東海大學

資訊工程研究所

碩士論文

指導教授：林祝興博士

結合屬性加密與角色基存取控制之雲端跨組織合作系統

A Cloud System for Cross-organization Cooperation Using Role-based Access Control and Attribute-based Encryption

研究生：齊軒

中華民國一零八年七月

東海大學碩士學位論文考試審定書

東海大學資訊工程學系 研究所

研究生 齊 軒 所提之論文

結合屬性加密與角色基存取控制之雲端跨組織合作系統

經本委員會審查，符合碩士學位論文標準。

學位考試委員會

召 集 人 ＿＿＿＿＿＿ 簽章

委 員 ＿＿＿＿＿＿

＿＿＿＿＿＿

＿＿＿＿＿＿

指 導 教 授 ＿＿＿＿＿＿ 簽章

中華民國 108 年 7 月 5 日

摘要

現今的各類產業分工細密，許多公司都有專職的技術或能力，尤其許多的中小企業更是經常依靠少數的專精能力在市場上生存。隨著產品的越趨複雜化，跨企業組織之間的合作也越加頻繁。因此，本論文將研究的目標設定在跨企業組織的整合上，提出一套兼具便利、簡潔與安全的雲端跨組織合作系統。通過角色基存取控制來分配與控制參與者的權限，並結合屬性加密來確保細粒度的資料安全性，以此為基礎設計一套系統架構並將之實作。

關鍵詞：屬性加密、角色基存取控制、雲端資料服務、跨組織合作。

Abstract

Currently various industries have a fine division of labor, different companies have their own specialized skills or capabilities. Many small or medium-sized enterprises often rely on a few specialized capabilities to survive in the market. As products become more and more complex, collaboration between organizations is becoming more frequent. Therefore, in this thesis, we set the goal on the integration of cross-organizations, and proposes a cloud-based cross-organization cooperation system that is convenient, concise and secure. We assign and control participant permissions through role-based access control, combined with attribute-based encryption to ensure fine-grained data security. Based on these, we designed and proposed a system architecture and implement it.

Keywords: attribute-based encryption, role-based access control, cloud data server, cross-organizational cooperation

目錄

圖目錄

表目錄

第一章　簡介

1.1 研究動機

隨著全球化時代的來臨，企業之間的互動與合作越趨頻繁，不少企業擁有單一的強大專業，然而卻不見得能獨自生產一個完整、且具備高實用性的產品，許多人們習以為常的產品都是多家企業合作的成果。然而，企業在合作時很容易就會遇到一些資訊安全的問題，譬如，資料應該存放在哪裡？哪些人可以看到？如何限制存取權限？

跨組織之間的合作首先會遇到的問題便是資料的存放問題，不同的企業或組織，尤其是以特定技術見長的企業，或多或少可能都會有不願被他人知曉的技術細節。這些資訊一般存放於公司內部的系統當中，合作開發時的資料若放在某一方的伺服器中，那麼存放方會需要對其他組織的參與人員開放伺服器的存取權限，這會造成資安上的隱患，有被對方從內部系統進行惡意攻擊的可能，因此，建立一套獨立的系統是一個較好的解決方案。然而，架設一套伺服器系統需要一筆硬體成本，對於大型企業來說不成問題，但對於規模較小的組織來說可能是一個負擔，尤其當合作結束後可能會變成閒置資源。這時，向第三方租借雲端伺服器就成了一個可靠的方案，可依據實際使用狀況彈性增減需求資源，且在於成本分攤上也較好分配。

要在雲端上建立伺服器首先要面對的便是雲端安全問題，早在 10 年前，相關的研究議題便已被陸續提出[1][2]，而本論文的研究方向就將著重在雲端安全的這個議題之上，旨在提出一套可為組織合作時所用，兼具安全性與實用性的雲端伺服器系統。

1.2 研究目的

本論文將提出一套便於跨企業跨組織(cross organizations)之合作使用，具備高安全性與高便利性的雲端存取控制系統，以角色基存取控制(role-based access control, RBAC)做為主要的存取控制系統，並結合屬性加密(attribute-based encryption)來提升其細粒度(fine granularity)的資料安全性。

