科技部補助

大專學生研究計畫研究成果報告

計　畫
　　：　雲端遊戲系統與安全
名　稱

執行計畫學生：崔元淇
學生計畫編號：MOST 109-2813-C-110-016-E
研 究 期 間 ：109年07月01日至110年02月28日止，計8個月
指 導 教 授 ：范俊逸

處 理 方 式 ：本計畫可公開查詢

執 行 單 位 ：國立中山大學資訊工程學系（所）

中 華 民 國　110年03月25日

目錄

摘要

「雲端遊戲」為透過網際網路將使用者經由搖桿、鍵盤和滑鼠等輸入，傳輸到遊戲服務端，使遊戲軟體的資料和運算皆在雲端伺服器中運作，再將遊戲畫面透過網路傳輸至使用者的電腦上顯示，其優點能讓使用者避免花費過多資源在升級硬體設備，並且不受平台的限制也可運行相同的遊戲，將對使用者的硬體需求最小化。

現今 5G 行動網路正在普及化，其具有比 4G 行動網路更佳的環境，無論是比 4G 行動網路快 10~100 倍的連網速度還是更低的網路能源消耗，都能讓雲端遊戲的體驗更佳。因此期許未來此計畫能夠使用 5G 行動網路，去除網路延遲造成的畫面撕裂。

然而雲端遊戲高度依賴網路的特性，讓其安全性顯得更加重要。透過多重身分驗證或透過防毒軟體及入侵防禦系統進行防護，而要如何在被侵入後，於最短時間內發現並解決問題，或在被入侵前阻絕其發生的可能，則是我們需要研究的重點。

關鍵詞：GamingAnywhere 伺服器、入侵防禦系統、阻斷服務攻擊、資料庫、雜湊鏈。

Abstract

The operation of Cloud Gaming is to transmit user input such as joystick, keyboard, and mouse to the games' server through the internet. After cloud server calculates with the user data and game data, it transmits the game scene to the user's computer via the internet for display. The advantages of cloud gaming are that they allow users to avoid spending too much money on upgrading hardware equipment and can run the same game without being restricted by the platform, minimising the hardware requirements for users.

Nowadays, 5G mobile network is becoming popular. It has a better environment than 4G mobile network, such as 10~100 times faster connection speed or lower network energy consumption than 4G mobile network, which can enable the experience of Cloud Gaming to be better. Thus, it is hoped that this project will be able to use 5G mobile network in the future to reduce screen tearing caused by network delay.

However, Cloud Gaming is highly dependent on the characteristics of the network, so network security is very important. Information security and network security are protected by multi-factor authentication or through antivirus software and intrusion prevention system. Thus, how to find and solve the problem in the shortest time after being intruded, or to prevent the intrusion before being invaded, is the focus of our research.

Keywords: GamingAnywhere server; Intrusion Prevention System; Denial of Service attacks; Database; Hash chain

研究計畫內容

一、前言

雲端遊戲提供一個平台給使用者進行遊玩，其運作方式為在雲端伺服器上執行計算複雜的遊戲，再將經過伺服器算圖後的遊戲場景通過網際網路傳輸到用戶端的設備上，而輸入設備的控制訊號被傳送回雲端伺服器控制遊戲。雲端遊戲平台通常被一個或多個數據中心的雲端伺服器維護。雲端遊戲平台執行電腦遊戲，該平台大致可分為兩個主要的部分：

(i) 負責將玩家的控制指令在遊戲內執行，改變遊戲狀態的遊戲邏輯。

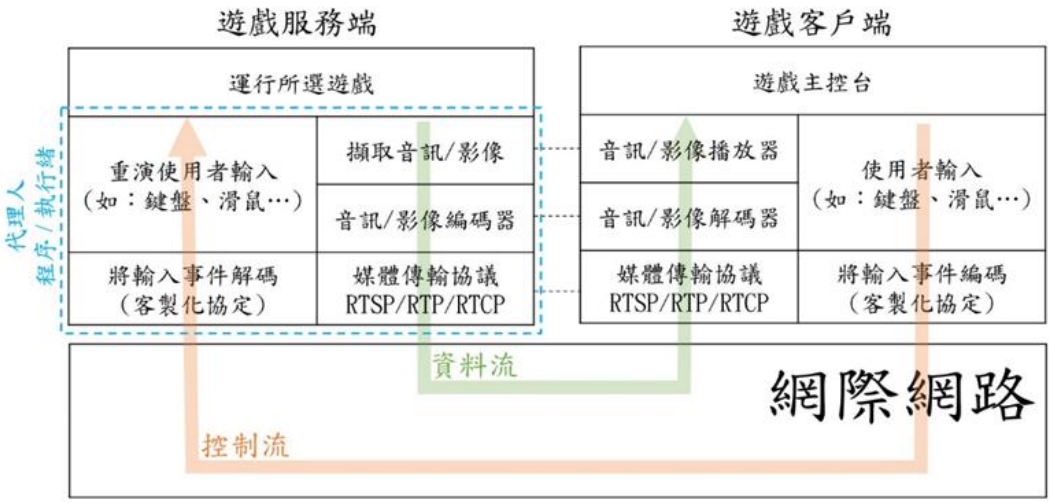
(ii) 生成即時遊戲場景的場景算圖引擎。

玩家指令來自指令直譯器，遊戲場景由影像擷取器擷取為影像，然後由影像編碼器壓縮。指令直譯器、影像擷取器和影像編碼器都為雲端遊戲平台的一部分。雲端遊戲平台將影像封包發送到用戶端，並從用戶端接收使用者對遊戲的輸入。它是簡單的用戶端機器，僅需要兩個低複雜度的零件組合：

