陳志誠、劉用貴(2016),『建構雲端環境資料安全存取模型暨績效評估』,
中華民國資訊管理學報,第二十三卷,第一期,頁 1-32。

建構雲端環境資料安全存取模型暨績效評估

陳志誠
大同大學資訊經營學系

劉用貴*
大同大學資訊經營學系

摘要

由於在雲端環境中越權存取的威脅日益嚴重,使網路服務的風險與日遽增,雲端服務提供者本身是否具有足夠能力確保客戶的資料安全、防範非授權使用者對資料的存取或破壞,已成為雲端使用者最關切的議題。為確保雲端用戶資料的機密性和完整性,在提升大量資料存取效率的同時,強化用戶資料傳輸和儲存的安全是極其重要的,本研究提出了一個新的作法,能夠使分散式資料庫存取更安全、更有效率的主動驗證與排程方法,內容包括「主動式身分驗證」、「安全隔離與資料交換」、「優先權多級排程控制」、「分散式存取方法」及「RC4 加解密技術」等。用戶作業必須透過私有雲主動驗證才能取得授權碼,其資料必須經過加密處理之後,再進入獨立通道透過安全隔離與資料交換,才能進入私有雲取得存取權限進行交易。為提升交易效率,我們建議結合優先權多級安全排程,進行分散式資料安全存取。經由實驗顯示,利他鎖定(Altruistic Locking, AL)排程原則能使分散式資料庫存取更有效率。經由檢視表將優先權及多級安全相互結合模擬,達到資料安全存取的目的。研究結果顯示,要做好存取控制,必先做好「讀」的控管,即可解決大部分不當存取的威脅,本研究並發現,做好「寫」的排程序列化,即可有效避免死結發生。研究顯示此一雲端資料安全存取架構能有效的遏止越權存取,也可提高交易並行性,增進資料存取效能性,透由實驗結果顯示,私有雲以優先權多級安全及分散式資料庫存取方式,AL 能更快更有效的完成交易,能盡快的將費時較短的交易完成(Commit),減少交易重新執行(Rollback),避免死結發生。經由兩組實驗比較,驗證私有雲分散式資料庫中「優先權多級安全及鎖定」AL 優於傳統的二階段鎖定(2-Phase Locking, 2PL),以 AL 作為排程的機制確實能獲得更佳的效能,說明了本研究架構之可用性。

關鍵詞:雲端運算、資訊安全、主動驗證、優先權多級安全、分散式資料存取

* 本文通訊作者。電子郵件信箱:alex51204612@yahoo.com
2013/05/11 投稿;2014/10/30 修訂;2015/01/15 接受

Chen, P.S. and Liu, Y.K. (2016), 'Construction and efficiency evaluation of a secure data access model in the cloud computing environment', *Journal of Information Management*, Vol. 23, No. 1, pp. 1-32.

Construction and Efficiency Evaluation of a Secure Data Access Model in the Cloud Computing Environment

Patrick S. Chen
Department of Information Management, Tatung University

Yong-Kuei Liu*
Department of Information Management, Tatung University

Abstract

Purpose—Due to the growing intelligent attacks, internet service providers are facing more and more risks. It has become a big concern, especially in the emerging cloud computing environment, whether the service providers have the capability to properly protect users' data from attacks and prevent unauthorized access.

Design/methodology/approach—In order to meet the information security requirements of confidentiality, integrity and availability with consideration of access efficiency in the presence of huge amount of data, we proposed an efficient and secure data access model covering active authentication, encryption/decryption, and access to databases.

Findings—Through experiments, we found that the control of "read" will solve most unauthorized access problems and serialization of "write" will avoid deadlocks.

Research limitations/implications—We designed a multi-layered, distributed database system and proposed a secure access model in which only two locking mechanisms, two-phase locking and altruistic locking, are compared. Other mechanisms are not considered in this study.

* Corresponding author. Email: alex51204612@yahoo.com
2013/05/11 received; 2014/10/30 revised; 2015/01/15 accepted

Practical implications－A prototype was implemented to test the applicability of the proposed model. The system first authenticates a user and then assigns him a ticket. This process accomplishes fined-grained access control. After analyzing the data obtained from the experiments, we found that the proposed data access model is well suited for the cloud computing environment in terms of security and efficiency.

Originality/value－This study proposes a new approach to system security, permitting distributed database access and efficient scheduling. The system allows active identity verification, secure data isolation and information exchange, multi-level scheduling based on priorities, distributed access control and use of encryption technology.

Keywords: cloud computing, information security, active authentication, multi-level security, distributed data access

壹、緒論

雖然雲端運算能提供多樣性的服務，但是資料洩漏和個人隱私被盜取的情況也成為嚴重隱憂，這也是也是企業進入雲端服務的最大阻礙因子（Lin et al. 2010）。此外，由於大量資料都存放在一起，存取效率也是重要考量。故如何提升雲端運算資料快速存取、強化資料保密及身分認證，以確保在雲端中用戶資料的機密性和完整性，是本研究主要目的。

本研究為改善雲端運算中資料存取安全問題，我們提出了主動式身分驗證機制；為確保雲端資料存取及傳輸安全，採用資料安全隔離技術；為提升雲端資料存取效能並避免死結產生，結合優先權多級排程控制方法；為保障隱私資料安全與多級安全分散儲存，採用分散式資料存取技術，以確保雲端環境資料安全。綜上，我們期望藉由這一套雲端環境資料安全存取模型，能改善雲端運算資料安全存取的問題，經由實驗顯示，本模型是可信任的。

本研究第貳節先針對雲端運算安全相關文獻加以探討；第參節說明如何建構雲端環境資料安全存取模型，並建立系統架構；第肆節對系統架構進行模擬及績效評估；最後於第伍節對本研究進行總結。

貳、文獻探討

雲端運算安全所牽涉的安全問題極多，依據 Brodkin（2008）發表的雲端安全研究報告－Assessing the Security Risks of Cloud Computing，使用者須注意資訊的完整性、資料復原、隱私性、稽核及管理承諾等。此外，不少學者，如 Goodhue 與 Straub（1991）、Kankanhalli et al.（2003）、Jung et al.（2001）等，曾指出組織及產業特性是影響資訊安全的重要因素。不同的產業有不同的資訊需求（葉桂珍 & 張榮庭 2006），比如傳統製造業、流通服務業等通常較重視資訊之可用性（availability），高科技產業較重視資訊之機密性（confidentiality），而金融服務業則較重視資料之完整性（integrity）（Jung et al. 2001）。陳志誠等（2009）提出資訊安全的漏洞往往不是技術性上的問題，大部分是由內部使用者違反規定導致資訊安全的漏洞；所以加強身分驗證管理及資料加密以保護資訊資產，成為銀行業者刻不容緩的工作。

綜上，由於雲端安全問題所涉及的層面甚廣，我們無意、也不可能解決所有問題，本論文僅針對其中許多先前研究都提到的使用者身分認證、資料儲存與安全隔離、雲端優先權多級安全排程、資料安全存取與隱私保護及資料安全傳輸等五個重要問題進行深入研究，提出我們的因應之道。有關以上所提雲端安全問題之文獻回顧分析如下：

一、使用者身分認證

雲端安全是指在一個開放的雲端網路平台上，為使用者提供一個安全可靠、穩定、持久的資料共享服務機制，以達成用戶資料的機密性（confidentiality）和完整性（integrity）的目標。雲端運算常發生資料被篡改、隱私資料越權竊取及敏感資料存取不安全等問題亟需克服（Lang 2010）。為了讓使用者快速建構具彈性且容易存取的私有雲基礎架構，以執行相關的商業行為與作業流程，許多公司如 BP、Intel、及 IBM 等，提供透過整合一些既有的技術或產品來達成私有雲的建置。惟對於私有雲隱私資料越權竊取及敏感資料存取之安全尚未見有完善的保護，引發本文研究動機。

二、資料儲存與安全隔離

利用分散式計算（Distributed Computing）可達成資源妥善應用之目的。雲端運算應用分散式技術和高速網路資源使用的效率，用戶之間可能存在著共用儲存資源或運算資源的操作模式，雲端安全聯盟（Cloud Security Alliance; CSA）於 2012 年發表可信任安全雲端運算計畫（trusted cloud initiative）提出私有雲端的安全重點集中在共享儲存、虛擬化、網路和雲端管理平台四個面向。

歐洲網路與資訊安全機構（European Network and Information Security Agency 2010）[1]公布的「雲端運算：利益、風險與資訊安全建議」，對企業運用雲端運算服務提出 24 項雲端風險項以及對運用雲端運算服務之安全措施提出三項建議：(1)於雲端運算服務的兩端須建立信任機制；(2)大型跨組織機構須執行電腦鑑識以及數位證據資料的保護措施；(3)建立大型電腦系統工程的資源隔離機制、不同雲端運算服務平台之間之溝通及系統回復能力機制。以上都與資料儲存與安全隔離有關。另外，Gartner（2010）針對雲端運算所面臨「資訊安全」，提出七項資安議題，提醒管理者注意並作控管，其中第(3)項是資料位置考量（data location）；第(4)項是資料的隔離（data segregation）。

三、雲端優先權多級安全排程

Pang、Carey 與 Livny（1995）提出即時資料庫的想法，但是若沒有優先權的觀念，所以當網路負載過量時，所有的處理都會受到影響。分散式資料庫的存取控制大致上可分為強制存取控制（mandatory access control; MAC）（Jeong et al. 2003）、自由裁量存取控制（discretionary access control; DAC）（Lewis & Wiseman 1997）及以角色為基礎存取控制（role-based access control; RBAC）（Ferraiolo et al.

