# Post-Meeting Thoughts on *Responsible Sharing of Federal Data*

David Evans, Professor of Computer Science, University of Virginia

7 November 2019

Thank you for the opportunity to participate in this effort. I am encouraged by the strong interest leaders have expressed in exploring technologies for enabling privacy-preserving data sharing, as well as the willingness shown to include academic scientists without vested interests in selling particular products in these discussions. I would like to clarify two important issues that emerged during the roundtable, regarding trust assumptions and the stated goals for a scalable data sharing approach.

## Trust Assumptions

There is an important and clear distinction between the two main threat models considered when we a perform a secure computation on sensitive data:

1. Participants agree on some entity they all trust (*common trusted entity*).
2. Each participant only relies on mathematics and things they control (*mutual distrust*).

**Common Trusted Entity.** For the first model, the solution is easy — everyone just gives their data to the one entity they all trust, which does the computation. In this setting, the design goal is to minimize the size and complexity of the trusted entity to increase the likelihood that the mutual trust the parties have in it is warranted. This is the goal of secure enclaves: they are designed to ensure that the only components that have access to the sensitive data are trustworthy. Recent approaches have sought to build trusted execution environments into commodity processors (e.g., Intel's SGX and ARM's TrustZone).

When parties agree to trust their sensitive data to a process running in a secure enclave, they are trusting the hardware design of that secure enclave, the manufacturing process that produced the physical hardware they are using that implements that design, the key generation and management process for the secure enclave and the entity responsible for that key management and distribution, the physical security of the location where the trusted environment is executing, and all the software that runs inside the enclave. Making the secure enclave open source can reduce the need to trust the hardware design since it can be independently verified, but participants still need to trust the manufacturing, key management, physical security, and software.

**Mutual Distrust.** For the mutual distrust model, participants do not use any common trusted entity, but rely on the security of the cryptographic protocol they use, which can be independently and formally verified building on widely accepted mathematical assumptions.

In the ideal case, each participant should select and procure in their own trusted way all the hardware and software they use that touches their sensitive data, and the only way that data is exposed is through the cryptographic protocol used to perform the joint computation. When secure computation is used in practice, however, this idea is rarely achieved — instead, typically all the participants end up trusting a single software vendor that provides the protocol implementation (although more sophisticated users will carefully audit that software and verify that the software they are running is the same as the one they audited). (A variation on the mutual distrust model is

what is known as the *semi-honest model*, which assumes that participants can trust each other to follow the protocol exactly as specified. This is a useful threat model in academic work, and there are methods for transforming certain kinds of semi-honest protocols to work in the mutual distrust model, but adopting solutions in the semi-honest model directly never makes sense in practice.)

**Output Disclosure.** Regardless of whether or not a computation is done securely, a separate issue is what is done with the output of that computation. Any function that is computed on sensitive data potentially produces an output that leaks sensitive information about that data. This is the main privacy issue if the results of the computation will be made public or released to people who are less trusted than the original data owners. Differential privacy mechanisms and synthetic data generation are designed for this goal. They replace the output of the computation on sensitive data, with a different output (with carefully generated random noise added), with the goal of limiting what can be inferred about the sensitive data. This presents difficult and use-case specific tradeoffs, since any output that has any value to a user must reveal something about the sensitive data.

## Scalability

There was a lot of discussion about the goal of developing a scalable method, which was used to mean something that makes it easy to conduct new joint data analyses between different agencies on different data sets (not scalability in terms of handling large data sets and complex functions). There are two requirements for this type of scalability: (1) the technology used is general purpose and can be easily adapted to many different problems, and (2) the bureaucratic process needed to apply the solution to a new problem is easy. Although certain problems and threat models are still challenging, the first requirement is well handled by many existing technologies, and a wide range of problems.

In my view, the second requirement should never be met — it should always require a difficult process for organizations to release sensitive data that is entrusted to them. There is no one-size-fits-all way to determine the right tradeoffs for a given use case, and getting them right is essential for protecting privacy while providing utility. The decisions about what threat model is acceptable and what needs to be done to prevent inference about sensitive information while providing enough information in the output to enable a given use, are complicated and depend on the specifics of the data and the intended use of the outputs.

In reflection on the meeting, I am increasingly disturbed by the discussion that the main difficulties are that CIOs won't sign off on data sharing using data they are responsible for, so the main thing we need is a way to either absolve them of that responsibility, or provide a technical and bureaucratic solution that allows the data sharing without anyone being responsible for it. Any solution to scalability that removes responsibilities from data owners has great potential for abuse and misuse. That said, I would fully support any efforts to educate CIOs and others responsible for data on the opportunities and risks associated with these technologies, and in providing a clearer framework for making these decisions. But, it is essential that data owners are still responsible, and that any data sharing that exposes data beyond what it was collected and authorized for, goes through a careful process where decision makers fully understand that the technologies can and cannot guarantee.