(i) 指令接收器：~~，~~它連接到遊戲控制器，例如搖桿、~~，~~鍵盤和滑鼠。

(ii) 影像解碼器：~~，~~可以~~可以是~~使用大量生產的解碼器晶片~~可以~~實現。

圖一展示了雲端遊戲服務的運作方式。



（圖一：雲端遊戲運作方式）

雲端遊戲平台與用戶端之間的溝通是透過 Best-Effort Internet 進行的，這反過來使支持即時反應的電腦遊戲變得頗具挑戰性。歷史上有兩個事件加劇了雲端遊戲市場的推進：第一，主要開發遊戲機的公司——Gaikai 在 2012 年被 Sony 收購，其次是 Sony 的 PlayStation Now（PS Now）和 Nvidia 的 GeForce Now 之間的競爭。

雲端遊戲的巨大普及可能歸因於對於遊戲玩家、遊戲開發商和服務提供商的一些潛在優勢。對於遊戲玩家而言，雲端遊戲使他們能夠：

(i) 隨時隨地訪問他們的遊戲

(ii) 依照需求購買或租借遊戲

(iii) 避免定期升級其硬體。

對於遊戲開發者而言，雲端遊戲使他們能夠：

(i) 專注於一個平台，從而降低了移植和測試成本

(ii) 繞過零售商以獲得更高的利潤率

(iii) 與更多遊戲玩家接觸

(iv) 避免盜版，因為遊戲軟體永遠不會下載到用戶端電腦。

對於服務提供商而言，雲端遊戲：

(i) 帶來了新的商業模式

(ii) 對已經部署的雲端資源提出了更多要求

(iii) 展示與子其他遠距執行應用程式的潛力，因為雲端遊戲實施了最嚴格的要求各種計算和網路資源的限制。

儘管雲端遊戲具有巨大的機遇，但在其充分發揮潛力以吸引更多遊戲玩家之前，遊戲開發商和服務提供商必須解決一些挑戰。我們將最重要的方面總結如下。首先，必須構建雲端遊戲平台和測試平台以進行全面的性能評估。評估包括對服務品質（QoS）指標（例如能耗和網路指標）以及體驗品質（QoE）指標（例如玩家感知體驗）的度量。建立平台和測試平台，設計測試方案以及執行評估需要大量的工作，而分析 QoS 和 QoE 指標之間的複雜相互作用則更加困難。

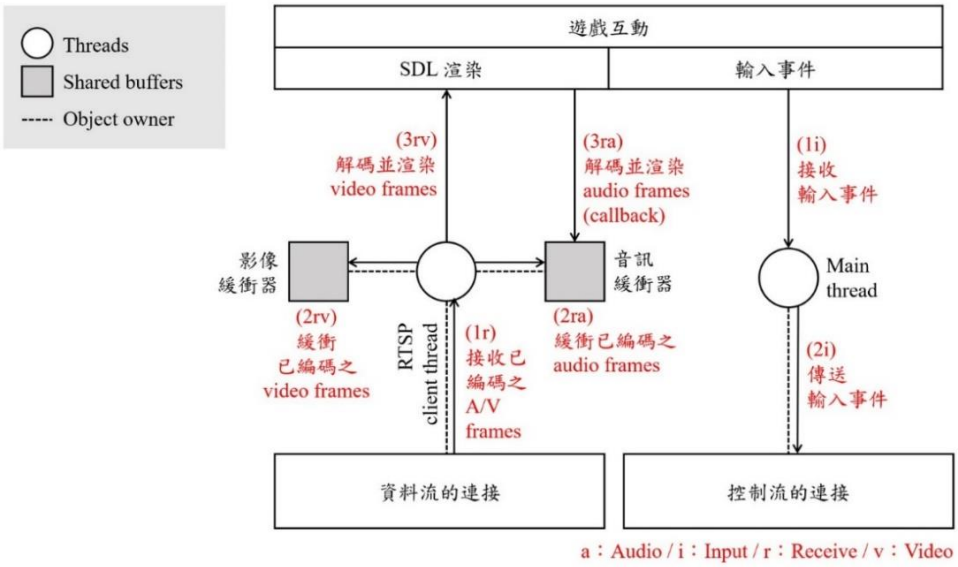
其次，雲端遊戲平台和評估程序使研究團體可以優化各種元件，例如雲端伺服器和通信渠道。更具體地，用於以下方面的優化技術：

