A Private Framework for Distributed Computation

Edward Tremel*, Ken Birman*, Robert Kleinberg*, and Márk Jelasity†

There is a growing class of distributed systems applications in which data stored on client platforms must be aggregated or analyzed without revealing private information to the operator. Systems such as the smart power grid, control systems for energy-efficient buildings, and traffic analysis in large cities all depend on the analysis of data supplied by measurement devices, yet the clients being tracked are unwilling to reveal such measurement data directly to the system owner, who might be "curious" about private client information. These systems thus may elicit public opposition despite their useful features because of a perceived privacy risk.

There are ways to upload sensitive data to an aggregator without compromising privacy, but existing options have limitations. One possibility is to keep the data encrypted with keys known only to the clients, but this requires expensive homomorphic encryption if the aggregator is to compute directly on it. Another is to employ a mechanism to de-correlate client identifiers from their data, as Chen et al. do in [4], but this imposes restrictions on the kind of aggregation that can be done. Instead, it would be beneficial to execute needed computation directly on the client platforms, so that the system operator or analyst only sees aggregate results. This approach would provide a better alternative to central aggregation provided it is privacy-preserving, robust, and efficient.

A data aggregation system based on client-side computation suggests a purely peer-to-peer architecture [10], which many systems have used to avoid centralized control [8, 7, 11]. However, peer-to-peer systems have problems of their own, even if we set privacy concerns aside. By eschewing centralization entirely, they can no longer take advantage of the powerful management tools developed for today's cloud computing model. In traditional peer-to-peer systems, clients are isolated network hosts rather than devices within a single administrative domain, and often have difficulty maintaining connections to each other through firewalls and address translation barriers. Determining the membership of a peer-to-peer network is a surprisingly difficult problem, since there is no one entity that knows the identities of all the clients, and changes in membership may not be detected and propagated in a timely fashion [1]. Without a centralized service to assign and manage node identities, Sybil attacks [6] are always possible, so a peer-to-peer system is extremely vulnerable to a few malicious peers becoming a majority of the apparent nodes in the system. Even choosing peers fairly becomes difficult, because peers usually do not store the entire membership list locally, and it is fairly easy for malicious peers to poison local membership views so that they will be preferred as neighbors by honest nodes [9, 3, 2].

Since neither completely centralized aggregation nor a completely peer-to-peer system is adequate for our purposes, we explore a new approach that combines the features of these two extremes. Although the idea of a communication system that combines some centralized control with a peer-to-peer overlay is not new, we are the first to use such a system to preserve privacy while computing on sensitive data. This combination is a sensible tradeoff for the kinds of systems we target, in which there is an owner or operator who can be trusted to provide basic services such as node identification and membership tracking but not to see non-aggregated raw client data. Essentially, we treat the system operator as an honest-but-curious adversary, who will keep the system running correctly but cannot be allowed to see more information than he or she needs to know.

In our paper [12], we introduce a method for constructing a communication overlay among the client nodes that can safely be used to perform aggregation and computation on private data. Although this overlay is set up and operated by the system owner, it provides minimal opportunity for the owner to learn any information about the data being aggregated other than the final result of the computation. When combined with differential privacy techniques, to protect the aggregation results themselves, it can be used to ensure that no query made to the system reveals the contribution of any particular node.

Our overlay network looks a bit like a gossip infrastructure [5], and can be used to run gossip-like protocols, with the key difference that the random peer selection of gossip is replaced with a completely deterministic function. Nodes are assigned virtual IDs that are either integers or finite field elements, and each node uses a function based on either modular arithmetic or finite fields to compute the order in which it should communicate with the other nodes. We construct this function to ensure that the network is optimally robust and efficient, converging in logarithmic time and tolerating message failures with minimal delay. Nodes use public-key cryptography to encrypt messages, ensuring that the the system operator cannot infer anything about the data being aggregated by observing network traffic. Even the communication pattern is completely predictable and hence reveals nothing. In this approach, malicious nodes cannot significantly deviate from correct behavior without being detected, so the network encourages the operator to behave correctly, and it even tolerates Byzantine failure by a small minority of clients. This ensures that important queries will not be corrupted or blocked by compromised devices, and that an adversary cannot compromise the privacy of client data by gaining control of a few devices in the system.

*Cornell University
†University of Szeged, Hungary

Acknowledgements

This work was supported, in part, by a grant from the NSF Smart Grids program.

References

[1] André Allavena, Alan Demers, and John E. Hopcroft. Correctness of a gossip based membership protocol. In *Proceedings of the Twenty-fourth Annual ACM Symposium on Principles of Distributed Computing*, PODC '05, page 292–301, New York, NY, USA, 2005. ACM.

[2] Emmanuelle Anceaume, Yann Busnel, and Bruno Sericola. Uniform node sampling service robust against collusions of malicious nodes. In *2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 1–12, June 2013.

[3] Edward Bortnikov, Maxim Gurevich, Idit Keidar, Gabriel Kliot, and Alexander Shraer. Brahms: Byzantine resilient random membership sampling. In *Proceedings of the Twenty-seventh ACM Symposium on Principles of Distributed Computing*, PODC '08, page 145–154, New York, NY, USA, 2008. ACM.

[4] Ruichuan Chen, Istemi Ekin Akkus, and Paul Francis. SplitX: High-performance private analytics. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, page 315–326, New York, NY, USA, 2013. ACM.

[5] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. Epidemic algorithms for replicated database maintenance. In *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing*, PODC '87, page 1–12, New York, NY, USA, 1987. ACM.

[6] John R. Douceur. The sybil attack. In Peter Druschel, Frans Kaashoek, and Antony Rowstron, editors, *Peer-to-Peer Systems*, number 2429 in Lecture Notes in Computer Science, pages 251–260. Springer Berlin Heidelberg, January 2002.

[7] P. Th. Eugster, R. Guerraoui, S. B. Handurukande, P. Kouznetsov, and A.-M. Kermarrec. Lightweight probabilistic broadcast. *ACM Trans. Comput. Syst.*, 21:341–374, November 2003.

[8] Márk Jelasity, Alberto Montresor, and Ozalp Babaoglu. T-man: Gossip-based fast overlay topology construction. *Computer Networks*, 53(13):2321–2339, August 2009.

[9] Gian Paolo Jesi, Alberto Montresor, and Maarten van Steen. Secure peer sampling. *Computer Networks*, 54(12):2086–2098, August 2010.

[10] Róbert Ormándi, István Hegedűs, and Márk Jelasity. Gossip learning with linear models on fully distributed data. *Concurrency and Computation: Practice and Experience*, 25(4):556–571, February 2013.

[11] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware 2001*, pages 329–350. Springer, 2001.

[12] Edward Tremel, Ken Birman, Robert Kleinberg, and Márk Jelasity. A private framework for distributed computation. August 2014. To appear. Preliminary version available upon request.