

UC Riverside

UC Riverside Previously Published Works

Title

Trustworthy Scientific Computing

Permalink

<https://escholarship.org/uc/item/40r1229t>

Author

Peisert, Sean

Publication Date

2021

Peer reviewed

► Terry Benzel, Column Editor

Security

Trustworthy Scientific Computing

Addressing the trust issues underlying the current limits on data sharing.

DATA USEFUL TO science is not shared as much as it should or could be, particularly when that data contains sensitivities of some kind. In this column, I advocate the use of hardware trusted execution environments (TEEs) as a means to significantly change approaches to and trust relationships involved in secure, scientific data management. There are many reasons why data may not be shared, including laws and regulations related to personal privacy or national security, or because data is considered a proprietary trade secret. Examples of this include electronic health records, containing protected health information (PHI); IP addresses or data representing the locations or movements of individuals, containing personally identifiable information (PII); the properties of chemicals or materials, and more. Two drivers for this reluctance to share, which are duals of each other, are concerns of data owners about

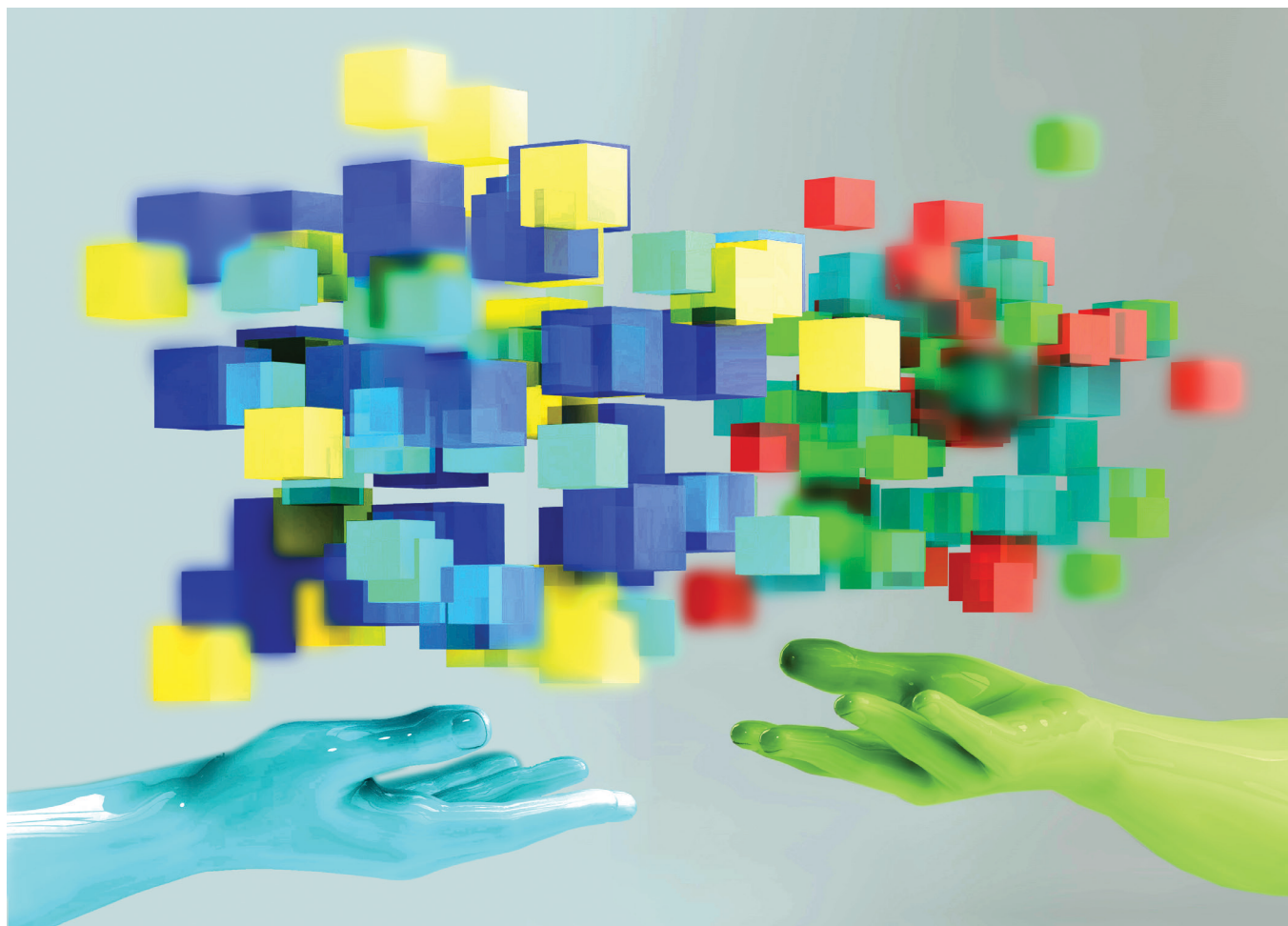
Hardware trusted execution environments can form the basis for platforms that provide strong security benefits while maintaining computational performance.

the risks of sharing sensitive data, and concerns of providers of computing systems about the risks of hosting such data. As barriers to data sharing are imposed, data-driven results are

hindered, because data is not made available and used in ways that maximize its value.

