leidos

Privacy-Enhancing Technologies (PETs):

Protecting Data In Use

Liv d'Aliberti, Evan Gronberg, Joe Kovba

Standard means of data encryption provide a high level of protection for data at rest and in transit. However, when the data needs to be used, whether for human analysis or as part of an automated system, it must be decrypted. Decrypting data creates an opportunity for plaintext to be exposed to unauthorized parties, whether intentionally by malicious actors or unintentionally by honest but careless adversaries, within the system. To address this data layer threat, system architects are incorporating privacy-enhancing technologies (PETs) into their systems.

What are privacy-enhancing technologies (PETs)?

PETs are technologies that aim to protect *privacy* and *confidentiality* of *data in use* without reducing necessary system functionality. Specifically, PETs are designed to do the following:

- Allow parties to collaborate while guaranteeing that any shared data will be used only for its intended purposes
- Glean insights from private data without revealing the sensitive contents of the data
- Carry out trusted computation in an untrusted environment
- Secure access to shared machine learning (ML) models without revealing sensitive data
- Add quantum-resistant data protections to the system
- Maintain complete control of the data throughout its lifecycle

We rely on the following definitions of privacy and confidentiality:

Confidentiality—The protection of any information that an entity has disclosed in a relationship of trust with the expectation that it will not be divulged to unintended parties. [1]

Privacy—Control over the extent, timing, and circumstances of sharing personal information. Requires special protections around the ways personal information is collected, used, retained, disclosed, and destroyed. [2]

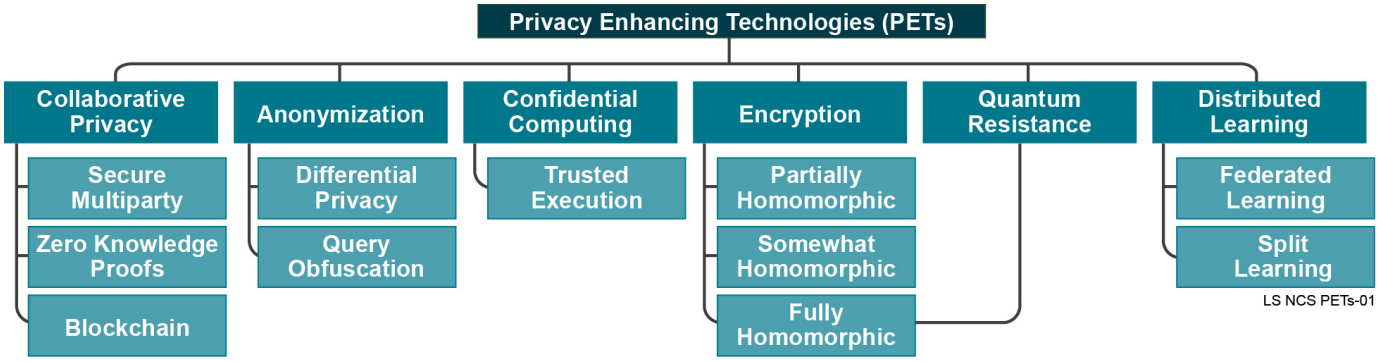


Figure 1. A Diagram of PETs Organized into Groups Based on Each Technology's Primary Mechanism

The primary difference between *privacy* and *confidentiality* is that the former protects personal information, while the latter protects data designated as sensitive. Moreover, confidentiality protects against the unauthorized use of information already in the hands of an organization, whereas privacy protects the rights of an individual to control the information that the organization collects, maintains, and shares with others. This separation of terms is important, because it lends itself to a separation of requirements for a system's architecture.

Data in use is data that is currently being updated, processed, erased, accessed, or read by the system. Examples of activities that require data in use include humans reading files, generating statistics, and carrying out active processes, but it also includes processes originating from humans that are automatically carried out by the system. Artificial intelligence (AI) models, as well as the data run through those models during inference, both represent data in use, for the following reasons:

- It is possible to extract information about the data used during training from that original model. This issue is highlighted in the U.S. Copyright Office's recent decision to not grant copyright to AI-generated art [3] because of the model's ability to replicate artist style without appropriate accreditation.
- A model is a highly refined, structured, and trained representation of data; both the way a model is structured and its training curriculum are key to its performance. For example, TikTok's algorithm is deemed highly successful at captivating users' attention. The model, which is the core of ByteDance Ltd.'s business, performs uniquely well because of its structure and its use of training data. In 2022, the Wall Street Journal used passive bots [4] to attack TikTok's model. The bot-based information collection campaign hijacked TikTok's data-in-use process to uncover sensitive model structure information.
- A model trained on private data could be used in ways that are unintended by the people who provided the data. Part of privacy is control over how and when personal information is used. Technical guarantees, in the form of PETs, need to be put in place to ensure that models trained on private data will only be used as permitted by the data owners.

