Secure MPC for Analytics as a Web Application

Andrei Lapets, Nikolaj Volgushev, Azer Bestavros, Frederick Jansen, Mayank Varia
Email: {lapets, nikolaj, best, fjansen, varia}@bu.edu
CS Dept., Boston University, 111 Cummington Mall, Boston, MA USA 02215

I. Introduction

Companies, government agencies, and other organizations have been analyzing data pertaining to their internal operations with great effect, such as in evaluating performance or improving efficiency. While each organization's own data is valuable internally, aggregate data from multiple organizations can have value to the organizations themselves, policymakers, and society. Unfortunately, an organization's data is often proprietary and confidential, and its release may be potentially deleterious to the organization's interests. Secure multi-party computation (MPC) resolves this tension: aggregate data may be computed while protecting each contributor's confidentiality. Theoretical constructs have been known for decades [1]–[3] and recent efforts aim to deliver them to end-users [4]–[6].

II. Scenario and Requirements

The Boston Women's Workforce Council (BWWC) initiated a study of gender and ethnicity wage gaps among employers within the Greater Boston Area; compensation data must to be collected from privately held companies in order to calculate an aggregate statistic (sum) over the data. Each company submits employee earnings aggregated by gender and job category. BWWC may view the aggregate totals across *all* companies, but individual company numbers must remain private.

We implemented and deployed an MPC protocol as a web-based service to compute the statistic without requiring the companies to trust BWWC or Boston University (BU) with sensitive data. The user interface provides a familiar spreadsheet that can be filled with data manually or via copy-paste. We successfully deployed this service twice (in 2015 and 2016) to analyze compensation data from a collection of 40–70 employer organizations [7].

We consider three roles in the deployed protocol: (1) an unknown quantity of *contributors* who contribute private data for the calculation; (2) an automated, publicly-accessible *service provider* that sees only encrypted data and connects all other participants without requiring them to maintain servers (or even to be online simultaneously); and (3) one or more *analyzers* who receive the output of the analytic. For an outline of the protocol we refer the reader to the Appendix. Several *security* and *usability* considerations drove protocol design and implementation.

Security: We rely on MPC with passive (semi-honest) security and without collusion [8]. This suffices in our scenario because the service provider and analyzer lack incentives to falsify the results or to learn private inputs: completing the study successfully is directly beneficial to BWWC (as the study initiator) and to BU (as an institution reliant upon a reputation of integrity). Additionally, obtaining any private contributor data (by colluding or actively deviating from the protocol) creates a liability risk for the service provider and analyzer. The semi-honest model is natural in this case: service providers are protected from the legal risks of processing sensitive data if the protocol is followed.

Usability: A secure MPC protocol only has value if multiple parties trust it and use it. The pay equity scenario involves individuals with a wide range of technical backgrounds utilizing computing resources that are outside of our control and governed by a variety of organizational constraints. Thus, our protocol and web service must satisfy many usability goals: comprehensibility (to drive adoption); transparency (open-source code); easy deployability (no specialized software, hardware, synchronization, or continuous network access); idempotent resubmission; input validation in the client interface; and others [9].

Off-the-shelf Tools: The past few years have seen several successful deployments of MPC [10]–[12] and a number of software frameworks are available [4]–[6], though they fall short of meeting our usability requirements (*e.g.*, non-expert comprehensibility and easy deployability). Our full technical report [9] provides a thorough evaluation of existing frameworks (*e.g.*, VIFF and Sharemind) and their limitations in this context, including assembly of exploratory prototypes using such existing frameworks.

III. Deployment and Future Vision

Practical deployment difficulties included browser and OS incompatibilities, human errors and associated support activities, and scheduling of the data collection; performance was not an issue given the procotol and scale of data [9]. The simplicity of the protocol and implementation helped decision makers feel confident that they adequately understood their operation, security guarantees, and risks.

Informed by our experience, we envision an MPC-as-a-service platform that provides powerful computing and networking capabilities to "thin client" users (having nothing more than a web browser) such that trust is inversely proportional to computing power. The security community can support use cases such as ours by combining MPC and cloud computing in a unique way that allows the most powerful computing entity to be the *least trusted*.

Appendix

The protocol developed for this application is a variant of a technique that allows multiple parties to securely compute a sum of their private inputs [13], though the naïve secure sum protocol could not be deployed as-is:

- participants must pass data along in sequence, requiring a sophisticated software infrastructure involving multiple client/server applications communicating with one another and maintaining state;
- participants must run the application for the duration of the computation (spanning hours or days); and
- if one participant makes an error and wishes to resubmit, the entire protocol would need to be restarted because updates are not idempotent.

These requirements are avoided in the adjusted protocol.

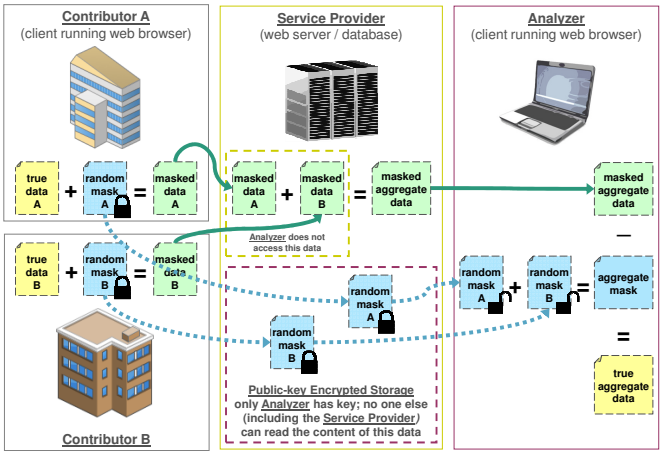

Fig. 1. Diagram of protocol deployment for two contributors used to explain the protocol to potential participants.

