Final Project: FHE-secured CAPTCHA Solver

蕭凱鴻 B13902046, 蔡兆豐 B13902110

1. 目標

模擬一個場景，使用者的 CAPTCHA 輸入會加密，伺服器對加密輸入與正確答案進行匹配驗證，不需看到明文。挑戰：如何在 FHE 下做字元等值比較。

2. 實作架構

我們全部都使用 python 實作，搭配 zama 官方提供的 concrete API 來達到 FHE 加密。程式分為 4 個檔案：`app.py`、`client.py`、`server.py`、`service.py`。

啟動 `python app.py` 以後會聯絡 `server.py`。server 會生成隨機的 6 個字元字串並編譯好用來加密比較兩個字串的 circuit，把 circuit 的計算部份和加密後的正確答案交給 `service.py`。然後 app 會在 `127.0.0.1:5000` 啟動 flask 框架的網頁，顯示 captcha 圖片給使用者。

使用者輸入完以後，會由 `client.py` 加密完把東西傳給 `service.py` 做運算，和正確答案做逐字比較，要每個字元的 FHE 加密都和答案一樣才算成功。比較完成以後會把結果回傳給 client 和 server，就能進到認證成功或失敗的頁面。

3. 結果

我們有提供一個簡單的網頁介面，使用者可以在上面看到 CAPTCHA 圖片，輸入答案後會顯示認證成功或失敗。如果成功的話就會連接到成功的接面並聯接到新的頁面以供後續使用，如果失敗的話就會顯示錯誤訊息並有重新輸入的按鈕。

4. 一些問題

關於實作上將 captcha 的產生及明文儲存放在 server 端，而不是直接由 service 端產生字串並向 client 驗證。由於 zama concrete 並沒有支援單獨傳送 public key 的功能，為了避免密鑰外洩，只能將鑰匙統一交給 server 端管理，service 只向 server 必要的加密文本。如果要達成更自然的寫法，將 public key 單獨傳送給 service 端讓其可以使用的話，只能用由 rust 寫的 TFHE-rs 套件。

5. Github

`https://github.com/GrandTiger1729/BDA-FHE.git`