(i) 在雲端伺服器上可以實現更好的資源分配和分散式結構。

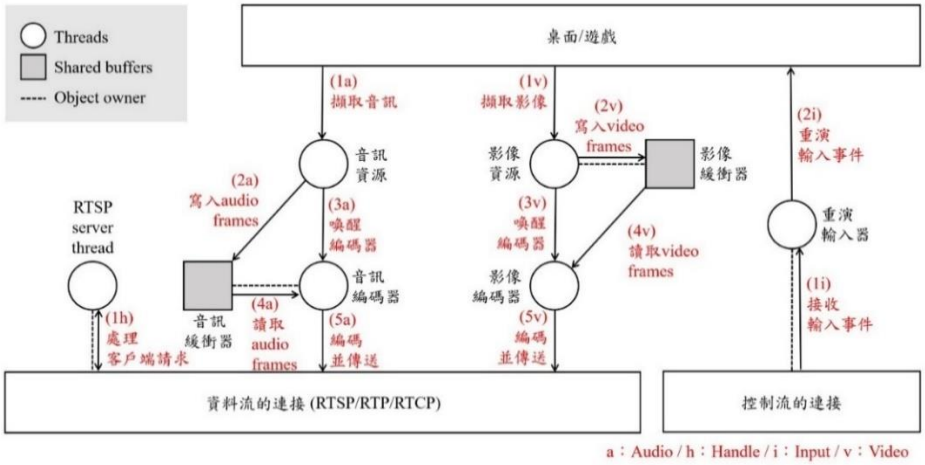
(ii) 在通信通道中可以實現最佳的內容編碼和自適應傳輸。

第三，電腦遊戲具有多種遊戲類型。這些體裁可以根據兩個要素進行分類：視角和主題。視角是玩家觀察遊戲場景的方式。它確定螢幕上算圖影像的變化性。最常見的視角包括第一人稱、第二人稱、第三人稱和全知視角（omnipresent）。第一人稱遊戲採用從遊戲中角色的角度呈現的視角，例如《絕對武力》。第二人稱遊戲是從遊戲中角色的背面渲染的，因此遊戲玩家可以在螢幕上看到這些角色，就像《俠盜獵車手》中一樣。第三人稱遊戲將遊戲者的視線固定在投影到 2D 空間的 3D 場景上。現代的第三人稱遊戲通常採用天空視圖，也稱為上帝視圖。經典的第三人稱遊戲包括《暗黑破壞神》,《紅色警戒》,《FreeStyle》等。全知視角使遊戲玩家可以從不同角度和距離觀看感興趣的區域（RoI）。最近的許多戰爭遊戲，例如《世紀帝國 3》,《絕地要塞 2》和《魔獸爭霸 III》，都屬於此類。遊戲主題確定遊戲玩家如何與遊戲內容互動。常見主題包括射擊、格鬥、體育，基於回合的角色扮演（RPG）,動作角色扮演（ARPG）,基於回合的策略,即時策略（RTS）和管理模擬。儘管視點可能會受到遊戲主題的限制，但是通常可以通過視角和主題來描述遊戲類型，例如第一人稱射擊、第三人稱 ARPG、可觀全局的 RTS 等。其中，節奏快第一人稱射擊遊戲具有最高的場景複雜性，這對雲端遊戲服務提供商而言是最具挑戰性的遊戲。相比之下，基於第三人稱回合的 RPG 遊戲對延遲的敏感度最低，因此更適合雲端遊戲。

至於雲端遊戲系統，現有一開源軟體——GamingAnywhere，其透過模組化設計構建成可跨平台（如：Windows、Linux、Android……）的系統，主要運作為使用者與雲端遊戲服務端皆使用相同區域網路進行資料傳輸與處理，圖二與圖三分別為 GamingAnywhere 的用戶端與服務端的流程圖，其中運用到 SDL 渲染技術為影像與音訊進行渲染，並對其編碼及解碼，為資料流的傳輸進行處理。



（圖二：GamingAnywhere 用戶端）



（圖三：GamingAnywhere 服務端）

二、研究動機與目的

市面上 3A 大作，對電腦配備有一定要求，然而雲端遊戲是個能讓使用者無需高規格硬體便能享受高品質遊戲的作法；但這種高度依賴網路的架構，可能會被有心人士竊取、竄改資料或透過網路攻擊將病毒置入使用者電腦或阻斷其服務，甚至針對該雲端系統內部的漏洞進行攻擊，上述安全問題，對雲端遊戲普及至關重要。因此我們決定在現有的開源雲端遊戲基礎上，架設一個更加安全的雲端遊戲系統，其中包含透過將密碼 hash 過後傳送確保使用者的資訊安全，研究應對網路攻擊的方法並撰寫演算法，以建置一個安全且使用者易使用的雲端遊戲系統。

再者，想在良好的環境下遊玩雲端遊戲，與網路之間的關係密不可分，根據

雲端遊戲需要大量頻寬的特性，在未來也可以嘗試在連線上使用 5G 行動網路，結合其特性——低延遲、高傳輸速度、低網路能源消耗等，將雲端遊戲的遊玩流暢性再進一步的提升，讓使用者感受到比在 4G 行動網路環境下更加優越的體驗。然而，當雲端遊戲利用架設的遊戲系統作為運作的基礎架構，更顯示出網路架構安全的重要性。從最基礎的遊戲體驗，包括連線的穩定性、遊戲操作的流暢程度、各項遊戲數值是否準確的顯示和讀取，到個人資訊的安全性，甚至當遊戲內容需要付費時，相關的帳戶資料或信用卡號碼都可能遭到有心人士的竊取和盜用。以上都是未來雲端遊戲可能會面臨到的難題及困境，如何提供一個安全舒適的遊戲環境也是相當重要的議題。

三、文獻探討

(一) 雲端遊戲系統之選擇

GamingAnywhere 是第一個開源的雲端遊戲系統，它具有可擴展性、可移植性以及可配置性，可使研究者去實現並測試他們的想法，也能使服務端發展雲端運算服務，還能讓玩家自行設置私有的雲端遊戲系統（Huang et al.,2013）。GamingAnywhere 將模擬未來的雲端遊戲運作與即時互動分散式系統。玩家不必再額外升級硬體資源以達到遊戲最低的硬體需求，便能在雲端運行想玩的遊戲。

(二) 雲端遊戲之建構

雲端遊戲運算可以分為四個部分，概觀、平台、優化與商業化。概觀可以分為普通雲端遊戲與特殊雲端遊戲兩個部分，特殊雲端遊戲這部分即是指行動裝置等。平台可以再延伸出 QoS 評估、QoE 評估，藉此來達成客戶滿意程度。優化有兩大方向，可以從服務端的基礎設施著手，亦可從傳輸資料時的資料壓縮等進行優化（Cai et al., 2016）。

透過在虛擬環境 VMware 或 VirtualBox 中架設 GamingAnywhere 系統，在該系統中實施虛擬套件來評估其效能，並測量服務端之 CPU 能量消耗以及觀察如何影響其效能。該論文也建議若在虛擬機上實施更好的排程演算法，則能有更好的表現（Vishnumolakala et al., 2018），因為在虛擬機內之能量損耗能夠透過 energy-aware 排程演算法來調節。

(三) 資料庫與其安全性

由於電子商務和網際網路的普及，網路安全開始受到重視，透過 NIDS Snort 實作入侵偵測系統來分析網路攻擊行為之程序，並將其事件紀錄分級，將來源之 IP 位址標示為可疑、威脅、惡意等三種狀態存放於 IP 狀態資料庫中，再以不同紀錄分級與 IP 位址狀態提供網管人員紀錄資料（劉明輝 et al., 2007），以解決記錄過多的問題。

(四) 探討加密方法

在將數據發送到雲端服務提供商之前，需先對其進行加密。然，當雲端服務提供商要對數據進行處理時。在執行所需的計算之前，用戶端將需要向伺服器提供私密金鑰以解密數據，這可能會影響雲端儲存的數據的機密性（Tebaa et al., 2012）。同態加密方法能夠執行加密數據的操作，而無需將其解密。在這項工作中，我們專注於同態加密方法在雲計算安全性上的應用，尤其是無需加密即可執行加密數據計算的可能性。

