Homomorphic Encryption (HE)

What is Homomorphic Encryption

Homomorphic encryption encrypts the data in a way that computation/learning can happen on the data w/o decrypting the data. Once data is encrypted in this fashion it can be moved to a central location for training/analytics or compute without sacrificing privacy. HE, is expensive to implement requires lot of compute, as of now the GPU (Graphic Processing Unit) is best suited to run the HE algorithm.

Why was it developed and how it works.

In today's data-driven landscape, safeguarding sensitive information is paramount. The shift to cloud computing and storage has transformed data processing, offering unparalleled resource availability and accessibility while reducing workload significantly. This revolution has spurred immense demand for outsourced applications, allowing clients to upload data to the cloud for processing and accessing results conveniently. While this offers substantial benefits, it also raises concerns about exposing sensitive data to third-party service providers.

Traditional encryption methods typically require data to be decrypted for processing, exposing it in its original form. However, homomorphic encryption breaks this norm by enabling computations on encrypted data, delivering encrypted results to users. This groundbreaking approach not only allows processing of encrypted data but also maintains utmost privacy throughout the entire process, ensuring confidentiality while enabling seamless computation. The egress cost associated with data leaving the cloud can we very significant, HE can play a vital role to perform analytic on the data without the egress cost.

Homomorphic encryption enables mathematical operations directly on ciphertexts, generating encrypted results that, upon decryption, align with the outcomes of operations conducted on plaintext

- Enable computing on encrypted data.
- Enables multiple parties to collaborate without sharing data.
- Apply Artificial Intelligence/Machine learning/Data Analytic

