組名 小組2

組員名單(姓名+學號)

1.簡廷瑋 112550193

2.梁恩齊 110550096

題目方向

simulation CSE with AES and 2FA

動機與背景

現今網路安全的大環境下，資料隱私和使用者資訊保護議題重要性逐漸上升。隨著隱私權概念的提升，使用者開始有除了伺服器端加密外的客戶端加密需求。客戶端加密確保資料以密文形式儲存在伺服器上，以此保障隱私在服務供應商資料洩漏或不法取用的情況下不受侵犯。

AES 為Advanced Encryption Standard(先進加密標準)，一種對稱式加密演算法，應用廣泛，安全性相當高。2FA 則是雙重認證，要求使用者有第二種方式來解鎖，避免單一密碼洩漏或遭破解時資料受到竊取，這裡我會實作TOTP (Time-based One-Time Password)。TOTP 會使用 secret key 與現在時間產生hash值並生成驗證碼，時間限制下安全性得以保障。

問題定義與目標

Client-Side Encryption(客戶端加密)的精神在於伺服器僅儲存加密後的數據，所以資料會先加密，載到本地端後要另外解密，為資料提供多一層保護。資料明文會經過AES加密後成為密文，但若要解密則需要透過2FA驗證才能解密。AES保障了密文不會隨便被破譯，2FA則是保護解密程式不被外人使用。

預期成果

一個能使用的Client-Side Encryption，由兩部分組成:

1. encry.py
2. decry2fa.py

預期能處理的檔案預設以壓縮檔(zip)為主，但保留只以txt檔實現的可能。

驗證的部分只需要確定能成功加解密即可，省略上傳雲端和下載的步驟。

技術規劃

預計採用 python 實作 Client-Side Encryption，技術上有兩個主題:

1. AES cipher 預計引用cryptography 函式庫實作。
2. 2FA (TOTP) 預計引用 pyotp 函式庫，生成一個 qrcode (內含secret key)再用手機程式掃碼，輸入的6碼要和decry2fa.py算一樣才解密AES。

流程:

1. 本地加密

   做AES加密得到密文。
2. 上傳雲端

   上傳密文到雲端硬碟。
3. 下載密文

   下載密文到本地端。

4. 雙重認證

    輸6碼的TOTP做雙重認證。

5. 解密獲得明文

    雙重認證通過後進行AES解密得到明文。

架構分佈:

1. encry.py
2. decry2fa.py

參考資料

GitHub Copilot (Version 1.311.0) (輔助程式撰寫, VScode的extension)