[1] http://www.enisa.europa.eu/

2001; Sandhu et al. 1996）這三大類，其中的優缺點比較如表 1 所示：

表 1：存取控制規則比較

存取控制規則	優點	缺點
MAC	安全性佳	效率不佳
DAC	有彈性	建立修改存取清單耗時，安全性不佳
RBAC	建立快速有效率	不適用於大型資料庫系統

由表 1 觀之，因本研究強調安全性，決定使用 MAC 來做存取控制的規則，至於效率不佳的缺點，可透過優先權及利他鎖定來改善。Wood、Summers 與 Fernandez（1979）提出作業系統與資料庫間的存取機制，此機制是以 MAC 為基礎，它必須有安全政策，適用在多級安全資料庫環境下，於是我們在私有雲後端採用分散式多級安全資料庫，以增強資料安全性與改善效率不佳的問題。

交易（transaction）是雲端分散式資料庫處理的最小邏輯單位，必須滿足以下特性（Garcia-Molina et al. 2008）：(1)單元性：可以將交易分成數個子交易，但一定要全部完成或全部不完成；(2)一致性：如果任一交易失敗，它必須退回到交易開始之前狀態；(3)隔離性：一個交易在尚未完成之前，不可和其他交易重疊；(4)持久性：一但交易完成，就必維持穩定一致狀態。為了保持資料庫的一致性而有了鎖定（locking）的功能設計，但常常會有死結及效能不佳的問題。

四、資料安全存取與隱私保護

雖然企業採用雲端運算有其利基，大部分仍在觀望之中，尤其是金融機構對雲端運算架構一直持謹慎態度（陳志誠 & 王瀞慧 2011）。在雲端運算架構下，隱私保護的問題起源於個資儲存於（或是移轉於）機器之間，而這些機器不是使用者所擁有或能控制的，隱私資料必須由雲端運算端用戶和雲端運算端主機共同來維護。

五、資料安全傳輸

為改善企業內部資訊需藉由雲端服務供應商所提供的管理機制進行資料存取及分析之問題，劉家驊與洪士凱（2010）提出以架構導向雲端運算服務之資訊安全防護機制，在傳送企業內部重要資訊前，先以架構導向將資料依企業目標與任務做整體性的規劃與分類，分類不僅須與企業目標相互呼應，同時又可運用資料進行適當的整理以決定適當的資訊安全防護機制。

為了解決以上所提之五項雲端安全問題，本研究將於下一節提出雲端環境資料安全存取模型及系統架構，以為因應。

參、系統架構

由上節的文獻探討與分析，我們為了解決有關使用者身分認證、資料儲存與安全隔離、雲端優先權多級安全排程、資料安全存取與隱私保護及資料安全傳輸等五個問題，提出本研究之模型設計及系統架構。

一、模型設計

本研究為改善雲端運算中，資料被篡改或被竊取等存取權限安全問題，並以建構雲端資料安全目標為研究目的，我們依序下列問題進行研究：(1)為解決雲端運算中，有關使用者身分認證存取權限安全問題，我們提出了「主動式身分驗證」，以主動式驗證身分方法，經驗證符合給予交易授權碼，始可依權限存取資料。(2)為確保雲端資料儲存與安全隔離，採用「安全隔離與資料交換」，防止外部與內部資料交換時企圖強制登錄，確保資料存取完整性。(3)為提升資料存取效能並避免死結產生，我們提出「優先權多級排程控制」，改善資料存取發生死結問題。(4)為保障雲端資料安全存取與隱私保護，採用「分散式存取方法」，防止資料被越權存取。(5)最後，為遏止雲端資料遭駭客竊取之風險，資料由公有雲傳輸到私有雲過程皆以 RC4 密碼基礎設施為整個系統交易完成提供加解密技術服務，確保資料安全傳輸。

綜上，本研究流程如圖 1 所示，由動機生成確立研究主題，透過文獻探討針對使用者身分認證等五項安全問題分析，建構包含「主動式身分驗證」等五項組合之模型，最後以分散式資料庫 My-SQL Query Analyzer 實驗模擬的方式證實我們所提模型架構的可行性。

透由研究流程，我們提出一套「雲端環境資料安全存取模型」如圖 2 所示，期望能改善雲端運算隱私資料安全存取的問題，經由實驗驗證顯示，本模型是可信任的。

二、建立系統架構

本研究引用 Bell 與 LaPadula 在 1976 年所提出的多級安全（multi-level secure; MLS）架構（Bell & LaPadula 1976），並結合客戶優先權（priority of client; PClient）的觀念，提出優先權多級安全模型（priority/multi-layered secure model; PMLS），若能賦予這些計算資源較貧乏的客戶端有較高的優先權，便能儘早將優先權較高及

週期性短的交易結束，產生死結的機會也會相對的減少，便能使客戶端的優先權相互結合，如此一來多級安全便可與資料庫相互分離出來，系統的模組化程度也會提高（Hinke & Schaefer 1975）。

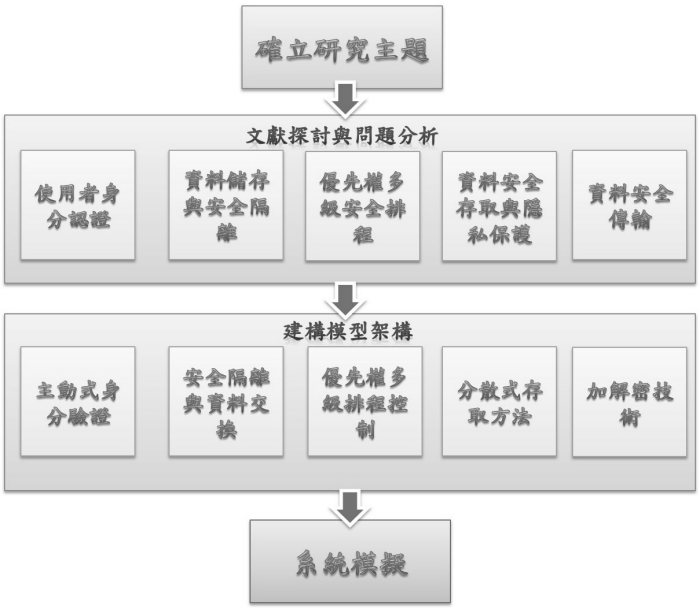

圖 1：研究流程

另搭配雲端客戶端（PClient）及使用者優先權（PMLS）的觀念，並給予優先權不同的使用者及客戶端在存取不同安全層級的資料提供不同的服務品質。當雲端壅塞時，先處理優先權高的交易，而將優先權低的交易暫時擱置。另（Pfleeger & Pfleeger 2002）將使用者視為 subject，存取標的物視為 object，其架構的主要觀念是，一個 subject 被授權（clearance）的安全層級為 C（S），一個 object 被分類（classification）的安全層級為 C（O），安全層級依 Lattice 理論組合，其中兩個主要精神為：(1)簡單的安全情況：subject 的安全級別必須高於 object 時，亦即 C（O）≤ C（S），才可讀取該資料，亦即不可向上讀（No Read-up）；(2)＊-property：當 subject 讀過 object O 時，欲寫入該資料 P，它的安全級別必須低於 P，亦即 C（O）≤C（P），才可寫入該資料，亦即不可向下寫（No Write-down）。其中目前廣為使用的多級安全資料庫架構可分為四個層級，分別是 TS（Top Secret）極機密、S（Secret）機密、C（Confidential）密及 U（Unclassified）普通（Niemeyer 1997）。

採用多級安全分散式資料存取並以兩階段鎖定（two-way phase lock; 2PL）方法，2PL 屬於並行控制（concurrency control）的技術，它可以確保在處理兩個表格或分散在各地不同雲端分散式資料庫中的資料時，不會因交易失敗，而導致雲

端資料不一致的情況發生，2PL 就是限制每個交易中全部的鎖定動作，並且會依交易的特性（讀或寫），給予適當的鎖定機制（讀鎖或寫鎖）。此種機制可被分為擴展（expanding）階段及收縮（shrinking）階段（David & Son 1993），簡介如下：(1)擴展階段：在此擴展階段，容許加入雲端分散式資料庫項目的新鎖定，但不允許解除任何鎖定。此階段容許升級，如讀鎖狀態升至寫鎖狀態；(2)收縮階段：在此收縮階段，現存的鎖定容許被解除（unlock），但不允許取得新的鎖定。此階段通常壓縮成交易結束時的單一指令 commit 或 rollback 指令。此階段容許降級，如寫鎖狀態降為讀鎖狀態。

Salem、Garcia-Molina 與 Shands（1994）提出了利他鎖定 AL（）想法，AL 其實和先前提的 2PL 完全一樣，只是多了贈與（donate）的觀念，此機制可以將已使用完的資料贈與出來，贈與給其他交易使用。Kim 等（2001）則提出了雙向贈與鎖定（two-way donation locking; 2DL）的架構。為優化雲端分散式資料庫效能避免資料存取時發生死結問題，本系統架構採用 2PL 並行控制與 AL 利他鎖定二種技術結合應用修改，最後透過 SQL Query Analyzer 模擬。

本研究系統架構如圖 3 所示，內容項目結合雲端運算之「動態性」、「多用戶性」及「安全性」的特點，包括「主動式身分驗證」等五項技術組合並闡述雲端環境資料安全和隱私保護，實現對資料和隱私權的每階段進行安全保護和績效評估。系統架構流程步驟如表 2 所示。