And yet, as emphasized widely in scientific communities,^{3,5} by the National Academies, and via the U.S. government's initiatives for "responsible liberation of Federal data," finding ways to make sensitive data available is vital for advancing scientific discovery and public policy. When data is not shared, certain research may be prevented entirely, be significantly more costly, take much longer, or might simply not be as accurate because it is based on smaller, potentially more biased datasets.

Scientific computing refers to the computing elements used in scientific discovery. Historically, this has emphasized modeling and simulation, but with the proliferation of instruments that produce and collect data, now significantly also includes data analysis. Computing systems used in science include desktop systems and clusters run by individual



investigators, institutional computing resources, commercial clouds, and supercomputers such as those present in high-performance computing (HPC) centers sponsored by U.S. Department of Energy's Office of Science and the U.S. National Science Foundation. Not all scientific computing is large, but at the largest scale, scientific computing is characterized by massive datasets and distributed, international collaborations. However, when sensitive data is used, computing options available are much more limited in computing scale and access.⁸

Current Secure Computing Environments

Today, where remote access to data is permitted at all, significant technical and procedural constraints may be put in place, such as instituting ingress/egress "airlocks," requiring "two-person" rules to move software and data in or out, and requiring the use "remote desktop" systems. Archi-

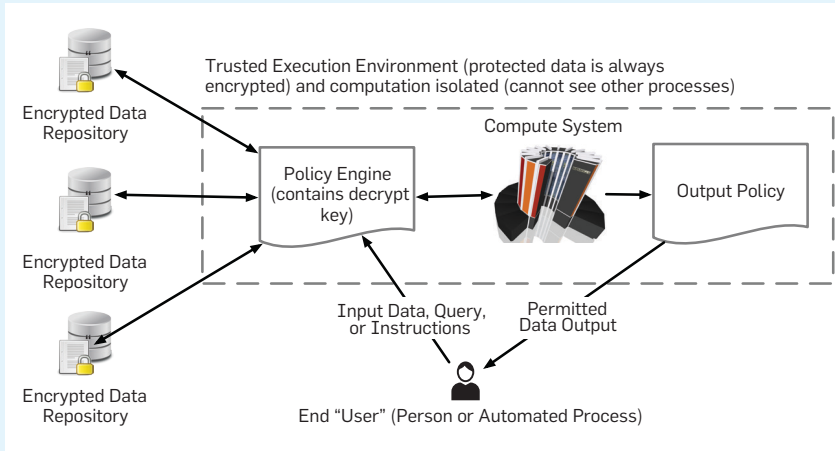
tectures like this are becoming more and more common as means for scientific computing involving sensitive data.⁸ However, even with these security protections, traditional enclaves still require implicitly trusting system administrators and anyone with physical access to the system containing the sensitive data, thereby increasing the risk to and liability of an institution for accepting responsibility for hosting data. This security limitation can significantly weaken the trust relationships involved in sharing data, particularly when groups are large and distributed. These concerns can be partially mitigated by requiring data analysts to be physically present in a facility owned by the data provider in order to access data. However, in all these cases, analysis is hindered for the scientific community whose abilities and tools are optimized for working in open, collaborative, and distributed environments. Further, consider the current pandemic in which a requirement of

physical presence in a particular facility for analysis would be a public health risk.

