國立交通大學

理學院網路學習學程

碩士論文

跨網域之校務單一登入系統

Cross Domain School Administrative-affair Single sign-on System

研究生：林裕峰

指導教授：蔡文能　教授

中華民國　九十七　年五月

跨網域之校務單一登入系統

Cross Domain School Administrative-affair Single sign-on System

研 究 生：林裕峰　　Student：Yung-Feng Lin

指導教授：蔡文能　　Advisor：Wen-Nung Tsai

國 立 交 通 大 學

理學院 網路學習學程

碩 士 論 文

A Thesis

Submitted to Degree Program of E-Learning

College of Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master

In

Degree Program of E-Learning

July 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年五月

跨網域之校務單一登入系統

研究生:林裕峰　　指導教授:蔡文能教授

國立交通大學理學院網路學習學程

摘　　要

隨著網際網路的急速發展，教育行政部門為了提供多元化的網路服務，紛紛開發以網頁為基礎的應用程式(Web-Based Application)來滿足使用者需求，但因每個網站系統開發平台或語言的不同，因此擁有各自的安全認證的機制，導致使用者在取得每一項服務前，均需提供系統不同的登入帳號、密碼等相關資訊，這對使用者來說是相當不便的；相對的，若使用者把所有服務網站的帳號、密碼都設成一樣，在安全上將會有相當大的風險。再者，國小行政人員每學年的工作通常都會有所調整及異動，面臨眾多網站的服務，行政人員往往無法有效管理登入的帳號及密碼，造成業務移交不順利。

本研究將針對以上問題，提出一種以網頁為基礎，且能達到跨網域之校務單一登入系統(Cross Domain School Administrative-affair Single sign-on System,CDSASS)，讓使用者只要通過一次身分認證，就可以取得不同網站系統的服務，而不需要再重複身分驗證的動作。另外，我們依照國小的行政運作體制，以角色為基礎的存取控制（Role_Based Access Control,RBAC）整合使用者、群組、團隊與角色的對應關係，讓業務帳號、密碼移交能夠更順利、簡便且任務執行上也更有效率。

關鍵字: 網頁式單一登入、跨網域單一登入、角色為基礎的存取控制、代理登入

Cross Domain School Administrative-affair Single Sign-On System

Student: Yung-Feng Lin Advisor: Wen-Nung Tsai

Degree Program of E-Learning
College of Science
National Chiao Tung University

ABSTRACT

With the rapid development of the Internet, educational and administrative departments in order to provide a wide range of Internet services, have developed a web-based applications to meet the needs of users, but each site or system development platform for the different language, have their own security authentication mechanisms, leading users to get each of the services, required a different account, password and other information, which the user is very inconvenient; As to, if all users of the site services account, password are set to the same, the security will be considerable risks. Furthermore, the primary school administrative staff of each school year, the work will normally be adjusted and changes facing the many sites of service, executives are often unable to effectively manage the login ID and password, resulting in the transfer of business is not smooth.

This study will address the above problems, propose a web-based, and can achieve Cross Domain School Administrative-affair Single sign-on System. Let users only pass an identity authentication,can obtain the services of different sites, and does not need to repeat movements that the identity verifies again. In addition, we are according to the administrative operation system of the primary school, role-based access control combine users, groups, teams and corresponding relation of role, let business accounts, passwords to transfer more smooth and Easily, on the implementation of tasks more efficient.

Keywords: WSSO, CDSSO, RBAC, Agents Sign

誌 謝

在此首先要感謝蔡文能教授兩年來的諄諄教誨，讓我在研究的過程中學到了嚴謹的研究方法和正確的學習態度，使我獲益匪淺，在論文研究上由於老師的啟發與幫助，才使得本論文能夠順利完成，在此由衷的感謝。在本論文撰寫期間，我所服務的單位主管及同事們對本論文也提出許多寶貴的建議，另外也包括我的兩位同學吳振遠與顏士哲，對本系統完成建置也貢獻良多。

最後，特別感謝我的另一半中文，謝謝妳對我的包容與支持，在我最無助的時候給予我鼓勵，最盲目的時候給予我指引，最孤獨的時候給予我陪伴，最失落的時候給予我慰藉，感謝妳無怨無悔的陪我渡過求學生涯。

林裕峯 謹誌

2008年5月

目錄

表目錄

圖目錄

第一章 緒論

隨著網際網路的急速發展，教育行政部門依各自需求，開發以網頁為基礎的應用程式(Web-Based Application)提供服務，導致使用者必須向各應用程式完成註冊，取得帳號、密碼後，才能使用該網站提供的服務。在國小行政工作中，行政人員每天都要登入到教育部或縣市教育局所建置的網站，進行查詢或填報作業等工作，但因每個網站系統開發平台或語言的不同，因此擁有各自的安全認證的機制，導致使用者必須反覆的輸入帳號、密碼，甚至為了方便使用相同的或過於簡單的帳號、密碼，造成資訊安全的一大漏洞。再者，行政人員每學年的工作通常都會有所調整及異動，面臨眾多網站的服務，行政人員往往無法有效管理登入的帳號及密碼，造成業務移交不順利。

單一登入(Single Sign-On,SSO)技術的出現[1]，讓使用者只通過一次的身分驗證後，即可存取所有已授權的網路資源，而不必再重複身分驗證的動作，且可以將帳號、密碼整合為一組，提高使用者帳號、密碼的安全性，以及管理上的方便。另外，以角色為基礎的存取控制（Role_Based Access Control,RBAC）[2]是一個管理方便、擴展性高以及具有彈性的存取控制方法，正可以簡化職務異動的移交業務。不過由於大部分的單一登入系統建置的成本太高，且傳統的 RBAC 並不適用於國小行政體系。因此本研究將針對此問題，提出一個可解決這些問題的服務架構，並根據此一架構進行實作，來證明此一架構的可行性。

1.1 研究動機

隨著科技成熟的腳步加促，各教育部門也以電子化為目標，根據各自的需求，先後建立多元的網路服務，但因為服務網站擁有各自的安全認證機制，因此各校行政人員得依照各部門的要求，完成使用者註冊，才可使用該網站的服務。然而，時間一久，各部門建立之網站系統越來越多，再加上校內發展的服務主機亦不在少數，造成使用者必須重複登入並記憶多組帳號、密碼，才可以順利使用各項服務；而使用者為了怕忘記或方便起見，在每個網站都使用同一組或過於簡單的帳號、密碼，這樣不僅造成了使用者的不便，也提高了資訊安全上的風險。再者，由於國小教師人員編制的問題，行政人員的職務異動也相對的頻繁，因此易造成帳號、密碼交接不實，影響業務之推展。

雖然很多學者提出跨網域單一登入的解決方法，不過其技術門檻與建置成本較高，亦不見得適用於國小校務行政，因為教育部門所提供的服務網站並不允許更改其登入介面或

認證機制；傳統的 RBAC 提供了一個管理方便、擴展性高以及具有彈性的存取控制方法，但應用在國小校務行政現場卻仍嫌不夠完備，例如：對於相同屬性的使用者，必須逐一指派相同角色給他們，若使用者角色需異動時，仍要逐一重新指派角色，非常沒有效率。另外在某些狀況下，上級指派之任務和填報工作需要各處室人員分工合作才能完成，而每個人負責的任務則依照自身被指派的角色來決定，傳統的 RBAC 並無法達到分工合作的機制。

我們希望在不更動現有系統環境下，提供讓使用者只要成功登入一次網站，即可在不同網站之間存取各種網路服務的架構，並透過整合群組（Groups）、團隊及任務（Teams and Tasks）的概念，除了讓角色管理權限機制適用於國小校務行政，也讓行政人員之職務調整，不必去交接各服務網站的帳號、密碼，只要調整新、舊任職者應有的職務即可，進而減輕行政交接的繁瑣與提升行政工作效率。然而，運行此架構並不會增加學校經費負擔，亦不需改變使用者之使用經驗，更不需修改服務網站認證機制，亦可達到跨網域單一登入的功能。

1.2 研究目的

本研究的目的為利用現有的硬體設備，在不變更使用者環境、不增加額外費用，且兼顧使用者經驗、安全性及易於部署的考量下，提出一個適用於國小行政現場的跨網域單一登入系統架構。本系統架構預期可達到之功能如下：

1. 網頁式單一登入（Web-Based Single Sign-On）介面：
   此功能提供使用者透過瀏覽器（Browser）便可由單一入口登入取得各網站服務，而不需要記憶多組不同的帳號、密碼，以減少資安問題的發生。另外，管理者可以進行使用者資料管理、代登入網站管理、授權與權限管理與稽核管理等功能。
2. 保有各網站的認證機制：
   由於系統採用代理登入（Agents Sign）的方式來替使用者進行登入動作，因此可以保有原服務網站的認證機制，而不需要進行登入介面的修改。
3. 富有彈性的權限管控機制：
   我們依照國小的行政運作體制，以角色為基礎，運用使用者、群組、團隊與角色的對應關係，搭配權限指派給角色的概念，讓業務帳號、密碼移交能夠更順利、簡便且任務執行上也更有效率。
4. 行政業務定期填報通知
   此系統除了省去記憶多組帳號、密碼的困擾，也提供行政人員自行針對業務需要，

設定每月定期填報的通知，以便提醒行政人員定期完成填報業務。

希望透過此架構提供的功能，來簡化行政業務，降低行政負擔，提高行政效率。最後，我們也將根據此一系統架構進行實作，來證明此一架構的可行性。

1.3 研究範圍

本研究將範圍設定在桃園縣國小教育行政現場，並蒐集縣內相關的行政填報與個人服務網站，再完成代理登入程式設計，以代替使用者進行登入服務網站的程序。另外，為了不增加額外的費用，本文實作之系統，採用開放原始碼（Open Source）的伺服器軟體（Apache）及伺服器端網頁程式語言（PHP）來設計跨網域單一登入系統，並以資料庫伺服器（MySQL）來儲存相關資料，最後透過 Cookie來儲存相關的會議金鑰（Session Key），再利用安全性高且被廣泛使用的 AES、RSA、SHA-1等三種演算法，來處理網頁內容與使用者認證的加解密及驗證程序，以強化系統的安全性。

1.4 章節介紹

本論文共分為六個章節，第一章的緒論說明本篇論文研究的出發點，第二章則針對單一登入的相關技術做說明。第三章將對各學者所提跨網域單一登入的方法分析比較。第四章為我們所提出的跨網域單一簽入系統架構。第五章則對本研究所提的方法設計、實作，並對系統做評估分析。最後，在第六章中依據先前章節所介紹之技術或系統，與本系統做比較分析，並對本研究做結論與未來的研究方向。

第二章 背景知識

本章節主要介紹及說明本論文主題相關的理論與技術，其內容包括網頁式單一登入(Web-Based Single Sign-On)所解決的問題及優點；HTTP Cookie 與 Session 運作模式有何不同，並如何應用在身分認證上；對稱式加密演算法（Symmetric Cryptographic Algorithm)、非對稱式加密演算法（Asymmetric Cryptographic Algorithm）與單向雜湊函數（One-Way Hash Function）的介紹；安全通訊層（Secure Socket Layer）的架構與提供的安全服務；角色為基礎的存取控制(Role-Based Access Control)與其他存取控制策略的比較，最後談及 Transcoding Reverse Proxy 的運作模式與運用在單一登入上的特性。

2.1 網頁式單一登入(Web-Based Single Sign-On)

隨著網際網路的發達，更多網路應用程式(Web Application)被開發，並組成了龐大的服務網路脈絡，讓使用者享受網路科技帶來之便利，也創造了龐大的商機。但由於服務網站過於分散，且各自擁有獨立的資料庫與認證機制，讓使用者必須重複登入各網站，才能取得權限使用該服務，其多重登入示意圖，如圖2.1。

分散的網路服務具有下列幾項缺點：[1]

1、使用者操作不便。

2、系統帳號、密碼管理不易，且造成資安問題的發生。

3、系統效能不佳。

4、系統管理成本增加。

5、異質系統資源整合不易。

為了解決這些問題，單一登入（Single Sign On，SSO）機制就隨著因應而生。

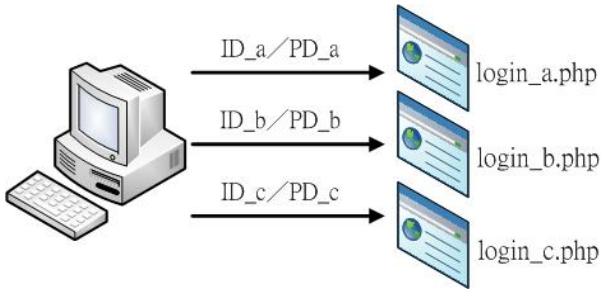

圖 2.1 多重登入示意圖

「單一登入」（Single Sign-On，SSO）[1][5][19]即只要一組識別碼和密碼就可以讓使用者登入不同的系統，使用者就不需要為了不同的系統記住多組識別碼和密碼。網頁式單一登入系統（Web-Based Single Sign-On,WSSO）[3][24]，是指使用者透過瀏覽器，在經過一次的身分驗證後，便可以存取不同網站的系統資源，而不需要再重複輸入帳號、密碼。由於以網頁為基礎(Web-Based)的介面，具有跨平台、介面簡單、無國界的特性，在加上單一登入的優點：

1、使用者用一組帳號、密碼即可存取不同網站之服務，而不需要記憶多組帳號、密碼，也不用反覆執行登入的動作，以節省使用者時間。

2、集中管理使用者帳號及密碼，以遏止資訊安全問題發生。

3、透過一次性認證，即可取得各網路應用程式資源，大大提升系統效能。

4、經由集中式的認證機制與權限控管，減少維護的時間，以降低管理成本。

5、串聯不同網域的網路應用程式，以整合異質系統之資源。

因此，很快的使網頁式單一登入系統（Web-Based Single Sign-On）被廣泛的應用。網頁式單一登入示意圖如圖2.2。

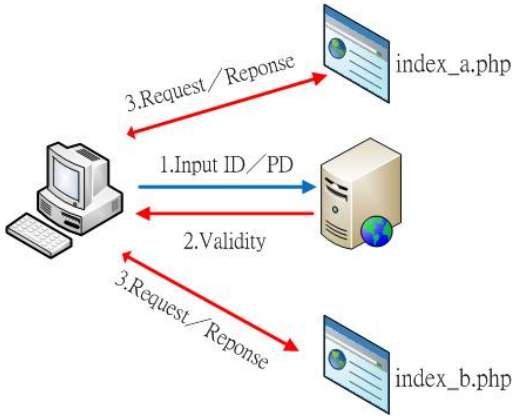

圖 2.2 網頁式單一登入示意圖

2.2 HTTP Cookie與Session

隨著網際網路及全球資訊網的快速發展，越來越多人透過瀏覽器來尋找他們所需的服務與資源，而瀏覽器主要是透過 HTTP 協定（Hypertext Transfer Protocol）來連結網頁伺服器取得網頁；HTTP 是以請求與回應模型為基礎[3][26]，即當用戶端透過瀏覽器發出一個服務請求（Request），伺服器就會回應（Respose）該請求，當每次的請求與回應完成後，使用者的瀏覽器和伺服器間的連線就會中斷，因此我們也稱 HTTP 是一種無狀態（Stateless）的協定，而 Cookie 正是一種可以解決 HTTP 無狀態（Stateless）協定的方法，且因為 Cookie 的傳遞都是透過 HTTP Header 來實行，所以也有人稱它為HTTP Cookie[4][27]。由於伺服器（Server端）可以透過 Cookie，將一小段資訊儲存在用戶端的電腦裡（Client端），以便下次連線時可以存取，所以常被應用在認證、追蹤與記錄使用者的資訊，因此很多網站都會使用 Cookie 來維持使用者的連線狀態。例如：線上購物網站、線上銀行、個人化的服務網站(iGoogle)…等。

Cookie 一般而言可以分為持續性（Persistent）與暫時性（Transient），兩者的差別在於持續性 Cookie （Persistent Cookie）會存在使用者的電腦，直到使用期限（Expiration Date）到了或使用者自行刪除才會消失；而暫時性 Cookie （Transient Cookie）是暫存於伺服器中，一旦使用者關閉瀏覽器，暫時性 Cookie 就會消失，因此，暫時性 Cookie 也常被稱為「Session Cookie」。

Cookie的規格有六個部分，這六個部分分別為：name、value、expire、domain、path、secure，其結構分別描述如下：

◆Name

Name 是 Cookie 的名稱，其內容不可以包含逗號、分號及空白字元。

◆Value

Value 是上述 Cookie name 的值，這個字串會以16進制編碼後存入 Cookie 。這些參數中，只有 Name 和 Value 是必需的。

◆Expire

Expire 是設定 Cookie 使用期限的參數， Expire 的值必須符合 GMT 格式;若不指定此參數，則 Cookie的使用期限，就是關閉瀏覽器時。

◆Domain

Domain 為設定那個網域可存取 Cookie，只有在與此設定相同的網域或子網域內才能存取該 Cookie 。如果不指定此參數， Domain 就指定為設定該 Cookie 的網頁所在的網域。

◆Path

Path 為指定可以存取該 Cookie 的有效路徑，URL 經過 Domain 的比對符合之後，再與 Path 的設定比對，符合路徑的才能存取該 Cookie 。一般最常用的的 Path 設定值為”/”。如果不指定此參數，Path 就指定為設定該 Cookie 的網頁所在的路徑。

◆Secure

Secure 是指定是否使用安全認證，若 Secure 被設定，則Cookie只會在使用HTTPS（HTTP over SSL）的連線間傳輸。若以 PHP 語法來設定 Cookie 時，Secure 值為1，代表須使用 HTTPS 連線；若為0，則代表不使用 HTTPS 。

Session 的功能和 Cookie 一樣，都是用來記錄使用者的習慣與資訊，而其不同點在於，Cookie 的資訊是存在用戶端（Client），Session 是存在伺服器端（Server）。一般來說，Session 在一段時間沒有再開啟另一個網頁，或是執行登出功能，工作階段就會結束，為了能夠長期記錄這些資訊，Session 都會搭配 Cookie 來使用，因此 Session 也算是一種 Cookie 的改良;不過有時用戶端會了安全考量，關閉瀏覽器 Cookie 的功能，使得瀏覽器必須使用網址改寫(URL Rewriting)的方式來傳送 Session ID，也讓有心人士有機會從網址列中取得某些資訊，進行進程追蹤，造成資訊安全問題。例如：
https://testsso.myftp.org/admin.php?sid=1A2A0BFA32B0025C8BA00A5E697A9D12