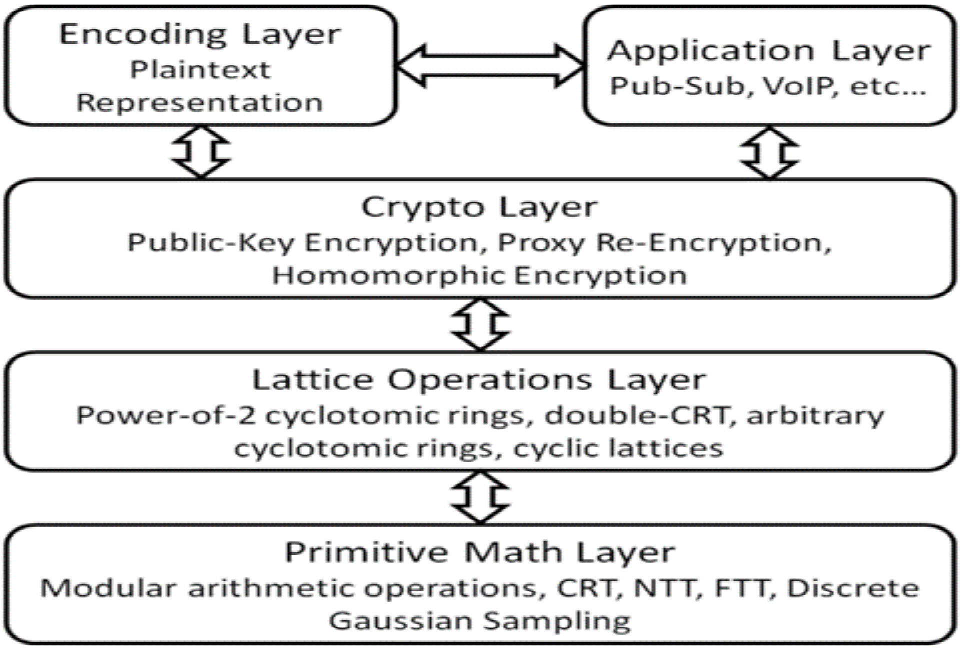
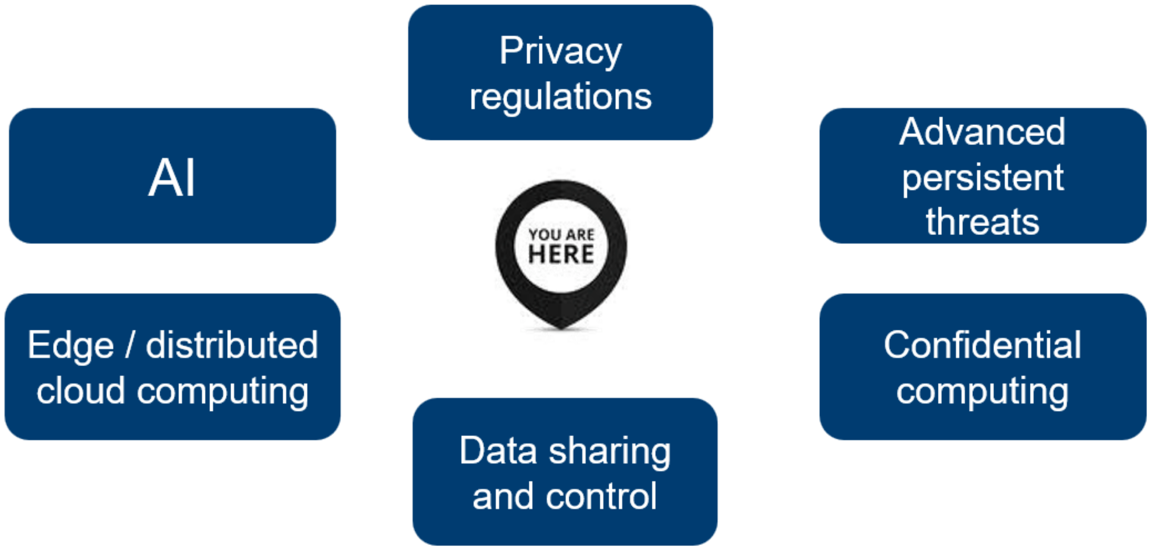
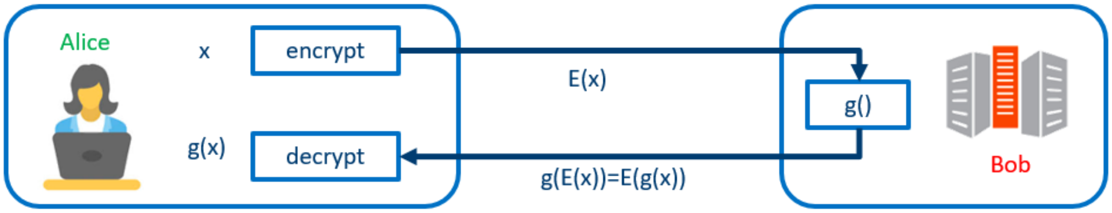
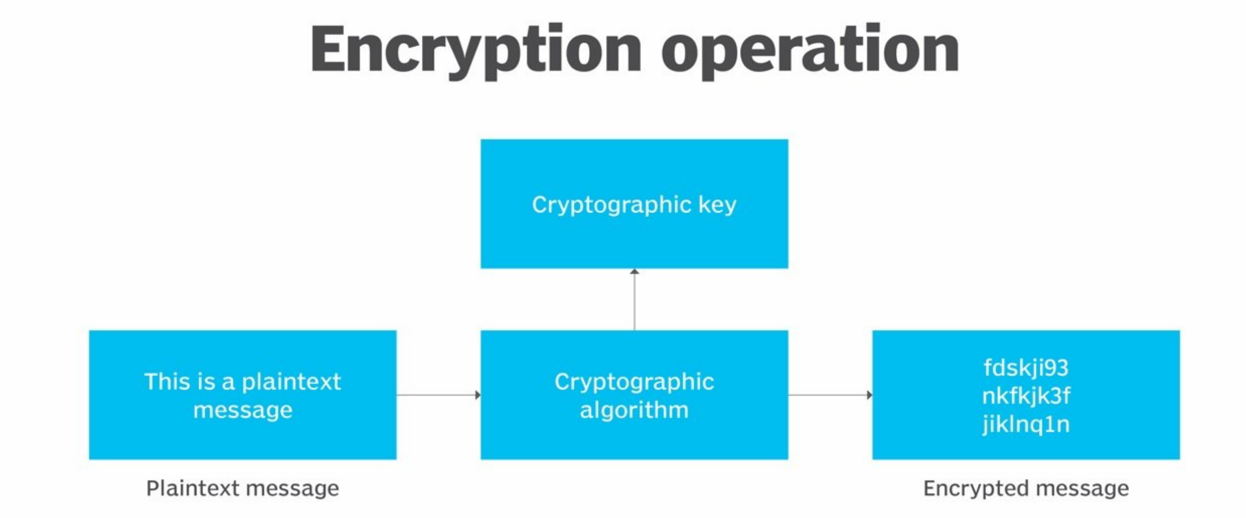

Encoding Layer
Plaintext Representation
Application Layer
Pub-Sub, VoIP, etc...
Crypto Layer
Public-Key Encryption, Proxy Re-Encryption, Homomorphic Encryption
Lattice Operations Layer
Power-of-2 cyclotomic rings, double-CRT, arbitrary cyclotomic rings, cyclic lattices
Primitive Math Layer
Modular arithmetic operations, CRT, NTT, FTT, Discrete Gaussian Sampling

Figure 1: High-level PALISADE architecture





Secure collaboration between Alice and Bob



Types of Homomorphic Algorithms:

Partially Homomorphic Encryption (PHE): Allows either addition or multiplication operations on encrypted data, but not both. Examples include RSA and ElGamal encryption schemes.