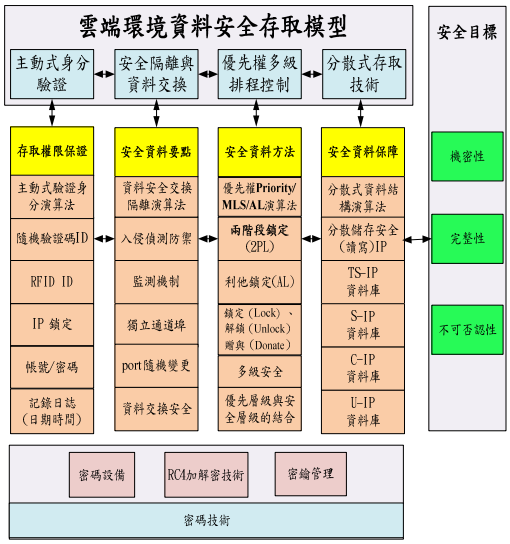

圖 2：雲端環境資料安全存取模型

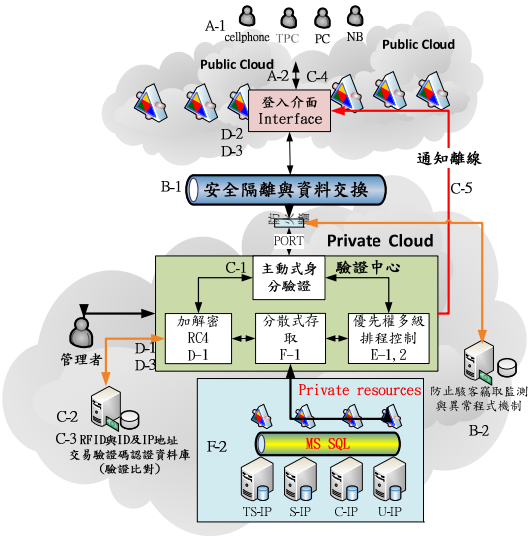

圖 3：系統架構

表 2：系統架構流程步驟

登入介面階段	
A-1	用戶經由瀏覽器連結伺服器，私有雲主動由介面偵測使用者設備判斷來源設備等級傳回伺服器，送出 IP 至伺服器產生驗證碼，伺服器將加密演算框架（script）及驗證碼以 RC4 方法加密後傳給使用者驗證使用，系統驗證資料庫會記錄 IP 位址及設備等級清單，限定已知的裝置才能安全存取資料，驗證平台伺服器會依使用者 IP 位址清單比對符合回傳驗證碼，如圖 5、圖 6 所示。
A-2	用戶取得驗證碼及加密演算框架（script），用戶將 ID 放入加密演算框架（script）中並以驗證碼當 Key 加密及設備等級比對後再傳給伺服器認證記錄。
安全隔離與資料交換階段	
B-1	透過安全隔離與資料交換技術，才能進入私有雲取得服務交易碼如圖 7 所示
B-2	使用者先經防止駭客竊取監測與異常檢查程式機制偵測結果如圖 8、圖 9 所示，進入 Port 隨機變更安全機制，使用者登入時隨機從不同的埠（設定 3 組隨機埠如圖 4 所示），再轉至 1433 獨立通道進入私有雲驗證平台，檢查符合後取得授權交易碼以保障安全性。
主動式身分驗證階段	
C-1	用戶必須透過主動式身分驗證，經驗證比對身分合格授予授權碼才能進入私有雲取得服務交易碼，同時驗證中心回傳交易授權碼，作為使用者存取權限。
C-2	導入至驗證資料庫中進行比對驗證工作包含（帳戶：高、中、低使用者及 IP、驗證碼），依據判斷使用者身分多級授予權限及容許工作內容、指令 instruction（查詢、新增、刪除、修改）。
C-3	RFID 碼認證與交易授權碼依據權限授予（查詢、新增、刪除、修改等權限）。
C-4	經由驗證合格後用戶取得交易驗證，用戶取得交易驗證（例如：授權序號 UR9012S）→經比對驗證合格後主動式驗證即會產生交易授權碼回傳使用者。
C-5	一次性驗證，當完成交易通知斷線處理，使用者必需重新登入驗證取得驗證交易驗，否則無法存取。
RC4 加解密階段	
D-1	驗證碼及加密演算法框架（script）一併加密後為數位簽章值（Cipher.txt）經由加密演算法進行加密處理，完成加密即回傳給用戶登入系統。
D-2	當使用者收到回傳驗證碼後同時解密及輸入帳號、密碼等資料，再進入私有雲驗證平台比對合符後，伺服器即發送交易授權碼給使用者。

D-3	使用者取得交易授權碼進入系統操作存取作業。
優先權多級排程控制階段	
E-1	用戶才能進入 2PL/AL 中進行優先權排程控制作業。
E-2	(1) 會先辨識客戶端的優先權，再將這些交易依所使用資料之安全層級來排序，完成後再將這些排序好的查詢交由 AL（）來鎖定。 (2) 2PL 以資料優先權讀寫並行控制存取欄位。
分散式存取階段	
F-1	分散式資料存取技術依授權層級不同，於不同資料庫伺服器存取（TS-IP、S-IP、C-IP、U-IP）。
F-2	存取資料必須依優先權排程控制，依交易授權碼定義資料表欄位如圖 18 所示，並由 MS-SQL 資料庫分散式存取技術分配存取至 TS、S、C、U 四個多級安全資料庫伺服器中。

表 2 系統架構流程步驟敘述如下：

（一）登入介面階段

使用者由登入介面瀏覽器通訊協定 TCP/IP 交握後 ClientHello 如圖 5、圖 6 所示，應用使用者優先權觀念，並授予不同優先權的服務品質，登入時先偵測使用者螢幕解析度判斷設備（例如 A：智慧型手機；B：平板電腦；C：筆記型電腦；D：桌上型電腦）的優先多級排序，並傳送使用者 IP 請求驗證服務，IP 加上伺服器端產出亂數值產生驗證碼，伺服器將加密演算框架（script）及驗證碼以 RC4 方法加密後傳給使用者驗證使用，系統驗證資料庫會記錄 IP 位址、設備清單並儲存，將限定已知的裝置才能安全存取資料，驗證平台伺服器會依使用者記錄清單比對符合回傳驗證碼。

（二）安全隔離與資料交換階段

由於使者在隔離應用交換中可以區分「用戶」、「資料」，將接收方的行為資訊和資料資訊分成了兩個不同的 Port 路徑，從而實現了應用資料的受控單向及隨機 Port 流動，資料與資訊必需經由獨立通道 Port 隨機變更安全機制，由系統自動隨機分配登錄系統 Port 89、8081、88 防止外部企圖強制登錄，驗證成功即由私有雲內的 1433 Port 來進行資料交換如圖 4 所示，資料交換過程如下：

1. 安全隔離交換過程步驟說明如圖 7 所示：(1)用戶端由獨立通道（Port）對外系統自動隨機分配登錄系統 Port 89、8081、88 防止外部企圖強制登錄根據權限指令使用功能。(2)進入私有雲中先解密驗證碼，再確認用戶權限。(3)驗證成功即由私有雲內的 1433 埠來進行資料交換，將判定合格交易驗證碼，傳回用戶端接收後解密依據交易驗證的用戶權限進行資料接收方將接

收到的資料進行存儲。(4)資料接收方將接收到的資料進行存儲。(5)通知斷線處理，等待下一個資料交換。

2. 安全隔離交換過程方法說明：安全隔離與資料交換演算法如表 3 所示。

表 3：安全隔離與資料交換演算法

資料交換演算法	資料交換演算法說明
Transfor（ID, RC4（ID）） If（ID is Allow） Return（do（ID）） Else Return 0,	公有雲加密並傳送自己 IP 與 ID 或 RFID 碼以及加密演算法框架，私有雲接收解密後，依據該交易授權碼授給權限在私有雲中存取操作。 私有雲回傳結果至公有雲，通知離線。

3. 獨立通道隨機變更安全機制

我們設計一個因應使用者登錄系統網頁時，由系統自動隨機分配隨機 Port（89、8081、88 三組 Port）以防止外部企圖強制登錄如圖 4 所示，驗證成功即由 1433 獨立通道（Individual Port Channel）進入私有雲，達到公有雲與私有雲間的資料交換和傳輸。此外，透過 Port 隨機變更安全機制，可以達到公有雲與私有雲間同步存取控制目的。

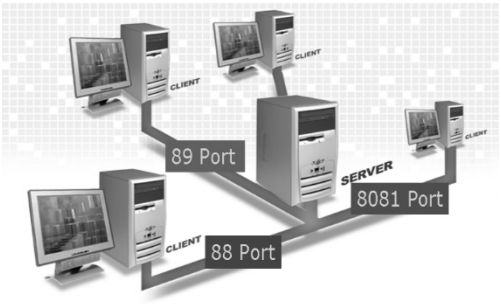

圖 4：登錄系統網頁 Port 隨機變更安全機制

圖 5：主動身分驗證實驗平台

圖 6：RFID 主動身分驗證

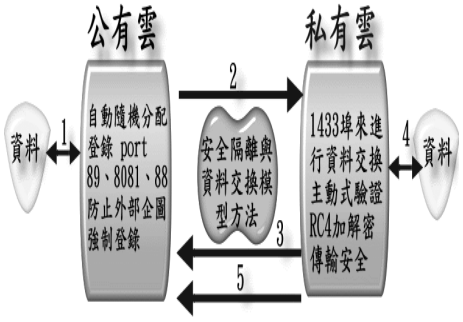

圖 7：安全隔離交換過程

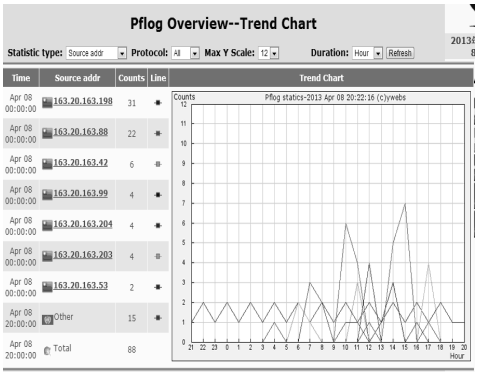

圖 8：異常 IP 記錄資料防禦處理

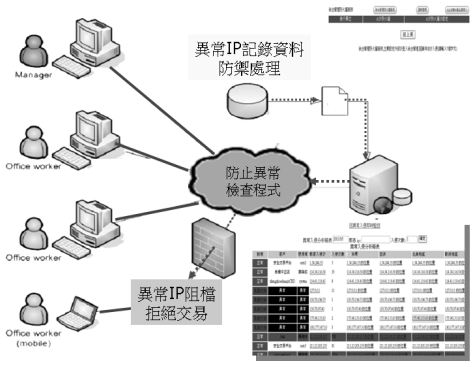

圖 9：防止異常檢查程式

4. 資料防止駭客竊取及監測機制（SQL Injections）

(1) 異常 IP 記錄資料防禦處理如圖 8 所示，網路封包的角度，大致分成上傳、下載的連線數量（Connect Seesion）、流量（Flow）跟持續時間（Time），藉由偵測這些數量的組合，推估使用者是正常使用網路或是有異常的行為（資料庫比對）。當發現內部使用者異常行為後，管理者可以採取立即限制他的最大頻寬、啟用協同防禦機制通知交換器將他封鎖或是通知管理者就好。