由於 Session 的資訊是儲存在伺服器端（Server），因此也常應用在使用者身分認證上。透過 Session 認證流程如圖2-3。

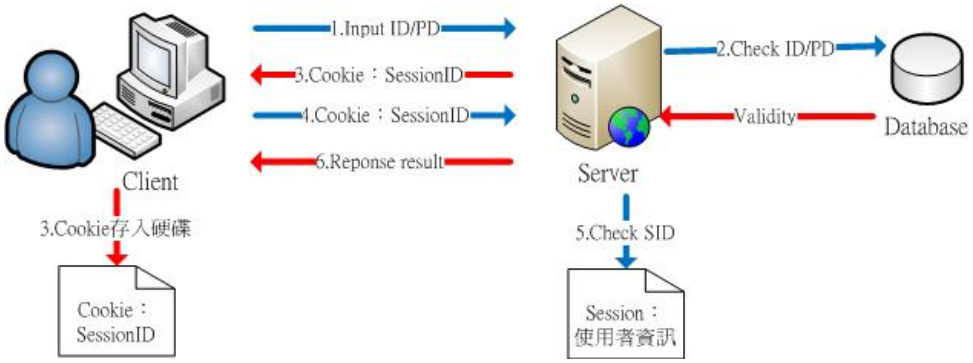

圖 2.3 Session 認證流程圖

1、用戶端在登入網頁中輸入帳號、密碼。

2、伺服器核對帳號、密碼與資料庫的資料是否一致。若不一致則重導回登入畫面。

3、伺服器產生一組隨機的 Session ID並經過 MD5加密後，存入 Cookie中回傳給用戶端。

4、用戶端透過瀏覽器使用 Cookie、POST 或 GET 的方式將加密過的Session ID傳給伺服器並請求服務。

5、伺服器依照 Session ID 取出參數值，並確認用戶端是否已完成認證及 Session ID存活時間是否過期。

6、回應請求的結果。

2.3 加解密技術

本節針對本系統會使用到的加解密技術分成三個部份作一簡單介紹。第一部分為對稱式加密演算法；第二部分為非對稱式加密演算法；第三部分則為單向雜湊函數。

對稱式密碼演算法（Symmetric Cryptographic Algorithm）

對稱式密碼演算法，亦稱為私密金鑰加密演算法[15]。此方法在進行加、解密時，都使用同一把金鑰。所以不知道金鑰的第三者，將無法得知加密後的資料內容，因此才被稱為對稱式加密演算法。由於加、解密時都用同一把金鑰，所以處理速度快是其優點，但是如何把金鑰安全的送到接收者的手上進行解密以及傳遞資料人數多時金鑰的管理，是此演算法最大的問題所在。目前較為知名的對稱式加密演算法有DES、IDEA、Rijndael(AES)等。

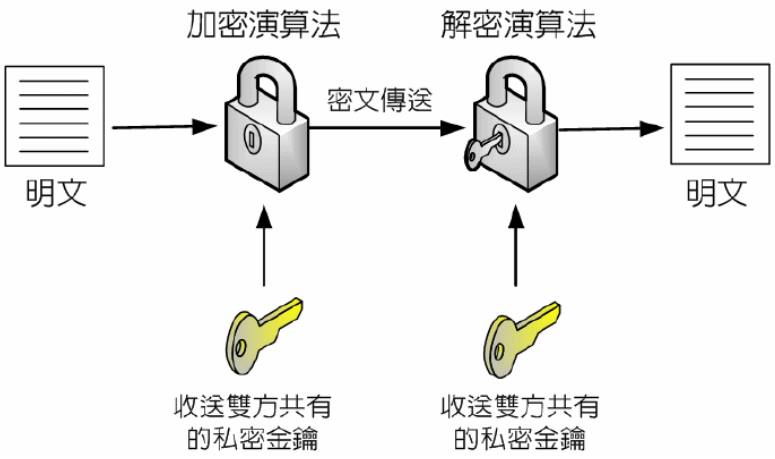

圖 2.4 對稱式密碼演算法

資料來源：陳軒正(2004)[15]

非對稱式密碼演算法（Asymmetric Cryptographic Algorithm）

非對稱式密碼演算法，亦稱為公開金鑰密碼演算法[15]，此方法在進行加、解密時，是透過兩把不同的金鑰，而此兩把金鑰是唯一的配對關係，一把稱為公開金鑰(Public Key)，可以公開分享給人知道；另一把稱為私密金鑰（Private Key），由產生公開金鑰的使用者擁有。經過公開金鑰加密的資料，只能用私密金鑰來解密。由於加、解密是使用不同的金鑰，因此在處理的速度上比對稱式密碼演算法來得慢。目前較有名的非對稱密碼演算法有RSA、ELGamal及Diffie-Hellman金鑰交換等。

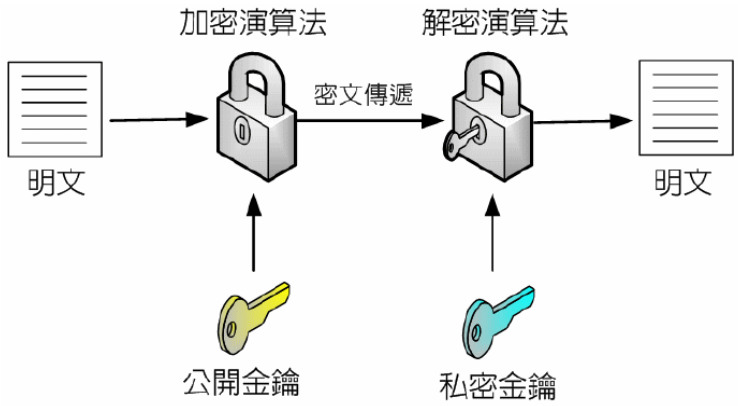

圖 2.5 非對稱式密碼演算法
資料來源：陳軒正(2004)[15]

單向雜湊函數（One-Way Hash Function）

所謂單向雜湊函數就是將任一長度的訊息 m，經過此函數的運算輸出 H(m)，此輸出訊息稱為雜湊值（Hash Value）、訊息摘要（Message Digest）或數位指紋（Digital Fingerprint），如圖2.6。

單向雜湊函數具有兩個特性：

1、單向性（One-Way）或不可逆：

想要在合理的時間範圍以及有限得資源限制下，由H(m)推算出m是不可行的。

2、抗碰撞性（Collision Resistance）：

將任意兩訊息m，n經過單向雜湊函數運算後，不會產生相同的雜湊值。

所以此加密法常被應用在確保資料傳送的完整性、數位簽章及驗證訊息等。目前常見的單向雜湊函數有MD5及SHA-1。

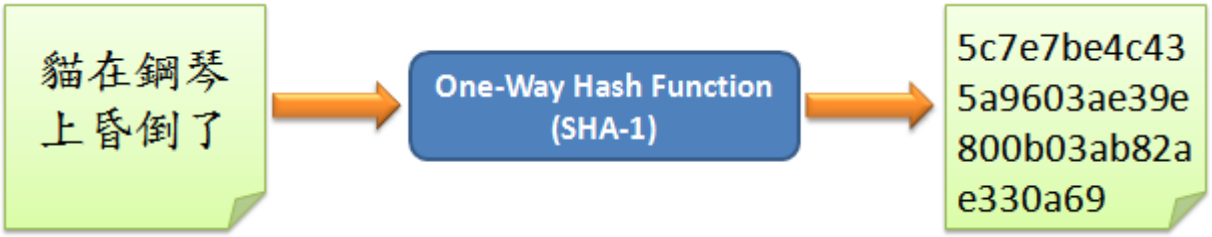

圖 2.6 單向雜湊函數

2.4 安全通訊層(Secure Socket Layer)

Secure Socket Layer（簡稱SSL）是Netscape所提出的一種網路通訊安全協定[5] [28]，其目的在確保通訊雙方資料傳輸之安全，以避免資料在傳輸過程中被攔截、竊取或偽造，因此SSL廣泛被應用在網頁傳輸安全協定（HTTP + SSL = HTTPS）上。

SSL 是介於傳輸層（Transport Layer）與應用層（Application Layer）間的安全機制，主要分為四個子協定：SSL 紀錄協定（SSL Record Protocol）、SSL 握手協定（SSL Handshake Protocol）、SSL 改變特定加密協定（SSL Change Cipher Spec Protocol）、SSL 警報協定（SSL Alert Protocol），如圖 2.7。

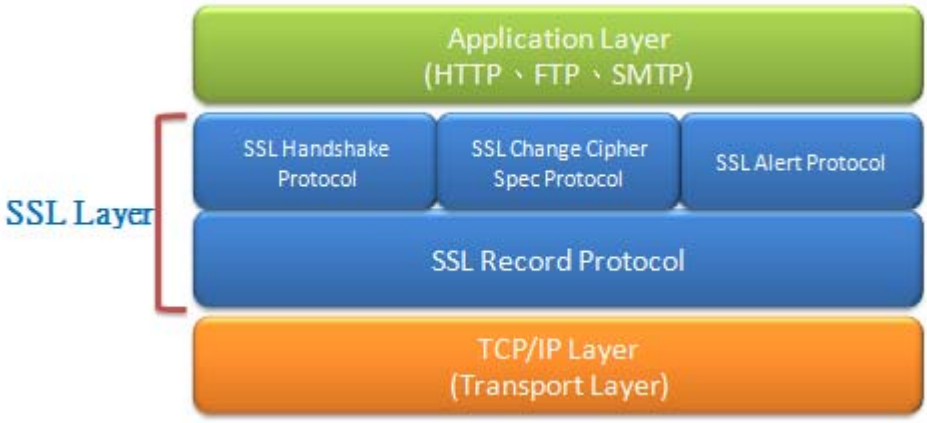

圖 2.7 SSL 協定架構圖

透過 SSL 紀錄協定(SSL Record Protocol)及 SSL握手協定(SSL Handshake Protocol)可達到下列三個目的：

1、身分驗證（Authentication）：讓用戶端和伺服器可以互相驗證對方的身分。

2、訊息機密性（Confidentiality)：提供通訊兩端一個經過加密具有機密性的連線。

3、訊息完整性（Integrity）：透過雜湊函數加密，以確保訊息未被竄改。

2.5 角色為基礎的存取控制(Role-Based Access Control)

存取控制（Access Control）[16]是一種管理模式，詳細規定了使用者可以從事哪些動作，並限制使用者做出危害系統安全的動作。存取控制在資訊管理中是一項非常重要的課題，最基本的是「存取控制列表」，列出使用者對於那些物件有什麼權限。一般常見的存取控制策略(Access Control Policy)[31]有：自由選擇存取控制、強制存取控制、以角色為基礎的存取控制。

自由選擇存取控制（Discretionary Access Control,DAC）

在 DAC 中的環境中，資源的擁有者，可自行決定把權限轉移給他人，接受到權限的使用者，不需要原資源擁有者的同意，就可以任意的將權限，再授與給其他的使用者。

強制存取控制（Mandatory Access Control,MAC）

強制存取控制是一種集中式的安全性控制，屬於管理者基礎的存取控制，所有資源都交由特定的權限中心來管理，這個權限中心記錄著每位使用者所能使用的資源及每項資源可以提供給哪些使用者來存取。

角色為基礎的存取控制（Role-Based Access Control,RBAC）

Role-Based Access Control(RBAC)[2][23]是在1992年被提出，與前面兩種存取控制策略不同的地方在於，RBAC 是依照使用者的角色、職務來決定存取權限，而非他的身分，且每個使用者可以擁有一個以上的角色，被授與此角色的使用者，就可取得此角色被授與的資源，使用者、角色與權限的關係如圖2.8。同一個使用者能夠分屬於不同的角色，就如同校園中，教師的身分不僅是教師，也同時是組長或主任。

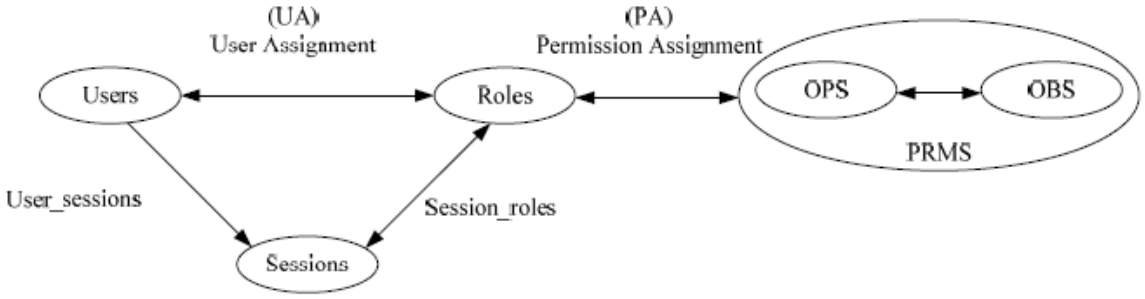

圖 2.8 使用者、角色與權限關係示意圖

資料來源：D. Ferraiolo ,R. Sandhu and R. Kuhn（2001）[6]

以上三種存取控制策略都有其適用範圍及優缺點，下列表格針對不同的存取策略應用在服務網站上做一比較。

表 2.1 各 Access Control 策略應用於網站服務時之比較表

	DAC	MAC	RBAC
權限管理基本單位	個人	個人	角色
是否可分類管理	否	否	是
權限設定複雜度	極高	高	低
可歸屬單一群組	否	否	是
可歸屬於多群組	否	否	是
整體管理複雜度	高	高	低

資料來源：李長庚(2002)[17]

2.6 Transcoding Reverse Proxy

Transcoding Reverse Proxy 又稱為 Transcoding Surrogate[6]，當使用者透過瀏覽器要求遠端主機的服務時，必須先透過 Transcoding Reverse Proxy 來代理使用者向遠端主機要求存取服務，或代理使用者登入遠端主機，以便使用者執行接下來的動作。其運作雛型如圖2.8：

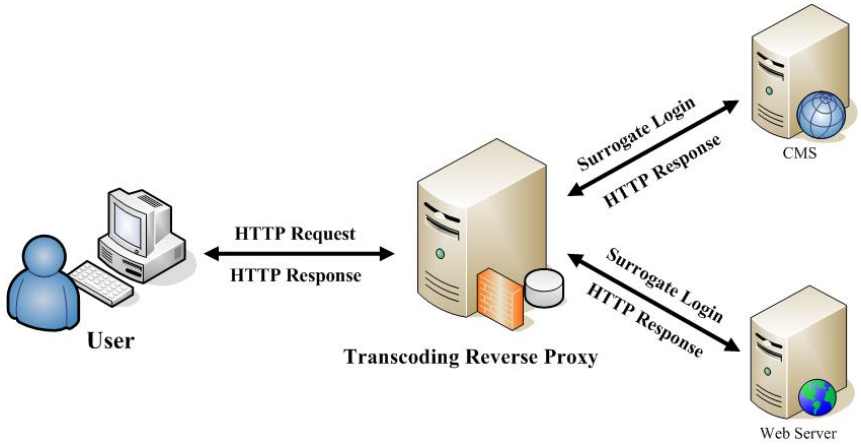

圖2.9 Transcoding Reverse Proxy運作雛型

Transcoding Reverse Proxy應用在單一登入上的特點：

1、 遠端服務主機，可保有本身的認證機制、存取規則和會期追蹤（Session-Tracking）的方式。

2、非常容易部署（可用開放原始碼軟體架設）且可依安全等級，任意改變認證機制。

3、使用者的認證及授權皆由 Transcoding Reverse Proxy 來控制。

4、Transcoding Reverse Proxy 儲存使用者登入各遠端主機的帳號、密碼，以便代替使用者登入遠端主機，達到單一登入的目的。

5、連至各遠端主機的超連結屬性（Hyperlink Attributes）將會改變。

第三章 相關研究

本章共分為三個小節，在 3.1 節中，探討各學者所提的跨網域單一登入機制，並針對其架構比較優缺點。在 3.2 節中，說明 Cookie 如何應用在身分認證上，並探討各學者所提安全 Cookie 機制，做一安全性分析比較。在 3.3 節中，將探討使用者要透過何種方式取得角色屬性以及何種類型的 RBAC 較能符合本系統之需求，並針對各學者所提之 RBAC 類型，做一整理比較。

3.1 跨網域單一登入機制之研究

當使用者處於一個提供多種服務或系統的環境中，為了存取這些服務或系統，使用者必須記憶不同的帳號、密碼，經過一次又一次的身分驗證，造成了使用者的不便。此時單一登入概念的興起，造成各式網站整合單一登入蔚為風潮，其就是為了解決此問題，讓使用者只要輸入一次帳號、密碼，經過一次的身分驗證程序，就可以取用各種服務或系統的資源，節省使用者寶貴的時間，提升網路的安全，進而創造更大的商機。

目前最被廣泛使用的單一登入系統機制就是 Miscrosoft 於1999年所提出的 .NET Passport[8]， .NET Passport是以密碼為基礎的單一登入服務（Password-Based Single Sign-In Service），採用麻省理工學院所提出的 「Kerberos」為其安全機制，其目的是讓使用者只要以一組帳號與密碼就可以通行加入 Passport 的網站，並且能夠存取以 .NET Passport 為基礎的 Web Service，但由於 .NET Passport 本身仍有非常多的漏洞需要修正，而這些漏洞也造成使用者的不安。因此有些隱私保護組織強烈要求美國聯邦貿易委員會（FTC）讓 Microsoft 停止通過 Windows XP 和 Passport 認證服務去追蹤並且監視用戶，避免引起嚴重的隱私權問題，故 Microsoft [28]在2006年將 .NET Passport做重大的更新後，改名為 Windows Live ID。

由於 .NET Passport 採用 Kerberos 為其認證系統，因此系統採取可信任的第三者驗證方式（Trust Third-Party Authentication）對使用者進行身分驗證，且為了不影響加密效能，主要利用 DES（Data Encryption Standard）加密演算法來加解密私密金鑰，再透過 Kerberos 提供跨網域的功能，讓 .NET Passport 的服務可以更安全、可靠、透明且有彈性。不過因為 .NET Passport 在驗證使用者身分後，隨即簽發一組會議金鑰給使用者，待合作網站接收到使用者會議金鑰後自行以和 .NET Passport 共同持有之私密金鑰作

對稱式加解密及驗證，於安全性上較為薄弱。

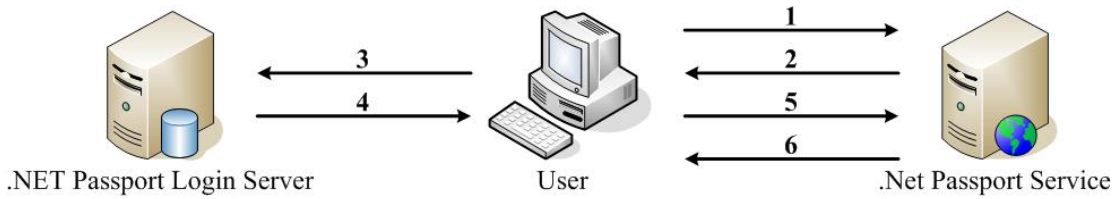

圖 3.1 Microsoft .NET Passport 使用者認證流程示意圖

Microsoft .NET Passport 使用者認證流程如圖3.1 所示：

1、 使用者連結到 .NET Passport 相關之合作網站請求服務。

2、 合作網站將使用者重導至 .NET Passport 驗證網站進行身份驗證。

3、 若使用者沒有.NET Passport 帳號則合作網站導引使用者至 .NET Passport 驗證網站進行資料登錄。

4、 使用者取得 .NET Passport 的認證及授權。

5、 使用者持授權證明向合作網站請求服務。

6、 合作網站經由 .NET Passport Manager 判讀授權無誤後允許使用者進入。

由於 Kerberos 系統之中，仍是採用傳統密碼系統（Convention Cryptosystems）[21]，但隨著公開金鑰密碼系統（PKC，Public Key Cryptosystem）的發明與公開金鑰基礎建設（PKI,Public Key Infrastructure）的建立，讓使用者認證服務與資料的傳遞更加安全。因此朱建達[16]在2000年便提出以公開金鑰基礎建設(Public Key Infrastructure)來設計單一登入系統，透過 PKI 的機制讓訊息傳遞與資料交換過程中，能夠確保資料完整性、資料來源辨識、資料隱密性、不可否認性等四種重要的安全保障。