As data collection, collaboration, and usage increase, it is increasingly important to acknowledge the vulnerabilities that data in use can introduce into a system, as well as the ways PETs can help address data layer vulnerabilities. The term "privacy-enhancing technologies" encompasses a wide variety of tools—hardware and software, local and cloud-based—all centered on providing increased protection for data in use. **Figure 1** shows various PETs, organized by their primary defense mechanism, demonstrating the breadth of functionalities and tools under the PETs umbrella.

PET Name	Degree of Privacy	Encryption	Scalability	Data Type Compatability	Speed	AI/ML Training/ Inference	No Masking and Hashing	Hardware Dependencies	Quantum Resistance
Zero-Knowledge Proof (ZKP)									
Trusted Execution Environments (TEE)									
Split Learning (SL)									
Secure Multi-Party Compute (SMPC)									
Query Obfuscation									
Privacy-Preserving Blockchain									
Federated Learning (FL)									
Fully Homomorphic Encryption (FHE)									
Differential Privacy (DP)									

Perfect Fit Potentially Viable Show Stopper

LS NCS PETs-02

Figure 2. A Selection of PETs Evaluated Against Important System-Level Variables. *This diagram indicates potential reasons to consider a particular PET for a given use case.*

Multiple PETs may operate in tandem to address a particular security concern or set of security concerns more fully. As **Figure 2** shows, each PET has its own strengths and weaknesses—there is no "one size fits all" PET.

The columns represent a system-level capability that could be enhanced or harmed by the addition of a PET. The intended capabilities of each PET are shown on this chart as a *perfect fit*. The *potentially viable* notation means that a PET can be modified to better fit the system. The chart also shows reasons that a given PET might not be used; for example, fully homomorphic encryption (FHE) can be detrimental to a system's speed and scalability, so it is classified as a *show stopper*. The following paragraphs present brief descriptions of some of the PETs Leidos is using to create differentiated systems.

Fully Homomorphic Encryption (FHE)

FHE allows users to perform computation on encrypted data and models—data in use is never decrypted. FHE offers incredible, perhaps even seemingly impossible, security. However, in its current state, it must be implemented very conscientiously. Realistic use cases for FHE include performing secure ML inference over smaller models, fully privatized database queries, and encrypted collaborative statistical analysis. **Figure 3** shows how FHE could be used to privatize an external remote process.

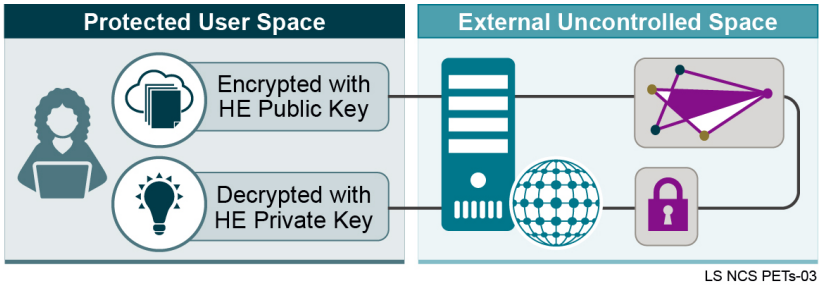


Figure 3. Example of an FHE Pipeline. *With FHE, users can send their data in an encrypted format for secure external remote processing. Insights generated by the remote processes remain encrypted until securely back in the hands of the data owner.*

Federated Learning (FL)

Federated learning uses multiple clients' data to build a shared ML model while keeping each client's training data local—no client can see any other client's data. Each client trains a model locally then sends that model (along with the number of examples that were used to train the model) to the server. The server then performs a weighted aggregation of all the models and sends the aggregated model to each client in the federation. This architecture relies on edge nodes with enough compute capacity to update their local model and may risk data deanonymization if not used in tandem with another PET. **Figure 4** shows an example of how FL could be used in a centralized system to update user models without sharing user data.