Reducing Data Sensitivity Using "Anonymization" Techniques

Sometimes attempts are made to avoid security requirements by making data less sensitive by applying "anonymization" processes in which data is masked or made more general. Examples of this approach remove distinctive elements from datasets such as birthdates, geographical locations, or IP network addresses. Indeed, removing 18 specific identifiers from electronic health records satisfies the HIPAA Privacy Rule's "Safe Harbor" provisions to provide legal de-identification. However, on a technical level, these techniques have repeatedly been shown to fail to preserve privacy, typically by merging external information containing identifiable information with quasi-identifiers in the dataset to re-identify "anonymized" records.⁶ Therefore,

A portion of a system leveraging a trusted execution environment in which data is stored encrypted on disk; a policy engine, controlled by the data owner, and running in the TEE, contains the mapping for what data is to be made available for computing by each authenticated user; and an output policy, also specified by the data owner, dictates what information is permitted to be returned to the user. An output policy might be based on differential privacy, or be access-control based, or be some combination of these or other functions.



de-identification does not necessarily address the risk and trust issues involved in data sharing because re-identification attacks can still result in significant embarrassment, if not legal sanctions. In addition, the same masking used in these processes also removes data that is critical to the analysis.⁶ Consider public health research for which the last two digits of a ZIP code, or the two least significant figures of a geographic coordinate are vital to tracking viral spread.

Confidential Scientific Computing

Hardware TEEs can form the basis for platforms that provide strong security benefits while maintaining computational performance (see the accompanying figure). TEEs are portions of certain modern microprocessors that enforce strong separation from other processes on the CPU, and some can even encrypt memory and computation. TEEs have roots in the concepts of Trusted Platform Modules (TPMs) and Secure Boot, but have evolved to have significantly greater functionality. Common commercial TEEs today include ARM's TrustZone, introduced in 2013; Intel's Secure Guard Extensions (SGX), introduced in 2015; and AMD's Secure Encrypted Virtualization (SEV), introduced in 2016 and revamped several times since then to include SEV-ES (Encrypted State) in 2017 and SEV SEV-SNP (Secure

Nested Paging) in 2020. All three vendors take extremely different approaches and have extremely different strengths, weaknesses, use cases, and threat models.

TEEs can be used to maintain or even increase security over traditional enclaves, at minimal cost to performance in comparison to computing over plaintext. TEEs can isolate computation, preventing even system administrators of the machine in which the computation is running from observing the computation or data being

Trusted execution environments can be used to maintain or even increase security over traditional enclaves, at a minimal cost to performance in comparison to computing over plaintext.

used or generated in the computation, including even from certain "physical attacks" against the computing system. They can implement similar functionality as software-based homomorphic and multiparty computation² approaches, but without the usability issues and with dramatically smaller performance penalties.

The use of TEEs to protect against untrustworthy data centers is not a novel idea, as seen by the creation of the Linux Foundation's Confidential Computing Consortium¹⁰ and Google's recent "Move to Secure the Cloud From Itself."⁷ Google has comparing the importance of the use of TEEs in its cloud platform to the invention of email.⁹ However, TEEs have not yet seen broad interest and adoption by scientists or scientific computing facilities.

The envisioned approach is to leverage TEEs when data processing environments are out of the direct control of the data owner, such as in third-party (including DOE or NSF) HPC facilities or commercial cloud environments, in order to prevent exposure of sensitive data to other users of those systems or even the administrators of those systems. Data providers can specify the configuration of the system, even if they are not directly the hosts of the computing environment, to specify access control policies, a permitted list of software or analyses that can be performed, and output policies to prevent data exfiltration by the user. The notion of being able to leverage community HPC and cloud environments also enables the use of data from multiple providers simultaneously while protecting the raw data from all simultaneously, each potentially with their own distinct policies.

Researchers at the Berkeley Lab and UC Davis have been empirically evaluating Intel SGX and AMD SEV TEEs for their performance under typical HPC workloads. Our results¹ show that AMD's SEV generally imposes minimal performance degradation for single-node computation and represents a performant solution for scientific computing with lower ratios of communication to computation. However, Intel's SGX is not performant at all for HPC due to TEE

memory size limitations. Importantly, NERSC-9 and a number of other modern HPC centers will contain AMD processors that support the SEV TEE, and thus it is our hope that our results will provide some of the evidence needed to justify the use of TEEs in scientific computing.

Looking to the Future

Although numerous commercial TEEs exist, no TEEs yet exist in processors other than CPUs, such as in GPUs and accelerators, although Google has indicated that it plans to expand “Confidential Computing” to GPUs, TPUs, and FPGAs.⁹ There are also issues with low-latency communication between TEEs, and also the cost of virtualization, that must be addressed to enable HPC at scale.¹ In addition, promising RISC-V efforts such as Keystone⁴ exist that carry both the promise of broadening the scope of processors that contain TEEs, while also being open source and possible to formally verify. However, RISC-V based TEEs have not yet been developed that target scientific computing. Most likely, an entirely new TEE architecture tailored for scientific computing and data analysis applications will be needed.

