fully secure solution, could be security-aware leakage of data access patterns, accompanied by a formal analysis of how this information satisfies security and privacy requirements. It is plausible that in many scenarios it can be formally shown, e.g., using tools from programming languages among others, that revealing some information about the access pattern is acceptable.

Output leakage. The goal of MPC is to protect the privacy of inputs and intermediate results, but at the end of the protocol the output of the function is revealed. A separate research field has developed around the complementary problem where there is no need to protect the input data, but the output must be controlled to limit what an adversary can infer about the private data from the output. The dominant model for controlling output leakage is *differential privacy* (Dwork and Roth, 2014) which adds privacy-preserving noise to outputs before they are revealed. A few works have explored combining MPC with differential privacy to provide end-too-end privacy for computations with distributed data (Pettai and Laud, 2015; Kairouz *et al.*, 2015; He *et al.*, 2017), but this area is in its infancy and many challenging problems need to be solved before the impacts of different types of leakage are well understood.

Meaningful trust. When cryptographers analyze protocols, they assume each party has a way to executed their part of the protocol that they fully trust. In reality, protocols are executed by complex software and hardware, incorporating thousands of components provided by different vendors and programmers. The malicious secure protocols we discussed in this book provide a user with strong guarantees that if their own system executes its role in the protocol correctly, no matter what the other participants do the security guarantees are established. But, in order to fully trust an MPC application, a user also needs to fully trust that the system executing the protocol on its behalf, which is given full access to their private data in cleartext, will execute the protocol correctly. This includes issues like ensuring the implementation has no side channels that might leak the user's data, as well as knowing that it performs all protocol steps correctly and does not attempt to leak the user's private input.

Providing full confidence in a computing system is a longstanding and challenging problem that applies to all secure uses of computing, but is

particularly sharp in considering the security of MPC protocols. In many deployments, all participants end up running the same software, provided by a single trusted party, and without any auditing. In practice, this provides better security that just giving all the data to the software provider; but in theory, it is little different from just handing over all the plaintext data to the software (or hardware) provider.

One appealing aspect of MPC techniques is they can be used to alleviate the need to place any trust in a computing system—if private data is split into shares and computed on using two systems using MPC, the user no longer needs to worry about implementation bugs in either systems exposing the data since the MPC protocol ensures that, regardless of the type of bug in the system even including hardware side channels, there is no risk of data exposure since the private data is always protected by cryptography withing the MPC protocol and is not visible to the system at all.

Finding ways to meaningfully convey to end users how their data will be exposed, and what they are trusting in providing it, is a major challenge for future computing systems. Making progress on this is essential for enabling privacy-enhancing technologies like MPC to provide meaningful and understandable benefits to the vast majority of computing users, rather than just to sophisticated organizations with teams of cryptographers and program analysts.

We wish to conclude this book on an optimistic note. The awareness that personal data can be compromised in a data breach, or can be abused by companies whose interests do not align with those of their users, is increasing. New regulations, including the European Union's *General Data Protection Regulation*, are making holding personal data a liability risk for companies. MPC has emerged as a powerful and versatile primitive that can provide organizations and individuals with a range of options for designing privacy-preserving applications. Many challenges remain, but MPC (and secure computation broadly) is a young and vibrant field, with much opportunity to create, develop and apply.

Acknowledgements

The authors thank Jeanette Wing for instigating this project; our editors at Now Publishers, James Finlay and Mike Casey, for help and flexibility throughout the writing process; and Alet Heezemans for help with the final editing. We thank Patricia Thaine for particularly helpful comments and corrections.

Vladimir Kolesnikov was supported in part by Sandia National Laboratories, a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

Mike Rosulek was supported in part by the National Science Foundation (award #1617197), a Google Research award, and a Visa Research award.

David Evans was supported in part by National Science Foundation awards #1717950 and #1111781, and research awards from Google, Intel, and Amazon.