Draft Design of Privacy and Safety Modules

In our project, we have embraced a strategy where the client outsources the training task to a worker with a GPU. While this approach leverages computational power and speeds up data processing, it also introduces a new challenge - the protection of training data and scripts' privacy. This document details the design of our privacy and safety modules, offering comprehensive measures to address these challenges.

1. Data Privacy

1.1 Data Encryption

We will use end-to-end encryption for data in transit and at rest, ensuring unauthorized parties cannot access sensitive data. We will employ strong encryption algorithms like AES-256.

1.2 Secure Multi-Party Computation (SMPC)

SMPC enables multiple parties to jointly compute a function over their inputs while keeping those inputs private. The worker with the GPU will perform computations without ever seeing the raw data.

1.3 Differential Privacy

Differential privacy, which introduces statistical noise into the output of data queries, will be integrated. This offers data privacy while still allowing for valuable analytics.

1.4 Federated Learning

Federated learning allows model training on edge devices using decentralized data. In this way, raw data never leaves the client's device, reducing data exposure risks.

1.5 Fully Homomorphic Encryption (FHE)

Fully Homomorphic Encryption allows for computations to be performed on encrypted data without decrypting it first. This means that the worker can train models on the data without actually seeing the data in its raw, unencrypted form. FHE preserves privacy by ensuring that no one, including the worker, can access the actual data, but can still perform meaningful computations on it.

2. Compilation Safety

The following features will ensure the safety and integrity of scripts during the compilation process:

2.1 Code Obfuscation

Code obfuscation techniques will be employed to protect the scripts, making it difficult for anyone to reverse-engineer or understand the code logic without the proper decryption key. Obfuscation involves making changes that conceal the code's purpose or logic, without affecting its functionality. Techniques include control flow obfuscation, data obfuscation, and layout obfuscation.

2.2 Runtime Code Execution

Instead of running precompiled code, runtime code execution can be utilized. This approach compiles code fragments just before execution, reducing the time hackers have to analyze the code, thereby making reverse engineering more difficult.

2.3 White-Box Cryptography

White-Box Cryptography involves the integration of cryptographic keys within the code itself, in such a way that even when an attacker has full access to the code, they still can't access the keys. This method can be utilized to protect crucial parts of the script.

2.4 Code Signing

Code signing is a method that uses a digital signature to verify the code's author and ensure the code has not been altered since it was signed. This can provide a level of trust that the code has not been tampered with and is from a known source.

3. Secure Runtime Environment

The running environment, particularly for the outsourced worker, is critical. To enhance the safety and allow better monitoring of the worker's operations, we propose the following measures:

3.1 Containerization

Using containers to isolate the worker's workspace prevents potential cross-contamination or unauthorized access to other system parts. This technique contributes to a secure and controlled runtime environment.

3.2 Monitoring and Auditing Worker's Operations

Comprehensive monitoring and auditing of the worker's operations within the container will be implemented. These measures, including real-time monitoring, audit logging, anomaly detection, access controls, and incident response, will contribute to ensuring a safe and secure operational environment.

4. Future Consideration

Looking ahead, we plan to continuously improve and update our privacy and safety mechanisms. This includes monitoring advancements in privacy-enhancing technologies and integrating them into our system as and when appropriate. We also recognize the importance of training and awareness, and will strive to ensure all users and administrators are educated about the best practices for privacy and data safety.In our project, we have embraced a strategy where the client outsources the training task to a worker with a GPU. While this approach leverages computational power and speeds up data processing, it also introduces a new challenge - the protection of training data and scripts' privacy. This document details the design of our privacy and safety modules, offering comprehensive measures to address these challenges.