Output policies are another area that deserve investigation. While TEEs protect against untrusted computing providers, and can provide certain measures of protection from malicious users, output policies determine what data is returned to the user. Differential privacy is a particularly interesting approach to providing strong privacy protection of data output. Differential privacy is a statistical technique that can guarantee the bounds on the amount of information about a dataset that can be leaked to a data analyst as a result of a query or computation by adding “noise” and enforcing a “privacy budget” that bounds information leakage. It is now a mainstream solution, with production use by Apple, Google, and the U.S. Census Bureau, the existence of several open source distributions, and successful application to a diverse range of data types. However, differential privacy is not appropriate everywhere, and applying it is currently challenging, requiring a high degree of expertise and effort. Thus, differential privacy is


Trusted execution environments enable sensitive data to be leveraged without having to trust system administrators and computing providers.

highly useful today, albeit in a limited set of situations for datasets that have sufficiently wide use to justify the time and expense required. Work is needed to advance the usability of differential privacy so it can more easily be broadly leveraged.

Summary and Next Steps

In contrast to traditional secure enclaves, TEEs enable sensitive data to be leveraged without having to trust system administrators and computing providers. However, while the application of TEEs has now been widely heralded in cloud environments, TEEs have not been discussed for use in scientific computing environments, despite the significant concerns frequently expressed by both data providers and computing facilities about hosting sensitive data. Operators of scientific computing facilities are notoriously conservative for good reason—they are frequently evaluated on the degree of utilization and amount of uptime of the systems they run, and so the margin for error is low. But TEEs are here, they are available, and until we start making use of them in scientific computing, data is not shared as much as it should or could be by leveraging TEEs to address the trust issues underlying current limits on data sharing.

What is missing is a connection to the particular infrastructure used in scientific computing, including identity, access, and authentication systems; remote direct memory access (RDMA);

batch scheduling systems in HPC; HPC I/O subsystems; custom scientific workflows; highly specialized scientific instruments; community data repositories, and so on. Therefore what is needed is a conversation between processor manufacturers, system vendors (for example, Cray, HPE), and scientific computing operators regarding enabling the TEE functionality already present in the AMD EPYC processors—and presumably in other, future processors—into scientific computing environments. However, the path forward is not solely technical. It requires the community to build infrastructure around TEE technology and integrate that infrastructure into scientific computing facilities and workflows, and into the mind-set of operators of such facilities. I hope this column helps to start that conversation. For more on TEEs, see the Singh et al. article on p. 42. —Ed. 

References

1. Akram, A. et al. Performance analysis of scientific computing workloads on trusted execution environments. In *Proceedings of the 35th IEEE International Parallel & Distributed Processing Symposium*. (2021).
2. Choi, J.I. and Butler, K. Secure multiparty computation and trusted hardware: Examining adoption challenges and opportunities. *Security and Communication Networks* 1368905 (2019).
3. Hastings, J.S. Unlocking data to improve public policy. *Commun. ACM* 62, 10 (Sept. 2019), 48–53.
4. Lee, D. Keystone: An open framework for architecting trusted execution environments. In *Proceedings of the Fifteenth European Conference on Computer Systems (EuroSys)* (Heraklion, Greece, 2020), Article 38.
5. Macfarlane, J. When apps rule the road: The proliferation of navigation apps is causing traffic chaos. It's time to restore order. *IEEE Spectrum* 56, 10 (2019), 22–27.
6. Narayanan, A. and Felten, E.W. No Silver Bullet: De-identification Still Doesn't Work. (2014); <https://bit.ly/3loBvMd>
7. Newman, L.H. Google moves to secure the Cloud from itself. *Wired* (July 14, 2020); <https://bit.ly/2NnLLru>
8. Peisert, S. An examination and survey of data confidentiality issues and solutions in academic research computing. *Trusted CI Report* (2020); <https://bit.ly/300ajx9>
9. Potti, S. and Manor, E. Expanding Google Cloud's Confidential Computing portfolio. (2020); <https://bit.ly/38Nozuo>
10. Rashid, F.Y. The rise of confidential computing. *IEEE Spectrum* 57, 6 (2020), 8–9.

Sean Peisert (speisert@lbl.gov) leads computer security R&D at Lawrence Berkeley National Laboratory and is an Associate Adjunct Professor at University of California, Davis, USA.

The author thanks Venkatesh Akella, Ayaz Akram, Jim Basney, Jason Lowe-Power, and Von Welch for their valuable feedback on this Viewpoint and the ideas in it. This work was supported by the Director, Office of Science, Office of Advanced Scientific Computing Research, of the U.S. Department of Energy, and by Contractor Supported Research (CSR) funds provided by Lawrence Berkeley National Laboratory, operated for the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors of this work.

Copyright held by author.