(2) 監測機制 SQL Injection 部分，程式完成時會先經過特別的防駭程式（防止異常檢查程式），對程式碼作分析，使得程式碼對於 SQL Injection 的防禦能力達到最好，加上程式會依照所偵測 IP 進行黑名單（異常 IP 記錄資料）比對，防止異常檢查程式如圖 9 所示，負責偵測網路中使用者異常行為，程式將啟動立即限制封鎖的程式。

(3) 異常 IP 阻檔拒絕交易：本系統會先建立使用者的正常剖繪（Profile），定義正常的標準值，若偵測到某些行為超過該標準值，則判定有入侵行為發生，針對異常 IP 阻檔拒絕交易。

使用者從統一對外窗口 Web server 80port 登入後，會經由驗證程式分配 IP 進行後續操作，再次加強隔絕外界對 SQL server 的威脅，以增強資料的完整性及可用性。

（三）主動式身分驗證階段

1. RFID 身分驗證流程

RFID 身分驗證流程如圖 10 所示，步驟如下：(1)伺服器主動取得使用者 IP 至驗證平台產生驗證碼。(2)使用者經由讀卡機讀取 RFID ID 值。(3)伺服器將加密演算框架(script)及驗證碼以 RC4 方法加密後傳給使用者驗證使用。(4)將 RFID ID 嵌入加密演算框架（script）中並以驗證碼為 Key 加密後產生 ID 回傳至伺服器進行驗證作業。

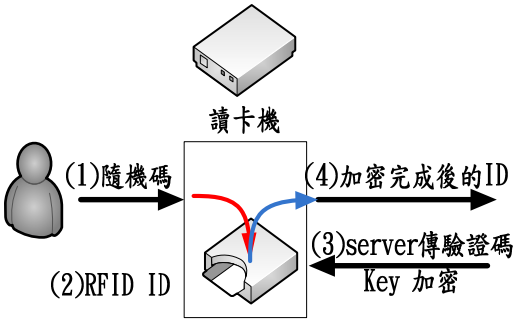

圖 10：RFID 身分驗證流程

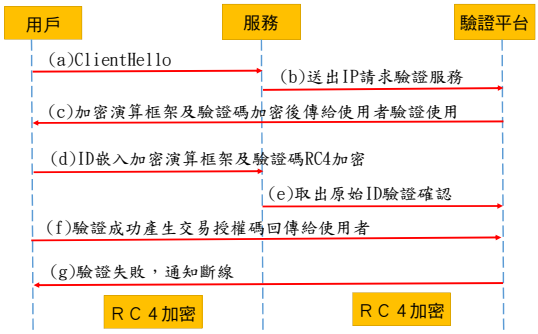

圖 11：主動式身分驗證流程

2. 主動式身分驗證流程

(1) 傳統 SSL 交握前 3 步驟如下：

(a) 發送一個「ClientHello」消息，說明它支持的密碼演算法列表、壓縮方法及最高協議版本，也發送稍後將被使用的隨機數。

(b) 然後收到一個「ServerHello」消息，包含伺服器選擇的連接參數，源自客戶端初期所提供的「ClientHello」。

(c) 當雙方知道了連接參數，客戶端與伺服器交換證書（依靠被選擇的公鑰系統）。這些證書通常基於 X.509，不過已有草案支持以 OpenPGP 為基礎的證書。

(2) 本研究採身分驗證法（主動式身分驗證）與傳統 SSL 通訊協定不同之處說明如下：本研究將前項之 SSL 交握前 3 步驟進行改良，修改成為主動

式交易授權碼來驗證，系統連接到用戶端只需接收到「ClientHello」，伺服器取得使用者 IP 嵌入至隨機產生密鑰中即會產生唯一驗證碼，和加密演算法框架（script）一併加密（RC4）後為數位簽章值（Cipher.txt）傳送到用戶端，用戶端收到即將 ID（帳號密碼、隨機碼或 RFID ID）值放入加密演算框架（script）中並用驗證碼當 Key 值加密後回傳至伺服器端溝通，伺服器收到即用驗證碼當 Key 解密後取出用戶原始 ID 進行對比驗證，驗證成功即產生交易授權碼經 RC4 加密後回傳，用戶取得交易授權碼後依授予權限存取資料操作。本研究以 RC4 加密演算法進行加密，嗣後如有新的加密演算方法，可以直接應用，無須確認用戶端是否支援，故在擴充即時性方面勝於傳統 SSL。

(3) 主動式身分驗證流程如下：主動式身分驗證流程如圖 11 所示：(a)用戶經由瀏覽器的通訊 TCP/IP 交握後 ClientHello。(b)使用者送出 IP 請求驗證服務。(c)伺服器主動取得 IP 後加伺服器端產出亂數值產生驗證碼 Key，伺服器將加密演算框架（script）及驗證碼以 RC4 方法加密後傳給使用者驗證使用。(d)用戶登入時會取得驗證碼及加密演算框架（script），用戶將 ID（帳號密碼或 RFID ID）嵌入加密演算框架（script）中並將驗證碼當 Key，加密後再傳給伺服器驗證。(e)當伺服器認證平台收到密文時利用驗證碼當 Key 解密後，再由演算框架（script）取出原始驗證碼 Key 及 ID 儲存於驗證資料庫做比對驗證確認工作。(f)驗證平台資料庫比對驗證碼、IP、RFID ID、帳號密碼等四項同時驗證成功，即產生交易授權碼回傳給用戶授予存取權限進存取操作。(g)本系統採用一次性驗證方法，當驗證失敗時會通知斷線，斷線後用戶會失去所有權限，必須再次的登入驗證才會授予新的權限。

（四）加解密階段

資料加密技術（例如 RC4），其加密演算法說明如下[2]：

1. 密鑰排程（key scheduling algorithm; KSA）：KSA 主要是初始化一個陣列值 S，作為產生 Key 密鑰使用。S 陣列長度可任意，但一般是 256，其過程是先排定一個陣列，然後再打散依其順序排序，其過程如圖 12 所示。圖中 KSA 為第二矩形框中的 K 是初始給定的 Key 密鑰，L 是密鑰 K 的長度。

[2] Brute Force Attack on Cryptographie Keys[EB／OL]．http://www.c1.cam.tic.uk/~mel/brute.htm1.

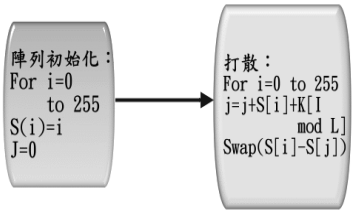

圖 12：KSA 示意

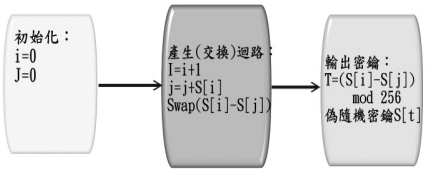

圖 13：PRGA 示意

2. 偽隨機密鑰產生（pseudo random generation algorithm; PRGA）：PRGA 把兩個變量 i 和 J 初始化為 0，然後把 i 和 J 值放入迴圈產生 Key 偽隨機密鑰。過程如圖 13 所示。
3. 產生密文：圖 13 產生的隨機密鑰 Key 與明文加密，即可得到密文。解密演算法與加密演算法相似，用密文與偽隨機密鑰 Key 相加即可生成原始的明文。

綜上，就機密性而言，將會利用在驗證碼階段所得到的驗證碼加密；至於完整性，驗證碼演算法來確保往來的訊息並沒有被篡改。

（五）優先權多級排程控制階段

1. 優先權（Priority/MLS/AL）技術
   (1) 優先權讀寫並行控制（2PL）：為了簡化交易的複雜度，一般將讀取物件視為並行的（concurrently），寫回物件則視為循序的（sequentially）。當考慮到鎖定時，我們以共享鎖定（shared lock）來處理讀鎖（read lock; RL）；以互斥鎖定（exclusive lock）來處理寫鎖（write lock; WL）。但是我們發現在排程（schedule）時會發生衝突，如圖 14 所示，根據 AL 的精神，當 T1 用完物件 a 時，便將 a 贈與出來供其他交易使用，因此 T2 可以寫鎖交易而不會有衝突，但是 T3 卻會有衝突發生，因為物件 b 並不在 T1 的贈與清單內。為了解決這種問題，而有負債（Indebtedness）的新觀念產生。假設 T3 鎖定一個由 T1 贈與的物件，則 T3 要負債於 T1，只有在兩大原則成立之下【a：個別鎖定的交易有衝突發生。b：在交易衝突鎖定之間有其他鎖定發生。】產生如表 4 之 AL 演算法及說明。

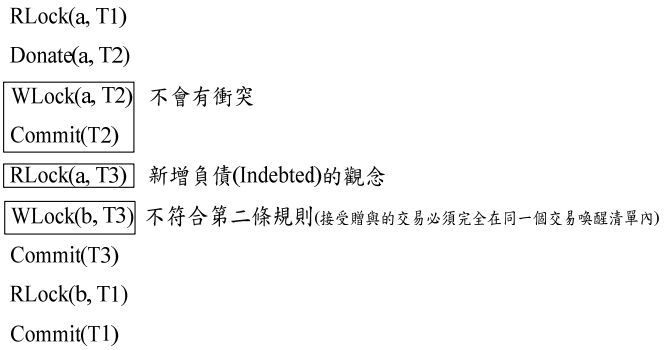

圖 14：利他鎖定的衝突參考 Salem 等（1994）本研究修改

表 4：AL 演算法及說明（加入讀寫及負債）

AL 演算法（加入讀寫及負債）	AL 演算法說明（加入讀寫及負債）

RLock（a,T）{
i←ι（T）∪wl（a）;
w←ω（T）∩d（a）;
if i≦w then{
rl（a）←rl（a）∪{T};
ι（T）←i;
ω（T）←w;
return（accept）;}
else return（reject）;
}
	交易 T 要讀鎖物件 a
	將寫鎖的物件 a 加入交易 T 並指向 i 暫存集合
	將喚醒清單內沒有贈與的交易移除
	為了確保第二條規則，則
	交易 T 可讀鎖物件 a
	更新 ι（T）
	更新ω（T）
	傳回交易 T 可讀鎖物件 a 的訊息
	若失敗則傳回交易 T 無法讀鎖物件 a