Somewhat Homomorphic Encryption (SHE): Enables a limited number of both addition and multiplication operations on encrypted data. The operations are constrained, making it suitable for specific computations. Examples include the Gentry's first level of Fully Homomorphic Encryption (FHE) and the TFHE (Tripartite Fully Homomorphic Encryption).

Fully Homomorphic Encryption (FHE): Allows unlimited iterations of both addition and multiplication operations on encrypted data. This advanced encryption form allows arbitrary computations on encrypted data without decryption.

Use of HE in Artificial Intelligence/Machine and Deep Learning.

Deep Learning, a subset of Machine Learning, employs Deep Neural Networks (DNNs) composed of multiple layers to learn representations of data with varying levels of abstraction. Its transformative impact across diverse domains is attributed to its ability to automatically discover intricate patterns within vast datasets. The core principle of Deep Learning involves training these neural networks on extensive datasets, refining their parameters by minimizing a predefined loss function. This process, called backpropagation, adjusts the model's weights iteratively to enhance its ability to make accurate predictions or classifications. Several breakthroughs in different domains showcase the prowess of Deep Learning:

Image Recognition: Architectures like ResNet, Inception, and others have significantly improved image recognition tasks. These models can identify and classify objects within images with remarkable accuracy, often surpassing human performance on specific datasets.

Natural Language Processing (NLP): Transformers like BERT, GPT (Generative Pre-trained Transformers), and their variants have revolutionized NLP tasks. These models excel in language understanding, sentiment analysis, text generation, machine translation, and more by leveraging massive amounts of text data.

Speech Generation and Recognition: Technologies such as WaveNet and its successors have elevated speech synthesis by producing human-like voices, enabling advancements in speech generation and recognition systems.

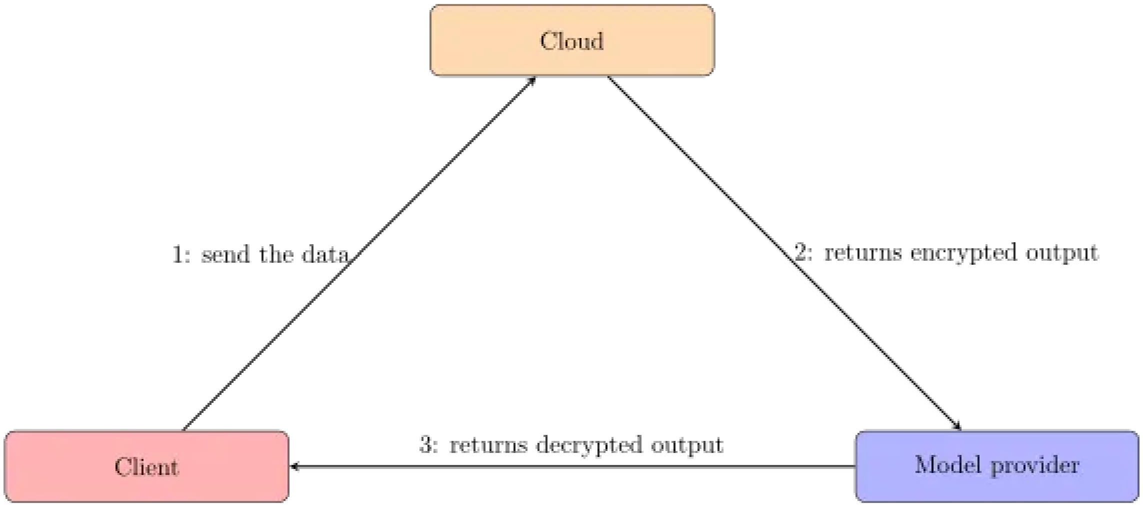
Fully Homomorphic Encryption (FHE) stands out as a potent solution for Privacy-Preserving Machine Learning (PPML) due to its robust security and efficient communication capabilities. FHE is an encryption method where ciphertexts can undergo processing using deep Boolean or arithmetic circuits without revealing the underlying data. This property enables computations directly on encrypted data, preserving privacy throughout the computation process. FHE ensures strong security in the cryptographic sense, allowing for computations on encrypted data without the need to decrypt it, thus safeguarding sensitive information. This approach offers a way to perform computations while maintaining the confidentiality of the data, which is pivotal in addressing the privacy concerns associated with machine learning applications.

Machine Learning as a Service (MLaaS), trust among the model owner, host, and client is paramount. Each party has specific concerns that need to be addressed to ensure the security and confidentiality of data and models.