由於本論文欲提出的系統是要提供企業組織使用，因此需要一套存取控制系統來區別不同用戶的權限。常見的存取控制機制總共有三種：任意存取控制(discretionary access control, DAC)、強制存取控制(mandatory access control, MAC)、角色基存取控制(role-based access control, RBAC)[3]，其中，最適合做為組織管理系統的當屬角色基存取控制。任意存取控制讓資料擁有者來決定哪些用戶可以存取資料，甚至可以把這個對資料的管理權限委託給其他用戶，這並不適合用在企業組織上。在企業組織中的資料，所有者應為企業組織本身，並非建立資料的用戶，任意存取控制提供給用戶的權限過大，會產生許多的弊端與資安風險；強制存取控制則由中央管理系統全權決定資料的存取規則，以及用戶的存取權限，此種作法雖然能提供較高的安全性，但卻需要龐大的管理成本，每當有人事異動或資料的新增、更新、權限調整等，管理人員都需花費大量時間去逐一進行調整與設定，這對於一個專案的開發流程會造成阻力。

角色基存取控制通過創造多個角色(roles)，並將權限(permission or privileges)指派給角色，再將角色授予給使用者(users)的存取控制模式(access control model)。先針對各種需求設定好角色的權限，再根據職位需求將角色授予給使用者，這其中的角色有點類似於打包的概念，預先設想各種情況，將各種情況會需要使用的權限先行打包，再把一包一包的權限分派給使用者。如此可節省大量重複的授權作業，提升權限管理的效率，因此，本論文將角色基存取控制用來作為核心的存取控制系統，提升管理的便利性。

與傳統的硬體環境不同，雲端伺服器的硬體由不可信任的第三方提供，存取控制主要是用作企業組織內部的系統所使用，具有防止外部入侵的能力，卻缺乏

抵禦從內部竊取資料的能力，這部份的議題已經被探討過[4][5]。本文將採用屬性加密來增強系統的安全性，避免來自伺服器供應商的潛在威脅，所有檔案在上傳至雲端伺服器前都會被進行加密。

屬性加密是一種公開金鑰加密法(public key ciphering)，所有使用者都會具備各自的屬性(attributes)，並依據各自所具備的屬性來產生私鑰(private key)。在利用公鑰(public key)加密時，加密者可以設定存取結構(access structure)，只有具備符合存取結構要求屬性的使用者可以解密。此種加密法通過屬性的設置實現了一把公鑰對應多把私鑰的加密形式，很適合應用在公司系統的檔案文件加密上，其概念與角色基存取控制有些許雷同之處，很適合與角色基存取控制進行搭配，本論文決定將其應用至系統中，增加在雲端環境上的資料安全性。

第二章　背景知識與相關文獻

2.1 角色基存取控制

角色基存取控制最早於 1992 年由 David F. Ferraiolo 與 D. Richard Kuhn 提出概念[6]，後由 Ravi Sandhu 等人在其基礎上進行改良並提出角色基存取控制模型[7]，之後由美國國家標準局(NIST)重新定義角色基存取控制模型(RBAC model)[8]，在 2004 年歸類採用為一種標準(ANSI/INCITS)，並在 2012 年修訂版本為 INCITS 359-2012。

角色基存取控制是一套以虛擬的角色為媒介來授予使用者權限的存取控制系統。相較於直接授予權限給使用者，角色基存取控制在權限的管理上更加便利，權限需要更改時僅需針對角色進行修改，無須一個個更改使用者的權限。當參與者的職位更動時也僅需重新授予角色即可，可以大幅簡化權限的管理流程。

2.1.1 角色基存取控制模型

角色基存取控制中包含了使用者(user)、角色(role)、權限(permission)、會期(session) 共 4 個主要元素，其交互關係如圖 2.1 所示。