WLock（a,T）{
i←ι（T）∪rl（a）∪wl（a）;
 	
w←ω（T）∩d（a）;
if i≦w then{
wl（a）←wl（a）∪{T};
wl（a）←wl（a）∪rl（a）;
ι（T）←i;
ω（T）←w;
return（accept）;}
else return（reject）;
}
	交易 T 要寫鎖物件 a
	將讀寫鎖之交易加入所有可能的喚醒清單並丟到 i 暫存集合
	將喚醒清單內沒有贈與的交易移除
	為了確保第二條規則，則
	交易 T 可寫鎖物件 a
	再加入負債的觀念
	更新 ι（T）
	更新ω（T）
	傳回交易 T 可寫鎖物件 a 的訊息
	若失敗則傳回交易 T 無法寫鎖物件 a

我們根據上述的兩條規則做了適度的修改以符合並行控制的原則：(1)兩個交易不能同時衝突鎖定（conflicting lock）同一物件，除非其中一個

交易先贈與該物件；(2)如果 T3 負債於 T1，則 T3 必須完全在 T1 的喚醒清單中，直到 T1 解鎖。

(2) 利他鎖定說明（AL）：以下便以四個簡單的範例說明 AL 的並行控制處理機制，其中 T1 及 T2 代表交易，分別維護 ι（T1）ω（T1）及 ι（T2）ω（T2），而物件 A 負責維護 wl（a）、rl（a）及 d（a）如圖 15 所示，分別執行 RL（a,T1）RL（a,T2）、RL（a,T1）WL（a,T2）、WL（a,T1）RL（a,T2）及 WL（a,T1）WL（a,T2）這四個範例，我們將演算法的執行過程以下表說明，最後執行的結果發現只有 RL（a,T1）RL（a,T2）可成功執行。換言之，只要有寫的部分，都必須循序地處理交易。

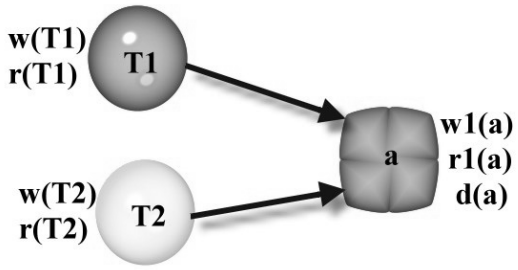

圖 15：說明並行控制之範例圖(參考陳志誠 & 宋子傑 2005）本研究修改

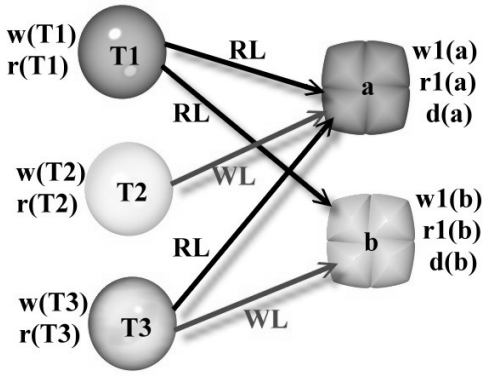

圖 16：AL 範例圖（參考陳志誠 & 宋子傑 2005）本研究修改

上面的敘述說明了 AL 可以做到並行控制，本研究將上述範例擴張詳如圖 16 所示，其中 T1、T2 及 T3 為交易，a 及 b 為資料物件，其交易順序為 RL（a,T1）→Donate（a,T1）→WL（a,T2）→Commit（T2）→RL（a,T3）→WL（b,T3）→Commit（T3）→RL（b,T1）→Commit（T1）。

2. 多級安全（PMLS）技術

雲端通訊工具例如智慧型手機、平板電腦、筆記型電腦及桌上型電腦，由高到低分別賦予優先級別 A、B、C 及 D 作為研究對象。若能賦予這些計算資源較貧乏的客戶端有較高的優先權，便能儘早將優先權較高及週期性短的交易結束，產生死結的機會也會相對的減少。

本節參考 Chen、Li 與 Liu.（2010）提出之研究，將優先權與多級安全

（prioritized multi-level security; PMLS）相互結合，說明如下，優先權的觀念必須搭配客戶端及使用者，假設客戶端分四個優先層級（I=4），使用者（如決策人員、經理人員，一般員工）分三個優先層級（J=3），資料庫分成四個安全層級（K=4）。其主要的精神是依序比較 I、J 及 K，系統會先依客戶端的裝置 I（A：最高、B：高、C：低、D：最低）給予不同的優先權，再依使用者的優先層級給他一個權限識別碼 J（1：高、2：中、3：低），先針對 I 及 J 排序，排序完後再針對物件的安全層級 K（ts：極機密、s：機密、c：密、u：普通）排序，為了容易理解，以圖 17 來表示。假設依序執行 T_1：D、T_2：D、T_3：C、T_4：B、T_5：D、T_6：A 及 T_7：D 這些交易客戶端，看客戶端優先權，由於 A 的優先權最高，所以把它放到最上面，再來是 B、C 及 D；T_6、T_4 及 T_3 這三個客戶端（代號：A、B 及 C）沒有重複，所以交易順序固定，至於 T_1、T_2、T_5 及 T_7 所使用的客戶端都是桌上型電腦（代號：D），所以要針對它們做第二次排序，由於使用者的優先層級 1 的優先權最高，因此將它往上移，其次才是 2 及 3；最後再依資料的安全層級排序，由於 ts 是最高的，因此將它排到上面，最後排序的結果依序為 T_6、T_4、T_3、T_5、T_2、T_7 及 T_1。

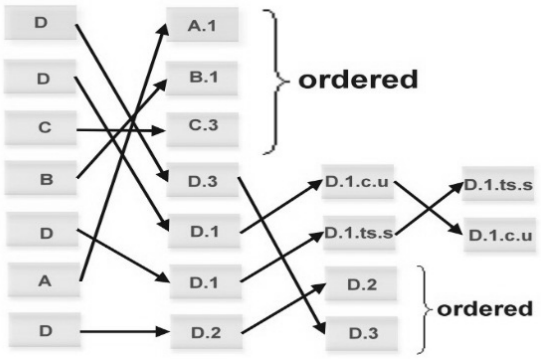

圖 17：優先層級與多級安全層級的結合範例

3. 優先權（Priority/MLS/AL）結構

綜合前面所介紹的 PClient、PMLS 及 AL，當系統同時處理許多交易時，會先辨識客戶端及使用者的優先權，再將這些交易依所使用資料的安全層級來排序，完成後再將這些排序好的查詢交由 AL（）來鎖定。一般而言，鎖死的發生通常肇因於兩個交易平權，以致僵持不下，客戶端及使用者優先權的概念導入後，可以適時有效的減少這種現象發生。

（六）分散式存取階段

資料表定義如圖 18 所示，步驟如下：

步驟 1： 分散式存取資料表主檔定義：定義私有雲中資料表多級安全（U、C、S、TS），將上述四個資料庫分散儲存至四個不同的 IP 伺服器（U-IP、C-IP、S-IP、TS-IP）之資料表讀寫資料欄位，讀寫優先權限 U、C、S、TS 等級設定。

步驟 2： 使用者透過外部公有雲登入認證程序：(1)第一層 ID 密碼認證 ID，並記錄登錄時間；(2)第二層 RFID 認證 ID；(3)記錄登錄使用者 IP；(4)使用者帳號密碼驗證；(5)經由驗證合格後用戶取得交易授權驗進入優先 2PL/AL 模型；(6)依授予優先順序，進行資料欄位存取操作。

步驟 3： 透過主動式驗證平台 ORDER（順序）送出交易授權碼（DATA SERVER IP、使用讀取欄位權限、2PL/AL 優先順序、驗證服務碼）。

步驟 4： 私有雲資料驗證平台接收交易授權碼，進行權限驗證處理：(1)ID、RFID、登錄時間、使用者 IP、驗證授權碼記錄比對等驗證工作。(2)驗證符合後，回傳接受結果。(3)使用者依驗證授予權限（U、C、S、TS）進入分散式資料表存取資料。(4)使用者僅能存取授權之資料而無法讀取權限外資料。

