國立臺北科技大學機械系專題製作成果四頁精簡報告

智慧系統演算法開發

1、專題類別：□新創型專題　■延續型專題

2、本版是否為最終版？請勾選　■是　□否

3、製作期間：110 學年度第 2 學期至 111 學年度第 1 學期

4、指導老師姓名及職稱：李仕宇副教授

5、指導老師簽名或蓋章處：

6、參與學生班級/姓名/學號：

機械四乙/張喻翔/109300219
機械四乙/賴群展/109300205
機械四乙/林柏帆/109300214
機械四乙/王致鈞/109300222

本實務專題製作依據（請勾選、可複選）：

□國科會計畫
□產學計畫
□業界需求
□參與競賽
■創新構想
□機械專業整合
□學碩一貫學程

繳交日期：中華民國 110 年 10 月 6 日

國立臺北科技大學機械系專題精簡報告

智慧系統演算法開發
Algorithm Development for Intelligent Systems

製作期間：111 年 10 月 22 日至 112 年 9 月 30 日
指導老師：李仕宇副教授
參與學生：
機械四乙/張喻翔/109300219
機械四乙/賴群展/109300205
機械四乙/林柏帆/109300214
機械四乙/王致鈞/109300222

一、中文摘要

本專題致力於深入研究混沌密碼學在資訊安全領域的應用，並評估其在數據保護方面的可行性、優勢和局限性。本研究為開發加密演算法應用於手機通訊軟體並使用雲端伺服加密。我們期望本研究的結果能夠為資訊安全方面提供進一步的貢獻。
關鍵詞：資訊安全、混沌演算法、雲端加密

Abstract

This research project is dedicated to a comprehensive exploration of the application of chaotic cryptography in the field of information security ,and it aims to assess the feasibility, advantages, and limitations of chaotic cryptography in data protection. The study is focused on developing the encryption algorithms for use in mobile communication software with cloud server encryption. We anticipate that the outcomes of this research will contribute significantly to the field of information security.
Keywords: information security、chaos algorithm、Cloud encryption

二、緣由與目的

在當今數位化發展迅速的時代，資料的安全性和保護變得至關重要，這推動了加密技術不斷的發展。[1]混沌加密法，作為一種新興的加密技術，混沌系統可以描述為非線性且複雜的動力學行為, 可以滿足密碼學中對解密防護的概念與隨機性高、難以預測的概念，其高度複雜和無序的特性使其被稱為混沌算法。同時也因其對密鑰的參數靈敏度高而受到廣泛關注。混沌演算法的特點就是靈敏度高，因此只要稍微的改變密鑰值就會發現解密後的結果完全不同，即使解密者已掌握了加密算法，也無法進行解密。混沌密碼學的研究不僅能夠加深我們對其在數據保護方面的理解，還有助於發現潛在的應用領域。

因為大多數的加密軟體常應用在電腦上，所以才想做加密軟體應用在手機上。利用加密演算法結合手機的通訊軟體進行加密與解密的功能，可使用手機將加密後的圖片傳送給對方以及獲得一組金鑰，且對方須使用這組金鑰才能去解開加密的圖片，獲取解開後的清晰圖片。

三、結果與討論

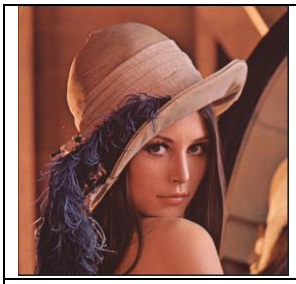
因為此專題需開發一手機應用程式，以及使用一套加解密演算法。所以這裡分兩組進行專題研究，一組為開發手機應用

