Proposal: Privacy Computing with Trusted Execution Environments

Dahui Li
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, China
lidh2021@mail.sustech.edu.cn

Sicheng Zhou
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, China
zhousc2021@mail.sustech.edu.cn

I. Problem Statement

As data becomes a crucial asset in decision-making, organizations—both public and private—are increasingly interested in leveraging large datasets to train machine learning models and gain valuable insights. In particular, government agencies are often custodians of vast amounts of sensitive, privacy-critical data, such as healthcare records, demographic information, and social security data. While this data can be used to improve public services and drive innovation, concerns about privacy violations and data misuse create significant barriers to its utilization by external entities.

Businesses and research institutions are eager to collaborate with government agencies to use this data for various purposes, such as training models for healthcare diagnostics, public policy analysis, and market predictions. However, the sensitive nature of the data poses challenges in terms of privacy and regulatory compliance, such as adherence to the General Data Protection Regulation (GDPR) [1] or Health Insurance Portability and Accountability Act (HIPAA) [2]. These regulations require strict safeguards to ensure data privacy and confidentiality throughout the processing pipeline.

Traditional approaches, such as anonymization or differential privacy, have limitations in protecting data during computation, especially when machine learning models require detailed data granularity. This presents a need for innovative solutions that allow external entities to use the data without compromising privacy.

II. Related Work

A. Confidential Computing Application Framework

The typical confidential computing application framework mainly involves four participant roles: platform owner, data owner, code owner, and consumer [3]. **Data owner** is responsible for storing, managing, and controlling large amounts of sensitive data, and has absolute ownership over the data. **Code owner** creates programs, algorithms, models, and other code for accessing, computing, and analyzing. **Platform owner** constructs a cloud environment and establishes a confidential computing platform to provide confidential computing functions and services to external entities. **Consumer** has actual demand for the confidential computing platform and services. They rely on data provided by data owners and code provided by code authors and use the management functions and software provided by platform owners to complete their confidential computing needs.

B. Privacy-Preserving ML with crypto-based methods

Fully Homomorphic Encryption (FHE) is an advanced encryption method that enables a non-trustworthy third party to perform computations on encrypted data without ever revealing the underlying confidential information. Lee et.al [4] explores the promising potential of FHE as a tool for enhancing privacy in machine learning applications. Zama [5] develops an open source privacy-preserving machine learning (PPML) set of tools that automatically turn machine learning models into their homomorphic equivalents. Functional Encryption(FE) is another cryptographic methods which enables data owners to grant third-party access to perform specified computations without disclosing their inputs. Different from FHE, it also provides computation results in plaintext. Panzade et.al [6] provide a survey of PPML works based on FE, and analyze the performance and usability of the available FE libraries and their applications to PPML.

C. Privacy-Preserving ML with TEE

Wang et.al [7] propose a hybrid framework integrating SGX and HE, called HT2ML, to protect user's data and models. Narra et.al [8] presents Origami, which provides privacy-preserving inference for large deep neural network (DNN) models through a combination of enclave execution, cryptographic blinding, interspersed with accelerator-based computation.

D. Applications

1) BigDL Privacy Preserving Machine Learning: BigDL PPML creates a trusted cluster environment that allows users to run standard big data and AI applications without compromising privacy or security by leveraging Intel SGX and integrating various hardware and software security technologies [9].

2) Confidential Google Kubernetes Engine Nodes: Confidential GKE Nodes are a security feature offered by Google Cloud that enhances data protection for containerized workloads running on Google Kubernetes Engine [10] [11]. Built on top of Compute Engine Confidential VMs, this technology leverages AMD Secure Encrypted Virtualization (SEV) [12] to encrypt data-in-use within the memory of nodes and workloads.

3) Scalable Open Financial Architecture Enclave: SOFAEnclave [13] is a confidential computing middleware developed by Ant Financial as part of their SOFAStack financial-grade distributed architecture [14]. It aims to address key challenges in confidential computing by improving usability and enabling cluster deployment. SOFAEnclave consists of three main components: Occlum, a memory-safe LibOS for Intel SGX enclaves that allows unmodified applications to run securely; KubeTEE, a framework for deploying and managing confidential computing workloads on Kubernetes; and security testing and analysis tools.

4) WeBank Data Privacy Rights: WeDPR [15] is a suite of privacy-preserving solutions developed by the blockchain team at WeBank, designed to address privacy concerns in digital business processes. It provides efficient, scenario-based frameworks for privacy protection, including encrypted ledgers, multi-party encrypted decision-making, ranking, computation, and selective ciphertext disclosure. These solutions are applicable to various fields such as finance, healthcare, auditing, and digital identity.

5) Federated AI Technology Enabler: FATE [16] is an industrial-grade open-source framework that facilitates secure data collaboration for enterprises through homomorphic encryption and multi-party computation (MPC), developed by FedAI. It supports various federated learning scenarios with algorithms like logistic regression, tree-based methods, deep learning, and transfer learning.

III. Proposed New Solution

We propose a detailed third-party platform that enables companies to train ML models on sensitive government datasets using Trusted Execution Environments (TEEs). We will simulate the entire workflow, encompassing four key stakeholders: data owners, code owners, platform owners, and consumers. Data owners securely upload their sensitive datasets to the platform, where they remain encrypted and accessible only within the TEE. Code owners contribute their proprietary algorithms and model architectures, which are executed within the secure enclave, ensuring intellectual property protection. The platform owner manages the infrastructure, orchestrating the secure computation environment and facilitating interactions between parties. Consumers, who may be researchers or businesses, can request specific analyses or model training without directly accessing the underlying data or algorithms.