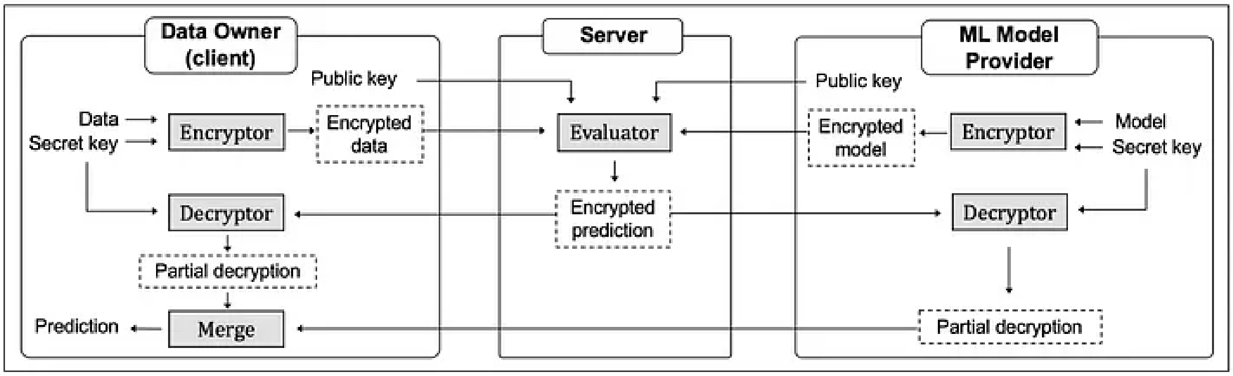
Model Owner (e.g., financial institution): The model owner has invested resources in developing the model and wants to safeguard its intellectual property. They need assurance that the model won't be compromised or stolen.

Host (e.g., Cloud Provider): The host provides the infrastructure and computational power for model inference. They need to ensure the security and privacy of the data and models residing on their servers.

Client: The client seeks the benefits of the model without compromising the privacy of their data. They want assurance that their sensitive information remains secure and isn't exposed or misused.



Three party confidential computation : Financial model hosted on the Cloud.



The figure above provides an overview of the process. The main idea is that both the client and the model owner can encrypt their data before sending it to the server. This one will be able to do computation using those two inputs, then send partial results to each party. The model owner will then remove part of the noise that he introduced using her private key, send it to the client, who will remove the remaining noise and get the output. This model serving is very common practice in financial industry.

Homomorphic Encryption (HE) offers financial institutions a potent solution for securely analyzing sensitive data in third-party or cloud environments. By keeping data encrypted during processing, HE addresses concerns about data breaches, privacy regulations, and data localization requirements. It enables secure computation without the need to decrypt data, ensuring confidentiality and compliance while utilizing external capabilities.

• Current advantages and disadvantages of HE

Advantages:

- The egress cost associated with data leaving the cloud can we very significant, HE can play a vital role to perform analytic on the data without the egress cost. This is significant saving to any business. Furthermore, remember the Ingress data coming into the cloud is free. The Cloud service provider will charge for all the egress cost, considering generative AI application such as Falcon 40B or Llama2 which uses substantial corpus data can leverage HE to compute the token latency in AI deep learning without the egress cost.
- Homomorphic encryption helps minimize the cost associated with data breaches, once the data is stored encrypted using HE it will be required to decrypt it using decryption method which is revers of encryption method.
- Homomorphic encryption stands as a pivotal solution in cloud computing, allowing organizations to securely store encrypted data within a public cloud infrastructure. This setup enables leveraging the cloud provider's analytic services while keeping sensitive information entirely encrypted. It ensures data privacy and security, allowing computations and analyses to be performed on the

