EDITOR **MARK CAMPBELL**
EVOTEK; mark@evotek.com

# Privacy-Preserving Computation: Doomed to Succeed

**Mark Campbell,** EVOTEK

*We are on the quest for mainstream adoption of privacy-preserving computation solutions. As present challenges are solved, homomorphic encryption and related privacy-preserving techniques will change the face of information technology, security, privacy, and policy.*

Every secret shared is vulnerable at three times: when the secret is held, being sent, and being used. In today's digital parlance, we call these *data at rest*, *data in transit*, and *data in use*, respectively. Although we have sophisticated and efficient solutions for the first two cases, we lack a tractable solution to the third. If protecting data in use is the Holy Grail of data privacy, homomorphic encryption (HE) and related technologies are its Galahad.

## HE HISTORY

In 1978, a group of cryptographers noted the unintentional but intriguing property of their newly unveiled Rivest–Shamir–Adleman (RSA) algorithm by which operations could be applied directly to encrypted data without decryption and the resultant still be valid when decrypted—a mathematical property called *homomorphism*. In their seminal work "On Data Banks and Privacy Homomorphisms," Rivest, Adelman, and Dertouzos (RAD), outlined a set of special encryption functions they called *privacy homomorphisms*.[1] They used the example of a small loan company that encrypts their sensitive loan data (for example, loanees, balances, and payments) using a special privacy homomorphism, sends the encrypted data to an outside party, and makes requests on these data such as average loan balance outstanding and upcoming loan payments. The outside party performs these operations and returns the (still-encrypted) response. The loan company decrypts the response to get the answer to their query.

The beauty of this procedure is that data are shared and used without the outside party ever having access to the original data themselves, thereby guaranteeing privacy. Although the paper outlined a solid approach based on

*Digital Object Identifier 10.1109/MC.2022.3178169*
*Date of current version: 2 August 2022*

properties found in RSA, there was only one small catch: How would one build a special privacy homomorphism? This took more than three decades to solve.

In his 2009 doctoral dissertation, modestly titled "A Fully Homomorphic Encryption Scheme," Craig Gentry outlined how one could build special privacy homomorphism, now called a *fully HE* (*FHE*), using "ideal lattices," a noise injector, and recursion.[2] Finally, a conceptual scheme had been developed showing how one would build an FHE. The rub is that this procedure produced encrypted data that were tens of thousands of times larger than the originals and took millions of times longer to operate on. This relegates FHE to minute data sets and uncomplicated operational queries. Nonetheless, the treasure is still there today, and the quest continues.

Several derivative HE schemes have been developed to squeeze more encryption out of fewer resources, with the most common including fast fully homomorphic encryption over the torus,[4] Brakerski-Gentry-Vaikuntanathan,[5] Brakerski/Fan-Vercauteren,[6] and Cheon Kim Kim Song,[7] but others such as Palisade, software-optimized encryption algorithm by Microsof, homomorphic encryption library by IBM, and homomorphic encryption for arithmetic of approximate numbers have also gained popularity.

> Only one small catch: How would one build a special privacy homomorphism?

## HE TODAY

Since RAD and Gentry's publications, the cadre of academic and commercial HE research projects has grown steadily, producing orders-of-magnitude performance and storage improvements. Many of these efforts have explored relaxing the number of addition and multiplication operations allowed on encrypted data and loosely fall into the following four levels[3]:

1. *FHE*: an unlimited number of both addition and multiplication operations.
2. *Somewhat HE*: a limited number of either addition or multiplication operations, but not both.
3. *Partially HE*: only one addition or multiplication operation, but not both.
4. *Leveled HE*: relinearization and modulus reduction performed on a limited number of operations.