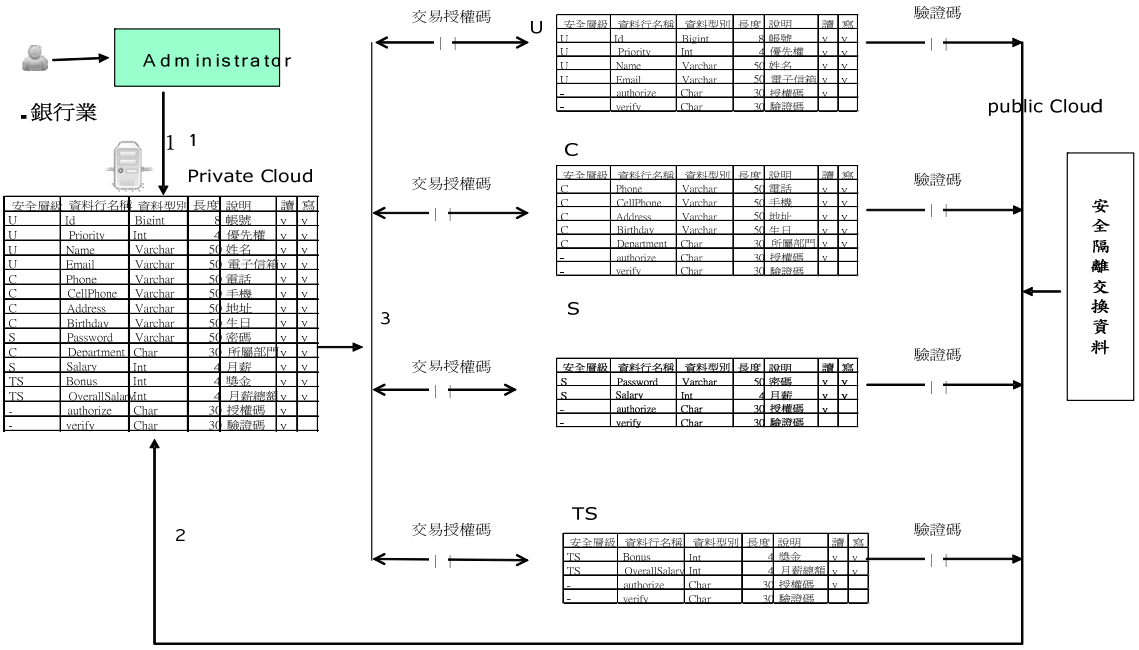

圖 18：資料表定義

本系統為保障個人隱私資料，以分散式多級安全資料庫管理，藉由主體的權限管理授權，可用來提供不同程度的資料公開等級，具分散而能靈活控管存取資料。因此本研究提出了一個適用於優先權多級排程控制（Priority/MLS/AL）並整合主動式驗證及分散式資料結構等技術形成本系統架構，最後將相關理論整合提出本研究系統架構以進行下一節的架構模擬及績效評估。

肆、架構模擬及績效評估

基於第參節所提出的系統架構，本節將透過實驗的方式，進行雲端環境資料安全存取情形之模擬，我們在雲端環境中賦予各種客戶端不同之權限，用來存取不同資料量之交易。其次，分散式資料庫中再分別以 2PL 及 AL 為排程原則進行交易處理，並比較兩者之優劣，最後，透過檢視表結合優先權及多級安全證實模型的可行性。

一、雲端環境資料安全存取模型模擬

依據第參節所提出的架構，透過 SQL Query Analyzer 實驗模擬，包含實驗環境及實驗內容。

（一）實驗內容

雲端環境資料安全存取模型主要分為 A、B 兩組實驗，其中實驗 A 是模擬兩階段鎖定（2PL）之排程機制的情形；而實驗 B 是模擬利他鎖定（AL）之排程機制的情形。

（二）A 組實驗（2PL 模擬）

在 A 組實驗又可分為四個子實驗，其中是為了判別長時間交易（long-running transaction; LRT）及短時間交易（short-running transaction; SRT）與讀寫之間的關係，其中關係如表 5 所示：

表 5：A 組實驗關係表

交易特性 / 子實驗	寫	讀/寫	讀/寫
實驗 A-1	TA11	TA12-讀	TA13-讀
實驗 A-2	TA21	TA22-讀	TA23-寫
實驗 A-3	TA31	TA32-寫	TA33-讀
實驗 A-4	TA41	TA42-寫	TA43-寫

實驗數據：每項實驗數據皆重複 128 次，每個值是由（$\frac{\text{總和}-\text{最大值}-\text{最小值}}{128}$）來計算，各項目所代表的涵義分別為，duration：交易所花的時間；CPU：交易花費 CPU 的量；read：交易讀取次數；write：交易寫入次數如表 6 所示。

表 6：A 組實驗數據表（註：四捨五入至小數點第一位）

項目 實驗		Duration	CPU	Read	Write	已傳送 TDS 封包	已接收 TDS 封包	累計的用戶端處理時間	累計的伺服器回應時間
實驗 A-1	TA11	197254.7	180423.5	89756	79234	3.6	88351.3	423.9	174466.3
	TA12	56.9	500.5	131	0	1.7	279.5	456.8	38937.3
	TA13	588	512.6	131	0	2.3	275.8	499.01	46449.5
實驗 A-2	TA21	238765.2	181245.2	89834	88923	4.3	97897.87	432.1	305563.8
	TA22	538.1	523.9	135	0.6	1.9	267.7456	434.2	49254.4
	TA23	交易資源已經被另一個處理鎖死並造成死結。							
實驗 A-3	TA31	238624.5	174256.8	89825	90926	5.0235	89005.6	543.6	254482
	TA32	交易資源已經被另一個處理鎖死並造成死結。							
	TA33	456.7	483.5	129	0	1.8	262.3	442.2	50562.9
實驗 A-4	TA41	264572.3	195687.3	92924	84828	5.8	97645.4	501.7	271961.1
	TA42	交易資源已經被另一個處理鎖死並造成死結。							
	TA43	交易資源已經被另一個處理鎖死並造成死結。							

發現只要有「寫」的交易（實驗 A-2、實驗 A-3 及實驗 A-4），就會有死結發生，這是因為 2PL 的並行性低。以 TA11 及 TA21 來看，發現 TA21 每個欄位的實驗數據都比 TA11 大，這表示碰到「寫」的交易都必須花更多時間來處理及等待。一旦長時間交易執行，「寫」的交易（TA23、TA32、TA42 及 TA43）因為較晚執行，所以都會產生死結。

（三）B 組實驗（AL 模擬）

在 B 組實驗也可分為四個子實驗，其中是為了判別長時間交易（LRT）及短時間交易（SRT）與讀寫之間的關係，其中關係如表 7 所示：

表 7：實驗 B 關係表

交易特性 / 子實驗	寫	讀／寫	讀／寫
實驗 B-1	TB11	TB12-讀	TB13-讀
實驗 B-2	TB21	TB22-讀	TB23-寫
實驗 B-3	TB31	TB32-寫	TB33-讀
實驗 B-4	TB41	TB42-寫	TB43-寫

實驗數據：每項實驗數據皆重複 128 次，每個值是由（$\frac{\text{總和}-\text{最大值}-\text{最小值}}{128}$）來計算，各項目所代表的涵義分別為，Duration：交易所花的時間；CPU：交易花費 CPU 的量；Read：交易讀取次數；Write：交易寫入次數如表 8 所示。

表 8：實驗 B 組數據表（註：四捨五入至小數點第一位）

項目 / 實驗		Duration	CPU	Read	Write	已傳送 TDS 封包	已接收 TDS 封包	累計的用戶端處理時間	累計的伺服器回應時間
實驗 B-1	TB11	66765.6	65722.7	92567	924	6.6	23500	167.2	144449
	TB12	146.5	75.7	112	0	1.3	13.9	32.8	17614
	TB13	125.4	73.6	112	0	1.5	17.6	44.5	13582.7
實驗 B-2	TB21	73454.2	68367.4	91463	945	7.8	25673.4	156.7	160081.9
	TB22	176.5	107.6	107	0	1.2	15.9	28.5	16540.1
	TB23	224.3	156.8	107	0	2.3	61.4	73	54965.2
實驗 B-3	TB31	73456.5	68345.9	91676	927	7.7	23367.5	133.9	163662.5
	TB32	207.3	127	103	0.1	2.3	51.3	53.5	50335.3
	TB33	178.2	122.1	103	0	1.2	16.6	28.2	18075.8
實驗 B-4	TB41	74578.3	64378.9	96821	923	7.6	23867.1	134.7	164670.7
	TB42	212.6	136.5	113	0	1.8	51.3	52.1	49800.9
	TB43	213.4	129.3	112	0	2.5	52.1	53.8	51189

若比較 TB11、TB21、TB31 及 TB41 這四個 LTT，發現 TB41 的值最大，這是因為 TB42 及 TB43 這兩個交易的性質皆是寫，相對要耗費的資源也比較多，導致 TB41 的實驗數據較大。在實驗 B 中沒有死結發生，這表示在 Altruistic_Lock 的機制中，在 LRT 執行中，不會因為 SRT 的加入而產生死結，這是因為 Altruistic_Lock 的交易的隔離等級為 READ COMMITTED，而且所宣告的 Cursor 為 OPTIMISTIC，這些程式碼都會提升交易的並行性及效率。

二、績效評估

針對 A、B 兩組實驗的數據進一步分析，包含效能改善分析及交易順序比較。

（一）效能改善分析

為了比較雲端資料庫兩階段鎖定及利他鎖定這兩種機制所耗用的資源，而有下面四個效能改善分析比較表。以下是實驗 A-1 及實驗 B-1 的比較，我們發現在效能改善的部分幾乎全部是正數，表示 AL 優於 2PL，所以 AL 作為排程的機制確實能獲得更佳的效能如表 9 所示。

表 9：實驗 A-1 及實驗 B-1 效能改善比較表

項目	實驗 A-1			實驗 B-1			效能改善		
	TA11	TA12	TA13	TB11	TB12	TB13			
Duration	177265.3	548.3	587	66663.4	142.3	137.2	62.39%	74.05%	76.63%
CPU	168625.6	473.8	512.2	63811.3	71.8	78	62.16%	84.85%	84.77%
Read	92356	131	132	92576	114	103	-0.24%	12.98%	21.97%
累計的用戶端處理時間	412.3	439.4	474.2	123.6	34.1	44.5	70.02%	92.24%	90.62%
累計的伺服器回應時間	172367.8	38756.3	44897.3	138992	16989	12896.5	19.36%	56.16%	71.28%