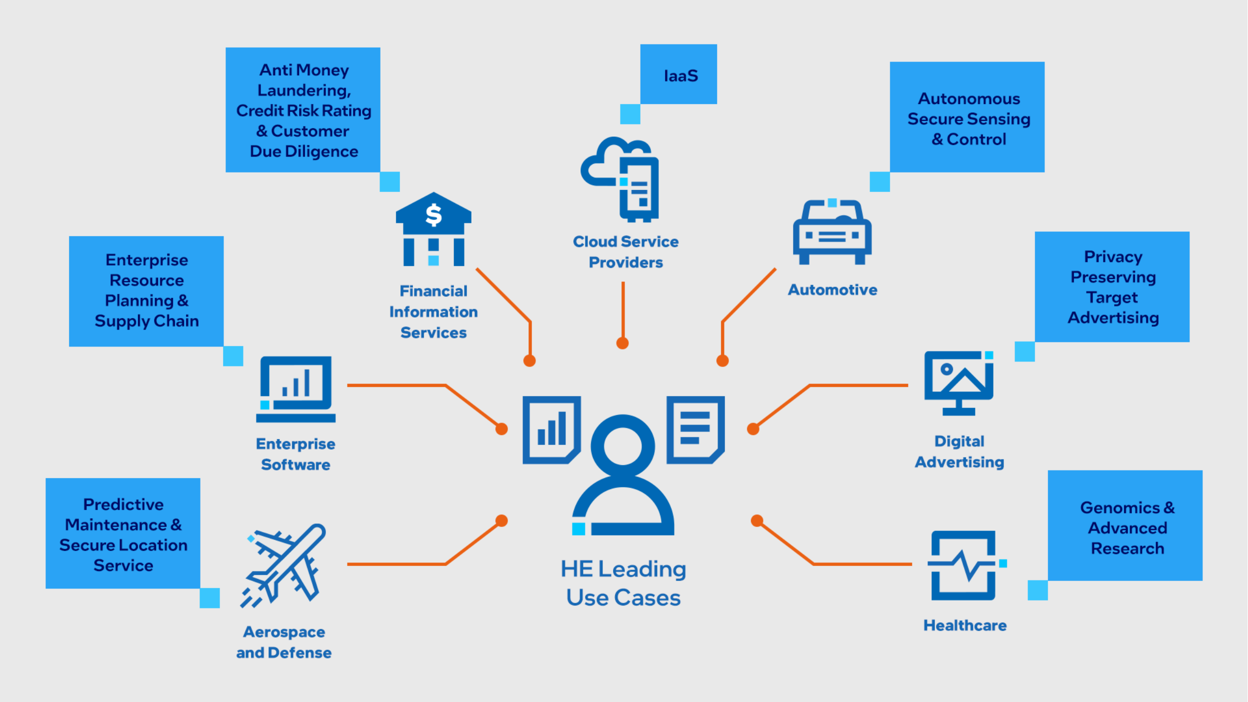
encrypted data without the need for decryption. This capability enables organizations to harness the benefits of cloud-based analytics without compromising on data confidentiality, maintaining control and privacy over their valuable information

  - Homomorphic encryption shields supply chains: Encrypting data for third parties ensures continuous protection. Even if breaches occur, data remains incomprehensible, securing the supply chain's integrity
  - Homomorphic encryption empowers companies like Meta to conduct analytics on user data without accessing the original information. This innovation enables more private targeted advertising, allowing analysis while preserving user data confidentiality

Disadvantages:

- Computational Overhead: Performing computations on encrypted data is significantly more computationally intensive than on plaintext data. This overhead can slow down operations and increase processing times. "Homomorphic encryption's significant drawback lies in its computational intensity, causing slower operations compared to plaintext. Encrypting, decrypting, and processing ciphertexts demands notably more resources and time. This limitation makes it unsuitable for real-time or high-throughput scenarios, potentially incurring substantial costs and latency.
- Key Management Security: Managing keys securely in homomorphic encryption systems is critical. Any compromise in key management could lead to a breach of the encrypted data.
- Computational Complexity: Implementing and working with homomorphic encryption requires specialized knowledge and expertise. Developing efficient algorithms and applications that use this encryption can be complex and challenging
- Performance Trade-offs: There's often a trade-off between security and performance. Increasing security measures in homomorphic encryption can lead to decreased performance or increased computational requirements.
- Resource Intensiveness: Deploying and maintaining systems that utilize homomorphic encryption can require significant computational resources, impacting scalability and cost-effectiveness.
- Usability and Interoperability: The usability and interoperability of homomorphic encryption face challenges due to the absence of common frameworks, libraries, and standards. This hinders ease of understanding, implementation, and compatibility among different schemes and cryptographic tools. Additionally, legal, ethical, and social concerns—like data ownership, consent, accountability, and trust—may arise, posing further obstacles to widespread adoption and implementation

● Use cases in financial services


Anti Money Laundering, Credit Risk Rating & Customer Due Diligence
IaaS
Autonomous Secure Sensing & Control
Cloud Service Providers
Financial Information Services
Automotive
Enterprise Resource Planning & Supply Chain
Privacy Preserving Target Advertising
Enterprise Software
Digital Advertising
Predictive Maintenance & Secure Location Service
Genomics & Advanced Research
HE Leading Use Cases
Aerospace and Defense
Healthcare

Homomorphic Encryption (HE's) applications in financial institutions are diverse, empowering secure and compliant data analysis, collaboration, and processing while preserving confidentiality, thereby addressing critical concerns related to data security and privacy. HE presents several compelling use cases within financial institutions:

- Secure Outsourced Computation: HE enables financial institutions to securely outsource computations to third-party providers or cloud services without exposing sensitive financial data. This allows for secure data analysis while maintaining confidentiality.

- Privacy-Preserving Analytics: It allows for performing computations on encrypted financial data, enabling various analytical tasks without compromising data privacy. For instance, banks can conduct risk assessments or analytics on encrypted customer data without decrypting it, ensuring privacy compliance.

- Secure Collaboration: Financial institutions often collaborate with external entities for joint analysis or auditing purposes. HE allows these collaborations while keeping the shared data encrypted, ensuring confidentiality throughout the collaborative process.