Today, HE solutions are joined by several other methods to help protect data in use. Collectively called *privacy-preserving computation techniques* (*PPCTs*), these hide information while allowing computation over it[3,8]:

- *Trusted execution environments*: hardware solutions that encrypt portions of process memory to prevent side-channel attacks.
- *Secure multiparty computation*: multiple parties provide encrypted input data, perform computations across data from all parties, and output a shared result to all parties.
- *Zero-knowledge proofs*: an iterative cryptographical construct in which a requester (prover) can prove to the server (verifier) that they have a given piece of knowledge without providing any information about the knowledge itself.
- *Federated learning*: allows artificial intelligence model training by different parties without revealing any party's data to another.
- *Differential privacy*: guarantees an individual's data privacy and anonymity while allowing the calculation of population-wide statistics.
- *Private set intersection*: allows multiple parties to compare encrypted versions of their data to compute and share the intersection of these sets.

## INCREASING DEMAND FOR PROTECTING DATA IN USE

In 2018, the Spectre and Meltdown breaches caught security teams flat-footed. These novel breaches exploited CPU design flaws that allowed bad actors to peek over the fence into their neighbor's memory space and view their unencrypted data as they were being processed.[8] This spotlighted the gaps in securing data in use and brought HE and PPCT to the top of security conversations.

As data privacy regulations like the General Data Protection Regulation and California Privacy Rights Act become increasingly restrictive on how data privacy must be treated, HE solutions show great promise in aiding compliance. By never allowing unencrypted user data to be transferred or seen, many privacy concerns are bypassed or obviated ... Imagine the benefit of refactoring existing services to leverage HE- and PPCT-based solutions so that they work on encrypted user data without privacy concerns.

Customers increasingly demand control, visibility, and discretion of their personal data and will leave services that do not provide adequate privacy protection. A service that provides value without using unencrypted user data has a distinct lead over those built on a "share and trust" model. The company that garners more trust may beat out competitors with more features and lower cost. In a privacy arms race, companies that can exploit the "share nothing" ethos of HE and PPCT will have a large competitive advantage.

HE and PPCT solutions can be readily applied to a growing number of compelling use cases. The advent of public cloud services enables joint computational

efforts among many diverse parties and data sources (some potentially untrusted), each governed by various data privacy regulations. Data-in-use security solutions enable these parties to collaborate on computational analysis without revealing their underlying data sets and subjecting them to privacy regulation restrictions. A genetic analysis across medical research firms and agencies, antifraud and antimoney laundering detection across national law enforcement agencies, and risk analysis and transfer across financial services institutions are only a few examples enabled by computation on encrypted proprietary data.

This groundswell of real-world uses enabled by HE and PPCT have led Gartner to predict that by 2025, "at least 20% of companies will have a budget for projects that include FHE, up from less than 1% today" and "60% of large organizations will use one or more privacy-enhancing computation techniques in analytics, business intelligence, or cloud computing."[9]

## REMAINING CHALLENGES

Several significant challenges must be met before HE and PPCT become a common architectural feature in applications and services, including

- *Performance*: The single greatest inhibitor of widespread HE adoption is undoubtedly performance. In 2018, IBM released a streamlined version of HElib C++ that sped up computation by 75 times over the previous version and two million-times faster than the version from three years before. Despite this dramatic performance boost, computation is still about one million-times slower than operations on plaintext.[10] So, a 4-ms plaintext operation will take its HE counterpart about an hour. This performance gap must be reduced by several orders of magnitude before it sees widespread adoption.
- *Consumability*: A cursory survey of HE and PPCT techniques on the market today reveals elaborate algorithms and concepts that require skills not found in most organizations. Although the HE community and skill pool increases steadily each year, mainstream adoption will require an arsenal of highly technical solution providers to make these techniques consumable and verifiable. Promising start-ups such as Baffle, Duality, Enveil, Inpher, Titaniam, and Tripleblind.ai are currently joining industry stalwarts like IBM and Microsoft to provide readily consumable products and services to keep pace with growing data-in-use protection demands. As these solutions mature and expand, wider adoption will follow.
- *Full query support*: The query operations supported on encrypted data varies by application and use case. Eventually, HE and PPCT solutions will need to support a full arsenal of query requests on par with today's Standard Query Language data management systems. Certainly, point applications can be built with more rudimentary operations, but the ability to "flip on the privacy switch" for a given data source will open data-in-use protection to most existing and future applications.
- *Standards*: Standards are presently being created to systematize frameworks for the research, development, and adoption of HE security, application programming interfaces, and applications. Although the 2018 "Homomorphic Encryption Standard" by community-based https://homomorphicencryption.org[11] is now de facto, there is a growing need for adapting and evolving the standard to apply to expanded parameter sets.[12] Others await government or international organizations such as the International Organization for Standardization or National Institute of Standards and Technology to publish sanctioned standards. Until mathematically sound uniformity is adopted across HE and PPCT solutions, a solution's overall security and interoperability will not be readily verifiable by the consumer or user.