程式，主要使用 Android Studio、Nodesjs、Google cloud platform；一組為研究演算法加密，主要使用 Matlab。

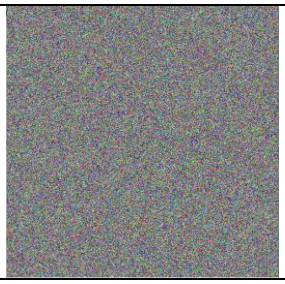
演算法研究的初期，著重在[2]混沌演算法，我們採用論文中 Chen's chaotic system;藉由不同的輸入參數會使得整體相對應的輸出會有所不同，在探討動態系統中無法用單一的資料關係，而必須用整體，連續的資料關係才能加以解釋及預測之行為。

之後我們找到[3]一篇論文能夠能快速進行加密彩色圖片，此論文的主軸加密算法為 3D Modular Chaotic Map(3DMCM)，主要是利用矩陣將原圖片的灰度值從一空間映射至另一空間使得像素位置被改變。而我們使用 3DMCM 類似的加密方式，從原本為從一空間映射至另一空間改為使用矩陣運算將圖片像素位置做交換打亂的動作。

我們修改後的加密方式是先將清晰的彩色圖片拆解成分別為紅、藍、綠三種顏色的灰度圖，接著再以灰度圖的亮度由淺至深化為 0 至 255 的灰度級，再利用混沌演算法將像素值打亂，再利用 3DMCM 將像素的位置做打亂即可得出加密圖片。



圖(一).a 清晰圖片



圖(一).b 加密後的圖片

手機軟體設計的部分，從原本的簡易預設專案中，開發出了許多功能，包括 Fragment、RecyclerView、MediaPlayer、Service、BroadcastReceiver、HTTP 通訊協定等等。剛開始接觸 Android Studio 時，其實對所謂的手機設計概念完全不熟，尤其是對 MVC 的架構。設計一個頁面的過程很艱辛，需要了解 manifest、gradle 是如何建構出整體架構。對元件的屬性、特性也須一一摸熟，接著為了實現選取照片，要透過系統自帶的 file system(為 content 開頭的 URL)呈現照片實體內容，接著會對照片的數據做處理，像是傳送至後端計算、前端呈現、資料庫存放等等，其中還使用 WebSocket 的技術來確保前後端的連線，這邊有牽扯到使用者的連線狀態，如果只要一方中斷，就更新 Database 中的登入狀態，並以 Boolean 來實現。

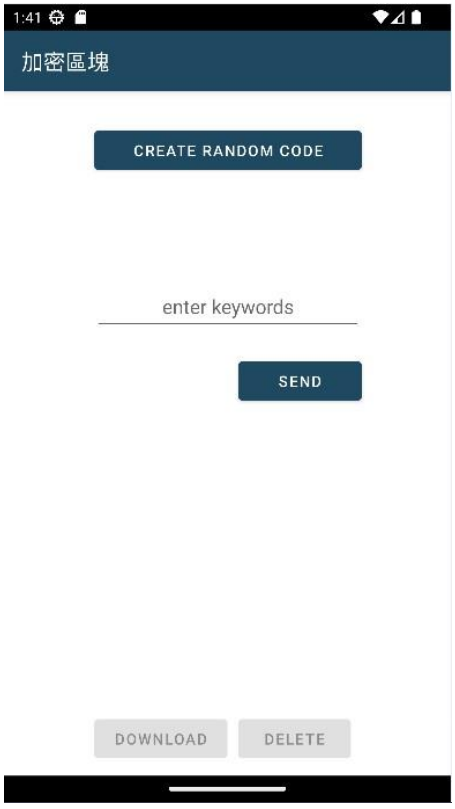
在設計中，遇到困難的點在於要遵守 TCP/IP 的協定來傳送資料，專題中就是以 HTTP 的方式來傳送照片，實現方式以 multipart/form-data 的結構、格式完成並傳送至後端處理，並且等待 Response，這裡還有為了考慮照片處理時間而設定了 Timeout 以避免超時發生錯誤。