- Fraud Detection and Anomaly Analysis: HE can be applied to encrypted transaction data for fraud detection and anomaly analysis. It enables computations on encrypted data to identify patterns without revealing specific transaction details.

- Compliance with Regulations: Financial regulations often mandate strict privacy measures. HE aids in complying with these regulations by allowing encrypted data to be used for compliance-related calculations without exposing sensitive information.

- Secured Machine Learning on Sensitive Data: Financial institutions can leverage machine learning models on encrypted data using HE. This facilitates training and inference without exposing the raw data, ensuring privacy while benefiting from advanced AI capabilities.

- Secure Client-Side Processing: HE enables financial apps or client-side tools to process sensitive financial information locally without the need to decrypt it. This ensures data remains confidential even during processing.

• How can it help increase collaboration to combat financial crime?

Federated Learning addresses the challenge of acquiring quality training data while respecting privacy and regulatory constraints. This approach enables multiple parties to collaboratively train a shared machine learning model without sharing their raw data. It not only enhances model generalization but also facilitates knowledge extraction from data that is not directly accessible.

However, stronger privacy measures are required with any financial institution, advancements beyond standard Federated Learning methods are necessary.

Secure Aggregation: Standard Federated Learning involves sending model updates to a central server for aggregation, potentially exposing information about individual updates. Secure aggregation techniques, like using HE protocols, allow for aggregating model updates in a way that conceals individual contributions, ensuring stronger privacy.

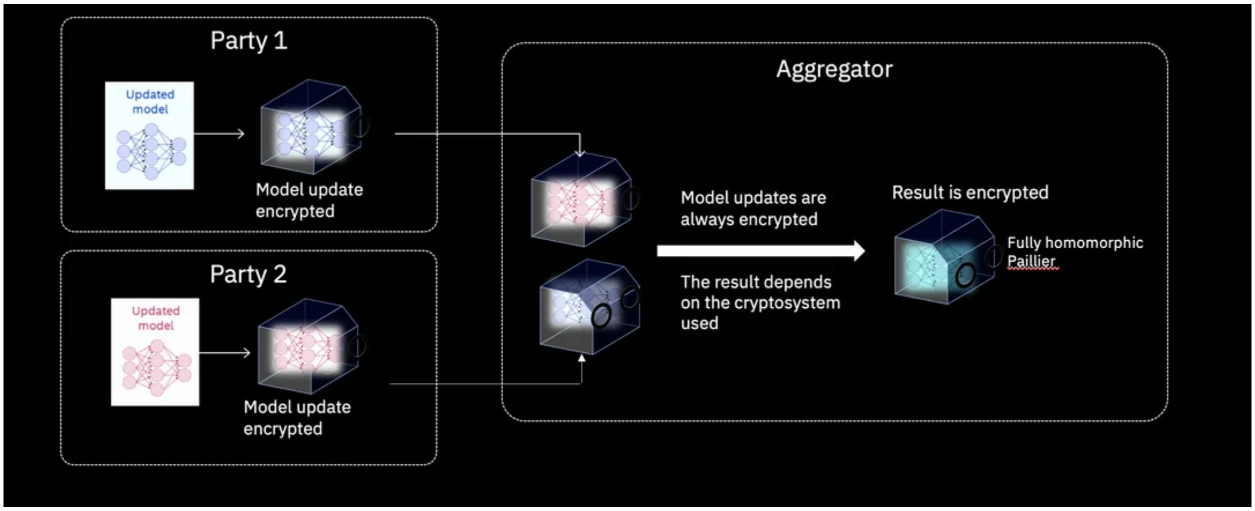
Differential Privacy: Incorporating differential privacy techniques into Federated Learning helps prevent the leakage of sensitive information. It adds noise or perturbation to the model updates before aggregation, preserving privacy without compromising the utility of the model.

Homomorphic Encryption: Integrating Homomorphic Encryption with Federated Learning allows computations to be performed on encrypted model updates without decrypting them. This ensures that sensitive data remains encrypted throughout the learning process, preserving privacy.

Local Model Updates: Instead of sending raw updates to a central server, local model updates can be aggregated on-device or within each participating entity's environment. Only the aggregated model update is shared, reducing the risk of exposing individual data.

Employing these advanced methods ensures stronger privacy protection while still enabling collaborative model training in Federated Learning setups. These approaches play a crucial role in preserving data privacy and meeting stringent privacy or regulatory requirements in various domains.

In federated environments with strict privacy requirements or regulatory constraints, additional protection mechanisms are crucial to prevent inference of private data. Transmitting model updates in plaintext can risk exposing sensitive information to potential adversaries. Fully Homomorphic Encryption (FHE) mitigates this risk by concealing the final model from the aggregator. It only reveals the aggregated version to involved parties. FHE is a cryptographic system enabling computations on encrypted data without decryption. It allows computation of functions over encrypted inputs, yielding results in encrypted form. This ensures sensitive data remains encrypted throughout computations, preserving confidentiality and meeting stringent privacy demands.