(五) 探討雲端遊戲現今威脅

雲端架構中 DDoS 的攻擊與防禦會比傳統的網路安全來的更為險峻。首先，當在雲端伺服器上出現漏洞時，他與單處理器所發生的安全性不同，因為使用者的連線使得每台電腦皆產生此問題，導致該漏洞在整個計算架構中被大量複製，因此成為傳統網路安全威脅的「漏洞放大器」（Neupane et al., 2017）；再者，存在新的 DDoS 攻擊手段，這些手段專門針對虛擬機（VM）以及 CSP 內部網路中的應用程序：多租戶易受攻擊區域中的雲端架構。而且傳統的檢測和反應防禦方式在受到 DDoS 攻擊時，關於保持服務水平協議（SLA）方面的效率很低，原因是因為他們缺乏應對攻擊檢測的速度、CSP 的成本及解決智能攻擊策略的技巧。其中 DDoS 攻擊為雲端環境主要威脅。傳統防禦方法由於效率較低，儲存量大等原因而無法輕鬆應用於雲端安全。

(六) 分析偵測攻擊演算法

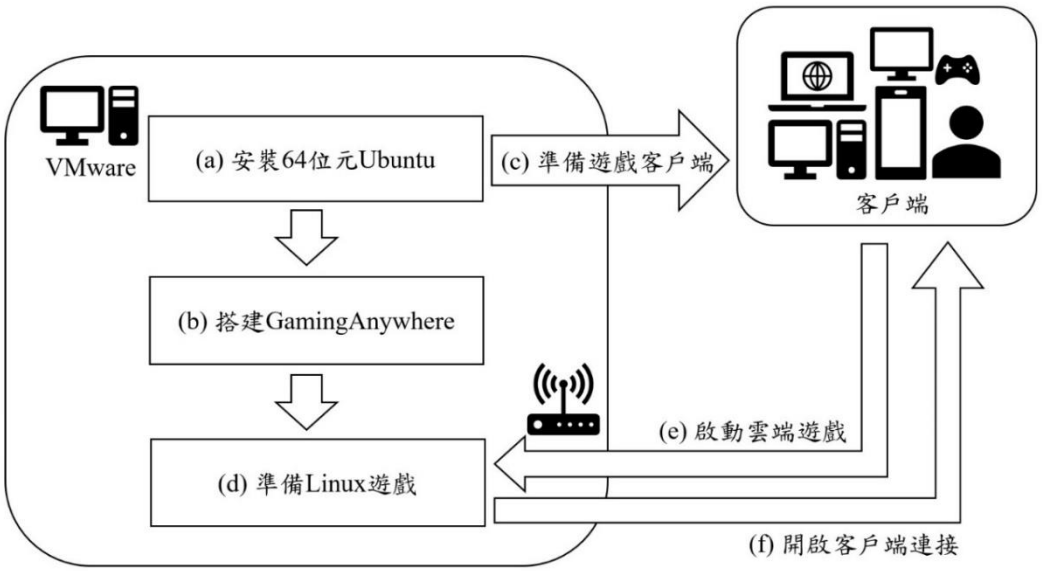
分散式拒絕服務攻擊（DDoS）是雲端環境的主要威脅。傳統防禦方法由於效率較低,儲存量大等原因而無法輕鬆應用於雲端安全。針對這一挑戰,本文針對雲端計算環境研究了一種基於可信度的過濾方法，即 Confidence-Based Filtering method（CBF）。該方法分為兩個週期，即非攻擊週期和攻擊週期。在非攻擊期間收集合法封包，以提取屬性對以產生成可信度值集合的檔案。藉由可信度值集合的檔案的標準，計算在攻擊週期特定封包的分數來評斷是否將其丟棄。經過測試，CBF 具有較高的評分速度，較小的儲存需求和可接受的過濾精準度（Chen et al., 2011），使其適用於雲端環境中的即時過濾。

對於 Web 服務在雲端計算環境中易遭受 SQL injection 的問題。因此提出了一種結合 Dynamic taint analysis 和輸入過濾的 SQL 檢測方法。並將其嵌入雲端環境中，以在雲端架構中實現對 Web 應用程序的保護。首先，該方法通過對 SQL 語句的詞彙規則進行分析來獲得 SQL 關鍵字。然後，它分析 SQL 語句的語法規則以創建決策樹。最後，在 SQL 語法規則建立的模型基礎上遍歷三元樹以檢測攻擊（Wang et al., 2016）。實驗結果表明，該方法是有效可行的。另外，通過添加檢測模組可以提高準確性。

四、研究方法

(一) 架設 GamingAnywhere

在伺服器端虛擬機上架構 GamingAnywhere 雲端系統，圖四為 GamingAnywhere 的搭建順序：



（圖四：GamingAnywhere 搭建流程圖）

(a) 事前準備：在 VMware 中安裝 64 位元 Ubuntu，並將 GamingAnywhere 的 source code 進行解壓縮。

(b) 搭建 GamingAnywhere:安裝其依賴項,再編輯配置文件與編譯,最後將動態連結函式庫裝載入系統變量。

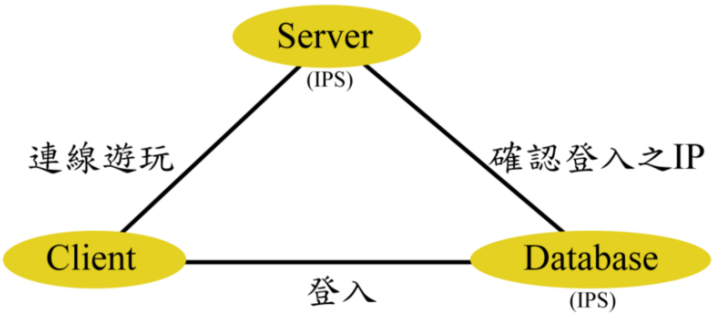
(c) 準備雲端遊戲用戶端。

(d) 準備 Linux 遊戲。

(e) 啟動雲端遊戲：在 Terminal 上輸入運行遊戲指令，並編輯配置文件，開啟 GamingAnywhere 服務端。

(f) 開啟用戶端連接。

(二) Server、Client 與 Database 間的連接，圖五為示意圖



（圖五：Client、Server、Database 三者連接圖）

(1) Client

使用 Qt 提供的 C++ API 撰寫 GUI，提供使用者圖形化介面操作，註冊與登入皆要先與 Database 連線，驗證其為合法使用者後，才能與 Server 連線遊玩遊戲。