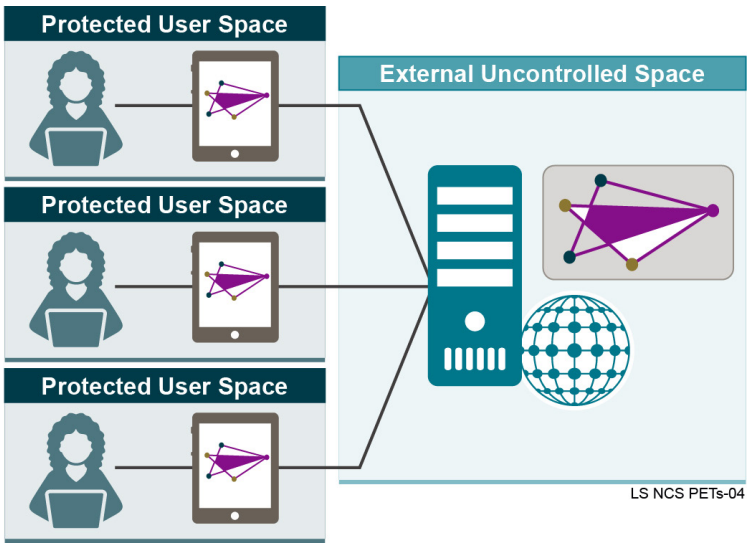


Figure 4. Example of an FL Pipeline. *With FL, users do not send training data, but instead only send model weights to a remote server. The remote server then aggregates those weights across all clients and returns the updated global weights to each client.*

Trusted Execution Environments (TEEs)

A TEE is an isolated compute space, separate from a host/parent instance, that relies on hardware-based encryption to protect data in memory and application-level code. Entities outside a TEE are unable to see or alter data or code during execution. A program running inside a TEE is cryptographically attested to be the program that is intended to run. **Figure 5** shows how a TEE could be deployed to ensure that data sent to a model is neither seen nor tampered with during remote inference.

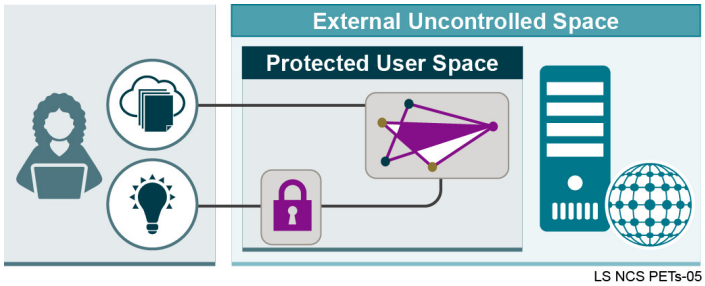


Figure 5. Example of a TEE Pipeline. *A TEE allows users to send data to a remote instance and perform computation within a "blocked off" area. The results are then returned to the user without revealing executed processes to the remote instance.*

Query Obfuscation

Query obfuscation allows users to derive alternative, less sensitive queries from a user query. These obfuscated queries do not directly reveal sensitive user information through the terms entered into a search engine. This method enables the investigation of sensitive topics over an uncontrolled network without revealing to the external party the true intent of the research campaign. The intent is to make it harder to tell who is asking the question and why the information might be important to the querier. Query obfuscation requires a balance between query privacy and specificity to ensure the search produces relevant results without revealing information about the person asking the question. **Figure 6** shows how a sensitive query might be broken down into obfuscated subcomponents, sent to an external search engine, and then returned and parsed to generate insights.