朱建達所提之論文是將系統整合公開金鑰基礎建設，讓使用者先經過「認證伺服器」（AS）驗證之後，取得一份由「憑證中心」（CA）所發的權限屬性票卷，透過此票卷便可向「權限屬性伺服器」（PAS）取得登入「應用伺服器」（V）的帳號、密碼，最後藉由「代理程式」（Agent）將取得的帳號、密碼轉換成為「應用伺服器」（V）可以接受的格式。其系統模型如圖3.2。

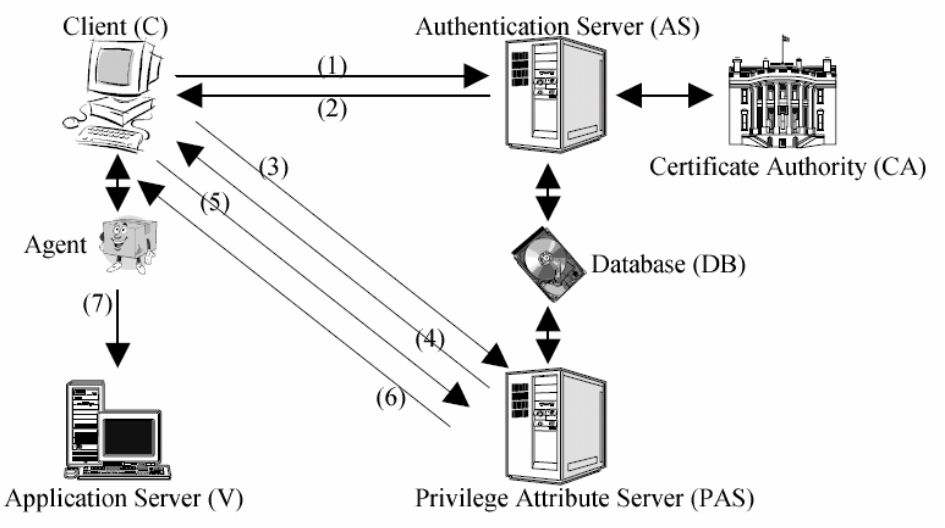

圖 3.2 「建立於公開金鑰基礎建設的單一簽入系統」系統模型圖

資料來源：朱建達(2000)[16]

若依此建置單一登入系統，至少需要「認證伺服器」、「憑證中心」、「權限伺服器」、「應用伺服器」四個伺服器與「代理程式」，此架構除了會增加管理上的難度，相對的建置成本也會提高。

對於目前市面上各單一登入服務套件之間，使用者認證資訊無法相容、大量資料對資料庫存取速度影響與缺乏內建的權限控制機制問題，李長庚[17]在2002年提出了「一個開放的 Web-Based Single Sign-On 服務架構」以解決上述問題。在這個服務架構中，透過專職對外聯繫的服務單位（SSO Embassy），來與其他的單一登入服務系統溝通，並採用以角色為基礎的權限控制機制（Role-Based Access Control,RBAC）與LDAP（Lightweight Directory Access Protocol）搭配，來提供良好的權限控制機制。

在這個架構中有四個主要角色：

1、 系統服務單位（System Units）：

系統服務單位中包含著本服務架構所提供的核心功能，諸如使用者認證、授權、使用者資訊存放、以及對外部單一登入服務的溝通元件等，皆屬於系統服務單位。

2、 服務網站（Service Site）：

即在本服務架構下提供使用者各項服務的服務網站。這些網站本身的認證資訊必須記錄在系統服務單位中；並且，這些服務網站也必須記錄該服務網站所屬之服務架構的相關資訊，如此方能建立可信任的關係，以利訊息交換。

3、 外部單一登入服務（Other SSO）：

代表本服務單位所將與之互動的外部單一登入服務網域。

4、 使用者（Users）：

即一般的使用者，泛指任何使用本服務架構、以及服務架構下之服務網站的使用者。

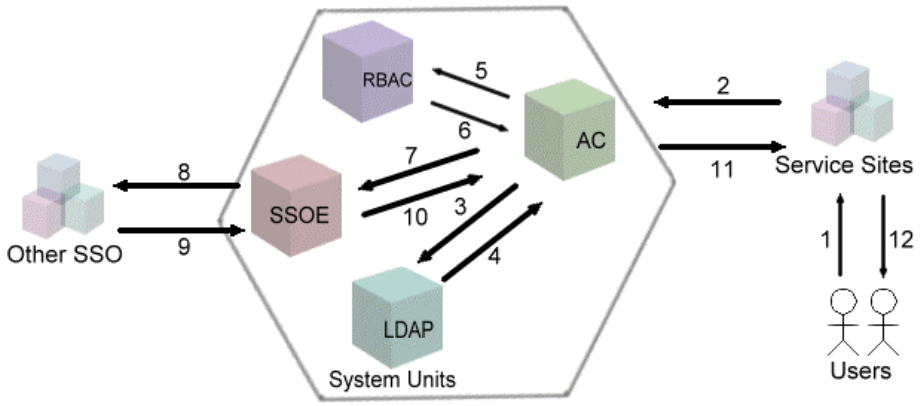

圖 3.3 「一個開放的 Web-Based Single Sign-On 服務架構」互動流程圖

資料來源：李長庚(2002)[17]

此服務架構互動流程如圖3.3：

(1) 使用者透過 Service Sites 進行登入動作。

(2) Service Sites 將使用者所輸入的認證資料送至AC。

(3) AC 根據Service Sites 所傳來的使用者認證資訊，向LDAP 取得該使用者的資料。

(4) LDAP 傳回給AC 該使用者的資料，AC 則根據這份資料進行使用者身分認證動作。若使用者所輸入的認證資料無誤，則繼續進行其他步驟。若使用者輸入的資料有問題，那麼就直接進行步驟11，並傳回錯誤提示訊息。

(5) AC 向 RBAC 索取該使用者的使用權限資料。

(6) RBAC 將該使用者的權限資料傳回給AC。

(7) AC 根據該使用者的決定來判斷是否替使用者透過 SSOE 到Other SSO 進行跨網域登入動作。若該使用者決定不進行此動作，則跳到步驟11，並繼續進行其餘的步驟。

(8) SSOE 依據AC 的要求，向Other SSO 進行該使用者的登入動作。

(9) Other SSO 傳回該使用者登入動作的執行結果給SSOE。

(10) SSOE 將工作執行結果回報給AC。

(11) AC 將使用者登入等一連串動作之執行結果回報給Service Sites。

(12) Service Sites 依據AC 所傳回的資訊決定該使用者是否有權使用Service Sites

所提供之服務。

此篇論文所提的 SSOE（SSO Embassy）是為解決各個單一登入服務網站間資訊交換格式不相容的解決方案。而為了提供使用者更多與外部單一登入服務溝通的選擇，需要實作多個 SSOE 以提供使用者選擇，不過實作 SSOE 是複雜而且困難的。

另外在1999年V.Samar[3]提出的論文中提到 Cookie 能實現單一登入，主要是基於以下幾點因素：

- Installation

  利用 Cookie 來實現單一登入，客戶端無需安裝額外軟體。

- User Experience

  使用者的感受沒有改變或極小的改變，因為當伺服器從舊有的認證機制改變成單一登入時，利用Cookie的方式可將認證機制模擬成舊有的方式，減少使用者適應新機制的麻煩。

- Independence with the authentication mechanism

  Cookie可以應用於各種認證機制。例如：透過http POST來傳遞username/password的方式、Basic http authentication、http digest authentication、Secure-ID authentication、PKI等。

- Maintainability

  由於 Cookie 的設定是由伺服器所控制，所以我們可以很容易的更改 Cookie 的格式和內容，而無需要求客戶端升級或調整設定。當遇到任何安全上的問題時，也可以快速的完成修復。

但由於 Cookie 本身的限制，使得V.Samar所提的機制只能局限於同一個網域內，因此在2005年王雅苓[18]提出了「一個新的跨網域單一登入服務架構」，利用 Cookie Center 的觀念來解決 Cookie 無法跨網域傳遞的限制。在此服務架構中，主要是建基於 Cookie 和 LDAP 之上，並且讓每一個參與單一登入服務的網域，都建置一個 Cookie Center，以便來發出認證的 Cookie 給它們自己網域的使用者。

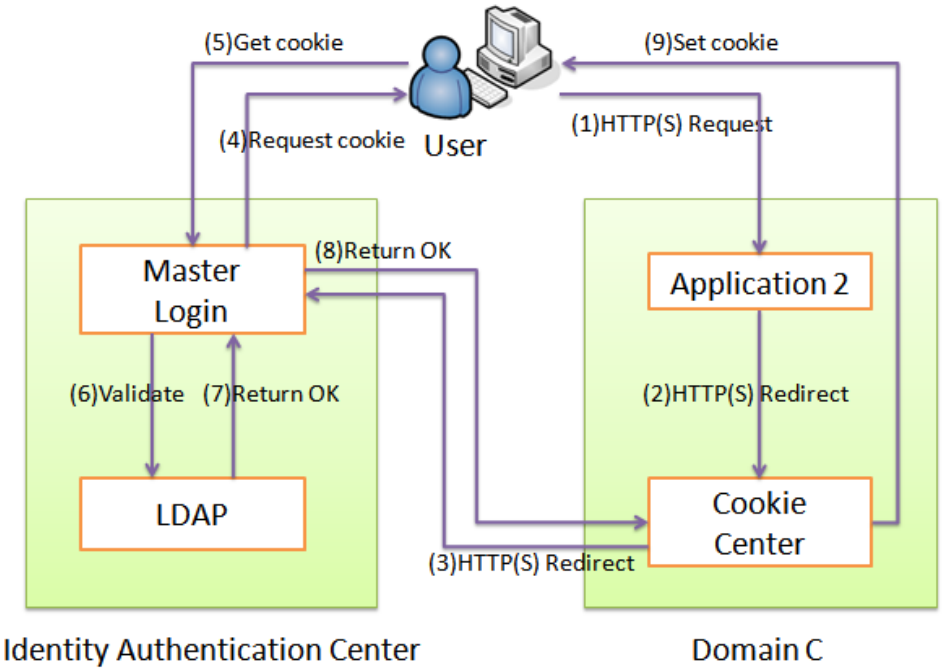

圖 3.4 使用者已通過認證，欲存取另一個網域應用系統之流程圖

圖3.4為此作者所提之跨網域單一登入服務流程圖，而其運作流程如下所示：

1、 使用者存取受保護的應用系統Application 2（Domain C）。

2、 Application 2重導到Cookie Center，Cookie Center檢查有沒有cookie。

3、 使用者雖然已通過認證，但沒有Domain C的cookie，因此重導到Master Login。

4、 Master Login要求使用者傳送 Cookie。

5、 Master Login取得 Cookie(Domain Identity Authentication Center)。

6、 Master Login將取得的 Cookie值SingOnKey傳送到LDAP進行身分認證。

7、 認證成功，回傳OK。

8、 Master Login 將認證成功的訊息送給 Cookie Center。

9、 Cookie Center 收到後，送出一個名為SingOnKey、value為X的cookie給使用者。

本論文雖然透過認證中心(Authentication Center)及 Cookie Center 兩個元件來簡化系統的架構，但因為每一個參與單一登入服務的網域，都必須建置一個 Cookie Center，而且所有合作的 Cookie Center 皆要向憑證中心來確認使用者身分，所以此做法除了增加管理者負擔外，也會加重網路流量的負荷。

最後，我們針對本節探討的四個服務架構，將其達到跨網域單一登入做法及缺點分析彙整如下表3.1：

表 3.1 各服務架構之做法及缺點分析表

.Net Passport (1999)[8]	做法：採用Kerberos為其認證系統，並將經身分認證取得的會議金鑰送交合作網站比對，以確認使用者的合法性。 缺點：合作網站利用與 .NET Passport 共同持有之私密金鑰作對稱式加解密，以完成驗證驗證，在安全性上較為薄弱。
朱建達 (2000)[16]	做法：系統採用公開金鑰基礎建設，利用憑證中心來發給每個系統元件和使用者一份憑證，讓雙方可以互相認證身分。 缺點：建置此架構至少需要「認證伺服器」、「憑證中心」、「權限伺服器」三個伺服器與「代理程式」，而此架構除了會增加管理上的難度，相對的建置成本也會提高。
李長庚 (2002)[17]	做法：實作SSOE來與其他單一登入服務交換使用者認證資訊，並內建角色為基礎的權限控制機制，與利用LDAP快速查詢使用者資料的優點來建置此架構。 缺點：要先與外部單一登入服務網站結為聯盟網域，且必須實作多個SSOE，不過實作SSOE是複雜而且困難的。
王雅苓 (2005)[18]	做法：讓所有參與單一登入服務的網域，都建置一個Cookie Center，透過Cookie Center向憑證中心確認使用者身分後，才發送該網域合法的cookie給使用者，以解決cookie無法跨網域傳遞的問題。 缺點：此做法除了增加管理者負擔外，也會加重網路流量的負荷。

對於表3.1所提到的缺點，為了能夠有效的解決這些問題，我們將實作一個跨網域單一登入系統，使用單一的網頁管理介面，並利用代理登入的方法來達到跨網域的目的，以降低管理者管理上的困難；而且各服務網站仍可保有原來的認證機制，進而提升認證安全；相對的也降低了建置成本與網路流量。

3.2 Cookie認證之研究

Cookie是被發明應用在維持Server和Browser之間的狀態，也就是說使用者登入一個系統，經過認證過程後，如果認證成功，伺服器會將認證字串(Message Authentication code,MAC)以Cookie的形式儲存在使用者的電腦中，待使用者再度登入同一個系統時，伺服器會從cookie中取出先前發給使用者的認證字串並進行驗證，而無須使用者再次進行登入的動作。不過若使用Cookie來驗證使用者身分，有一些安全上的議題是值得考慮的。

1、Clear Text Cookie：

未加密的cookie是很容易在傳送過程中，被攔截進而取得私密資料；即便是加密過的cookie被攔截後，也能利用重送攻擊(replay attack)來欺騙伺服器，因此cookie中儲存的內容與傳遞過程，已經出現了很大的漏洞。因此在設定cookie內容時，應該要避免儲存一些私密資訊；並且在資料傳遞過程中，最好使用SSL加密傳輸，以補強cookie被竊取的風險。

2、Cookie Storage Security

一般來說，使用者瀏覽器會將cookie存放在用戶端電腦中，此時惡意攻擊者便可透過外掛程式或其他方法來竊取cookie，因此使用in-memory cookie或session cookie來儲存資料比將cookie存放在用戶端電腦中安全許多。

3、Cookie Destination Control

當使用者透過瀏覽器要求伺服器派發cookie時，此cookie只能被與原伺服器同網域或子網域之伺服器所存取，但是透過用戶端的網域名稱伺服器(Domain Name Server)，卻可以修改原本不隸屬於派發cookie伺服器的網域，造成不合法的伺服器取得該cookie，我們也將此稱之為cookie-harvesting threat。

4、Client IP Address Restriction

在cookie中加入用戶端的 IP Address，可以避免伺服器遭受重送攻擊。不過在某些情況下，此機制可能會遇到一些困難，例如：在Network Address Translation(NAT)的環境中或使用者透過代理伺服器(Proxy Server)連接伺服器時，用戶端的 IP Address 可能會因此而改變，造成利用 IP Address 來驗證cookie的合法性失效。

基於以上的問題，Joon S.Park and Ravi Sandhu[9]在2000年提出三種型態的Secure Cookie，用以驗證使用者是否為cookie持有人，以防cookie被有心人竊取、重送，這三種型態的Secure Cookie分別為 Address-based authentication、Password-based authentication 與 Digital-signature-based authentication，如圖3.4。

	Domain	Flag	Path	Cookie_Name	Cookie_Value	Secure	Date
IP_Cookie	acme.com	True	/	IP_Cookie	129.174.100.88	False	12/31/2000
Pswd_Cookie	acme.com	True	/	Pswd_Cookie	hashed_password	False	12/31/2000
Sign_Cookie	acme.com	True	/	Sign_Cookie	Signature_of_Alice	False	12/31/2000

圖 3.5 三種型態的 Secure Cookie

資料來源：Joon S. Park and Ravi Sandhu(2000)[9]

Address-based authentication(IP_Cookie)：

當用戶端透過瀏覽器向伺服器要求存取時，伺服器會立即取得用戶端的IP Address並將之存入cookie(IP_Cookie)中。當用戶端再度向此伺服器要求存取時，伺服器會先核對用戶端的IP Address是否與IP_Cookie中的IP值一致；如果相同，伺服器就會相信此用戶端是IP_Cookie的真正擁有者。

Password-based authentication：

伺服器將用戶端提供的密碼經過 hash 函數的加密後，存放在cookie(Pswd_Cookie)中。當用戶端向伺服器要求存取時，必須先輸入一串密碼，再經過 hash 函數的加密，若此加密過的密碼與存放在Pswd_Cookie的密碼值相同，伺服器就會相信此用戶端是Pswd_Cookie的真正擁有者。

Digital-signature-based authentication：

如果伺服器知道用戶端的public key，像DSA或RSA的技術，伺服器就可以確認使用者是否為cookie真正的擁有者。

在2005年Alex X. Liu等人[10]針對 Joon S. Park and Ravi Sandhu 利用cookie所提出的三種認證方法，提出一個Secure Cookie Protocol，以解決此三種認證機制較無效率與不容易部署的問題，並滿足一個Secure Cookie Protocol該提供的四項服務：可認證性(Authentication)、機密性(Confidentiality)、完整性(Integrity)、防止重送攻擊(Anti-replay)。

- **認證性(Authentication)：**

Joon S. Park and Ravi Sandhu 所提出的三種Secure Cookie認證機制，在某些情況下，會遇到一些困難。

IP_Cookie：由於IP_Cookie是透過IP位址(IP Address)來確認使用者身分，因此會有會三個問題發生。

1、 惡意攻擊者利用IP偽造(IP Spoofing)的技術通過認證機制。

2、 用戶端透過DHCP(Dynamic Host Configuration Protocol)取得IP位址，導致IP位址可能會變動。

3、 在NAT(Network Address Translator)或代理伺服器(Proxy Server)的環境下，其他的用戶端可能也會取得相同的IP位址。

Pswd_Cookie：當用戶端瀏覽其他網站時，必須一再輸入密碼，以驗證用戶端的身分，這也造成了使用者的不便。

Sing_Cookie：此機制是讓每個用戶端皆擁有一組公開金鑰與私密金鑰，並利用資料庫的搜尋與公開金鑰加密演算法來完成身分認證，而此方法是不容易去部署且較耗費成本的。

因此Alex X. Liu等人提出另一種Secure Cookie，如圖3.5，此方法則可以解決上列三種認證機制所遇到的困難。

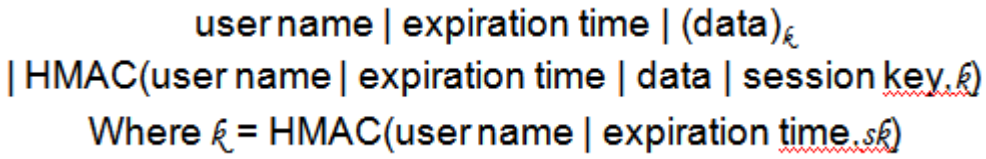

圖 3.6 A Secure Cookie Protocol

資料來源：Alex X. Liu等人(2005)[10]