後端的部分是利用 Nodejs 的環境來實現並以 Express 框架去撰寫整體架構，同時使用框架下附贈的 Router 去寫 API 的部分給前端呼叫並 Access data。

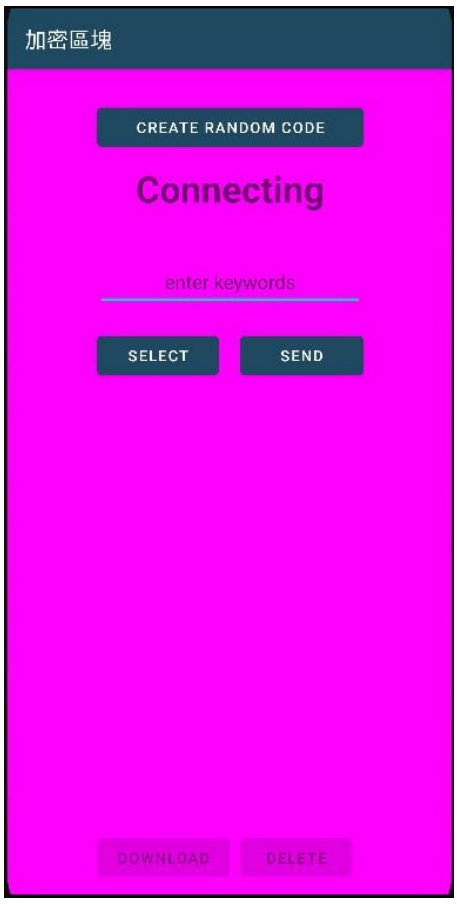
我們將 Matlab 也當作是一個 Server，當後端傳過來需要加密的照片時，利用設計好的加密演算法，再透過 Matlab 平行運算將多張圖片做加密，這邊用到一個技術是 Parallel Pool，可以比喻為有許多待工作的 Worker，一但有任務進來時，就會由系統分配工作給它們，而實際的技術是以 Threads 或 Processes 實現，而這次專題是用 Processes 為基礎，因為這樣 Worker 之間才不會因為太多任務而阻塞另一個正在進行加密的 Process。

我們利用 Google 開發的 GCP 平台去實現 Cloud server，在這次專題中租借了一台主機拿來做資料處理、運算。實現方式是以三個 Server，分別為 Matlab Server 和 Backend Server 以及 MySQL Server，個別為獨立的 Instance，所以並不會因為某一個 Server 阻塞而互相影響，它們擁有獨立的記憶體資源。另外有用到 UUID 的技術來區分出使用者資料以及新增排隊系統來分配資源，如此一來就不會一瞬間給 Server 太大的壓力以至於阻塞整個行程，同時也能依序的平行傳送結果給前端，一舉兩得!

圖(二)手機 app 登入畫面



圖(三)手機 app 功能畫面



圖(四)手機 app 雙方連線成功畫面

四、計畫成果

預期達成目標:完成在手機上將圖片加密後傳送給對方，並且對方可以使用密鑰解開加密圖片。

專題成果:使用 Matlab 實現混沌算法與新論文中演算法做結合得出創新的圖片加密與解密方法。使用 Android Studio 製作手機 APP，透過 APP 可以在傳輸任意資料時套上一層鎖以降低被竊取的風險，當使用者要查看被加密的資料時，只需輸入正確的密鑰即可。其中加密、解密都在雲端伺服器上實現，以減少在手機上 Memory leak 的現象發生，這裡以 Google Cloud Platform 為這次雲服務的使用對象。

五、參考文獻

[1] 百科知識，https://www.jendow.com.tw/wiki/混沌密碼學，(Sep.20.2023)。

[2] Tiegang Gao , Zengqiang Chen , *A new image encryption algorithm based on hyper-chaos* , (2008, January)。

[3] Ali Broumandnia , *Scale Invariant Digital Color Image Encryption Using a 3D Modular Chaotic Map* , (2021,July)