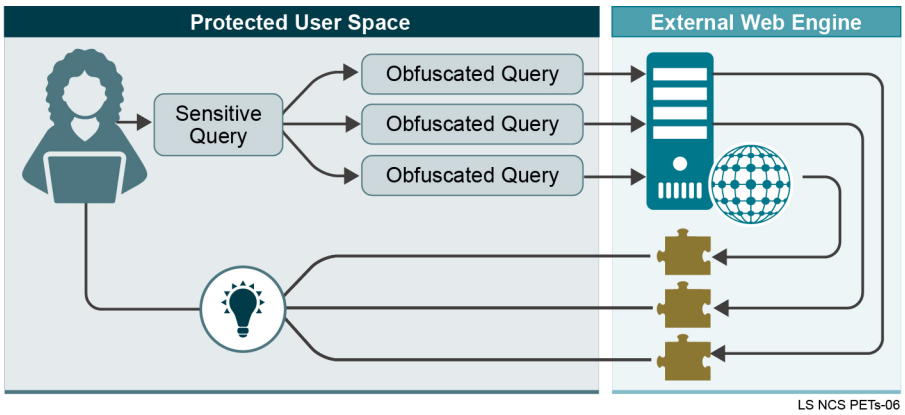


Figure 6. Example of a Query Obfuscation Pipeline. *In this example, a sensitive query is separated into nondescript component queries that separately do not reveal the true intent of the sensitive query. The component queries are then sent to an external web engine, and then reparsed as a set of insights for the user—returning results related to the desired sensitive query.*

What kind of attacks do PETs defend against?

The PET used depends on the type of adversary the system needs protection against, such as the following:

- **Careless Insider**—a person who has rightful access to a system but also the potential to leak or use the system's information in unintended ways
- **Curious Outsider**—a person who has no rightful access to a system but can still manage to connect to the data and models to learn more about the system without intent to harm the system or wrongfully distribute its data
- **Malicious Insider**—a person who has rightful access to a system but is intent on harming the system or wrongfully distributing its data
- **Malicious Outsider**—a person who has no rightful access to a system but is intent on leaking or using the system's information in unintended ways

Insider Threat Vulnerability

Insider threat attacks occur when an approved user gains authorized access to protected data resources and then shares that data in an unauthorized manner. A recent example of this kind of attack is the Discord leaks [5], where a malicious insider copied sensitive material and shared it over an unapproved messaging server. As **Figure 7** shows, data, even data that is secured through recommended security protocols, must normally be decrypted for any form of analysis, placing the data at risk of disclosure. Security policies, such as least access principles, minimize access to confidential data; however, few technological techniques have been employed to ensure that malicious insiders are left without the ability to gather and distribute sensitive information.

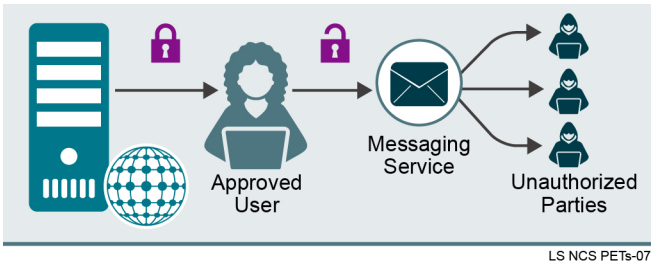


Figure 7. Example of Insider Threat. *In this example, an approved user can access sensitive information and then share it through a messaging service with unauthorized parties.*

We need ways to ensure that analysts can access and share data only as intended. PETs could be implemented to secure "need to know" systems, limiting the information a given analyst has access to in accordance with their explicit tasking.

Machine Learning (ML) Vulnerabilities

ML models are subject to the same challenges that data faces—namely, unauthorized access and manipulation, as shown in **Figure 8**. ML models are vulnerable to manipulation, known as adversarial attacks, in which examples are specially crafted to fool the model. Researchers highlighted the impact of an adversarial attack by confusing a Tesla Model S and forcing it to move into oncoming traffic. [6] PETs could help validate that inputs provided are not manipulated by a malicious user to force unforeseen model behaviors. This validation would be similar to the way cryptography maintains the integrity of data via hashing. The goal of hashing is to validate that data will accomplish its intended purpose, and it is important to carry those parallels into data-in-use scenarios for ML.

Another privacy threat when working with ML models and anonymized data is that the model can potentially be reverse-engineered to reveal sensitive characteristics about the data that was used to train the model, making it possible to relink the data used to create the model to the personally identifiable data in the model's training set. This scenario could result in an unintentional release of sensitive data, like in the Netflix Prize [7], a challenge in which researchers were able to take an anonymized dataset and relink people to their ratings of television shows. PETs can be implemented to better protect datasets while still ensuring that researchers can generate performant models.