※效能改善：$\frac{(Experiment\,1-1)-(Experiment\,2-1)}{Experiment\,1-1}$

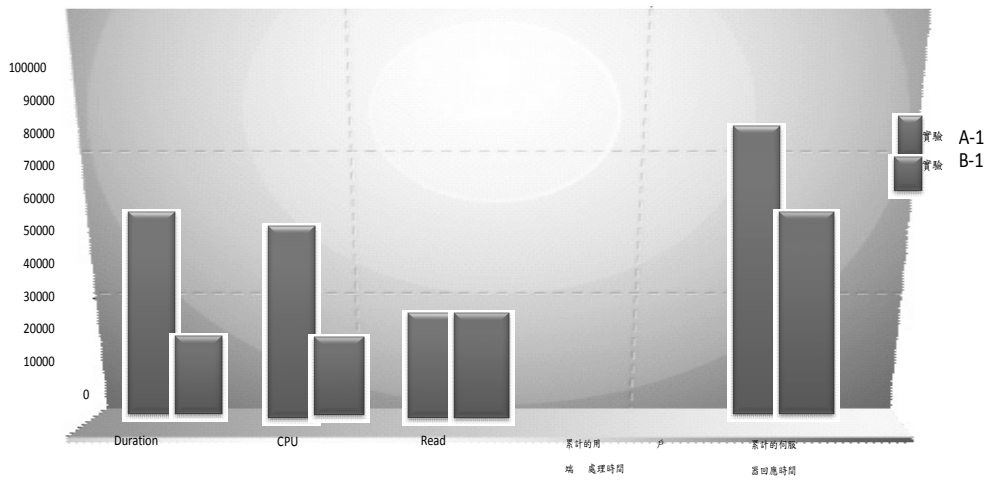

圖 19：實驗 A-1 及實驗 B-1 效能改善比較圖

以下是實驗 A-2 及實驗 B-2 的比較，我們發現在效能改善的部分全都是正數，明顯表示 AL 優於 2PL，所以 AL 作為排程的機制確實能獲得更佳的效能。但由於 TA23 為死結沒有數據，故 TB23 無法針對其改善的效能做比較如表 10 所示。

表 10：實驗 A-2 及實驗 B-2 效能改善比較表

測試項目	實驗 A-2			實驗 B-2			效能改善		
	TA21	TA22	TA23	TB21	TB22	TB23			
Duration	233576.3	567.4	交易在資源上已經被另一個處理程序鎖死。	71475.6	179.4	222.1	69.40%	68.38%	TA23 為死結，故沒有數據。TB23 以利他鎖定避免死結發生。
CPU	188646.3	527		68813.3	116.2	136.6	63.52%	77.95%	
Read	91768	129		91598	103	107	0.19%	20.16%	
累計的用戶端處理時間	452	435.3		147.3	27.2	68	67.41%	93.75%	
累計的伺服器回應時間	255679.7	51376.3		161121.3	16532.3	54955.3	36.98%	67.82%	

※效能改善：$\frac{(Experiment\,1-2)-(Experiment\,2-2)}{Experiment\,1-2}$

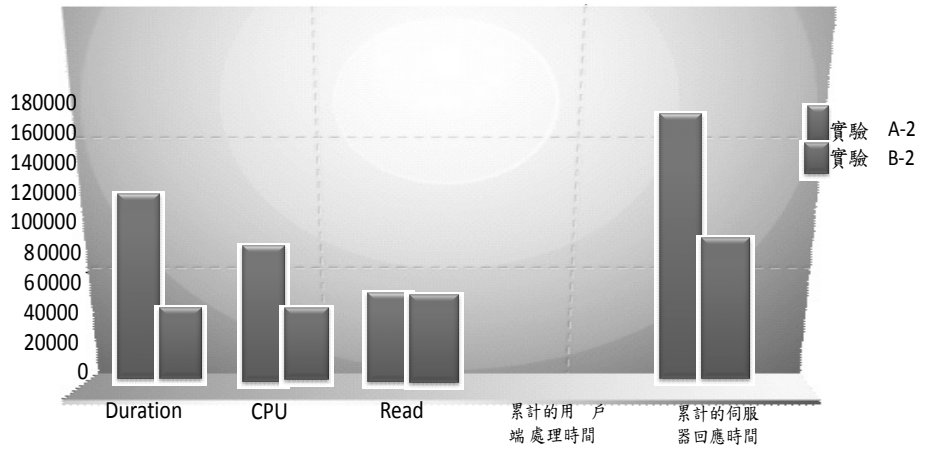

圖 20：實驗 A-2 及實驗 B-2 效能改善比較圖

以下是實驗 A-3 及實驗 B-3 的比較，我們發現在效能改善的部分全都是正數，明顯表示 AL 優於 2PL，所以 AL 作為排程的機制確實能獲得更佳的效能。但由於

TA32 為死結沒有數據，故 TB32 無法針對其改善的效能做比較如表 11 所示。

表 11：實驗 A-3 及實驗 B-3 效能改善比較表

測試項目	實驗 A-3			實驗 B-3			效能改善		
	TA31	TA32	TA33	TB31	TB32	TB33			
Duration	248805.1	交易在資源上已經被另一個處理程序鎖死。	556.3	72548.2	210.3	181.2	70.84%	TA23 為死結，故沒有數據。TB33 以利他鎖定避免死結發生。	66.65%
CPU	184123.7		528.1	68115.1	142	113.2	63.01%		78.62%
Read	91798		129	91791	113	106	0.01%		19.23%
累計的用戶端處理時間	450.3		448.1	139.9	53.1	30.2	68.93%		93.25%
累計的伺服器回應時間	254503		50559.8	163677.3	50339.2	18039.3	35.69%		64.25%

※效能改善：$\frac{(Experiment\ 1-3)-(Experiment\ 2-3)}{Experiment\ 1-3}$

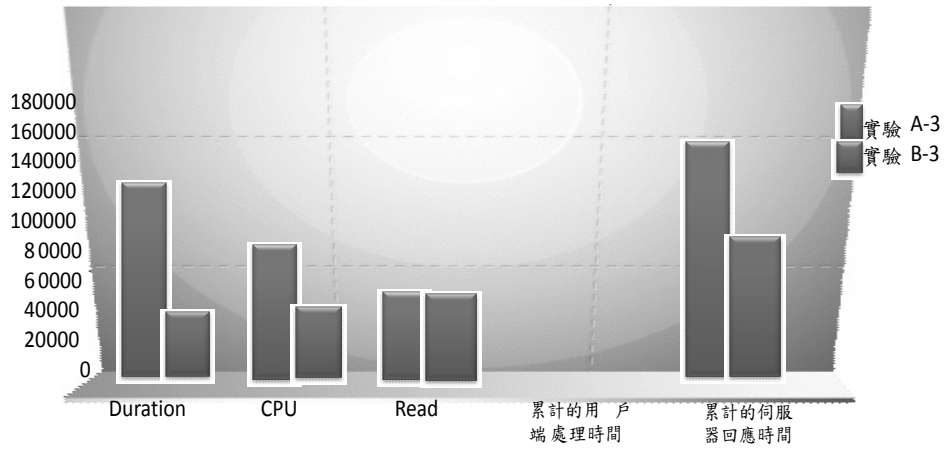

圖 21：實驗 A-3 及實驗 B-3 效能改善比較圖

以下是實驗 A-4 及實驗 B-4 的比較，我們發現在效能改善的部分全都是正數，明顯表示 AL 優於 2PL，所以 AL 作為排程的機制確實能獲得更佳的效能。但由於 TA42 及 TA43 都為死結沒有數據，故 TB42 及 TB43 無法針對其改善的效能做比較如表 12 所示。

表 12：實驗 A-4 及實驗 B-4 效能改善比較表

測試項目	實驗 A-4			實驗 B-4			效能改善		
	TA41	TA42	TA43	TB41	TB42	TB43			
Duration	270431.5	交易在資源上已經被另一個處理程序鎖死。	交易在資源上已經被另一個處理程序鎖死。	72355.4	209.7	226.1	73.24%	TA42 TB42 以利他鎖定避免死結發生。為死結，故沒有數據資料。	TA43 TB43 以利他鎖定避免死結發生。為死結，故沒有數據資料。
CPU	189911.3			66895.4	130.5	127.9	64.78%		
Read	91745			91601	105	120	0.16%		
累計的用戶端處理時間	502.4			149.3	53.1	54.8	70.28%		
累計的伺服器回應時間	280022.3			166753.4	48978.5	51206.3	40.45%		

※效能改善：$\frac{(Experiment\,1-4)-(Experiment\,2-4)}{Experiment\,1-4}$

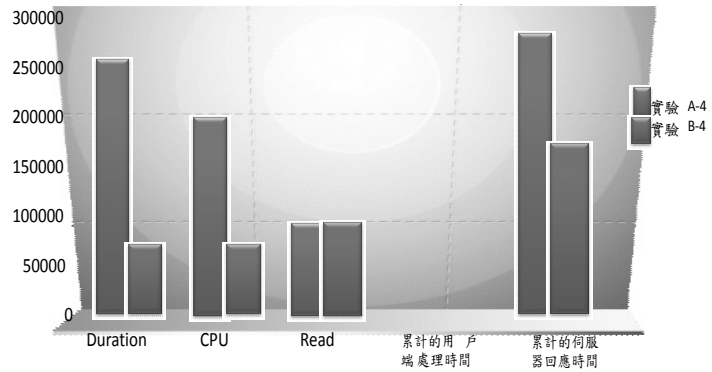

圖 22：實驗 A-4 及實驗 B-4 效能改善比較圖

（二）交易順序比較

由交易順序的比較，可以窺知排程器調度的行為。如果僅看實驗 A-3 及實驗 A-4 這兩個實驗，理論上 TA32 及 TA42 會先完成，不過卻是 TA33 及 TA41 最先完成，這表示這些「寫」的交易（TA32、TA42 及 TA43）不但導致死結，而且還會擾亂交易完成順序。

在實驗二交易完成順序的部分，都是第二個、第三個及第一個（例：TB12、TB13 及 TB11），而且沒有死結產生，這表示在 AL 的排程機制中，LRT 執行中，不會因為 SRT 的加入而改變執行順序，表示 AL 能提升交易並行性，並且保持交易完成順序正常執行如表 13 所示。

表 13：交易順序比較表

A 組實驗		
實驗 A-1	執行順序	TA11→TA12→TA13
	交易完成順序	TA12（○）→TA13（○）→TA11（○）
實驗 A-2	執行順序	TA21→TA22→TA23
	交易完成順序	TA22（○）→TA23（死結）→TA21（○）
實驗 A-3	執行順序	TA31→TA32→TA33
	交易完成順序	TA33（○）→TA32（死結）→TA31（○）
實驗 A-4	執行順序	TA41→TA42→TA43
	交易完成順序	TA41（○）→TA42（死結）→TA43（死結）
B 組實驗		
實驗 B-1	執行順序	TB11→TB12→TB13
	交易完成順序	TB12（○）→TB13（○）→TB11（○）
實驗 B-2	執行順序	TB21→TB22→TB23
	交易完成順序	TB22（○）→TB23（○）→TB21（○）
實驗 B-3	執行順序	TB31→TB32→TB33
	交易完成順序	TB32（○）→TB33（○）→TB31（○）
實驗 B-4	執行順序	TB41→TB42→TB43
	交易完成順序	TB42（○）→TB43（○）→TB41（○）