(2) Server

使用 C++ 撰寫 Server 端，負責接收 Database 傳來的合法 Client 的 IP address，並在合法 Client 與 Server 連接時，開啟雲端遊戲讓 Client 遊玩，並使用入侵防禦系統（Intrusion Prevention System）預防攻擊。

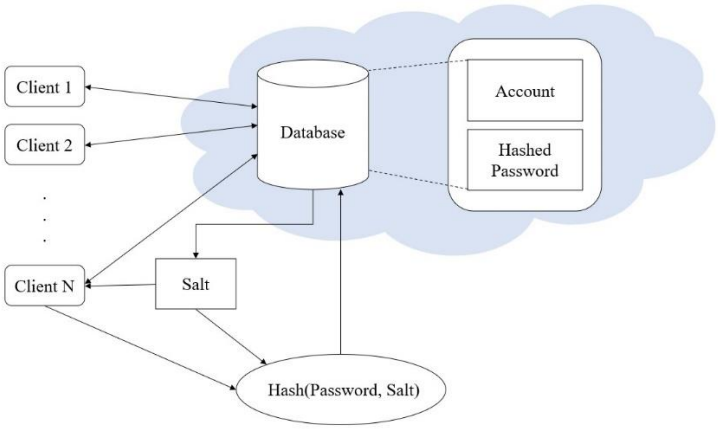
(3) Database

使用 MySQL 提供的 C API 架設一個資料庫存放使用者資訊，用來身分驗證，避免未註冊過的使用者直接與伺服器連線，造成伺服器資源耗盡、癱瘓、無法服務，在資料庫上使用入侵防禦系統（Intrusion Prevention System）預防攻擊。

(4) Database 與 Client 之間

1. Client 使用 hash chain（hash 次數每次遞減）的方式將資訊與 Database 生成的鹽值（Salt）合併 hash 後，傳送給 Database。

2. Database 接收 Client 傳遞的資訊，並產生隨機鹽值（Salt）與資訊進行 hash 後存入 Database，圖六為 Database 儲存 Client 資料流程圖。



（圖六：Database 儲存 Client 資料流程圖）

(5) Database 與 Server 之間

1.首先由 TLS 協議進行建立加密通道需要的協商和認證，Database 透過使用 TLS 協議的加密通道，傳送合法使用者 IP address 給 Server，保證通訊的私密性。

2. Server 把合法使用者 IP address 存放於 set 中，等候 Client 連線並與 set 中資訊比對是否為合法使用者。

(6) Client 與 Server 之間

1. Client 傳送請求給 Server。

2. Server 查詢 Client 的 IP address 是否存在於 set 中，若存在則開啟雲端遊戲給 Client 遊玩，反之則關閉連線。

(三) 入侵防禦系統

1. 使用入侵防禦系統協助預防攻擊，並從被攻擊紀錄中，研發更多抵擋他種攻擊的防禦規則。

2. 撰寫對被攻擊紀錄進行特徵比對的程式，偵測到攻擊時，自動寫入 IPS 的規則中，並使用 IPS 丟掉來自異常 IP 的封包。

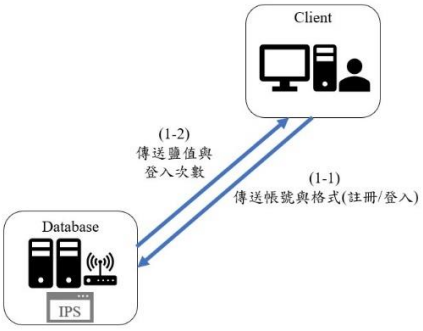
五、研究成果

經過此實驗，使得使用者能暢遊各式各樣的遊戲，而不需具備高規格硬體，且攻擊者不能輕易竊取使用者資料、攻擊資料庫及伺服器，使雲端遊戲系統更加安全。

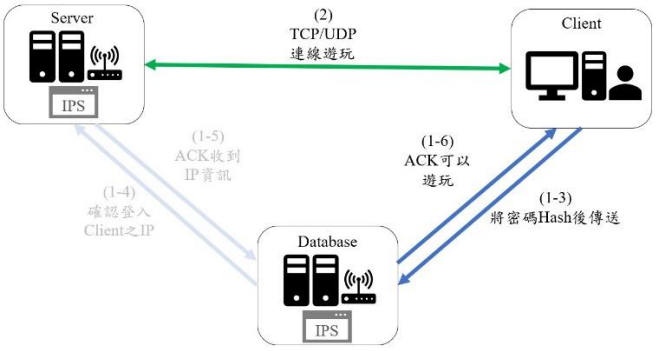
以下為本次實驗各個部份之分析：

(一) Client：

在註冊與登入階段，如圖七所示先藉由傳送帳號與格式告知 Database 此使用者需要註冊/登入，並等待 Database 傳送鹽值與登入次數後將使用者密碼進行 hash chain 回傳給 Database，最後等待 Database 發送許可即可與 Server 進行遊玩連線，圖八為上述的示意圖，圖九、圖十為使用者註冊與登入介面。