- **機密性(Confidentiality)：**

在需要高等級得機密的環境中，Joon S. Park and Ravi Sandhu 利用伺服器隨機產生的Session Key來加密cookie中的資料，再利用伺服器的公開金鑰(Public Key)來加密Session Key，並存入cookie(Key_cookie)中。由於此種方法使用到公開金鑰的演算法，因此會使得cookie protocol變的複雜且沒有效率。因此，Alex X. Liu等人利用HMAC(user name | expiration time,sk)來當作加密金鑰 k，以確保資料的機密性，而此方法有三個優點：

1、 密鑰唯一性：由於每位使用者的名稱與cookie過期時間皆不相同，因此產生的加密金鑰也會不同。

2、 密鑰不可偽造：因為server key是無法被得知的，因此加密金鑰k也就無法被偽造。

3、 密鑰不用儲存：由於加密金鑰k是利用user name、expireation time及server key動態產生，因此加密金鑰 k 不用儲存在伺服器端或是cookie中。

- **完整性(Intergrity)：**

  Alex X. Liu等人利用金鑰雜湊訊息認證碼(keyed-hash message authentication code ,簡稱HMAC)來驗證資料是否有被竄改，只要cookie中的值有被修改過，其產生的HMAC便會與原先在cookie中的HMAC不同。

- **防止重送攻擊(Anti-replay)：**

  Alex X. Liu等人將 SSL 的session key加入到訊息認證碼(HMAC)中，再經由加密金鑰k加密，如HMAC(user name | expiration time | data | session key,k)，則此時HMAC就具有session 的特性，因此即使此cookie被竊取來進行重送攻擊，也無法成功。

即使各學者所提的安全Cookie，能有效的達到資料的可認證性、機密性、完整性與防止重送攻擊，不過隨著科技日新月異，即使再安全的設計，總有被破解的一天，所以我們儘可能不要把重要或敏感的資料存放在cookie中，以避免私密資料外洩的問題發生。最後，我們針對Alex X. Liu等人所提的Secure Cookie Protocol加以簡化改良，詳細的設計將在第四章中加以說明。

3.3 單一登入使用RBAC之研究

由於網路服務的多元性，傳統的存取控制已經無法滿足現今的系統，主要是因為傳統存取控制只有針對授權(Authorization)決定策略來決定使用者的權限，傳統的權限控管系統，大多是使用者和權限直接對應，當使用者和權利都越來越多後，系統管理者要管理每位使用者和權利之間的關係會增加許多的複雜度。由於 RBAC 將傳統指派權限給使用者的方式，修改為使用者對應角色，角色對應權限，如此一來，當使用者權限有變動時，系統管理者只要改變使用者和角色的對應即可，進而降低了管理上的成本。另外，因為 RBAC 包含了三個基本安全原則：[2][11][20]

1、 最少權限(Last Privilege)：

一個角色中會擁有許多不一樣的權限，一個使用者也可能被派發多個角色，權限會視角色的需要來派發，而且擁有多重角色的使用者無法利用其他的角色來進行特定角色的職務，以避免權限會遭到濫用或是發生衝突。

2、 責任區分(Separation of Duties)：

這是屬於多人控制安全策略，對於有互斥性質的職務可以分配給不同的角色，並且限制使用者取得相互斥的角色，以避免發生有舞弊的情況發生，例如出納與會計的職務不能由同一個員工來擔任。

3、 資料抽象(Data Abstraction)：

傳統的存取控制限定於實際的操作動作，如讀取、寫入等。RBAC可以利用語意的方式來描述權限，如借款、貸款。

因此常被應用來提升存取控制的安全。

本篇論文除了探討如何使用安全 cookie 來認證使用者身分與利用代理登入的方式來達到跨網域的單一登入，也期望藉由角色為基礎的存取控制(RBAC)，來解決學校行政人員業務帳號、密碼移交的不便。因此使用者要透過何種方式取得角色屬性以及何種類型的RBAC較能適用本系統，將是現行需要解決的的兩個議題。

對於如何在網路上來取得使用者的角色屬性，在2001年Joon S. Park等人[12]提出兩種方法，分別為User-Pull與Server-Pull架構，其架構圖如圖3.7和圖3.8。

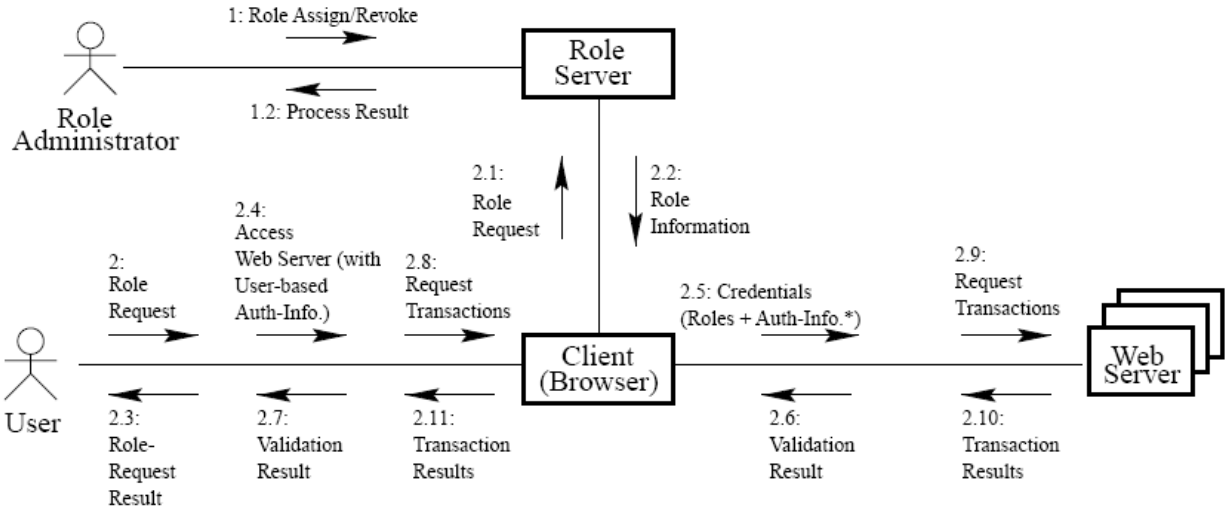

圖 3.7 User-Pull 架構合作圖

資料來源：Joon S. Park et al.(2001)[12]

在User-Pull架構中，使用者透過瀏覽器(Browser)向Role Server取得授予的角色，並儲存在使用者的主機中，接著將此角色與認證憑證一併提交給Web Server，待Web Server確認認證憑證無誤後，便可讓使用者透過瀏覽器(Browser)，來取得該角色所擁有的權限。

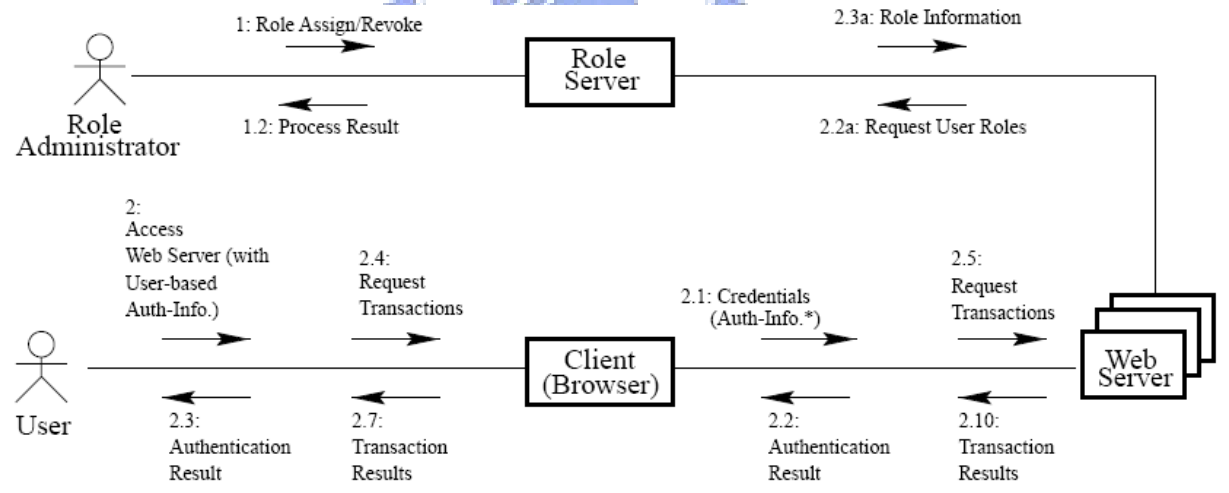

圖 3.8 Server-Pull 架構合作圖

資料來源：Joon S. Park et al.(2001)[12]

在Server-Pull架構中，使用者透過瀏覽器(Browser)將認證憑證提交給Web Server，待Web Server確認認證憑證無誤後，Web Server即向Role Server請求取得授予使用者的角色，接著使用者透過瀏覽器(Browser)請求服務時，Web Server便依照該使用者的角色，來

提供應有的權限。

User-Pull與Server-Pull架構最大的不同點在於，User-Pull架構是將Role Server所授予的角色儲存在使用者主機中，而Server-Pull架構是儲存在Web Server中；由於User-Pull架構是將授予的角色儲存在使用者主機中，因此使用者便可以利用此角色，在不同連線要求中或在不同的網頁伺服器(Web Server)中傳遞角色，以取得各種服務，直到角色的使用期限過期，這種方法雖然需要使用者的合作(例如：允許使用cookie)，但因為角色的可重複使用性，卻增強了網頁伺服器(Web Server)的效能。尤其對某些使用者而言，為了方便在不同的網站中存取服務，因而必須維持他的角色或頻繁使用他的角色，而User-Pull架構就是一個很好的解決方法。User-Pull與Server-Pull架構的比較，如表3.1。

表 3.2 User-Pull 與 Server-Pull 架構比較表

	User-Pull架構	Server-Pull架構
使用者便利性(User's convenience)	低	高
網頁伺服器效能(Performance)	高	低
可重複使用性(Reusability)	高	低
角色新穎度(Role freshness)	低	高
單點故障(Single-point failure)	低	高

資料來源：Joon S.Park et al.(2001) [12]

隨著網路的發達，使用網路服務的人數大量增加，讓採用RBAC來管理使用者權限的網路系統無法有效的減輕管理上的負擔，因此，在2005年Q. Li等人[13]提出一個以群組(Groups)為概念，分散式的角色為基礎存取控制(Decentralized RBAC,DRBAC)模型，以解決大量使用者對於傳統 RBAC 的衝擊。

分散式的角色為基礎存取控制(DRBAC)模型是由使用者(Users)、群組(Groups)、角色(Roles)、權限(Permissions)和會期(Sessions)等五個元件組成，其模型如下圖3.8。

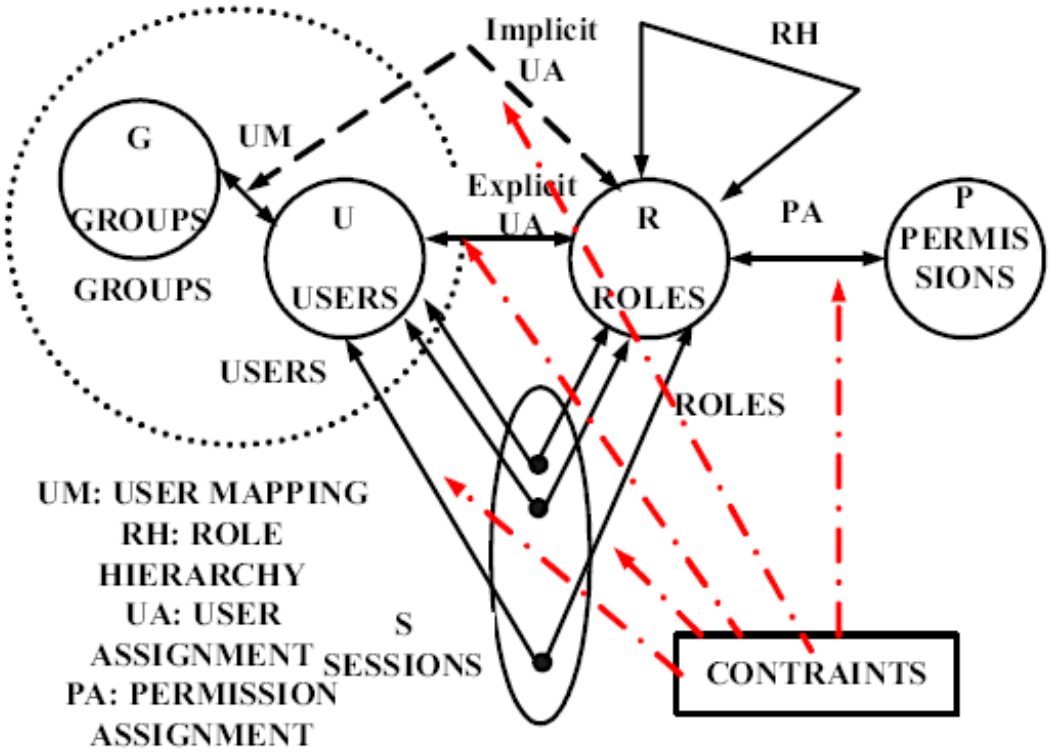

圖 3.9 DRBAC 模型圖

資料來源：Q. Li et al.(2005)[13]

由於傳統的RBAC是利用使用者對應角色，角色對應權限的方式來運作，一旦面臨大量的使用者與角色之間關係的轉換配置時，就會大大降低處理效率。因此透過群組(Groups)將具有相同屬性的使用者集合起來，並賦予群組(Groups)適當的角色，可以免去繁瑣的使用者角色配置。

在國中、小行政體制中，常用團隊或社群來執行上級所交辦之任務，並利用群組的概念將參與此團隊或社群的教師設定成群組角色，但是事實上，群組角色是讓每位教師所擁有的權限都一樣，這樣的做法卻與實際運作情況不盡然相符。因為在同一團隊或社群的教師，可能會透過分工合作的方式來執行任務，所以把這些教師設為同一個群組角色並不恰當，因為每位老師都會有自己負責的任務。因此在2007年Wei Zhou等人[14]便提出了一個以團隊與任務為基礎的角色存取控制(Team and Task Based RBAC Access Control，簡稱TT-RBAC) 架構來實行權限控制與認證管理，TT-RBAC增加了團隊和任務(Team and Task)兩元件，並整合NIST RBAC所提出的架構，因此TT-RBAC也承襲了RBAC角色階層的概念，透過角色階層(Role Hierarchy)的概念，使高層的角色可以繼承低層角色的權限，例如：教務主任擁有教學組長及訓導組長的權限。

在TT-RBAC架構中，使用者(Users)被分配到某些角色(Roles)或某些團隊(Teams)；而

角色(Roles)或任務(Tasks)被分配給某些團隊(Teams)，接著權限(Permissions)再被分配給角色(Roles)或任務(Tasks)，因此使用者可執行的權限是由他被賦予的角色或隸屬的團隊所決定，而此團隊被指派的任務，將決定參加此團隊的使用者所能執行的權限。例如：某位教師擔任六年級導師的角色，又因其專長而參加數學領域這個團隊，接著教務處將數學競賽這個任務指派給數學領域團隊來負責，而此團隊成員便各司其職，以完成此任務。以TT-RBAC為核心加上階層概念的架構圖，如圖3.9所示。

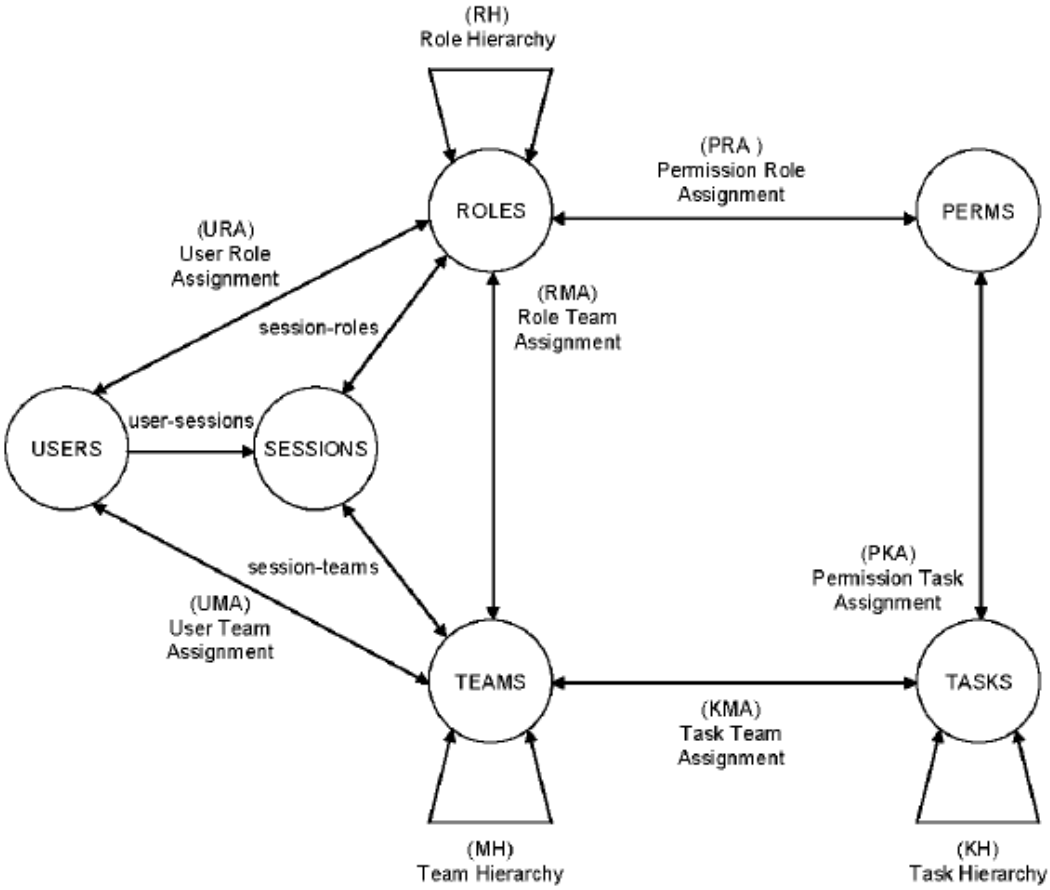

圖 3.10 階層的 TT-RBAC

資料來源：Wei Zhou et al.(2007)[14]

最後我們針對傳統RBAC、DRBAC與TT-RBAC的運作原理與缺點分析彙整如下表3.3：

表 3.3 傳統 RBAC、DRBAC 與 TT-RBAC 的運作原理與缺點分析表

傳統RBAC	原理： 利用角色來控管權限，再授予使用者應有的角色身分；而當使用者更換職務時，只需要改變它原有的角色即可，不用逐一修改使用者權限。 缺點： 若面臨大量的使用者與角色之間關係的轉換配置時，就會降低處理效率。
DRBAC	原理：將具有相同屬性的使用者集合起來，並賦予群組(Groups)適當的角色，免去繁瑣的使用者角色配置。 缺點：在合作式的環境中，群組的角色並無法有效的定義，每個角色所被賦予的權限。
TT-RBAC	原理：使用者可執行的權限是由他被賦予的角色或隸屬的團隊所決定，而此團隊被指派的任務，將決定參加此團隊的使用者所能執行的權限。 缺點：並非任何情況下，團隊成員都需要利用合作的方式來完成任務。