（三）雲端分散式資料庫存取安全控制

在第參節提到五種方法可以確保多級安全資料庫的機密性，本研究的架構類似於限制的檢視表，其特色就是依使用者的權限（Need to Know），透過「檢視表」限制他們所能看到的資料欄位如圖 23 所示。

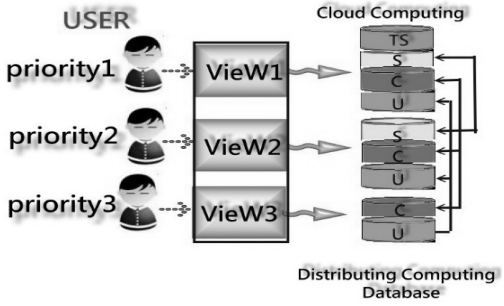

圖 23：Restricted Views 架構

第貳節中論及雲端環境中越權存取的攻擊日益嚴重問題（如文獻所述，網路銀行客戶資料外洩事件、2011 年 SONY PSN 遭駭客入侵個資外洩、IT 公司員工盜賣資料等事件），本研究將雲端環境資料安全存取模型進行模擬，並且順利完成兩組實驗，可以有效的遏止類似的越權存取攻擊。

經由實驗結果顯示，AL 能更快更有效的完成交易，因此特別適合雲端運算中智慧型手機及平板電腦這類計算機資源少的客戶端。因為雲端環境中大量資料存取死結問題，所以它能盡快的將 SRT 完成（commit），減少交易重新執行（rollback），避免死結發生。最後將主動式身分驗證、安全隔離與資料交換、優先權多級排程控制、分散式存取及 RC4 加解密技術相互結合，使本系統架構更為完備並能完成模擬評估。

伍、結論

為確保雲端中用戶資料機密性和完整性，同時支援動態資料優先排程控制以提高海量資料存取效率，強化用戶資料傳輸和儲存安全，在雲端環境中為使分散式資料庫存取更安全更有效率，本研究提出一套新模型暨績效評估方法以達到資料安全存取目的，包括「主動式身分驗證」、「安全隔離與資料交換」、「優先權多級排程控制」、「分散式存取」及「RC4 加解密」等設計，期找出解決私有雲越權存取問題。用戶於公有雲必須透過私有雲主動驗證方法授予授權碼，驗證符合後才能取得交易驗證並經加密處理，透過安全隔離與資料交換模型，才能進入私有雲依授權權限存取資料，並整合優先權多級安全排程控制，執行優先權多級安全排程，最後以分散式分級授權存取資料。實驗顯示，利他排程原則能使分散式資料庫存取更有效率。經由檢視表將優先權及多級安全相互結合模擬，達到資料安全存取的目的。通常要做好存取控制，只要做好「讀」的控管，即可解決大部分不當存取的威脅，本研究並發現，做好「寫」的排程序列化，即可有效避免死結發生。研究顯示此一雲端安全資料存取架構能有效的遏止越權存取，也可提高交易並行性，增進資料存取效能性，透由實驗結果顯示，私有雲以優先權多級安全及分散式資料庫存取方式，AL 能更快更有效的完成交易，能盡快的將費時較短的交易完成，減少交易重新執行，避免死結發生。經由二組實驗比較，驗證私有雲分散式資料庫中「優先權多級排程控制」AL 優於 2PL，以 AL 作為排程的機制確實能獲得更佳的效能，說明了本系統架構之可用性。因應雲端運算已呈海量資料庫的趨勢，建議未來研究方向可朝雲端海量資料搜尋最佳化排程演算機制進行研究，應可解決雲端海量資料搜尋效率。

參考文獻

陳志誠、王瀞慧（2011），『金融機構雲端運算架構下客戶資料防護之探討』，*2011 產業資訊應用暨個案競賽（CIIA 2011）*，明志科技大學，台灣，10 月 28 日。

陳志誠、林淑瓊、李興漢、許派立（2009），『資訊資產分類與風險評鑑之研究－以銀行業者為例』，*中華民國資訊管理學報*，第十六卷，第三期，頁 55-84。

陳志誠、宋子傑（2005），『在無線網路環境中基於用戶優先權與利他鎖定之多級安全資料庫存取控制』，*資訊安全通訊*，第十一卷，第三期,頁 51-67。

葉桂珍、張榮庭（2006），『企業之資訊安全策略與其產業別及資訊化程度關係探討』，*中華民國資訊管理學報*，第 13 卷，第 2 期，頁 113-143。

劉家驊、洪士凱（2010），『雲端運算資料安全防護機制之研究』，*2010 電腦視覺、影像處理與資訊技術研討會（CVIPIT 2010）*，清雲科技大學，台灣，6 月 9 日，頁 100-109。

Bell, D.E. and LaPadula, L.J. (1976), '*Secure computer systems: Unified exposition and multics interpretation*', Technical RePort MTR-2997, Mitre corp, Bedford MA. Brodkin, J. "*Gartner: Seven Cloud-computing Security Risks*," http://www.networkworld.com/news/2008/070208-cloud.html, Network World, 2008/07/02.

CSA (2012), Security Guidance for Critical Areas of Focus in Cloud Computing v3.0.https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf

Chen, P.S., Li, S.H. and Liu, Y.K. (2010), 'Scheduling the access to multi-level secure databases in a wireless network environment', International Journal of Innovative Computing, Information and Control, Vol. 6, N0. 12, pp. 5381-5403.

David, R. and Son, S.H. (1993) 'A secure two phase locking protocol', *Proceedings of the twelfth IEEE Symposium on Reliable Distributed Systems (SRDS 1993)*, Princeton, NJ, USA, October 6-8, pp.126-135.

European Network and Information Security Agency (2010), 'Cloud computing: benefits, risks and recommendations for information security', *European Network and Information Security Agency*, available at http://www.enisa.europa.eu/act/rm/files/eliverables/cloud-computing-risk-assessment (accessed 17 February 2014).

Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D. and Chandramouli, R. (2001), 'Proposed NIST standard for role-based access control', *ACM Transactions on Information and Systems Security*, Vol. 4, No. 3, pp. 224-274.

Gartner(2010), http://www.garther.com/technology/home.jsp.

Garcia-Molina, H., Ullman, J. and Widom, J. (2008), *Database Systems: The Complete Book*, Pearson Education, India.

Goodhue, D.L. and Straub, D.W. “Security Concerns of System Users:A Study of Perceptions of the Adequacy of Security Measures,” *Information & Management* (20:1,January), 1991, pp.13-27.

Hinke, T. and Schaefer, M. (1975), Secure Data Management System. Rome Air Development Center Technical Report, *System Development Corp*., pp. 75-266.

Jung, D. (2001). Transformational and transactional leadership and their effects on creativity in groups. *Creativity Research Journal*, pp. 13, 185-195.

Jung, J.-Y., J.L. Qiu and Y.-C. Kim (2001) ‘Internet Connectedness and Inequality: Beyond the “Divide”’, *Communication Research* 28(4): pp. 507-535.

Kim, H.W., Park, D.S., Rhee, H.K. and Kim, U.M. (2001), ‘Advanced transaction scheduling protocol for multilevel secure database in wireless mobile network environment’, *Proceedings of Joint fourth IEEE International Conference on ATM (ICATM 2001) and High Speed Intelligent Internet Symposium*, Seoul, Korea, April 22-25, pp. 240-244.

Kankanhalli, A., Tan, B.C.Y. and Wei, K.K. “Contributing knowledge to electronic knowledge repositories:An Empirical Investigation,” Mis Quarterly, Volume 29, Number 1, pp. 113-143.

Lang, U. (2010), ‘OpenPMF SCaaS: Authorization as a service for cloud & SOA applications’, *Proceedings of the second IEEE International Conference on Cloud Computing Technology and Science (CloudCom 2010)*, Indianapolis, Indiana, USA, November 30-December 3, pp. 634-643.

Lewis, S. and Wiseman, S. (1997), ‘Securing an object relational database’, *Proceedings of the thirteenth Anunual Computer Security Applications Conference (ACSAC 1997)*, San Diego, California, USA, December 8-12, pp. 59-68.

Lin, J., Lu, X., Yu, L., Zou, Y. and Zha, L. (2010), ‘Vega Warden: A uniform user management system for cloud applications’, *Proceedings of the 2010 IEEE International Conference on Networking, Architecture and Storage (NAS 2010)*, Macau, China, July 15-17, pp. 457-464.

Niemeyer, R. E. (1997), ‘Using Web technologies in two MLS environment: A security analysis’, *Proceedings of the thirteenth Anunual Computer Security Applications Conference (ACSAC 1997)*, San Diego, California, USA, December 8-12, pp. 205-214.

Pang, H., Carey, M.J. and Livny, M. (1995), ‘Multiclass query scheduling in real-time database systems’, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 7, No. 4, pp. 533-551.

Pfleeger, C.P. and Pfleeger, S.L. (2002), *Security in Computing*, Prentice-Hall Int., USA.

Salem, K., Garcia-Molina, H. and Shands, J. (1994), 'Altruistic Locking', *ACM Transactions on Database Systems*, Vol. 19, No 1, pp. 117-169.

Sandhu, R.S., Coyne, E.J., Feinstein, H.L. and Youman, C. (1996), 'Role-based access control models', *IEEE Computer*, Vol. 29, No 2, pp. 38-47.

Wood, C., Summers, R.C. and Fernandez, E.B. (1979), 'Authorization in multilevel database models', *Information Systems*, Vol. 4, No. 2, pp. 155-161.