（圖七：註冊/登入階段一）



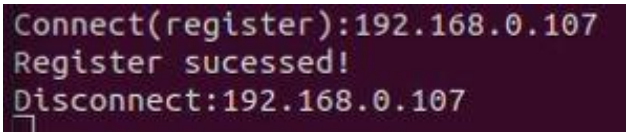
（圖八：註冊/登入階段二）

（圖九：Client UI 註冊介面）

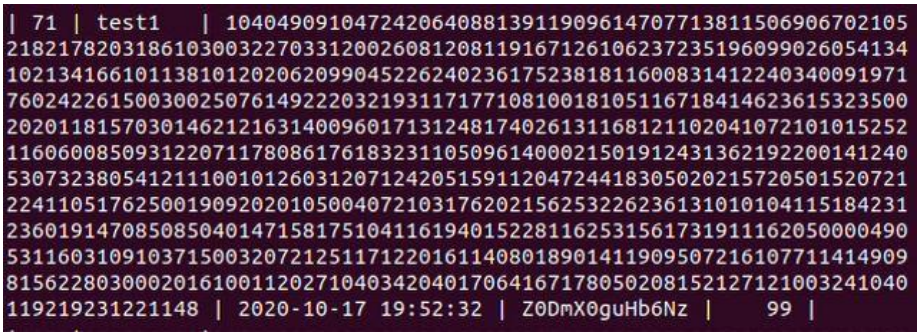
（圖十：Client UI 登入介面）

(二) Database：

註冊階段：等待 Client 連線，隨機生成長度最多為 15 bytes 的 salt（鹽值）並傳送給 Client 使其順利將密碼 hash 後回傳過來，會在 Database 顯示註冊成功與否的畫面，如圖十一。Database 藉由 Mysql 儲存的使用者資料格式，如圖十二依序為使用者編號、使用者帳號、使用者 hash 後的密碼、登入時間、Salt 與剩餘登入次數等。

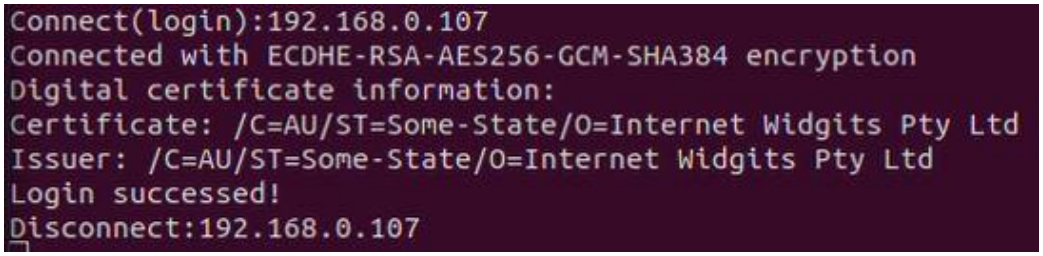


（圖十一：Database 中的註冊畫面）

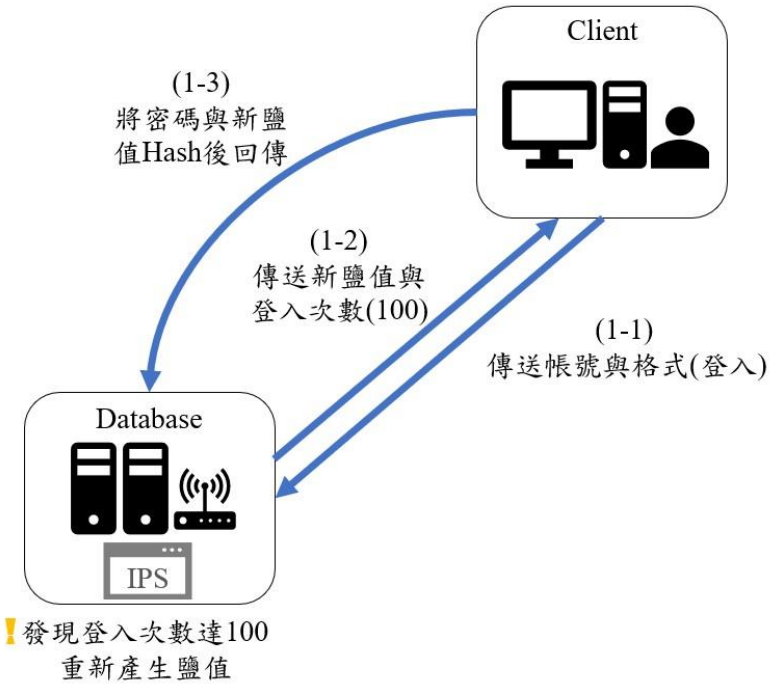


（圖十二：儲存格式）

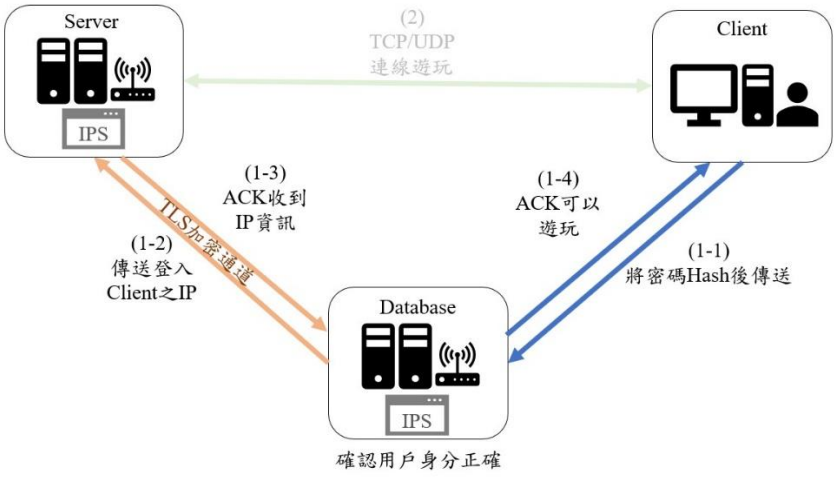
登入階段：確認該使用者身分正確時，將 Client 的 IP address 藉由與 Server 建立的 TLS 加密通道傳送，圖十三為 Database 中的使用者登入的畫面，其中包含與 Server 搭建 TLS 加密通道的證書內容。等待 Server 傳送可建立連線之回覆後，再將該結果告知 Client，使其能與 Server 正常連線。圖十四為當使用者到達登入次數上限時，Database 生成隨機鹽值並交給 Client 使其將密碼重新 hash 傳送回來。圖十五為 Client 連接 Database 傳送資料流程圖。