為了能讓本系統存取控制與授權機制更加富有彈性，我們將整合DRBAC與TT-RBAC的優點，讓相同屬性的使用者具有群組角色(Group role)的權限，而且在需要合作式的環境中，也能利用團隊(Team)分工的方式來達成任務(Task)，詳細的設計將在第四章中加以說明。

第四章 跨網域校務單一登入系統(CDSASS)

在過去網路尚未普及之前，文書資料或相關調查的傳遞上，均需親自或透過郵寄方式來作業；但隨著科技網路蓬勃發展，教育行政部門依各自需求，先後建立許多網路應用程式，讓行政人員透過網路即可完成資料傳送或填報作業，以提升行政效率；但由於教育部門委由廠商或自行開發的平台或語言的不同，因此擁有各自的認證機制，導致使用者必須反覆的輸入帳號、密碼，才能存取相關服務，造成使用者相當的不便。再加上行政人員職務異動的頻繁，如何有效管理眾多登入的帳號、密碼，而讓各項業務執行得更順利，成了另一個需要解決的問題。

4.1 CDSASS概說

本系統最重要的環節就是如何在不變更現有網路環境且不增加軟硬體經費下，讓使用者只需要登入一次系統，便可取得不同網域的服務，並設計一套適合國小行政業務的權限管理機制來提升行政效率。

基於以上問題，我們建置了一套能提升行政效率且兼具安全性的跨網域校務單一登入系統（CDSASS），如圖4.1。希望藉由單一登入的技術與代理登入的方式，讓使用者只需要登入一次系統，便可透過本系統來進行跨網域的代理登入作業，且不需要修改外部服務網站的登入介面，以減少使用者需要記憶多組帳號、密碼而產生的資安問題。另外，本系統以角色為基礎，運用使用者、群組、團隊與角色的對應關係，搭配權限指派給角色的概念，讓國小行政業務帳號、密碼移交與填報作業能夠更順利、簡便且任務執行上也更有效率。最後，透過系統稽核的功能來記錄使用者的行為，讓使用者非法竄改的登入密碼與角色權限後，能修正回復為原本的資料，以確保系統的安全性。

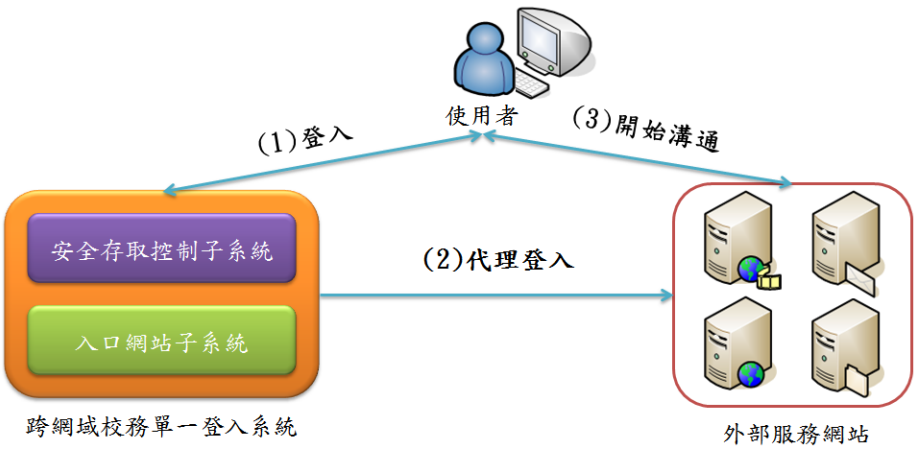

圖 4.1 系統示意圖

4.2 CDSASS架構

我們透過邏輯設計上的差異性與階段性，將系統分割為兩個子系統：入口網站子系統與安全存取控制子系統。而其主要核心功能由下列六個模組所組成：登入與登出模組、認證模組、單一登入服務模組、使用者資訊儲存模組、授權與權限管理模組與稽核管理模組，其系統架構如圖4.2。

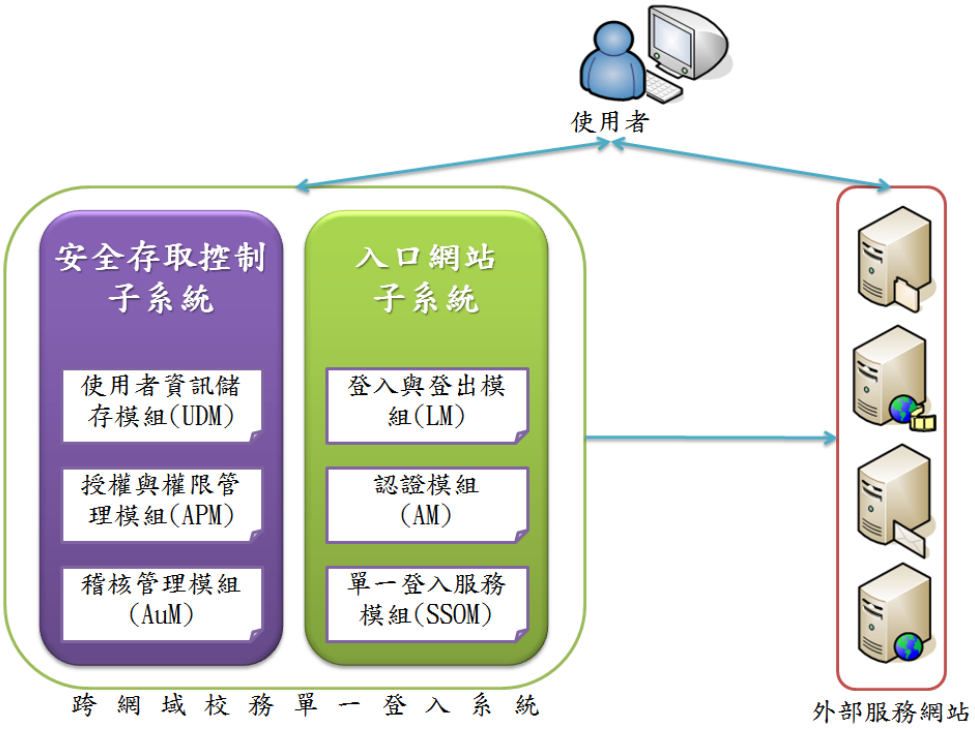

圖 4.2 CDSASS 系統架構圖

接著我們將對此六個模組的工作內容與設計理念做深入的說明。

1. 登入與登出模組（Login Module ,LM）：

- 工作內容：

(1) 提供各種使用者登入方式，例如：傳統的帳號/密碼、Certificate 認證、Smart Card 等。

(2) 必須提供資料安全傳輸管道，以確保資料不會在傳遞過程中被竄改或竊取。

(3) 提供刪除使用者端及伺服器端暫存資料之功能。

- 設計理念：

每個單一登入服務的網站都必須設計使用者登入的頁面，本系統是採用帳號/密碼

的登入方式，並在連線過程中，採用SSL加密通道，來確保資料傳輸的安全性。當使用者執行登出動作時，LM 便會刪除 Cookie 與 Session 中的資料，防止資料外洩的可能性。若使用者未執行登出動作，因Cookie 與 Session均有時間限制，故超過時間限制時，Cookie 與 Session 便會自動失效。

2. 認證模組（Authentiation Module ,AM）：

- 工作內容：

(1) 設定認證訊息與角色代碼於Cookie與Session中。

(2) 確認使用者送出的認證資料是否正確。

(3) 確認Session的合法性，並取得使用者角色身分。

- 設計理念：

雖然以 Cookie 來認證一直為人所詬病，不過考量簡化使用者登入步驟、提升行政效率之因素，在此我們仍以 Cookie 來當作認證的一環。若是以系統的安全性為主要考量的話，當然還是建議使用 Session Cookie的概念來執行認證程序。

使用者的登入模式可以分為首次登入與再度登入，而這兩種登入模式皆由認證模組負責來核對與確認。

第一次登入系統：

當使用者透過 LM 將帳號、密碼傳送至 AM 後，AM 依據 LM 所傳送過來的帳號與經過雜湊運算的密碼，和資料庫中存放使用者密碼的欄位值比對。若使用者輸入的認證資料無誤，則由資料庫存放角色對應表中取得該使用者的角色代碼，連同認證成功訊息一併存入session中，最後在cookie中存入經過雜湊運算的身分證字號、使用者IP位址、cookie的有效時間、以k當金鑰加密的session id及雜湊訊息確認碼(Hash Message Authetication Code ,HMAC)，提供使用者再度登入系統時，僅需傳送cookie值，系統便可確認使用者身分的合法性。本系統安全cookie 的協定格式如下圖4.3。

$$\text{Hash(ID)} \mid \text{IP} \mid \text{Expiration time} \mid (\text{session id})_k$$
$$\mid \text{HMAC}(\text{Hash(ID)} \mid \text{IP} \mid \text{Expiration time} \mid (\text{session id})_k, sk)$$
$$\text{Where } k = \text{HMAC}(\text{Hash(ID)} \mid \text{Expiration time}, sk)$$

圖 4.3 本系統之安全 cookie 協定

我們所設計的安全Cookie Protocol與 Alex[10] 在2005年所提的安全Cookie

Protocol，加密金鑰都是獨一無二，對於本系統來說，每一個使用者的身分證字號和 Cookie 有效期限是不同的，因此產生出來的加密金鑰也將不同；而且加密金鑰都無法偽造，因為加密金鑰有一般使用者無法得知的server key。不過在Cookie中存放重要資料仍是較不恰當的做法，即使資料是加密過的，因為加密過的資料只需要花費多一點的時間即可破解，但對於本系統所設計的安全cookie protocol，惡意攻擊者若取得加密過的session id，但無法在cookie或session過期的時間內解出，進行重送攻擊，取得的session id也不具任何意義；另外為了防止重送攻擊，我們也在cookie中加入的IP值，若使用者來源IP與cookie中的IP值不符的話，也是無法登入本系統。

已通過認證，再度登入系統：

當使用者通過帳號/密碼認證再度登入時，AM 會先比對使用者IP是否與Cookie中的IP值相符，再利用雜湊訊息確認碼檢查使用者所傳送的cookie是否合法，若cookie合法，則將存放在cookie中的session id解密後，可依此session id存取伺服器中session的認證訊息與授權的角色權限；若cookie不合法或session存活時間逾期，則將使用者重導至登入頁面，讓使用者重新輸入帳號/密碼認證。下圖4.4為使用者第一次登入與通過認證後，再度登入系統之認證流程圖。

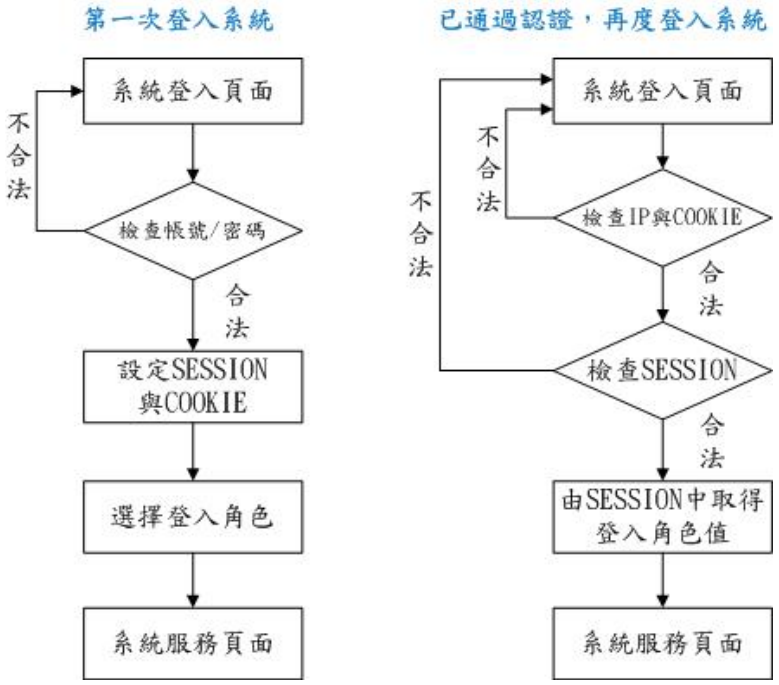

圖 4.4 登入系統之認證流程圖

3. 單一登入服務模組（Single Sign-On Module ,SSOM）：

- 工作內容：

(1) 提供管理者代理登入網站資料之新增、修改、刪除等管理功能。

(2) 提供登入網站之帳號/密碼加密，以確保存入資料庫之帳號/密碼的私密性。

(3) 能夠代替使用者登入外部服務網站，而不需要再輸入帳號/密碼。

(4) 定時通知使用者完成業務填報。

- 設計理念：

單一登入服務模組為本系統的主要核心功能，提供使用者與外部服務網站溝通的橋梁，因此資料庫中必須設計一個資料表，來存放網站基本資料、網站所屬部門、代登入程式檔名與經過加密的外部服務網站帳號/密碼，以便管理者可以維護外部服務網站的資料與代登入的帳號/密碼。本系統所設計的代登入網站資料表結構如表4.1。

表 4.1 代登入網站資料表結構

欄位名稱	欄位型態	欄位說明
site_id	int	網站編號
site_name	varchar	網站名稱
site_url	varchar	網站網址
site_info	varchar	網站說明
dept_id	int	網站所屬部門編號
login_agent	varchar	代登入的程式檔名
sso_id	varchar	經過加密的帳號
sso_pd	varchar	經過加密的密碼

在大部分的網站環境中，使用者欲使用網路服務前，必須要先呼叫網站的登入網頁，接著透過 HTML 表單(Form)中的 POST 傳送方式，將表單中輸入的訊息傳遞出去，因此可以不管外部服務網站的登入網頁是用何種語言所寫，只要取得外部服務網站登入表單所要傳送的欄位參數，加入要傳送的參數值，設計出一個代理登入程式，替使用者進行代填帳號/密碼的登入工作。另外，我們為了提升代理登入帳號/密碼的安全性，因此採用 AES 加密演算法（Advanced Encryption Standard，又稱Rijndael加密法）來對代登入帳號/密碼加密後才存入資料庫中，以抵抗目前已知的攻擊，強化系統的安全性，也避免影響到外部服務網站的系統運作。

雖然我們利用代理登入的方式讓行政人員不用反覆輸入帳號、密碼，不過對於眾多服務網站，行政人員也難免會有所疏失而忘記填報，所以也提供一個提醒的機制，讓行政人員皆能順利完成業務。

4. 使用者資訊儲存模組（User Data Module , UDM）：

- 工作內容：

(1) 提供使用者維護個人基本資料與登入密碼之變更。

(2) 提供管理者修改、刪除與查詢使用者資料。

- 設計理念：

使用者在首次登入進行認證程序時，需要核對使用者輸入的帳號/密碼，以確認使用者的身分是否合法，因此我們建立一個使用者資料表來儲存使用者的相關資訊，讓使用者可以自行維護基本資料，而管理者也可以針對特定的使用者來進行資料異動的工作。另外，為了確保資料的機密性，我們將 user_pass 欄位值透過SHA1的雜湊運算後，才儲存到使用者資料表中。使用者資料表結構如下表4.2。

表 4.2 使用者資料表結構

欄位名稱	欄位型態	欄位說明
uid	int	使用者編號
ID	char	身分證字號
user_name	varchar	使用者名稱
user_pass	char	使用者密碼
user_cname	varchar	使用者姓名
Email	varchar	電子郵件
Dept_id	int	使用者所屬部門編號
ip	varchar	登入IP位址
last_login	datetime	登入時間

5. 授權與權限管理模組（Authorization and Privilege Management Module ,APM）：

- 工作內容：

(1) 提供管理者新增、修改、刪除與啟用角色之功能。

(2) 提供管理者角色權限設定之功能。

(3) 提供管理者指派角色值給使用者、群組、團隊之功能。

(4) 提供管理者新增、修改、刪除群組及設定使用者與群組對應之功能。

(5) 提供管理者新增、修改、刪除群組及設定使用者與團隊對應之功能。

(6) 提供管理者新增與指派任務之功能。

- 設計理念：

授權與權限管理模組存放有系統架構中權限管理機制裡各個角色（Roles）、群組（Groups）、團隊（Teams）、任務（Tasks）與其對應的權限資訊，因此 AM 要授權使用者相關權限時，就必須從授權與權限相關資料表中取得資料。

由於在國小的行政環境中，會有一群人同時執行相同的工作或一群人分工合作執行一件任務的情形，因此我們依此需求規劃出更符合校務行政現場的角色存取關係，來取代傳統、群組或團隊的 RBAC 設計，在此我們共規劃了十個資料表來存放授權與權限管理資料，表4.3簡略說明了每個資料表所存放的資料內容。

表 4.3 授權與權限管理資料表

資料表名稱	說明
role	角色編號與名稱
role_assign	使用者、群組、團隊與角色的對應關係
role_permiss	每個角色被賦予的權限
group	群組編號與名稱
group_mapping	使用者與群組的對應關係
team	團隊編號與名稱
team_mapping	使用者與團隊的對應關係
task	任務編號與名稱
task_assign	任務與執行團隊對應關係
task_permiss	每個任務被賦予的權限

此十個資料表是以不同功能做為切割資料表的依據，以方便管理者可以針對不同的資料表來做角色指派與權限設定等維護工作，圖4.6是各資料表間交互參考的關聯圖。

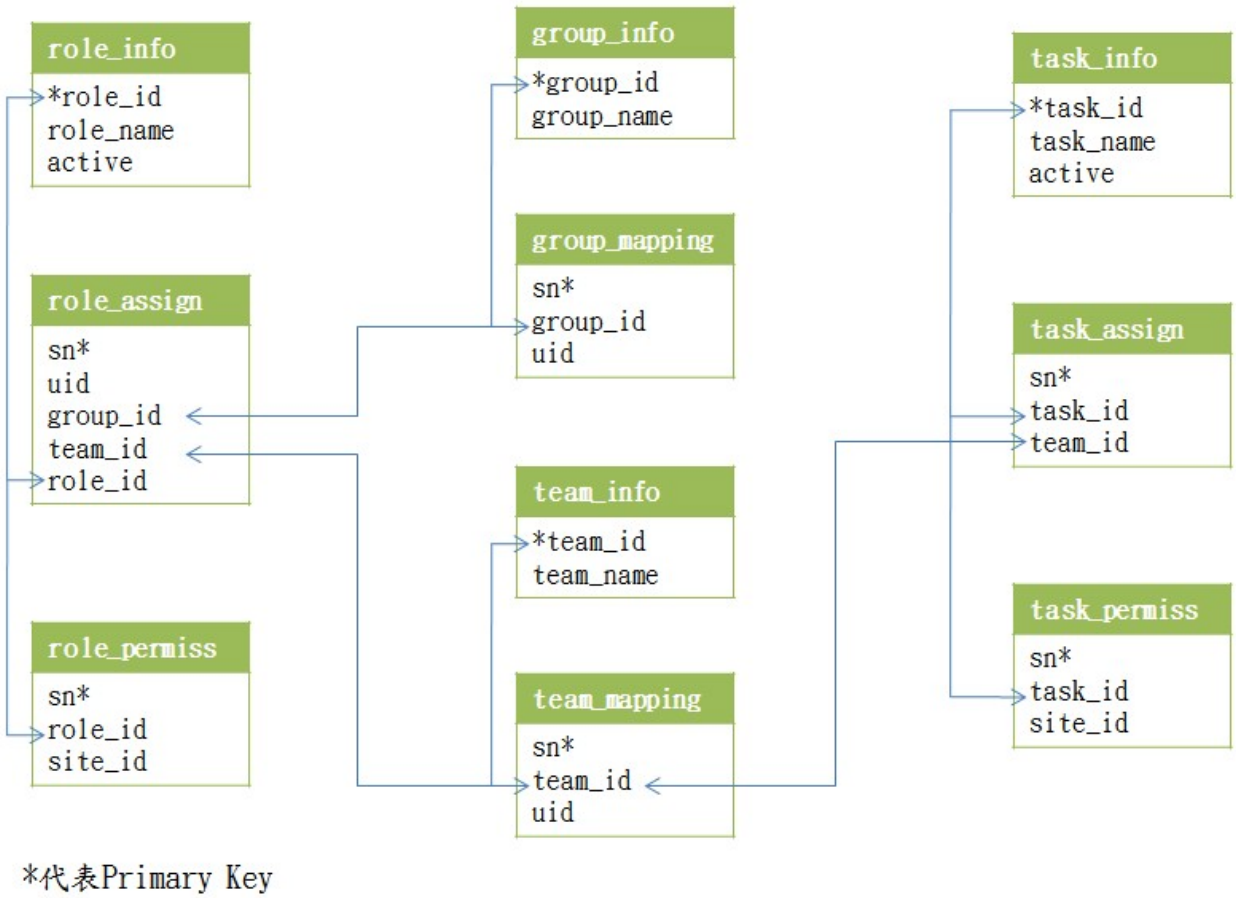

圖 4.5 授權與權限管理資料表關聯圖

6. 稽核管理模組（Audit Management Module，AuM）：

- 工作內容：

(1) 提供管理者自行設定需稽核之項目。

(2) 紀錄使用者變更資料前的密碼、時間與來源IP。

(3) 紀錄管理者變更角色指派前之角色值。

(4) 紀錄管理者變更角色權限前之權限值。

(5) 紀錄管理者變更代登入網站資料之網站編號與變更前之代登入帳號/密碼。

- 設計理念：

在開放的網路環境中隨時都有許多事件發生，這包括合法與非法的所有行為，所以沒有一套資訊系統可以保證百分之百的安全，因此必須採用一些事後補救的措施來補強不足的部份，稽核最主要的目的也就是要記錄所有的事件，如使用者登入時間、來源IP位址、密碼與角色變更、存取權限與代登入的帳號/密碼異動等，因此我們將稽