Despite current HE and PPCT challenges, the potential of protecting data in use is irresistible. Governments, medical establishments, universities, financial institutes, and law enforcement agencies "desperately want data-in-use security to become a reality."[13] This is coming.

## THE FUTURE OF PROTECTING DATA IN USE

In anticipation of meeting today's challenges, several trends indicate what tomorrow's data-in-use protection will look like:

- *On-demand services*: Several firms, including IBM,[14] are now offering HE services, and public cloud vendors will soon provide HE as a service too. These types of services will replace the specialized knowledge, systems, processes, and tools required to implement an HE and PPCT solution with a simple monthly bill.
- *HE on a chip*: Several efforts outlining how to build a custom fully homomorphic cryptoprocessor from the ground up have been around for more than a decade.[15] Recently however, several efforts have been launched to create a specialized processor optimized for HE computations. One such Indonesian effort uses low-cost programmable floating point gate arrays to achieve an order-of-magnitude

performance gain over traditional CPU-based systems.[16]

- *Data-in-use security systems*: An alternate approach uses a layered framework of HE and PPCT technologies engineered into a function system to accomplish what individual technologies are incapable of today. One such product recently made available to the market is Titaniam, built on differential privacy and multiparty compute with no HE at all. Titaniam's solution manages the entire data lifecycle in which sensitive data are always encrypted, protecting them from careless transfer, loss, or theft by a bad actor. "It's a system, not just an algorithm, and so we can rely on the engineering and not just the underlying math alone," notes Arti Raman, CEO and founder of Titaniam.[13] These systemic solutions not only hold the promise of bypassing current HE limitations, they also lay a firm foundation for layering other data-in-use security solutions as they emerge.

Another example is Baffle, which uses secure multiparty compute via database extensions to provide privacy-preserving data ingest, storage, and computation. By handling only encrypted data, Baffle's customers can move proprietary or protected information to the cloud without fear of disclosure. Baffle's underlying techniques add only a small percentage to the performance profile and allow databases, data warehouses, ingest pipelines, and visualization tools to operate in a "plaintext-free" environment. "We view this as a change in the data pipeline architecture, with security built in, rather than a security solution that is bolted on," comments Ameesh Divatia, CEO and cofounder of Baffle, Inc.[17]

- *Encryption by default*: Using history to illustrate a point, when Transport Layer Security (TLS) emerged in the early 2000s to encrypt Internet traffic, it added a modest operational overhead compared to sending messages in plaintext. Before implementing a TLS solution, an engineer needed to show that the security risks of plaintext traffic were greater than the encryption speed penalty. Today, almost all network traffic runs on TLS and an engineer proposing a plaintext solution would need to justify why they would jeopardize security for such a modest performance gain. Similarly, solving these PPCT challenges foreshadow a world where data handling would be "encrypted with privacy-preserving techniques by default and exceptions would need special justification," predicts Jon Callas, director of technology projects at the Electronic Frontier Foundation and cofounder of the PGP Corporation and Blackphone.[18]

These predictions are only a small subset of the changes that privacy-preserving solutions will bring to the information technology landscape.

Data-in-use protection and privacy-preserving schemes have come a long way from the first theoretical speculations to today's layered PPCT solutions. Yet, there is still a long way to go and daunting challenges to solve. The promise unlocked by privacy-preserving solutions, however, far outweighs the time, effort, and investment of surmounting today's challenges. This might take years to evolve, but privacy-preserving computation seems a certain destination. We are doomed to succeed.