（圖十三：Database 中的登入畫面）



（圖十四：Database 重新生成鹽值並傳送）



（圖十五：Database 與 Client 傳送資料過程）

(三) Server：

連接 Database：由 TLS 協議建立加密通道需要的協商和認證，在 Database 連線後，傳送所需的證書內容並等待資料，等接到 Client 的 IP address 時，將其資訊存入 IP table 中當作合法使用者，以用來判斷後面連線者是否為合法連接者。

連接 Client：判斷與 Server 連線之 Client 是否為合法連接者，確認成功後才開啟 GamingAnywhere 伺服器供給其使用。

(四) IPS：

圖十六為我們的程式碼，運作方式為每次讀取 500 筆網路流量資料進行運算，判斷各個 IP 在一段時間內的流量變化是否正常，透過結合 Fast Entropy Approach 和 Adaptive Threshold Algorithm 來偵測 DoS 攻擊，並將判斷為攻擊方之 IP 加入 IPS Rule 中，以丟棄來自該 IP 之封包。

程式碼變數用途解釋：

x.first：欲計算之 IP 的名稱。

x.second：欲計算之 IP 在 500 筆資料中的出現次數。

temp：所儲存的為上 500 筆資料中有出現 x.first 的數量。

H：Entropy。

sum：此次計算之出現 x.first 的數量。

mean：此次計算之 IP 平均出現的次數（sum/出現之 IP 總數）。

B：threshold。

stdDeviation：標準差。

```
for(auto x : mapper)
{
        double temp;
        if(premapper.find(x.first)!=premapper.end())
        {
                temp = premapper[x.first];
                if(x.second > temp)
                        H = log(temp/x.second)-log(x.second/sum);
                else
                        H = log(x.second/temp)-log(x.second/sum);
                if (H> 1.5*mean)
                        B++;
                if (H< 0.5*mean)
                        B--;
                if(fabs(mean-H) > B*stdDeviation)
                        cout << x.first<<endl;
        }
}
```

（圖十六：Entropy Approach）

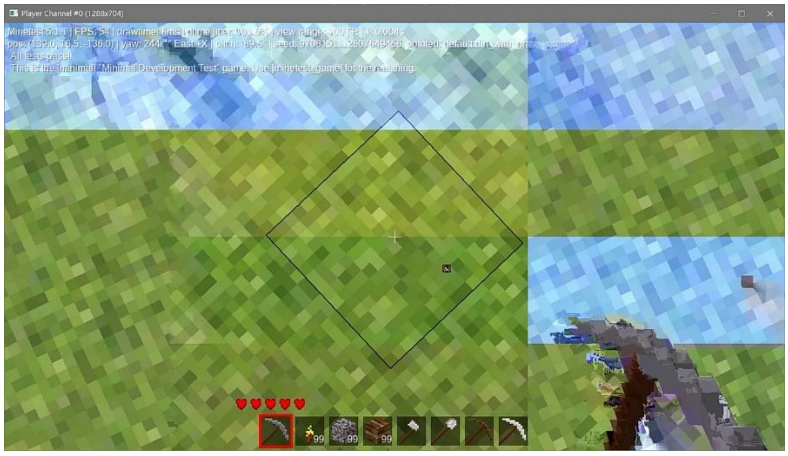
六、結論與展望

我們在開源雲端遊戲系統 GamingAnywhere 的基礎上，加入了原先沒有的入侵防禦系統，透過偵測 DoS 攻擊並將抵禦惡意 IP 的規則加入 IPS Rule 中，達到抵擋攻擊的效果，且在 Server 端加裝 IPS 可以抵禦已登入的使用者是惡意使用者的狀況，能夠適時防禦已受認證 IP 之攻擊；而 Database 的資訊傳輸，除了與 Server 間的 TLS 加密通道或是和 Client 之間的密碼 hash 傳送，都能增加其安全性。再加上將 Server 與 Database 分開架設，使得 Server 只需處理 Database 認證 IP 之封包，且如果其中任一方被攻擊，都不會影響對方，因此可以加強使用者遊玩品質。

在我們的專題期間，當網路中沒有惡意攻擊者時，使用者能在一個安全的環境下，很快的註冊、登入連線開始遊玩，但仍遇到了下列問題：

(1) 遊玩時延遲：

在 4G 行動網路中，因為頻寬不足、延遲，導致使用者與伺服器間能肉眼辨識出兩者無完全同步，如圖十七影像傳送時，網路過慢造成延遲。現階段 5G 行動網路正在普及，之後 5G 行動網路取代 4G 行動網路後，預計不會有此問題。



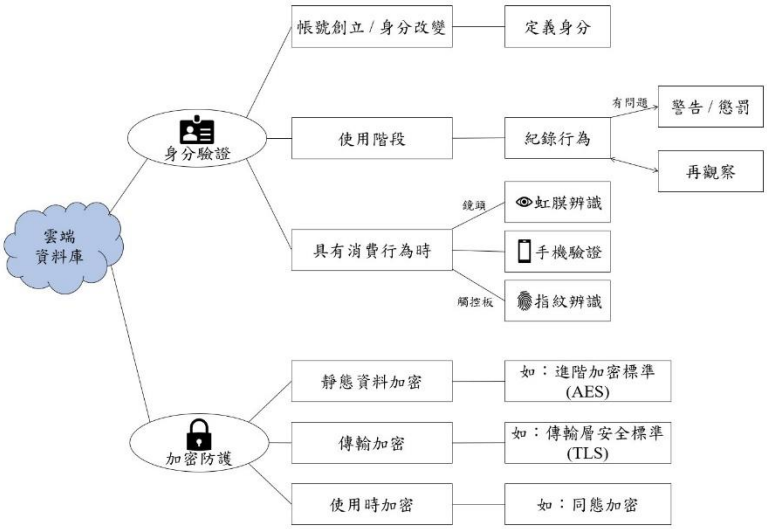
（圖十七：UDP 傳送封包延遲）

(2) Server 內部網路存在大量封包而堵塞：

Server 的內部網路遭受 DoS 攻擊後，網路中會留下大量惡意封包，而導致伺服器與使用者之間連線受到影響，使用者遊戲畫面 FPS 下降且畫面延遲，之後藉由上游流量清洗（可以透過外包或自己寫機器學習達到該效果），以及下游搭配防禦機制，預計能減緩此問題。