In Federated Learning with Fully Homomorphic Encryption (FHE), parties agree on a shared private key. Using this key, they encrypt their model updates before sending them to a central server for aggregation. The encryption ensures data remains secure throughout transmission and aggregation, allowing collaborative model training without exposing sensitive information.

Ref:

https://homomorphicencryption.org/introduction/

https://arxiv.org/pdf/2301.07041.pdf

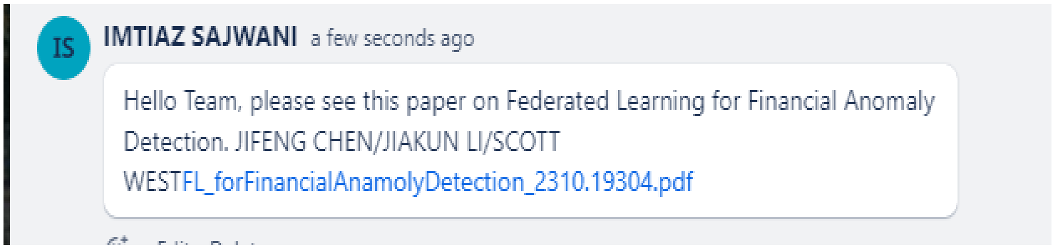
https://www.future-of-computing.com/zama-shaping-the-future-of-homomorphic-encryption-for-machine-learning/ (AI/ML)

https://eprint.iacr.org/2022/915.pdf (opne source FHE library)

Bank of America:
https://www.retailbankerinternational.com/data-insights/bank-of-america-files-patent-for-homomorphic-encryption-based-testing-system-for-application-testing/?cf-view

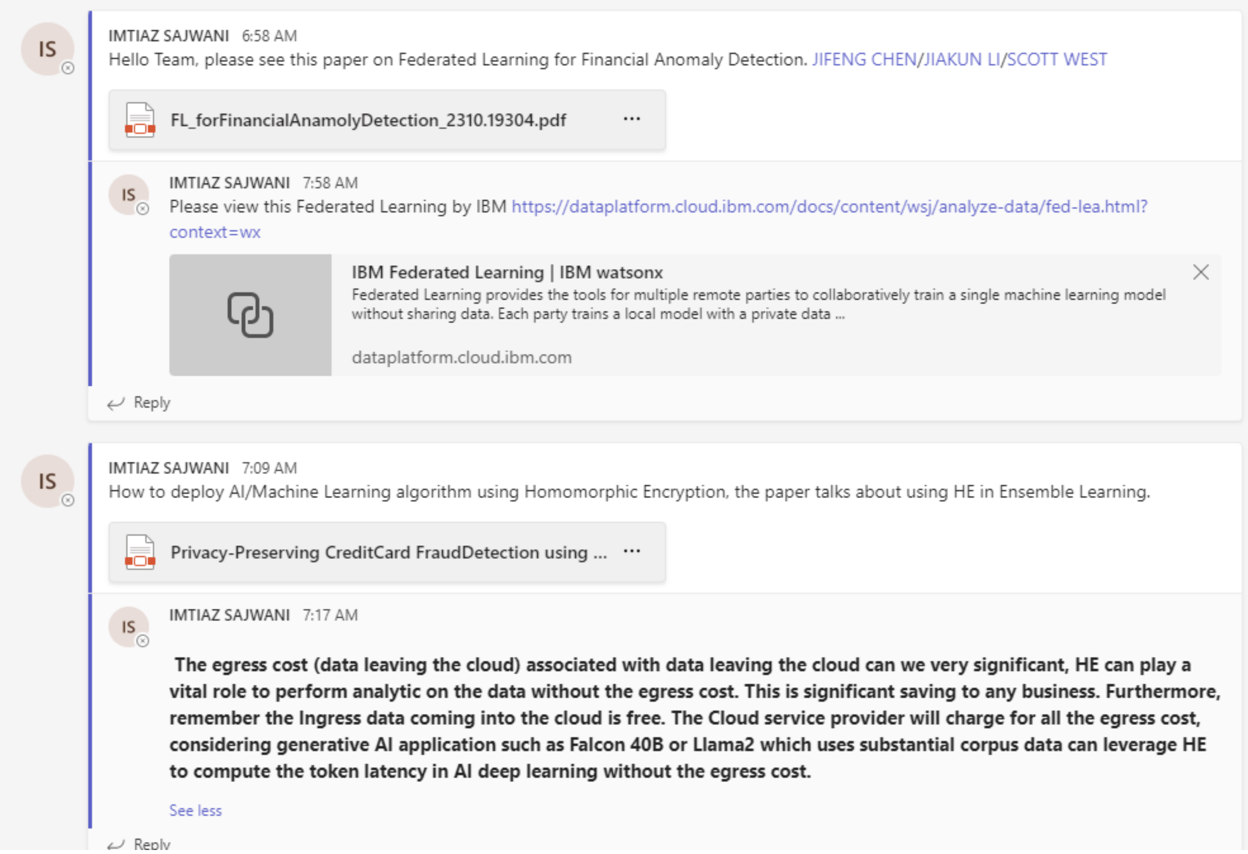
https://www.microsoft.com/en-us/research/project/microsoft-seal/