## REFERENCES


1. R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," Massachusetts Inst. Technol., Cambridge, MA, USA, 1978. [Online]. Available: http://people.csail.mit.edu/rivest/RivestAdlemanDertouzos-OnDataBanksAndPrivacyHomomorphisms.pdf
2. C. Gentry, "A fully homomorphic encryption scheme," Stanford Univ., Stanford, CA, USA, 2009. [Online]. Available: https://crypto.stanford.edu/craig/craig-thesis.pdf
3. J. Cabrero-Holgueras and S. Pastrana, "SoK: Privacy-preserving computation techniques for deep learning," in *Proc. Privacy Enhancing Technol.*, 2021, pp. 139–162, doi: 10.2478/popets-2021-0064.
4. I. Chillotti, N. Gama, M. Georgieva, and M. Izabachene, "TFHE: Fast fully homomorphic encryption over the torus," *J. Cryptography*, vol. 2020, no. 33, pp. 34–91, 2019, doi: 10.1007/s00145-019-09319-x.
5. Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in *Proc. 3rd Innov. Theor. Comput. Sci. Conf.*, Cambridge, MA, USA, 2012, pp. 1–36, doi: 10.1145/2090236.2090262.
6. J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," Katholieke Universiteit Leuven, Leuven-Heverlee, Belgium, 2012. [Online]. Available: https://eprint.iacr.org/2012/144.pdf
7. J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Proc. Int. Conf. Theory Appl. Cryptology Inf. Secur.*, Hong Kong, China, 2017, pp. 409–437, doi: 10.1007/978-3-319-70694-8_15.
8. S. M. Kerner. "Spectre and meltdown vulnerabilities & protection." eSecurity Planet. https://www.esecurityplanet.com/applications/spectre-and-meltdown-vulnerabilities/ (Accessed: May 17, 2022).

9. M. Driver, "Emerging technologies: Homomorphic encryption for data sharing with privacy," Gartner, Stamford, CT, USA, 2020. [Online]. Available: https://www.gartner.com/en/documents/3983970
10. "What is homomorphic encryption, and why isn't it mainstream?" KeyFactor. https://www.keyfactor.com/blog/what-is-homomorphic-encryption/ (Accessed: May 9, 2022).
11. "Homomorphuic encryption standard," Homomorphic Encryption, Nov. 21, 2018. Accessed: May 18, 2022. [Online]. Available: http://homomorphicencryption.org/wp-content/uploads/2018/11/HomomorphicEncryptionStandardv1.1.pdf
12. B. R. Curtis and R. Player, "On the feasibility and impact of standardising sparse-secret LWE parameter sets for homomorphic encryption," in *Proc. 7th ACM Workshop Encrypted Comput. Appl. Homomorphic Cryptography*, London, U.K., 2019, pp. 1–10, doi: 10.1145/3338469.3358940.
13. A. Raman, private communication, May 12, 2022.
14. "IBM security homomorphic encryption services," IBM, Armonk, NY, USA, 2020. [Online]. Available: https://www.ibm.com/downloads/cas/KQ27PWBO
15. P. T. Breuer and J. P. Bowen, "A fully homomorphic crypto-processor design," in *Proc. Int. Symp. Eng. Secure Softw. Syst.*, Paris, France, 2013, pp. 123–138, doi: 10.1007/978-3-642-36563-8_9.
16. I. Syafalni, G. Jonatan, N. Sutisna, R. Mulyawan, and T. Adiono, "Efficient homomorphic encryption accelerator integrated PRNG using low-cost FPGA," *IEEE Access*, vol. 10, pp. 7753–7771, Jan. 2022, doi: 10.1109/ACCESS.2022.3143804.
17. A. Divatia, private communication, May 19, 2022.
18. J. Callas, private communication, May 18, 2022.



**MARK CAMPBELL** is the chief innovation officer for EVOTEK, San Diego, California, 92121, USA. Contact him at mark@evotek.com.







*Digital Object Identifier 10.1109/MC.2022.3188070*