Let G be an appropriate additive group such as $\mathbb{Z}/2^{64}\mathbb{Z}$ and distinguish each contributor using an index $i \in \{1, ..., n\}$. A single *session* (execution) proceeds as follows:

1. the analyzer initiates the process by generating a secret and public RSA key pair (s, p) sending p to the service provider and all the contributors;
2. each of the n contributors possesses a secret *data* value $d_i \in G$ and does the following at least once:
   a. generates a secret *random mask* $m_i \in G$ and calculates the *masked data* $r_i = d_i + m_i$, and
   b. sends r_i unencrypted to the service provider and uses p to send an encrypted mask $c_i = \mathrm{Enc}_p(m_i)$;
3. the service provider computes the aggregate of the masked data $R = \sum_{i=1}^{n} r_i$;
4. the analyzer then retrieves R and all the $c_1, \ldots, c_n$ from the service provider, computes $m_i = \mathrm{Dec}_s(c_i)$ for all i, computes $M = \sum_{i=1}^{n} m_i$, and obtains the final result $R - M = \sum_{i=1}^{n} d_i$.

The service provider never sees the masks because they are encrypted, and the analyzer never sees the individual masked data values unless it colludes with the service provider. Our protocol guarantees that any malicious outsider that can observe and store all communications between all participants will gain no information beyond the aggregate being computed. We exploit this when deploying: the server housing the data can be commodity hardware purchased from any third-party provider.

This research was partially supported by the NSF under Award #1414119 and Award #1430145.

References

[1] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, ser. SFCS '82. Washington, DC, USA: IEEE Computer Society, 1982, pp. 160–164. [Online]. Available: http://dx.doi.org/10.1109/SFCS.1982.88

[2] 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*. ACM, 1987, pp. 218–229.

[3] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract)," in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*. ACM, 1988, pp. 1–10.

[4] "VIFF, the Virtual Ideal Functionality Framework," http://viff.dk/, [Accessed: August 15, 2015].

[5] D. Bogdanov, S. Laur, and J. Willemson, "Sharemind: A Framework for Fast Privacy-Preserving Computations," in *Proceedings of the 13th European Symposium on Research in Computer Security - ESORICS'08*, ser. Lecture Notes in Computer Science, S. Jajodia and J. Lopez, Eds., vol. 5283. Springer Berlin / Heidelberg, 2008, pp. 192–206.

[6] A. Rastogi, M. A. Hammer, and M. Hicks, "Wysteria: A programming language for generic, mixed-mode multiparty computations," in *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, ser. SP '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 655–670. [Online]. Available: http://dx.doi.org/10.1109/SP.2014.48

[7] R. Barlow, "Computational Thinking Breaks a Logjam," http://www.bu.edu/today/2015/computational-thinking-breaks-a-logjam/, [Accessed: August 15, 2015].

[8] O. Goldreich, *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press, 2004.

[9] A. Lapets, N. Volgushev, A. Bestavros, F. Jansen, and M. Varia, "Secure Multi-Party Computation for Analytics Deployed as a Lightweight Web Application," CS Dept., Boston University, Tech. Rep. BUCS-TR-2016-008, July 2016. [Online]. Available: http://www.cs.bu.edu/techreports/pdf/2016-008-mpc-lightweight-web-app.pdf

[10] P. Bogetoft, D. L. Christensen, I. Damgård, M. Geisler, T. Jakobsen, M. Krøigaard, J. D. Nielsen, J. B. Nielsen, K. Nielsen, J. Pagter, M. Schwartzbach, and T. Toft, "Financial cryptography and data security," R. Dingledine and P. Golle, Eds. Berlin, Heidelberg: Springer-Verlag, 2009, ch. Secure Multiparty Computation Goes Live, pp. 325–343. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03549-4_20

[11] I. Damgård, K. Damgård, K. Nielsen, P. S. Nordholt, and T. Toft, "Confidential benchmarking based on multiparty computation," Cryptology ePrint Archive, Report 2015/1006, 2015, http://eprint.iacr.org/.

[12] D. Bogdanov, L. Kamm, B. Kubo, R. Rebane, V. Sokk, and R. Talviste, "Students and Taxes: a Privacy-Preserving Study Using Secure Computation," *PoPETs*, vol. 2016, no. 3, p. 117Ű135, 2016. [Online]. Available: http://www.degruyter.com/view/j/popets.2016.2016.issue-3/popets-2015-0019/popets-2016-0019.xml

[13] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu, "Tools for privacy preserving distributed data mining," *SIGKDD Explor. Newsl.*, vol. 4, no. 2, pp. 28–34, Dec. 2002. [Online]. Available: http://doi.acm.org/10.1145/772862.772867