Secure KNN Computation on Cloud - Simulation Project

Project Overview

This project simulates secure query generation for cloud storage, based on the research paper "Secure KNN Computation On Cloud" by Tikaram Sanyashi, Nirmal Kumar Boran, and Virendra Singh. The simulation is implemented using Flask-based APIs, with three microservices that mimic the roles of Query Users (QU), Data Owner (DO), and Cloud Service Provider (CSP).

Project Structure

- **Microservices**:

  - **Query Users (QU)**: Code is available in the `Client/QU` folder.
  - **Data Owner (DO)**: Code is available in the `Server/DO` folder.
  - **Cloud Service Provider (CSP)**: Code is available in the `Server/CSP` folder.

- **Data Storage**:

  - `Server/DO/Data`: Contains data in plaintext form.
  - `Server/CSP/Data`: Contains encrypted data, which is initialized by DO on each run.

- **Public Variables**:

  - The `public-data.json` file contains several public variables used by the simulation:

    1. **n**: Number of data vectors (rows in the data matrix).
    2. **d**: Dimension of data vectors (columns in the data matrix).
    3. **c** and **epsilon**: Values used by the algorithm to calculate `eta = d + 1 + c + epsilon`.
    4. **max_value**: Maximum value a data point can achieve.
    5. **k**: Input parameter value for the K-NN algorithm.

How to Start the Application

1. **Start Data Owner (DO)**:

   ```
   python3 -B Server/DO/main.py
   ```

2. **Start Cloud Service Provider (CSP)**:

   ```
   python3 -B Server/CSP/main.py
   ```

3. **Start Query User (QU)**:

   ```
   python3 -B Server/QU/main.py
   ```

4. **Simulate the Process**:

   - Access the "/simulate" API by visiting the following URL in your browser.

     ```
     http://localhost:8000/simulate
     ```

   - Note: The simulation may take up to 10-15 seconds to complete.

● **Important**: The steps above must be followed in sequence to ensure proper functioning.

Flexible Features

● Customizable Configuration:

　　● You can modify the public-data.json file to suit your needs.
　　● If the application is running, restart the services following the steps provided above to apply the changes.

● Simulation Options:

　　● The /simulate API accepts two optional GET parameters:

　　　　● count: Specifies how many times you want to simulate the process. For example, count=3 means three different queries will be run on the same data points. The default value is 1.
　　　　● verify: A boolean parameter (True or False). If set to verify=True, the simulator will also apply the K-NN algorithm on plaintext data and queries, providing output for verification.
　　　　● Example usage:

```
http://localhost:8000/simulate?count=3&verify=True
```

Conclusion

This project provides a simulation of secure KNN computation on the cloud, demonstrating the concepts outlined in the referenced research paper. The flexible structure allows for easy modification of public variables and provides tools for verifying the correctness of the simulated secure KNN queries.