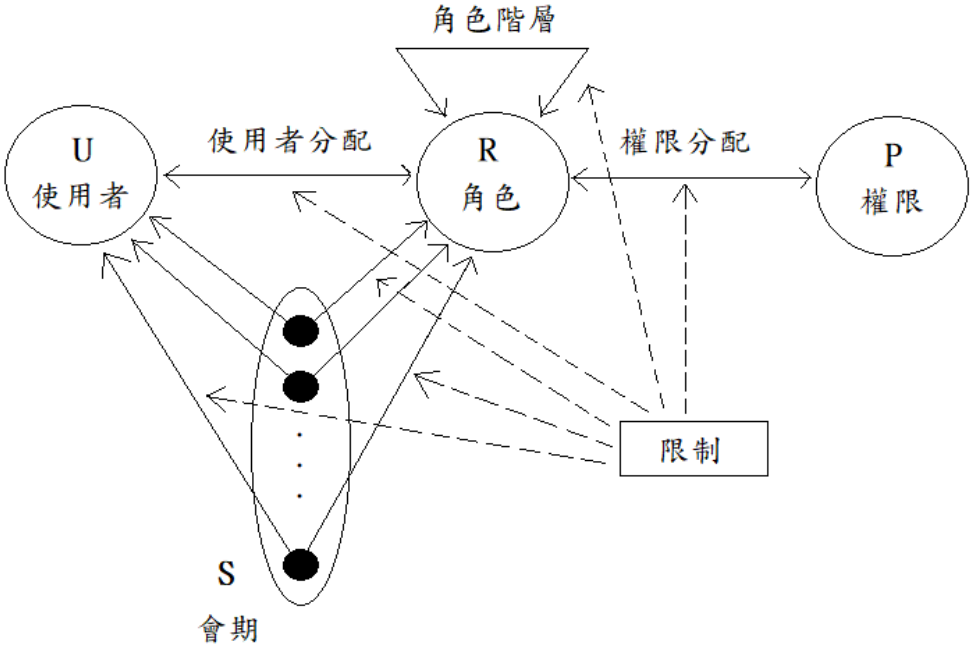


圖 2.1 角色基存取控制模型

I. 使用者 U(user)：一般來說是一個真實的人物，但也可以是一個智慧型的自動化系統，例如機器人、一台電腦，或甚至是一個網路，為求簡便，在本論文中單指參與計畫的人。

II. 角色 R(role)：一個虛擬的單位，角色通常是組織內的工作職能或職稱。

III. 權限 P（permission）：對物件進行存取或操作的許可權。

IV. 會期 S（session）：使用者對應到角色的過程，一個使用者可對應多個角色，一個角色也能對應給多個使用者，但是必須符合某些限制。

角色基存取控制模型除了 4 個主要元素之外還有使用者分配以及權限分配兩種交互關係。

i. 使用者分配(user assignment)：將角色授予給使用者，使用與角色之間是多對多的關係，一個使用者可被授予多個角色，而一個角色也可被授予給多個使用者，但必須符合某些限制。

ii. 權限分配(permission assignment)：將權限分配給角色，角色與權限之間是多對多的關係，一個角色可被授予多個權限，一個權限也能被授予給多個

使用者，但必須符合某些限制。

2.1.2 角色階層

有時角色與角色之間會有階層關係(hierarchical relation)，一般來說較為低階的角色會放在階層下面，而較為高階的會放在上面，高階的角色會繼承所有在其之下的低階角色的權限，通過角色繼承可以簡化權限的授權流程。如圖 2.2 所示，測試工程師必然會包含專案成員的身分，而專案總監則會具備所有小組成員的權限。