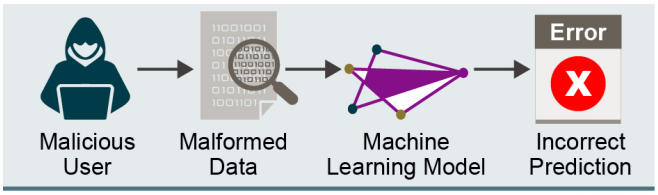


Figure 8. Example of Adversarial Attack. *In this example, a malicious user attempts to confuse an ML model by sending malformed data. This type of attack could have negative results as ML models take on increased control over operations.*

Data Sharing Risks

Without PETs, data must be decrypted before ML models can perform inference. Data leaks can occur when sensitive information is sent to a third-party model owner, as seen in **Figure 9**. Think of prompts sent to ChatGPT. As it stands, ChatGPT must have access to the plaintext prompt for it to produce a response. This requirement poses a confidentiality concern because data is necessarily exposed at some point in the process. Companies such as Samsung [8] and Northrop Grumman [9] are establishing policies for the careful use of ChatGPT and other large language models. The problem with these policies is that organizations cannot guarantee that they will be followed. One of the goals of PETs is to be able to use resources like ChatGPT while maintaining confidentiality.

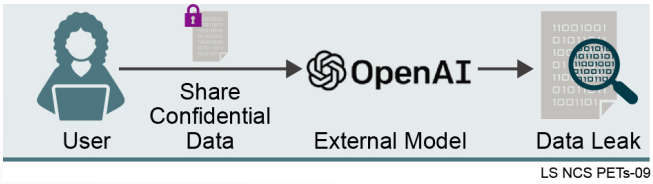


Figure 9. Example of a Data Sharing Risk. *A user shares confidential data in plaintext with an external ML model, creating a potential data leak in which OpenAI (manufacturer of ChatGPT) [10] may gain undue access to confidential data.*

How should PETs be evaluated?

Not all PETs perform the same security tasks, nor do they provide the same level of performance or protection, so it is important to evaluate the PET for system fit. In **Figure 2**, we listed several variables to consider while designing a solution that includes PETs for data layer security. In this section, we describe a holistic framework for determining how a specific PET implementation might affect the system.

Use Case Applicability

The first step of integrating a PET into a system is to clearly define a use case. The use case should define the technical objective, the threats to the system, and all relevant privacy and security requirements. Solution architects with experience using PETs will be able to select the appropriate PET (or collection of PETs) to protect against data-in-use vulnerabilities. To aid in the assessment of PETs against specific use cases, governments around the world are imploring public sector, private sector, and academic institutions to collect, curate, and publish use case repositories for open consumption. Consider the following questions to guide a use case applicability assessment:

- Does the chosen PET adequately address security requirements while maintaining the necessary level of accuracy of the data or model?
- Does the use case include privacy or confidentiality requirements?
- Does the selected PET include encryption, obfuscation, anonymization, or a different mechanism for enhancing the system's data security?

System Design, Integration, and Performance Characteristics

PETs enable innovative approaches to data security and privacy but may also require performance trade-offs. A clear set of standards for PET performance characteristics is still under development, although significant progress has been made in the past 5 years. [11, 12] These up-and-coming standards aim to define cryptographic schemes and security parameters. A key step of implementing PETs is understanding how compatible these standards are with existing system designs.

From an integration perspective, one must consider how the introduction of PETs affects the current system security level. PETs are still in their relative infancy, and new threats to their use will likely be developed in the future. PETs must be integrated in a way that allows for cryptographic agility; otherwise, they may have a net-negative impact on system security in the long run. However, with thoughtful use case design and a flexible integration plan, PETs have the potential to greatly enhance overall system security.

Privacy and performance are currently at odds with each other; enhanced privacy can come at the cost of downgraded performance, and vice versa. PETs have the potential to undermine intended system functionality, and it is important to consider that trade-off during evaluation. Consider the U.S. Census Bureau's effort to minimize disclosure risk through the use of differential privacy (DP). [13] Though DP is effective at protecting the identities of individuals represented in a dataset, the "noise" that it injects into data necessarily perturbs it and can reduce statistical accuracy. The Census Bureau determines voting districts and allocates federal funding based on population statistics. Small perturbations in census data have the potential to produce outsized impacts on real-world decisions. Thus, statistical safeguards that protect the fidelity of the data must exist to ensure the appropriate balance of privacy and accuracy. As another example, consider a healthcare use case in which a doctor needs immediate access to records from another hospital as part of patient care. If the use of a PET introduces an increased latency into a system, it could be detrimental to patient outcomes. Thus, it is critically important to match the performance characteristics of the PET to the well-defined use case. The following questions are helpful in determining PET-related trade-offs:

- Does the type of security provided by the PET integrate well with existing, standard security measures already provided by the system?
- How flexible is the system in its ability trade time, space, compute capacity, and performance for added data protections?
- Where is the system hosted—on premises, in the cloud, or as a hybrid of both?
- Who controls the data collection process?
- Is the system centralized or decentralized?