https://arxiv.org/pdf/2106.07229.pdf



IMTIAZ SAJWANI a few seconds ago

Hello Team, please see this paper on Federated Learning for Financial Anomaly Detection. JIFENG CHEN/JIAKUN LI/SCOTT WESTFL_forFinancialAnamolyDetection_2310.19304.pdf



IMTIAZ SAJWANI 6:58 AM

Hello Team, please see this paper on Federated Learning for Financial Anomaly Detection. JIFENG CHEN/JIAKUN LI/SCOTT WEST

FL_forFinancialAnamolyDetection_2310.19304.pdf

IMTIAZ SAJWANI 7:58 AM

Please view this Federated Learning by IBM https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/fed-lea.html?context=wx

IBM Federated Learning | IBM watsonx

Federated Learning provides the tools for multiple remote parties to collaboratively train a single machine learning model without sharing data. Each party trains a local model with a private data ...

dataplatform.cloud.ibm.com

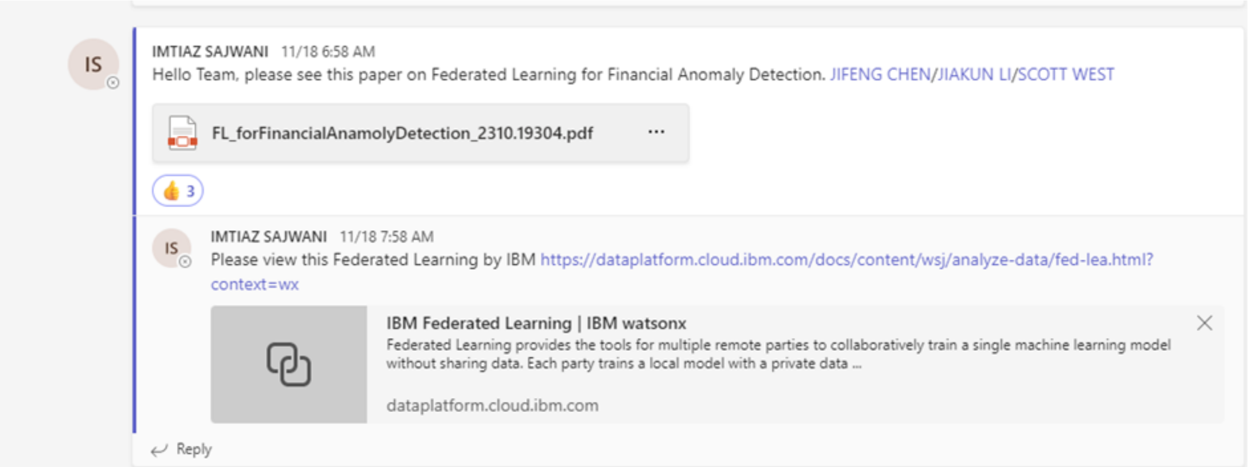
IMTIAZ SAJWANI 7:09 AM

How to deploy AI/Machine Learning algorithm using Homomorphic Encryption, the paper talks about using HE in Ensemble Learning.

Privacy-Preserving CreditCard FraudDetection using ...

IMTIAZ SAJWANI 7:17 AM

The egress cost (data leaving the cloud) associated with data leaving the cloud can we very significant, HE can play a vital role to perform analytic on the data without the egress cost. This is significant saving to any business. Furthermore, remember the Ingress data coming into the cloud is free. The Cloud service provider will charge for all the egress cost, considering generative AI application such as Falcon 40B or Llama2 which uses substantial corpus data can leverage HE to compute the token latency in AI deep learning without the egress cost.



IMTIAZ SAJWANI 11/18 6:58 AM

Hello Team, please see this paper on Federated Learning for Financial Anomaly Detection. JIFENG CHEN/JIAKUN LI/SCOTT WEST

FL_forFinancialAnamolyDetection_2310.19304.pdf

👍 3

IMTIAZ SAJWANI 11/18 7:58 AM

Please view this Federated Learning by IBM https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/fed-lea.html?context=wx

IBM Federated Learning | IBM watsonx

Federated Learning provides the tools for multiple remote parties to collaboratively train a single machine learning model without sharing data. Each party trains a local model with a private data ...

dataplatform.cloud.ibm.com