核的事件分成兩個階段來執行，分別為登入階段與資料異動階段。當使用者輸入的帳號或密碼有誤時，系統會將登入的錯誤訊息、登入帳號、使用者來源IP與登入時間存入稽核資料表中，讓管理者可以做進一步的管制措施；若使用者登入成功，也會將登入資訊存入稽核資料表，若使用者更進一步執行登入密碼的變更或以管理者身分重新指派角色與權限設定，也都會一一記錄在稽核資料表中。當意外事件發生時，我們便可以透過稽核管理模組提供的查詢與報表列印功能，調閱出事件發生的時間與變更的事項，並且將被使用者非法竄改的資料迅速修正或回復為原本的資料，以確保系統的安全性。表4.4與4.5分別為登入階段與資料異動之稽核記錄資料表結構。

表 4.4 登入階段之稽核記錄資料表結構

欄位名稱	欄位型態	說明
sn	int	流水編號
status	int	登入成功或失敗狀態
user_name	varchar	使用者登入的帳號
ip	varchar	使用者登入的IP位址
ch_time	datetime	使用者登入的時間

表 4.5 資料異動之稽核記錄資料表結構

欄位名稱	欄位型態	說明
sn	int	流水編號
user_cname	varchar	變更資料的使用者名稱
ip	varchar	變更資料的使用者IP位址
ch_time	datetime	變更資料的時間
ch_user_pd	char	變更密碼前的使用者密碼
ch_role	varchar	變更角色指派前的角色代碼
ch_permiss	varchar	變更角色權限前的權限代碼
site_id	int	代登入帳號/密碼變更的網站編號
ch_site_ssoid	char	變更前的代登入帳號
ch_site_ssopd	char	變更前的代登入密碼

4.2 CDSASS運作流程

本節針對前一節所提跨網域之校務單一登入系統架構的運作流程做說明，並將系統運作流程分為第一次登入與已通過認證再度登入兩種情況個別介紹，且詳細的描述使用者如何透過跨網域校務單一登入系統的模組來達到與外部服務網站的溝通。

使用者第一次登入本系統，利用代登入方式與外部服務網站溝通流程共分七個步驟，以下分別說明每一個步驟所交換的訊息內容，其運作流程如圖4.7所示：

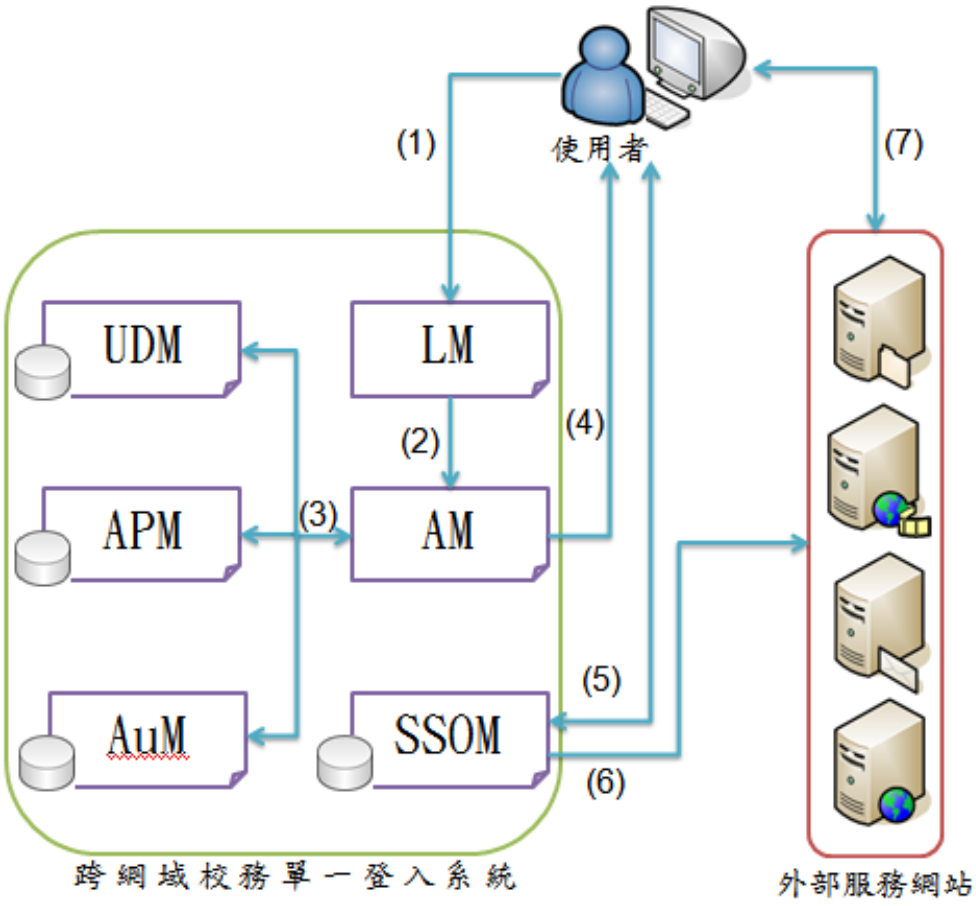

圖 4.6 第一次登入系統之運作流程圖

(1) 使用者 → LM

使用者透過 HTTPS 加密通道連至登入頁面，並輸入帳號、密碼等認證資料以進行登入動作。

(2) LM → AM

LM 將使用者輸入的帳號與經過雜湊函數運算的密碼送至 AM。

(3) AM → (UDM || APM || AuM)

AM 根據 LM 所傳來的使用者帳號與經過雜湊函數運算的密碼，向 UDM 取得該帳號所對應的密碼與之比對。若使用者輸入的認證資料無誤，AM 便向 APM 索取該使用者的角色權限代碼，連同認證成功的訊息存入session中，最後 AM 將使用者登入成功的紀錄存到 AuM 資料表中。若使用者輸入的認證資料錯誤，AM 會將畫面重導回 LM 的登入頁面，並將使用者登入失敗的紀錄存到 AuM 資料表中。

(4) AM → 使用者

AM 將經過雜湊運算的身分證字號、使用者IP位址、cookie有效的時間、加密的session id及雜湊訊息確認碼(Hash Message Authetication Code ,HMAC)儲存在Cookie中回傳給使用者。

(5) 使用者 → SSOM

使用者向 SSOM 請求服務， SSOM 依據 Session 中的角色代碼，取得角色權限並列出使用者可執行的代登入連結，以提供使用者選擇所需服務。

(6) SSOM → 外部服務網站

SSOM 根據使用者選定的服務，向 SSOM 資料表取出登入該服務網站的帳號、密碼進行解密後，執行跨網域代登入的動作。

(7) 使用者 → 外部服務網站

使用者與外部服務網站開始溝通，完成跨網域單一登入的動作。

若使用者已通過認證，再度登入本系統，則此過程將省去帳號/密碼的認證與角色的授權，而此種狀況的運作流程共分六個步驟，以下也分別說明每一個步驟所交換的訊息內容，圖4.8為已通過認證，再度登入系統的運作流程圖。

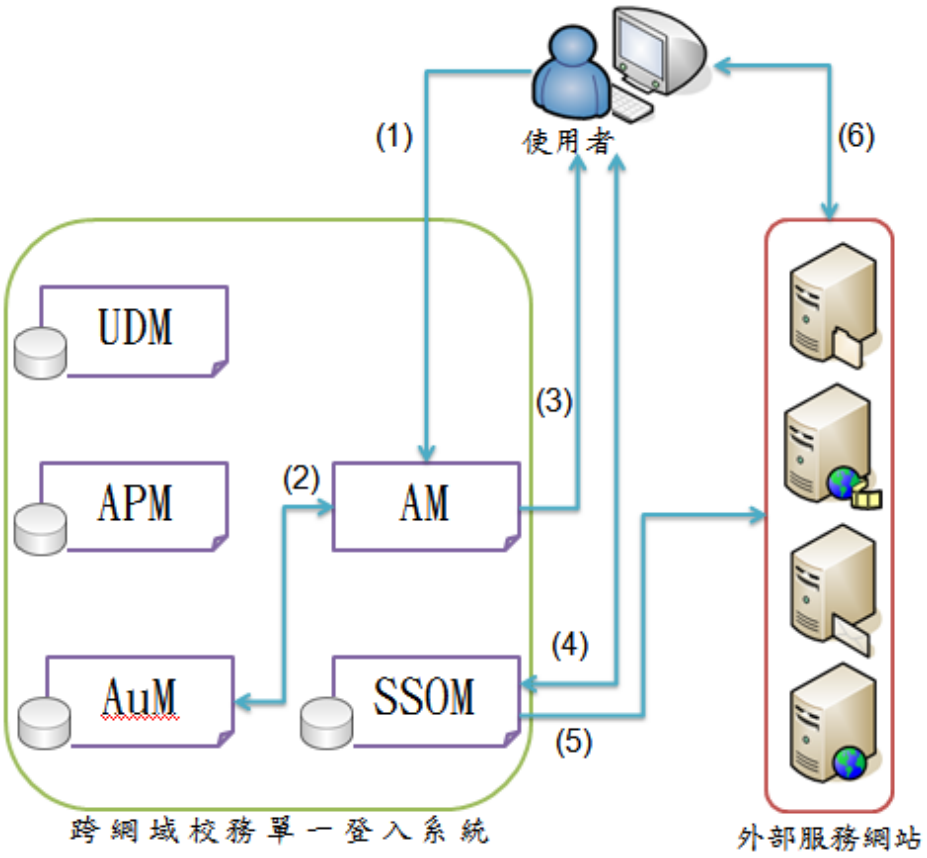

圖 4.7 已通過認證，再度登入系統之運作流程圖

(1) 使用者 → AM

使用者將前一次登入後所取得的cookie認證資料傳送給 AM ， AM 依序檢查以下項目：

- cookie是否過期。
- cookie中的IP位址是否與使用者來源IP相符。
- cookie中資料是否遭到竄改。

接著利用加密金鑰 K_s 將session id解密，若session未過期，就根據session id取得存放於伺服器的session值，並檢查此使用者身分是否合法。若使用者身分合法，則依照存在session中的角色權限執行該使用者的請求。

(2) AM → AuM

AM 會將使用者登入成功或失敗的相關資訊記錄到 AuM 資料表中。

(3) AM → 使用者

AM 將登入成功訊息傳回給使用者，並將會面重導至系統服務頁面。

(4) 使用者 → SSOM

使用者向 SSOM 請求服務， SSOM 依據 Session 中的角色代碼，取得角色權限並列出使用者可執行的代登入連結，以提供使用者選擇所需服務。

(5) SSOM → 外部服務網站

SSOM 根據使用者選定的服務，向 SSOM 資料表取出登入該服務網站的帳號、密碼進行解密後，執行跨網域代登入的動作。

(6) 使用者 → 外部服務網站

使用者與外部服務網站開始溝通，完成跨網域單一登入的動作。

4.3 CDSASS安全性探討

在使用者進行單一登入過程中，可能會遇到不可預知的攻擊或安全性的問題，我們將對本系統所提供的安全服務分為身分識別與認證、機密性、完整性、防止重送攻擊、權限存取控制、稽核、單一登入等七項來逐一探討，並說明本系統如何確保單一登入過程的安全性。

身分識別與認證（Identification and Authentication）

認證性是讓使用者和伺服器可以互相驗證對方的身分是否合法，並且不會遭受到第三者偽裝訊息而干擾。

本系統之認證性將針對登入階段與登入後服務請求階段加以說明：

- 登入階段：

  使用者透過 SSL 安全通道來檢查伺服器的憑證，以確認伺服器的合法性；伺服器則透過使用者輸入的帳號、密碼來確認使用者的身分是否合法，若合法則傳給使用者一個安全的 Cookie。

- 登入後服務請求階段：

  使用者利用此安全的 Cookie 向伺服器請求服務；伺服器檢查使用者所傳送的 Cookie 是否合法，若合法則回傳使用者請求的資源。

機密性(Confidentiality)

機密性是提供通訊兩端一個經過加密具有機密性的連線，來保護使用者與伺服器之間所傳送的資料。

在圖4.7的(1)、(4)、(5)三個步驟中，都將透過 SSL 安全連線來傳遞訊息，以確保使用者所輸入的帳號、密碼與 Cookie 中的資訊不會遭到竊聽的攻擊，進而達到資料傳輸的機密性。且 Cookie 中所儲存的數值，並不具有實質意義，因此即使儲存在使用者電腦中的 Cookie 被竊取，也不會直接對本系統造成威脅。

由於使用者的登入密碼與執行代登入的帳號/密碼儲存在資料庫中，如果有惡意攻擊者直接讀取該資料庫則有資訊外洩的問題。針對此問題，我們使用 SHA-1 雜湊演算法對使用者登入密碼加密，並將執行代登入的帳號/密碼透過 AES 對稱式加密演算法加密後才儲存，因此可以保障使用者帳號、密碼的安全。

完整性(Integrity)

完整性是要確保資料在網路傳遞的過程中，不會遭受無意或惡意的竄改，以便維持資料的正確性。保護資料完整性常見的做法是將資料經由雜湊函數運算後產生一個唯一且不可逆的雜湊值，由於雜湊值產生碰撞（Collision）的機率非常低，所以只要資料在傳送的過程中有些微的改變，都可以被偵測出來。

在圖4.8的步驟(1)中，為了確認使用者的 Cookie 是否在傳遞的過程中遭到修改，我們將 Hash（ID）、IP、Cookie 過期時間、加密的 Session id 與 Server key 透過 SHA-1 運算後與存放在 Cookie中的Hmac欄位值比對，以保障資料的完整性。

防止重送攻擊(Anti-replay)

重送攻擊是指第三者竊取通訊雙方的認證訊息，並原封不動地將訊息送出，以假扮成原訊息發送者的身分，進而取得相關服務。

本系統利用IP_cookie來做為防止重送攻擊的第一道防線。只要使用者來源IP與Cookie中的IP值不符，則無法成功登入；另外，由於使用者每次登入之後，均會派發新的會議金鑰（Session id），而且登出之後會將該次登入的相關會議金鑰廢止，所以並無法重複使用上一次所得到的會議金鑰。但並非每位使用者離開時都會點選登出選項，因此我們也提供使用者暫時性Cookie（Session Cookie）的功能，只要使用者關閉瀏覽器，系統便會刪除存放在使用者電腦中的Cookie，以防止重送攻擊的事件發生。

存取控制(Access Control)

存取控制（Access Control）[16]是一種管理模式，詳細規定了使用者可以從事哪些動作，並限制使用者做出危害系統安全的動作。

由於以角色為基礎的存取控制（Role-Base Access Control）具備三個基本安全原則[2][11][20]，因此本系統為了提升存取控制的安全性，以傳統 RBAC 為基礎，整合了DRBAC和TT-RBAC的優點，利用群組(Group)、團隊(Team)和任務(Task)來簡化角色的權限設定，讓我們的權限存取控制更具有彈性。由於我們的角色值與角色權限皆儲存在資料庫中，因此我們設計一個以網頁為基礎（Web-Based）的管理介面，來進行權限存取管控的動作，以提升管理的效率。

稽核(Audit)

稽核功能是記錄系統的重要訊息，讓管理者可以隨時注意系統是否有任何的異常狀況或資料庫資料是否被修改。

本系統在使用者進行登入時，便會記錄使用者的IP來源、登入時間以及登入成功或失敗的訊息。且當使用者異動密碼或惡意攻擊者以管理者身分重新指派角色與權限設定時，皆會記錄在稽核資料表中。因此管理者可以很快速的察覺系統的問題，進一步採取管制措施或將資料復原，以確保系統的安全性。

單一登入(Single Sign-On)

單一登入就是讓使用者經過一次身分認證後，而不需要重覆輸入認證資訊，就可以獲得適當的權限。

使用者第一次要使用系統所提供的服務時，需要在登入頁面輸入合法的帳號、密碼，這時使用者所得到的認證成功訊息及角色權限都會被儲存在 Session 當中，此後使用者在有效時間內，如果要登入其他外部服務網站時，就不需要再重新輸入帳號、密碼，因為系統會去讀取 Session 中的認證訊息及角色權限，以確認使用者身分的合法性，並代替使用者進行代理登入(Agents sign)的動作，以符合單一登入的特性，並減少資安問題的發生。

第五章 系統建置與分析

經過第四章詳細的介紹系統規劃與設計重點之後，接著就是將所有的想法付諸實現，我們將依照設計理念來建置一套完整的系統，之後再進一步來評估本系統的實用價值。因此本章節將先介紹本系統的建置環境，接著對選用的開發工具及相關的函式庫做說明，並以實作出來的畫面展示本系統的成果，最後對本系統實作後的應用做評估分析。

5.1 系統建置環境

本研究考量到跨平台的特性，因此在選擇網頁伺服器軟體就必須首先考量系統相容性的問題，並希望能以校園中原有的伺服器來安裝本系統，而不需要另外增加硬體支出。另外，系統的建置成本也是一大考量，選用自由軟體可以節省許多軟體購置經費，這對於經費較於拮据的中小學來說，也是較容易採納的一個方法。因此本系統採用目前使用率最高、免費且最穩定的 WEB 伺服器：Apache HTTP 伺服器，並且利用OpenSSL 所提供的 SSL 加密模組，讓 WEB 伺服器擁有更高的安全性。