Implementation Readiness

Implementation readiness concerns both the readiness of the technology itself and the readiness of an organization to adopt the technology.

PETs are currently in a transitional state as they go from being tested and prototyped in a lab to being deployed commercially in the real world. Organizations pushing PETs toward deployment include large companies—such as Intel Corporation, a leader in TEEs [14], specialized FHE hardware [15], and remote attestation [16]—and small companies—such as Duality, a vendor with mature service offerings [17] based on contributions to open-source FHE. [18] While typical measures for software acceptance (e.g., static and dynamic code analysis, fuzzing, penetration testing) are important for guaranteeing the security of such systems, their cryptographic nature also requires formal guarantees of their functional characteristics. Organizations adopting PETs should implement formal verification methods to ensure that the system operates as expected. To aid in such efforts, some organizations are publicly publishing PET maturity assessments to reduce the level of effort required to fully assess a PET's readiness level.

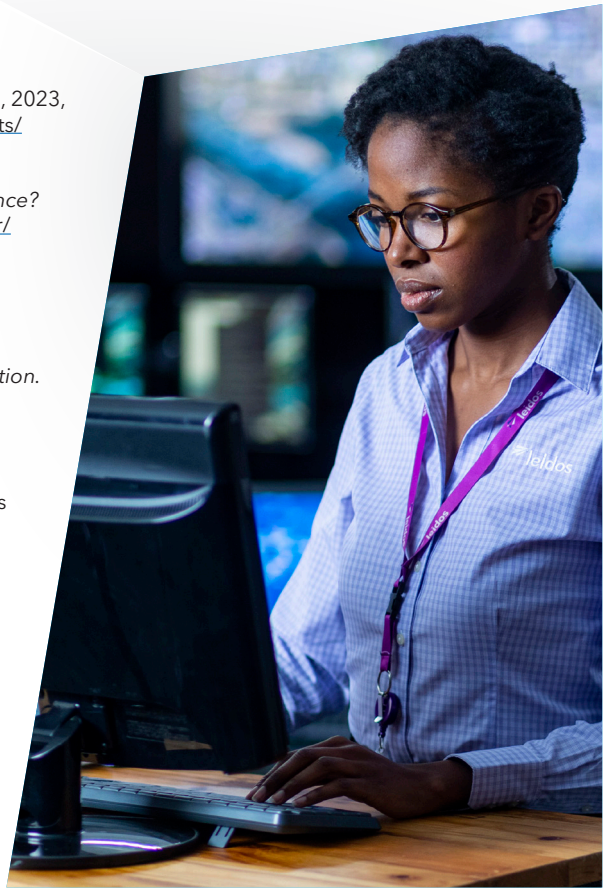
An organization must also carefully consider its own readiness for adopting PETs. Though PETs focus on data security and privacy, they have wide-ranging implications for enterprise data architecture, data governance policies, potentially multinational legal requirements, and ethical matters (which greatly affect customer confidence). Moreover, an organization must consider standard technical matters, such as staffing needs (at the development, integration, deployment, and maintenance levels), security analysis and accreditation, and technological strategy around data and AI/ML. These two groups of factors, organizational and technical, must be evaluated jointly to ensure the most effective adoption of PETs. The following questions could help guide a discussion about adoption readiness:

- Has the underlying mechanism been deployed elsewhere in the government?
- Does the PET provider have a proven record of deploying software?
- How mature is the PET?

Conclusion

PETs represent the next step forward in cybersecurity. No longer can we only protect data at rest and in transit. We must also protect data in use, closing the final gap and providing true end-to-end security. In this piece, we highlighted several PETs and motivating use cases, and provided a framework for PET evaluation. Our definitions for key terms and evaluation criteria emphasize a holistic, inclusive view of PETs and recognize that new PETs and new use cases are still emerging. Future work will include more in-depth, rigorous reviews of specific PETs discussed in this piece.

