange Secrets. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science (SFCS '86)*. IEEE Computer Society, USA, 162–167. https://doi.org/10.1109/SFCS.1986.25