由於 Apache支援多種作業系統平台，但為了考量系統整體的運作效能，筆者這邊則採用 Unix-like 的作業系統：FreeBSD 作為開發平台；另外在資料儲存上，我們則採用同樣是免費的資料庫軟體：MySQL來作為資料庫管理系統；並以跨平台、內建多樣化函式庫的 PHP5 來作為程式開發語言，且利用 PHP5 支援的mcrypt編碼函式庫及mhash雜湊函式庫，來確保資料庫中資料的安全。而本系統所採用的硬體及軟體套件版本詳列如表5.1，另有本系統採用的加密演算法及其功能如表5.2，可供系統建置之參考。

表 5.1 系統建置硬體及軟體套件版本

用途	套件名稱及版本
硬體環境	Pentium-4 2.0GHZ 256MB
作業系統	FreeBSD 6.2-Release
伺服器軟體	
-網頁伺服器	Apache 2.2.3
-資料庫	MySQL 5.0.27
開發工具	

-伺服器端網頁程式語言	PHP 5.1.6
-加解密工具	
-非對稱式加密(RSA)	OpenSSL 0.97e
-對稱式加密(AES)	mcrypt 2.5.7
-雜湊函數(SHA-1)	mhash 20060101
-資料庫管理介面	phpMyAdmin 2.9.0.2

表 5.2 本系統所採用的加密演算法及其功能

用途	演算法	加密長度(bits)	功能
對稱式加密演算法	AES (Rjindael)	256	負責資料庫資料之加密
非對稱式加密演算法	RSA	1024	負責解密金鑰之傳遞及使用者上傳資料之驗證
雜湊演算法	SHA-1	160	驗證傳送內容之正確性

5.2 系統模組開發

本系統是一套能滿足國中、小校務行政填報業務與個人化需求的跨網域單一登入系統，所以在模組開發上，我們盡可能依照現行的教育行政體制與操作簡單易懂的概念去建置，以利系統功能擴充與其他系統之整合，並採用 Web-Based 作為系統的介面，以方便使用者與管理者操作，最後為了整體開發成本之考量，決定採用開放原始碼所提供之函式庫或工具，並將其適當的整合運用至本系統中。圖5.1及圖5.2分別為使用者登入本系統後與管理者進入系統管理之畫面，畫面左邊為功能選項。

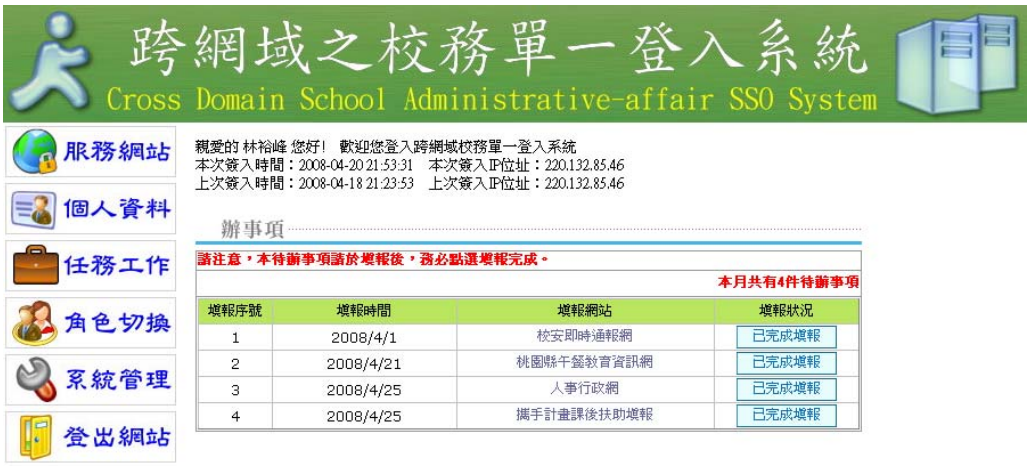

圖 5.1 使用者登入畫面

圖 5.2 管理者操作畫面

下面我們就針對各模組的開發逐一的介紹：

1. 登入與登出模組（Login Module ,LM）

由於本系統是採用帳號/密碼的登入方式，因此開發此模組的首要工作便是確保資料傳輸過程的安全，因此我們透過 OpenSSL 所提供的加密函式庫來實作 SSL 服務，以確保使用者輸入的資料與後續跟本系統溝通的安全，圖5.3為使用者經過 SSL 加密的登入畫面。

圖 5.3 SSL 加密的登入畫面

當使用者點選「登出網站」或傳送過期的 cookie 資訊欲登入時，我們利用PHP提供的Session函式庫：session_destroy與HTTP 相關函式庫：setcookie，來刪除伺服器端的session資料與 AM 所派發的cookie資訊。若使用者在登入時，Cookie有效期選擇即時的話，即使使用者未執行登出的動作，只要使用者直接關閉瀏覽器，Cookie 與 Session 也將會立即失效。

2. 認證模組（Authentiation Module ,AM）

認證模組主要負責使用者的認證工作，依運作流程階段分為帳號/密碼認證、Cookie認證與Session認證。

當使用者第一次登入本系統時，必須先通過帳號/密碼的認證，才能取得相關的服務。雖然資料在傳輸的過程中有 SSL 加密的保護，不過由於認證密碼都儲存在資料庫中，因此我們採用 mhash 函式庫中的 MHASH_SHA1 雜湊運算法來對登入密碼作加密，以確保資料庫中資料的安全。

若使用者輸入的帳號/密碼與使用者資料表中user_name、user_pass 的欄位值相

符，AM 便會向 role_assign 資料表取得該使用者的角色代碼，連同認證成功訊息一併存入 Session 中。認證成功的 Session 內容範例如下：

```
valid_info|s:2:"ok";uid|s:1:"1";user_cname|s:6:"林裕峰";role_id|s:1:"1";
```

（Session 內容就是「變數名稱｜變數類型：長度：內容；」的組合）

使用者認證成功後，AM 除了在伺服器端存入一個 Session 值外，也會產生一個安全的 Cookie 並回傳給使用者。依照我們第四章所設計的安全cookie， AM 會利用setcookie函數，將Hash(ID)、IP位址、Cookie過期時間、經過加密的session id及雜湊訊息確認碼(Hash Message Authetication Code ,HMAC) 存入Cookie中，讓使用者在有效時間內，能透過傳送 Cookie 值來確認身分的合法性，而不須再輸入帳號/密碼即可登入本系統。

最後為了強化資料的安全性，因此採用 PHP 提供的 mcrypt 編碼函式庫：MCRYPT_RIJNDAEL_256 加密演算法來對我們的session id加密，並且為了提高駭客破解的難度，以 Hash(ID) 加上管理者設定的server key當作加密金鑰，讓每位使用者所使用的加密金鑰皆不同。

當使用者通過帳號/密碼的認證再度登入時，AM 會先比對 Cookie 中的雜湊訊息確認碼，檢查使用者所傳送的Cookie是否合法，若 Cookie 合法，則將存放在 Cookie 中的 Session id 解密後，依照 Session id 取得原先存放在 Session 中的認證訊息與授權的權限資料，並將畫面重導至服務頁面。若 Cookie 不合法或 Session 有效時間逾期，則將使用者重導至登入頁面，讓使用者重新輸入帳號/密碼認證。

3. 單一登入服務模組（Single Sign-On Module ,SSOM）

本模組在單一登入功能設計上，分為行政人員業務網站與個人服務網站，因此必須提供管理者新增網站、網站維護與網站列表等功能來維護行政人員業務網站，如圖5.5。由於網路服務的登入頁面大都是透過 HTML 表單(Form)中的 POST 傳送方式將訊息傳遞出去，因此本模組在開發前，須事先蒐集各行政人員與個人常登入之服務網站，接著取得外部服務網站 HTML 表單所要傳送的欄位參數，加入從代登入網站資料表中取出的帳號、密碼，再搭配 Java Script 程式語法，先行將代登入的程式設計好，以便替使用者進行代填帳號/密碼的登入工作。圖5.4是以桃園縣網路公文系統為範例的代理登入程式。

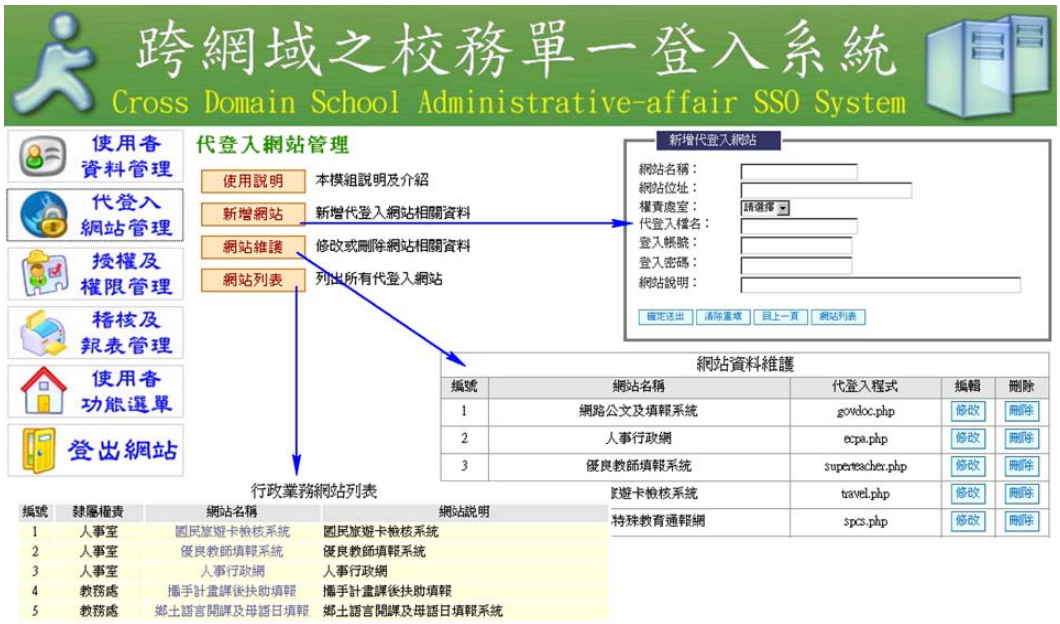

圖 5.4 代登入網站管理功能列表

```
<form action="http://doc.tyc.edu.tw/password.asp" method="post" name=Form>
<input type=hidden name=account size=8 value="<?echo $SSO_Site_ID;?>">
<input type=hidden name=password size=8 value="<?echo $SSO_Site_PD;?>">
<input type="hidden" name="menuname" value="">
<input type="hidden" name="menu" value="">
<input type="hidden" name="url" value="/govdoc/receivedoc.asp">
</form>

<script>
  document.Form.submit();
</script>
```

圖 5.5 桃園縣網路公文系統代理登入程式

另外，我們也提供行政人員可以自行設定填報業務定期提醒的功能，如圖5.6，讓行政人員面臨眾多填報的網站而不會忘記定時完成填報業務。

圖 5.6 定期填報業務提醒設定畫面

最後我們也將個人常用的服務網站轉換成代登入程式，提供一般使用者自行新增個人服務網站的功能，如圖5.7，以減少使用者反覆輸入帳號、密碼的次數。

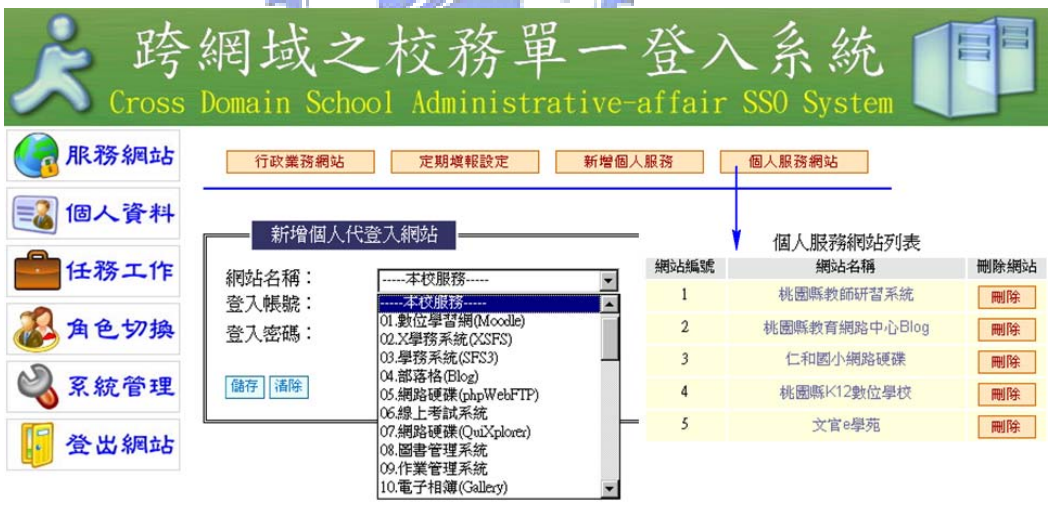

圖 5.7 新增個人代登入服務網站畫面

由於我們設計的代登入程式，都需要讀取存放在資料庫中的帳號、密碼，因此儲存在代登入網站資料表中的代登入帳號/密碼極為重要，若被惡意攻擊者所取得，將會產生資安風險，因此我們利用透過 AES 加密演算法（MCRYPT_RIJNDAEL_256）將代登入帳號/密碼加密，以防止惡意攻擊者直接看到代登入帳號/密碼。

4. **使用者資訊儲存模組（User Data Module , UDM）**

本模組設計的功能，以維護使用者個人基本資料為主，因此對於一般使用者而言，他們可以透過此模組來修改基本資料或變更登入密碼；相對的，管理者也可以利用此模組維護所有使用者的基本資料，圖5.6為使用者資料管理功能列表。

為了防止使用者短暫的離開而未登出系統，導至密碼被變更，因此使用者在變更密碼時，必須先輸入舊的密碼，才可以將新密碼存入資料庫中，並且變更的資訊也會被記錄至稽核資料表中，以防止惡意使用者的任何行為。

圖 5.8 使用者資料管理功能列表

5. **授權與權限管理模組（Authorization and Privilege Management Module ,APM）**

由於使用者、群組、團隊與角色及權限的關係，皆儲存在資料庫中，因此我們必須設計一個介面來提供管理者可進行角色資料維護、角色指派與權限設定等相關作業。本模組主要提供了四個主功能：角色維護、群組維護、團隊維護與任務維護。透過角色維護管理功能，管理者可以新增角色資料、編刪啟用角色、角色權限設定、使用者帳號與角色對應、群組與角色對應、團隊與角色對應之設定，如圖5.7。

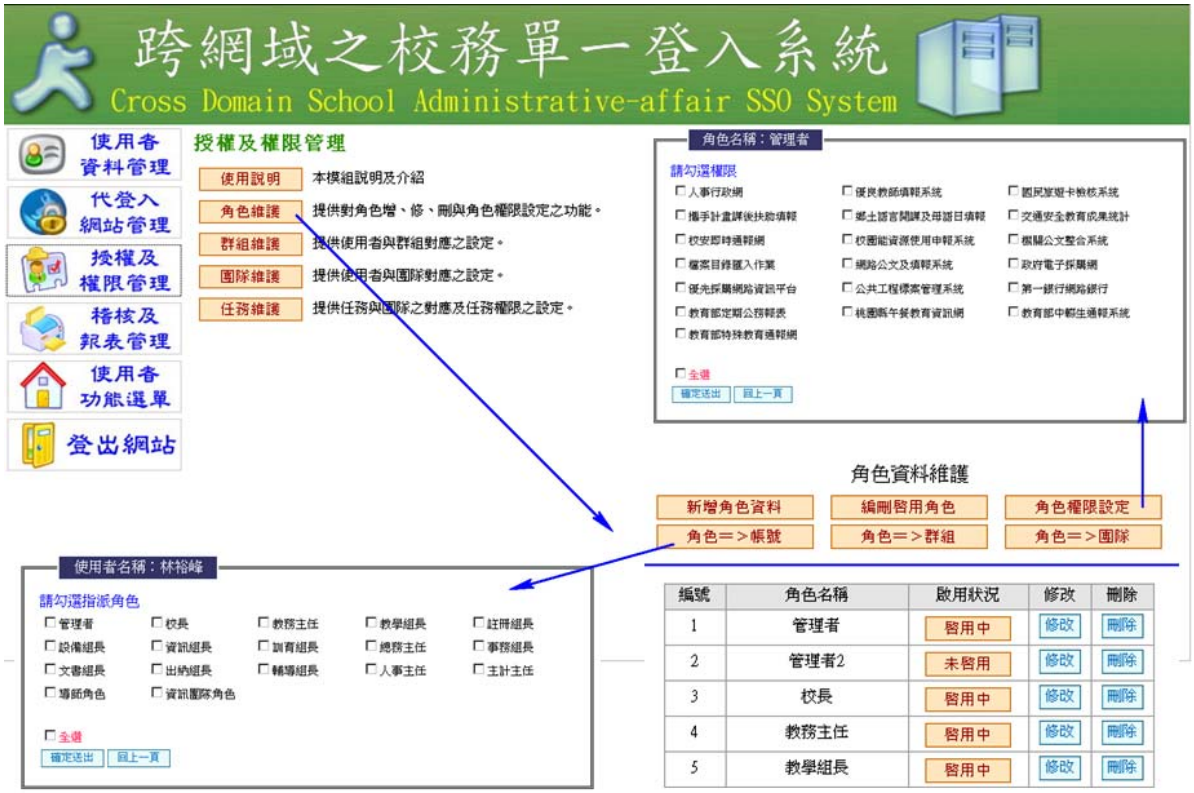

圖 5.9 角色維護管理功能選單

管理者可依照教育行政編制，將角色名稱先行建立好，接著設定每一個角色所擁有的權限，最後再將角色值指派給相關使用者、群組或團隊，讓使用者成功登入系統後，即可依照其角色身分來執行該角色所被指派的權限。

另外，對於角色屬性相同的使用者，我們把他們設為同一個群組，再將選定的角色指派給該群組，讓管理者不用重複設定相同角色給每位使用者；若相同角色屬性的使用者變更角色時，也只需要重新指派角色給群組即可。