References



1. Office of Research (n.d.). *Privacy and Confidentiality.* UCI. Retrieved June 12, 2023, from https://research.uci.edu/human-research-protections/research-subjects/privacy-and-confidentiality

2. (2019, October 10). *Privacy vs Confidentiality vs Security: What's the Difference?* EdTech. Retrieved June 12, 2022, from https://edtechmagazine.com/higher/article/2019/10/security-privacy-and-confidentiality-whats-difference

3. Copyright Registration Guidance: Works Containing Material Generated by Artificial Intelligence, 88, Fed. Reg. 16190 (March 16, 2023)

4. W.S.J. Staff (2021, July 21). *Inside TikTok's Algorithm: A WSJ Video Investigation.* Wall Street Journal. https://www.wsj.com/articles/tiktok-algorithm-video-investigation-11626877477

5. Richer, A. D., Tucker, E., & Merchant, N. (2023, April 14). *Suspect in military documents leak appears in court as U.S. Reveals case against him.* PBS News Hour. https://www.pbs.org/newshour/politics/billing-records-from-discord-server-helped-fbi-identify-suspect-in-military-docs-leak

6. Hao, K. (2019, April 1). *Hackers trick a Tesla into veering into the wrong lane.* MIT Technology Review. https://www.technologyreview.com/2019/04/01/65915/hackers-trick-teslas-autopilot-into-veering-towards-oncoming-traffic/

7. Narayanan, A., & Shmatikov, V. (2008). *Robust De-anonymization of Large Sparse Datasets.* 2008 IEEE Symposium on Security and Privacy, 111-125. https://doi.org/10.1109/SP.2008.33

8. Gurman, M. (2023, May 1). *Samsung Bans Staff's AI Use After Spotting ChatGPT Data Leak.* Bloomberg. https://www.bloomberg.com/news/articles/2023-05-02/samsung-bans-chatgpt-and-other-generative-ai-use-by-staff-after-leak

9. Ellis, L. (202, March 22). *ChatGPT Can Save You Hours at Work. Why Are Some Companies Banning It?* Wall Street Journal. https://www.wsj.com/articles/despite-office-bans-some-workers-still-want-to-use-chatgpt-778da50e

10. OpenAI (2022, November 30). *Introducing ChatGPT.* Retrieved June 12, 2023, from https://openai.com/blog/chatgpt

11. (2023, February 11). ISO/IEC WD 18033-8: *Information security — Encryption algorithms — Part 8: Fully Homomorphic Encryption.* ISO. Retrieved June 12, 2023, from https://www.iso.org/standard/83139.html

12. Martin Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, Satya Lokam, Daniele Micciancio, Dustin Moody, Travis Morrison, Amit Sahai, & Vinod Vaikuntanathan (2018). *Homomorphic Encryption Security Standard* [White paper]. https://homomorphicencryption.org/standard/

13. U.S. Census Bureau. (2023, March). *Why the Census Bureau Chose Differential Privacy.* U.S. Department of Commerce. Retrieved June 12, 2023, from https://www2.census.gov/library/publications/decennial/2020/census-briefs/c2020br-03.pdf

14. (n.d.). *Intel® Software Guard Extensions* (Intel® SGX). Intel. Retrieved June 12, 2023, from https://www.intel.com/content/www/us/en/architecture-and-technology/software-guard-extensions.html

15. Cammarota, R. (2023, May 31). Intel Labs Continues Focused Research and Standards Efforts to Make FHE Viable. Intel. Retrieved June 12, 2023, from https://community.intel.com/t5/Blogs/Tech-Innovation/Data-Center/Intel-Labs-Continues-Focused-Research-and-Standards-Efforts-to/post/1488532

16. (n.d.). *Strengthen Enclave Trust with Attestation.* Intel. Retrieved June 12, 2023, from https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/attestation-services.html

17. Rohloff, K. (2017, January 1). *Homomorphic Encryption – Making it Real.* DualityTech. Retrieved June 12, 2023, from https://dualitytech.com/blog/homomorphic-encryption-making-it-real/

18. Rohloff, K. (2022, July 7). *Duality Advances Homomorphic Encryption Landscape with OpenFHE.* DualityTech. Retrieved June 12, 2023, from https://dualitytech.com/blog/duality-advances-homomorphic-encryption-landscape-with-openfhe/


leidos