(3) 帳戶安全性問題：

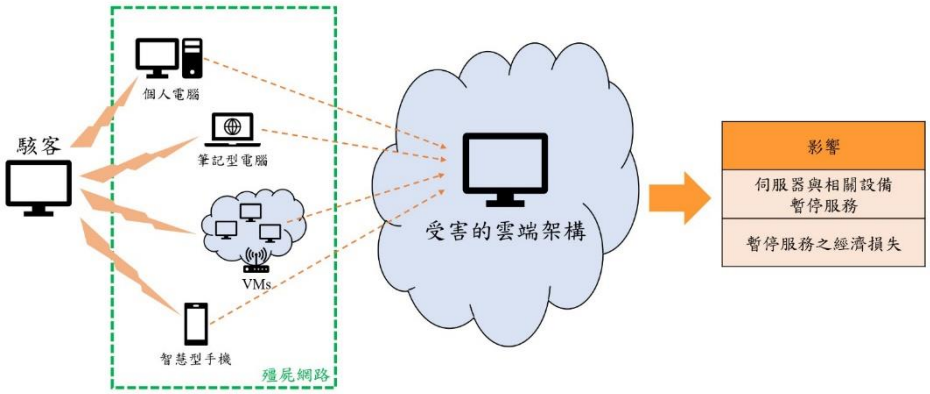
由於我們對於使用者的資料安全僅使用了密碼 hash 傳送的部分，在面對有心人士的惡意攻擊時難免會防不勝防；因此希望未來能透過圖十八中列出的安全防護方式，如：虹膜辨識或指紋辨識等，達到更嚴謹的防護，雖然過程繁雜但對於安全性而言可以達到良好的效果。



（圖十八：強化身分驗證之安全防護）

(4) 無法避免 DDoS 攻擊：

如圖十九所示的 DDoS 攻擊，目前雖然無法完全遏止，但未來期許能夠透過虛擬伺服器多開，轉移伺服器重心以避免攻擊，或透過機器學習盡可能地偵測並處理 DDoS 攻擊，以確保使用者的遊玩品質。



（圖十九：DDoS 對於雲端系統的攻擊與影響）

七、參考文獻

1. Huang, Chun-Ying, et al. "GamingAnywhere: an open-source cloud gaming testbed." Proceedings of the 21st ACM international conference on Multimedia. 2013.
2. Cai, Wei, et al. "A survey on cloud gaming: Future of computer games." IEEE Access 4 (2016): 7605-7620.
3. Vishnumolakala, Bandhavi. "Performance evaluation of Gaming Anywhere server in a virtual environment." (2018).
4. 劉明輝, and 蔡文賢. 網路安全監控行為之實作與分析. Diss. 2007.
5. Tebaa, Maha, Saïd El Hajji, and Abdellatif El Ghazi. "Homomorphic encryption applied to the cloud computing security." Proceedings of the World Congress on Engineering. Vol. 1. No. 2012. 2012.
6. Neupane, Roshan Lal, et al. "Dolus: cyber defense using pretense against DDoS attacks in cloud platforms." Proceedings of the 19th International Conference on Distributed Computing and Networking. 2018.
7. Chen, Qi, et al. "CBF: a packet filtering method for DDoS attack defense in cloud environment." 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing. IEEE, 2011.
8. Wang, Kuisheng, and Yan Hou. "Detection method of SQL injection attack in cloud computing environment." 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC). IEEE, 2016.
9. Miao, Rui, et al. "The dark menace: Characterizing network-based attacks in the cloud." Proceedings of the 2015 Internet Measurement Conference. 2015.
10. Malina, Lukas, and Jan Hajny. "Efficient security solution for privacy-preserving cloud services." 2013 36th International Conference on Telecommunications and Signal Processing (TSP). IEEE, 2013.
11. Bawany, Narmeen Zakaria, Jawwad A. Shamsi, and Khaled Salah. "DDoS attack detection and mitigation using SDN: methods, practices, and solutions." Arabian Journal for Science and Engineering 42.2 (2017): 425-441.

12. Agarwal, Ashish, and Aparna Agarwal. "The security risks associated with cloud computing." International Journal of Computer Applications in Engineering Sciences 1 (2011): 257-259.
13. 罗志强, 沈军, and 金华敏. "分布式 DNS 反射 DDoS 攻击检测及控制技术." 电信科学 31.10 (2015): 1-196.
14. Shea, Ryan, et al. "Cloud gaming: architecture and performance." IEEE network 27.4 (2013): 16-21.
15. Chen, Kuan-Ta, et al. "On the quality of service of cloud gaming systems." IEEE Transactions on Multimedia 16.2 (2013): 480-495.
16. Carter, J. Lawrence, and Mark N. Wegman. "Universal classes of hash functions." Journal of computer and system sciences 18.2 (1979): 143-154.
17. Gentry, Craig. "Fully homomorphic encryption using ideal lattices." Proceedings of the forty-first annual ACM symposium on Theory of computing. 2009.
18. Day, David, and Benjamin Burns. "A performance analysis of snort and suricata network intrusion detection and prevention engines." Fifth international conference on digital society, Gosier, Guadeloupe. 2011.
19. Bertino, Elisa, and Ravi Sandhu. "Database security-concepts, approaches, and challenges." IEEE Transactions on Dependable and secure computing 2.1 (2005): 2-19.