圖5.8為群組維護管理功能選單，管理者透過此功能選單，可減少設定所花費的時間，藉此也能減輕管理者的負擔。

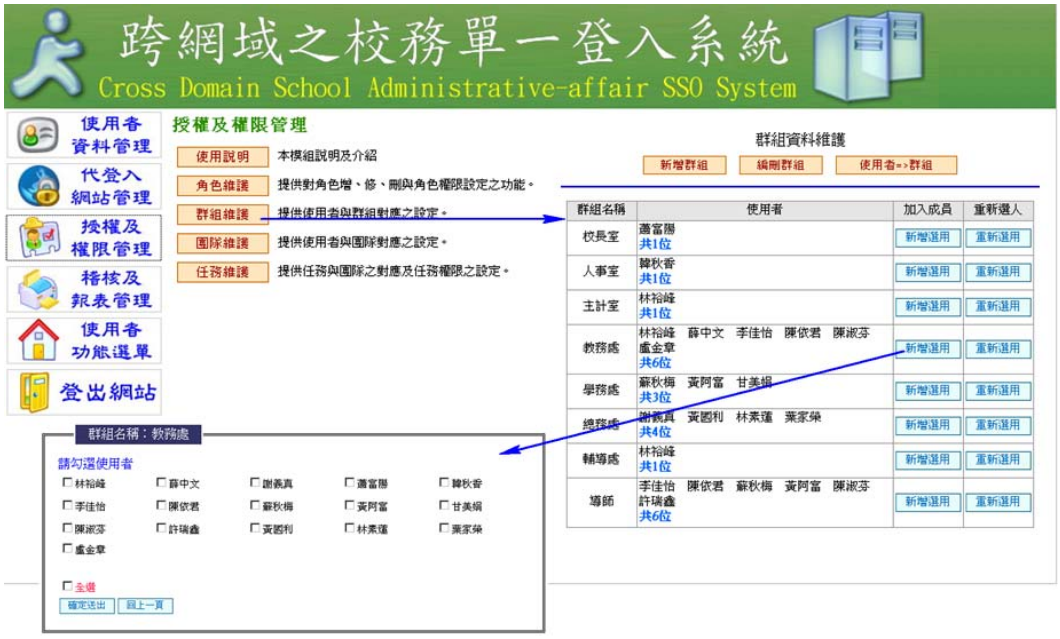

圖 5.10 群組維護管理功能選單

對於某些需要團隊分工合作才能完成的任務，我們也設計一個管理介面提供管理者新增團隊名稱或依需求將使用者選入同一團隊中之功能，讓每位使用者都能依照自己的專長或角色來完成任務，團隊維護管理功能選單如下圖5.9。

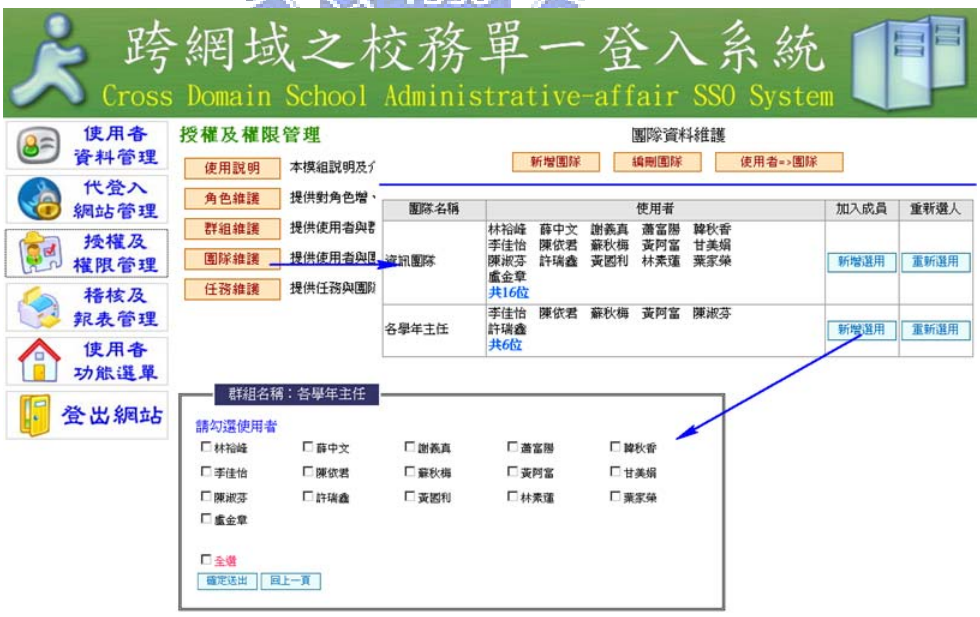

圖 5.11 團隊維護管理功能選單

最後，我們也必須提供管理者可以將新增的任務指派給執行團隊之功能，如圖5.10，讓此團隊可以依據任務的權限，完成自己本身負責之事項。

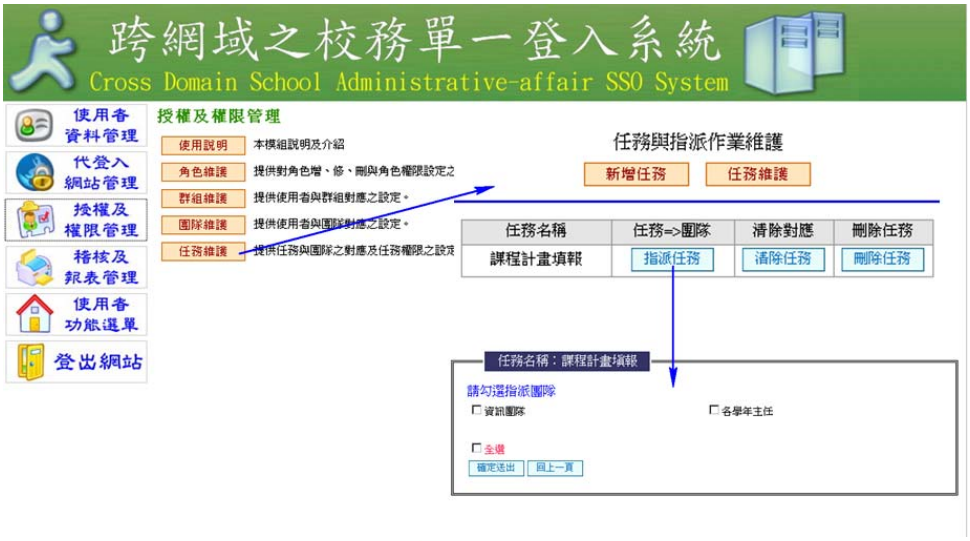

圖 5.12 任務指派功能選單

6. 稽核管理模組（Audit Management Module，AuM）

本模組的設計是為了讓管理者隨時掌握非法的操作行為，因此我們設計一個查詢與列印的管理介面，經由此介面可以呈現登入資訊、登入密碼異動、代登入帳號密碼異動、角色指派異動、角色權限異動等相關資訊，供管理者進行系統安全補強或資料復原等工作，圖5.11為系統稽核畫面。

圖 5.13 系統稽核畫面

5.3 CDSASS評估分析

本研究所提的跨網域校務單一登入系統是在不變更使用者環境、不增加額外費用，且兼顧使用者經驗、安全性及易於部署的情況下，希望能藉此簡化行政業務、提高行政效率。因此在使用界面上，我們採用網頁式（ Web-Based ）的操作介面，讓使用者透過一般瀏覽器即可操作運用，並利用代理登入（Agents Sign）的方式來進行跨網域登入，讓使用者不須改變任何操作行為。在程式設計方面，我們以模組方式來開發，讓系統更容易在不同環境中部署，且利用開放原始碼(Open Source)的軟體來設計，因此不用支出任何軟體費用；最後再搭配對稱式加密演算法（Symmetric Cryptographic Algorithm）、非對稱式加密演算法（Asymmetric Cryptographic Algorithm）、單向雜湊函數(One-Way Hash Function)來強化系統的安全性。在存取控制方面，我們以角色為基礎，運用使用者、群組、團隊與角色的對應關係，讓整個存取管控機制更具彈性，且更適用於國小教育現場。

本系統提供了許多便利服務與優點，但在實際運作當中也產生了一些問題，因此我們將本系統提供的服務與遇到的問題分析討論如下：

系統服務：

1、 提供網頁式單一登入介面：

透過網頁式單一登入介面來發送帳號、密碼，可以整合目前所有以傳統帳號/密碼登入為主的網站系統，而不需要記憶多組不同的帳號、密碼，以減少資安問題的發生。

2、 保有服務網站認證機制：

本系統能連結各種語言撰寫的網站系統，而且不需要根據某種標準而修改系統，也無需知會對方或是經過對方許可。

3、 具有彈性的權限管控機制：

以傳統 RBAC 為基礎，整合群組（Groups）、團隊（Teams）的概念，可提供符合各種狀況的解決方式。

4、 提供定時填報通知：

本系統提供使用者設定週期性的填報作業，可提醒行政人員定時完成任務。

5、 提供易於部署的系統架構：

本系統架構建置簡單，使用現行廣泛應用的技術方法即可完成建置，不論使用者端或是伺服器端均不受系統平台的限制。

6、 不變動使用者環境：

使用者在使用本系統時，不需要做任何環境的改變或另外安裝套件，只要是使用標準瀏覽器即可執行。

7、 保存使用者習慣：

透過本服務架構，不需要改變使用習慣，只要登入一次，即可存取所有外部服務網站。

系統遇到的問題：

1、 使用者自行修改服務網站帳號、密碼：

若使用者自行至服務網站修改登入帳號或密碼時，卻不告知系統管理者，將會造成代登入程式無法登入成功。為了確保業務帳號、密碼能夠順利移交，我們要求行政人員將業務帳號、密碼交出，並由本系統統一管制，以免業務承辦人遺失或忘記帳號、密碼，造成交接不實之情形，進而影響業務之推展。

2、 不以帳號、密碼認證的服務網站：

並非每一個網站都是利用帳號、密碼來認證，因此如果遇到透過智慧卡或非帳號、密碼來認證的網站，本系統就無法達到跨網域單一登入的功能了。

3、 使用者關閉 Cookie 之功能：

由於本系統是透過 Session 與 Cookie 來進行身分認證，因此使用者如果將 Cookie 的功能關閉，將造成系統無法認證使用者身分的合法性。為了讓使用者順利操作本系統，所以我們在登入畫面設計一個檢測 Cookie 是否啟用之功能，以便提醒使用者修改瀏覽器 Cookie 之設定。

第六章 結論與未來工作

由於目前資訊科技的蓬勃發展，使得人們透過網路無地域的限制讓知識分享的願景得以充分的實現在全球網際網路這個平台上，正因如此政府近年來也積極的推動全國 E 化政策，並提供多元化的網路服務來滿足大家的需求。不過面對日以劇增的服務網站，我們也必須逐步的將傳統各自獨立的資訊系統整合在一起，並藉由網路科技來提高競爭力，而不是帶給使用者的不便。

6.1 結論

經由第一章的研究動機說明，可以清楚了解本研究要做的方向，於是收集、研究了許多相關的文獻，如第二、三章所介紹，在探討的過程中，了解到單一登入相關技術、跨網域單一登入機制的特性。再對過去學者所提的方法做完問題分析之後，發現並無法滿足校務行政的需求，因此本研究利用現有的硬體設備，在不變更使用者環境、不增加額外費用，且兼顧使用者經驗、安全性及易於部署的考量下，提出一個適用於國小行政現場的跨網域校務單一登入系統服務架構，其主要目的在於：

1、解決使用者面臨眾多服務網站時，必須記憶多組帳號密碼的問題。
2、解決外部服務網站的安全認證機制無法修改的問題。
3、提出一個適合國小行政運作體制的權限管控機制，以解決業務帳號、密碼移交不確實的問題。

針對這些目的，本研究所提出的解決方案分別是：

1、利用網頁式單一登入介面（Web-Based Single Sign-On），讓使用者只需要成功認證一次身分，即可使用網路服務。
2、透過代理登入（Agents Sign）的方式來替使用者進行登入動作，且仍可以保有原服務網站的認證機制，而不需要進行登入介面的修改。
3、分析各個以角色為基礎的存取控制機制之優劣，以傳統 RBAC 為基礎，整合了 DRBAC 和 TT-RBAC 的優點，作為內建的權限控制機制。

除此之外，本研究根據本服務架構實作一套範例系統，來說明本服務架構的可行性，並證明這個服務系統的確能解決以上所提的問題。

最後，我們將本服務系統與第三章各學者所提的跨網域單一登入系統做比較，並將比較結果列於表 6.1：

表 6.1 本研究與各學者所提之跨網域單一登入系統比較表

	我們的系統	朱建達 (2000)	李長庚 (2002)	王雅苓 (2005)
是否需要額外建置驗證用伺服器	否	是	是	是
跨網域服務使用之技術	Agents Sign	Agent (附加在 Client)	SSOE	Cookie Center
實作的複雜度	低	高	高	高
是否能整合非授權管理之系統	是	否	否	否
每個網站系統是否保有自己的驗證機制	是	否	否	否
權限控制機制	整合 Group 與 Team 的 RBAC	RBAC	RBAC	無

相較於其他三位學者所提的跨網域單一登入服務架構，我們的系統是利用 Session 與 Secure Cookie 來執行身分認證，因此不需要額外建立驗證伺服器，以節省經費的支出。另外本系統透過代理登入（Agents Sign）的方式來達到跨網域單一登入的功能，與朱建達所提之 Agent 方式有異曲同工之妙，差別在於我們的 Agent 是附加在伺服器上，而不需要使用者下載更新套件，以免造成使用者困擾；除此之外，透過代理登入（Agents Sign）的方式，也能整合非授權的外部服務網站，而且不須修改該網站認證機制，亦能達到跨網域單一登入的目的。最後，我們以傳統 RBAC 為基礎，整合群組（Group）與團隊（Team）的概念，作為本系統的權限管控機制，藉此提供更便利且有彈性的管理。

在本服務架構當中，雖然使用者的登入密碼與代理登入網站的帳號、密碼都經過加密處理，但並未考量更進一步的網路安全問題。主要是因為一個服務網站在網路安全方面的考量，應該是在實作單一登入服務前，規劃網站架構與資訊基礎架構時就必須完成。因此我們也建議系統建置者，將本系統建置在學校內部網域中，限定只有校內使用者才能使用，以強化系統的安全性。

6.2 未來工作

本研究所提出之機制依然有未盡完美之處，對於未來工作方面，可以朝向三個方向來進行。

1、 RBAC 進階功能之實作：

雖然本系統的存取控制是以傳統的 RBAC 為基礎，但只實作了基本的授權與權限管理，讓具有合法角色的使用者均能夠使用服務；但在進階功能上，如權責的區分、角色的代理、繼承等，可以適用在複雜的環境之中，這些功能也值得深入研究。

2、 資料庫存取效能之改善：

由於本系統是在不變更現有系統環境下來設計，因此所有的系統資料都儲存於 MySQL 資料庫中，不過隨著資料量越大則存取速度將會有所影響；未來則可藉由輕量級目錄存取協定（Lightweight Directory Access Protocol,LDAP）快速查詢資料的特性，來解決資料庫效能隨著資料量增加而逐漸降低的困境。

3、 代理登入程式之自動化：

目前我們所設計的代登入程式是依據桃園縣國小行政人員經常登入的網站服務，以及教師個人經常使用的網路服務，經由人工分析先行設計好，以替使用者進行代理登入的程序。日後若可加入資料探勘（Data Mining）與自動分析欄位方法，來自動建立代登入程式，對程式系統的便利性一定有很大幫助。

參考文獻

[1] Ajay Lodha, Ram Sarma, "A Single Sign-On Approach", Avenue a razorfish, Inc., March 2006.

[2] R. Chandramouli, D. Ferraiolo, S. Gavrila, R. Kuhn, and R. Sandhu, "Proposed NIST standard for role-based access control, " ACM Transactions on Information and Systems Security, Vol. 4, No. 3, pp. 224-274, August 2001.

[3] Samar, V., "Single sign-on using cookies for Web applications ", Proceedings of the IEEE 8th International Workshops on Enabling Technologies:Infrastructure for Collaborative Enterprises , pp.158-163, 1999.

[4] D. Kristol, "HTTP Cookies: Standards, privacy, and politics",ACM Transactions on Internet Technology,Vol. 1, No. 2,pp. 151-198, 2001.

[5] T. Nykänen, "Secure Cross-Platform Single Sign-On Solution for the World-wide Web", Department of Computer Science and Engineering, Helsinki University of Technology, 2002.

[6] D. Ferraiolo ,R. Sandhu and R. Kuhn, "Proposed NIST Standard for Role-Based Access Control", ACM Transactions on Information and System Security, Vol. 4, No. 3, August 2001, Pages 224 - 274.

[7] Kurt Geihs et al., "Single Sign-On in Service-Oriented Computing", Springer Berlin, Heidelberg, 2003.

[8] Oppliger, R. , Microsoft .Net Passport: a security analysis, IEEE Computer Society, pp.29-35, 2003.

[9] PARK, J. S., SANDHU, R. , "Secure Cookies on the Web " , IEEE Internet computing, pp.36-44 , July-Aug. 2000.

[10] Liu, A.X. et al., "A secure cookie protocol", Proceedings of the IEEE 14th International Conference on Computer Communications and Networks, pp.333-338, 2005.

[11] R. Sandhu, D. Ferraiolo and R. Kuhn, "The NIST Model for Role-Based Access Control：Towards a Unified Standard", Proceedings of the 5th ACM Workshop on Role-Based Access Control, pp. 26-27, 2000.

[12] Park, J.S., Sandhu, R., and Ahn, G.J., "Role-Based Access Control on the Web", ACM Transactions on Information and System Sucurity, Vol. 4, No. 1, pp. 37-71,

[13] Q. Li, J. Shi and S. Qing, "An Administration Model of DRBAC on the Web," Proceedings of IEEE International Conference on e-Business Engineering, pp.364-367, 2005.

[14] Zhou, Wei and Meinel, Christoph, "Team and Task Based RBAC Access Control Model ", In Proceeding of Network Operations and Management Symposium, , Brazil, Sept. 2007 pp.84-94

[15] 陳軒正，結合IC卡之校園安全網頁系統的設計與實現，國立高雄師範大學資訊教育研究所碩士論文，2004。

[16] 朱建達，"建立於公開金鑰基礎建設的單一簽入系統"，國立交通大學資訊科學研究所碩士論文，2001

[17] 李長庚，"一個開放的Web-Based Single Sign-On服務架構"，國立交通大學資訊管理所碩士論文，2002。

[18] 王雅苓，"一個新的跨網域單一登入服務架構"，國立交通大學資訊管理研究所碩士論文，2004。

[19] 廖英彥，"網際網路單一簽入系統應用"，世新大學資訊管理學系碩士論文，2005 。

[20] 林孟勳，"結合RBAC授權之網站單一簽入機制研究"，世新大學資訊管理學系碩士論文，2006 。

[21] 王嘉宏，"植基於目錄服務的使用者管理系統"，國立交通大學電機資訊學院碩士論文，2004。

[22] Single Sign-On(SSO), http://andrewpatrick.ca/biometrics/singlesignon/singlesignon.shtml

[23] Role-Based Access Control,http://csrc.nist.gov/rbac

[24] Web Single Sign On Systems,http://www.cesnet.cz/doc/techzpravy/2006/web-sso/

[25] Cross-domain single sign-on solution, http://publib.boulder.ibm.com/tividd/td/ITAME/SC32-1134-01/en_US/HTML/amweb41_admin09.htm

[26] J. Gettys,J.,Mogul,H.,et al,"Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, Jun. 1999.

[27] PERSISTENT CLIENT STATE-HTTP COOKIES, http://wp.netscape.com/newsref/std/cookie_spec.html

[28] SSL 3.0 SPECIFICATION, http://wp.netscape.com/eng/ssl3/

[29] Windows Live CD,http://zh.wikipedia.org/

[30] 張源峰，Single Sign-On (單一簽入)技術介紹，
http://dbmaker.syscom.com.tw/mag/100/research_11.htm，2005
[31] 財團法人資訊工業策進會，單一簽入規劃報告，
http://www.gsp.gov.tw/downloadfile/guidebook/RFP0037.pdf ，2003 。
[32] PHP5網管實驗室，http://www.php5.idv.tw/
[33] 陳建勳，Apache 2 Server 徹底研究，博碩文化股份有限公司，台北，2002。
[34] 陳會安，[新觀念]資料庫系統 理論與設計實務，學貫行銷股份有限公司，台北，2006。