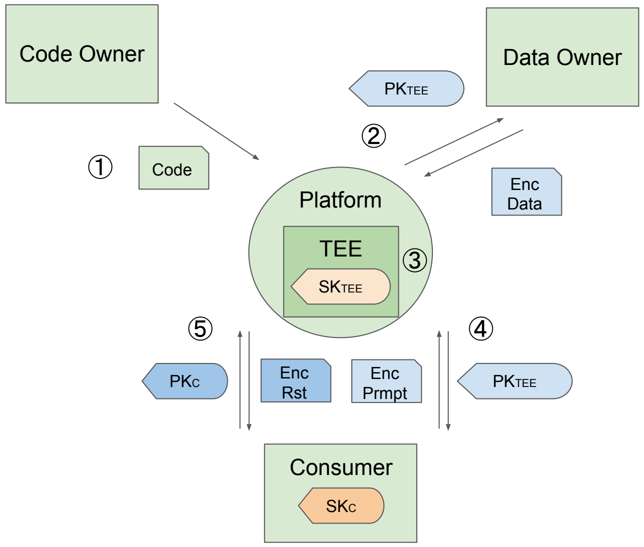

Fig. 1: The proposed workflow. PK: Public Key. SK: Private Key. 1. The *code owner* sends the code to the *platform*; 2. The *platform* sends the PK_{TEE} to the *data owner* and the *data owner* sends back the encrypted data which can only be decrypted by the *platform*; 3. The *platform* trains the model using the code and the data; 4. The *platform* sends PK_{TEE} to the *consumer* who uses this key to encrypt the prompt and then sends it back; 5. The *consumer* sends the PK_C to the *platform* which uses this key to encrypt the inference result and then sends it back.

Our platform will utilize remote attestation to verify the integrity of the TEE and the code running within it, establishing trust among all parties. We will implement secure multi-party computation protocols to enable collaborative model training while maintaining data confidentiality. Additionally, the platform will incorporate differential privacy techniques to further protect against potential inference attacks on the trained models. By simulating this end-to-end workflow, we aim to demonstrate the feasibility and effectiveness of our solution in enabling privacy-preserving ML collaborations.

IV. Evaluation plan

We will compare our platform against notable existing applications introduced in related works. Our evaluation will focus on several key metrics that are commonly used to assess the performance and security of privacy-preserving ML platforms.

For performance metrics, we will measure the computation time taken to train models of varying complexities, comparing our platform with non-TEE alternatives and other TEE-based solutions. We'll evaluate how well the platform scales with increasing data sizes and number of participants. For security and privacy metrics, We will assess the level of protection provided to sensitive data using standard cryptographic measures. The reliability and speed of the remote attestation process will be measured.

V. Timeline

Week05	Complete project proposal
Week06	Design detailed workflow
Week07	Design detailed workflow
Week08	Get familiar with TEE
Week09	Build up the TEE environment
Week10	Build up the TEE environment
Week11	Coding for the workflow
Week12	Coding for the workflow
Week13	Test the performance and analyse the security
Week14	Test the performance and analyse the security
Week15	Write report and prepare presentation
Week16	Submit project report and make project presentation

References

[1] European Commission. General data protection regulation, 2016. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679.

[2] U.S. Department of Health and Human Services. Health insurance portability and accountability act, 1991. https://www.hhs.gov/hipaa/index.html.

[3] Dengguo Feng, Yu Qin, Wei Feng, Wei Li, Ketong Shang, and Hongzhan Ma. Survey of research on confidential computing. *IET Communications*, 18(9):535–556, 2024.

[4] Joon-Woo Lee, Hyungchul Kang, Yongwoo Lee, Woosuk Choi, Jieun Eom, Maxim Deryabin, Eunsang Lee, Junghyun Lee, Donghoon Yoo, Young-Sik Kim, and Jong-Seon No. Privacy-preserving machine learning with fully homomorphic encryption for deep neural network. *IEEE Access*, 10:30039–30054, 2022.

[5] Zama. Concrete ML: a privacy-preserving machine learning library using fully homomorphic encryption for data scientists, 2022. https://github.com/zama-ai/concrete-ml.

[6] Prajwal Panzade, Daniel Takabi, and Zhipeng Cai. Privacy-preserving machine learning using functional encryption: Opportunities and challenges. *IEEE Internet of Things Journal*, 11(5):7436–7446, 2024.

[7] Qifan Wang, Lei Zhou, Jianli Bai, Yun Sing Koh, Shujie Cui, and Giovanni Russello. Ht2ml: An efficient hybrid framework for privacy-preserving machine learning using he and tee. *Computers & Security*, 135:103509, 2023.

[8] Krishna Giri Narra, Zhifeng Lin, Yongqin Wang, Keshav Balasubramaniam, and Murali Annavaram. Privacy-preserving inference in machine learning services using trusted execution environments, 2019.

[9] Intel. Bigdl privacy preserving machine learning (ppml) user guide, 2024. https://bigdl.readthedocs.io/en/latest/doc/PPML/Overview/ppml.html.

[10] Google. Google kubernetes engine (gke), 2024. https://cloud.google.com/kubernetes-engine/?hl=en.

[11] Google. Encrypt workload data in-use with confidential google kubernetes engine nodes, 2024. https://cloud.google.com/kubernetes-engine/docs/how-to/confidential-gke-nodes.

[12] AMD. Amd secure encrypted virtualization (sev), 2024. https://www.amd.com/en/developer/sev.html.

[13] Ant Financial. Sofaenclave github, 2024. https://github.com/SOFAEnclave.

[14] Ant Financial. Sofaenclave introduction, 2019. https://www.sofastack.tech/blog/sofa-enclave-confidential-computing/.

[15] WeBank. Wedpr github, 2024. https://github.com/WeBankBlockchain/WeDPR-Lab-Core/.

[16] WeBank. Fate github, 2024. https://github.com/FederatedAI/FATE/.