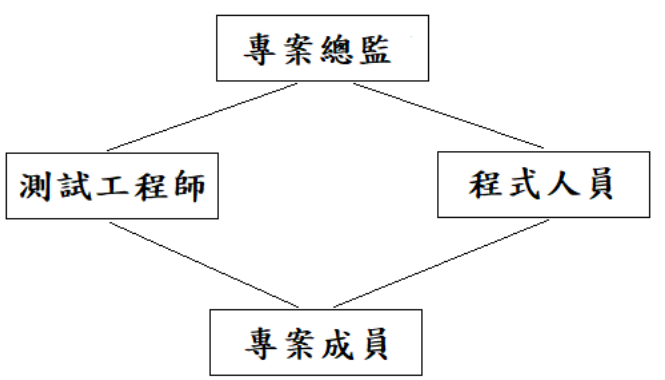


圖 2.2 角色階層

2.1.3 限制

為使計畫能更穩定的運作，並避免不必要的風險，我們通常會在權限與角色的授予上設定一些限制，這些限制通常分為三大類，互斥角色、數量限制以及前提條件。

I. 互斥角色(mutually exclusive roles)：某些角色之間的功能是互斥的，不能同時授予給同一個使用者，否則可能會產生一些權限上的弊端。例如：採購人員跟核銷人員這兩個角色不能授予給同一個人，否則可能會有舞弊的現象。

II. 使用者數量(cardinality constraint)：限制角色的授予數量，部分角色在同一時間內被授予的數量不宜太多，避免被不當利用。以專案總監為例，無論任何時間，能被授予的人數都只能有一個。

III. 前提條件(prerequisite constraint)：使用者須先滿足某些特定條件才能被授予角色，例如：必須年資滿 3 年才能被授予組長的角色。

2.2 屬性加密

屬性加密(attribute-based encryption)的概念最早由 Amit Sahai 與 Brent Waters 提出[9]，屬性加密是一套公開金鑰加密系統，與傳統公開金鑰相同需先通過公鑰加密，再利用使用者的私鑰來解密。但不同之處在於，早期的公開金鑰加密系統是採一對一的形式加密，一把公鑰對應一把私鑰，加密者必須先清楚對方的身分，再拿對方的公鑰來加密資料。而屬性加密再加入了屬性的概念後，每個使用者都具備各自的屬性(attribute)，通過這些屬性來產生私鑰，資料在加密時除了公鑰之外還須設定存取結構(access structure)，存取結構一般是一個由 AND 與 OR 構成的樹狀結構(如圖 2.3)，只有具備符合存取結構條件的屬性的使用者，可以使用其私鑰來解鎖密文，實現了只需一把公鑰就能對應多個使用者的一對多加密形式。

隨著後面多位研究者的改良，屬性加密又分為了，金鑰策略屬性加密(key-policy attribute encryption, KP-ABE)[10][11] 以及密文策略屬性加密(ciphertext-policy attribute encryption, CP-ABE)[12][13][14]。金鑰策略屬性加密將存取結構存放在使用者的私鑰上，密文的屬性符合使用者的存取結構時使用者才能解密密文。而密文策略屬性加密則將存取結構存放在密文上，符合存取結構要求的使用者才能解密密文。本文後續將採用的是密文策略屬性加密，因此先介紹之。

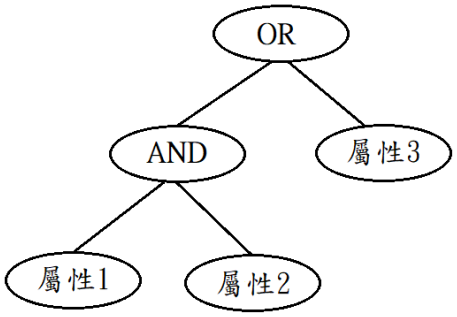


圖 2.3 存取結構範例

2.2.1 密文策略屬性加密

密文策略屬性加密共有 4 個階段的演算法，分別為 Setup、Encrypt、KeyGen、Decrypt。下面將會在密文策略屬性加密的基礎上針對這些步驟進行介紹，所有公式均參考自文獻[12]。

I. Setup：產生公鑰(PK)以及主密鑰(MK)。先選擇一個具備生成集合 g 的素數階 p 的雙線性群 G_0 接著選擇兩個隨機整數 $\alpha,\beta \in Z_p$，產生如下的公鑰跟主密鑰：

$$PK = (G_0, g, h = g^{\beta}, f = g^{1/\beta}, e(g,g)^{\alpha}) \quad (1)$$

$$MK = (\beta, g^{\alpha}) \quad (2)$$

II. Encrypt：透過公鑰(PK)和存取結構(T)將資料(M)加密。首先為樹 T 中的每個節點 χ 選擇多項式 q_{χ}，從根節點 R 開始，由上至下選擇這些多項式。對於樹中的每個節點 χ，將多項式 q_{χ} 的維度 d_{χ} 設置為比該節點的閾值 k_{χ} 小 1，即 $d_{\chi} = k_{\chi}-1$。從根節點 R 開始，隨機選擇整數 s 並設定 $q_R(0) = s$。每個節點 χ 都須符合 $q_R(0) = q_{parent(\chi)}(index(\chi))$。設 Y 是 T 中的葉節點集，最終可得初如下的密文(CT)：

$$CT = (T, \tilde{C} = M \cdot e(g,g)^{\alpha s}, C = h^{s}, \forall y \in Y : C_y = g^{q_y(0)},$$

$$C'_y = H\big(att(y)\big)^{q_y(0)}) \quad (3)$$

III. KeyGen：利用主密鑰(MK)以及使用者的屬性(S)來產生各自的私鑰(SK)。隨機選擇一整數 r，對每一屬性 j∈S 隨機選擇一整數 r_j 並計算出密鑰：

$$SK = (D = g^{(\alpha+r)/\beta}, \forall j \in S : D_j = g^r \cdot H(j)^{r_j}, D_j' = g^{r_j}) \qquad (4)$$

IV. Decrypt：利用私鑰(SK)將被加密的檔案(CT)解密。先定義一個遞歸算法 DecryptNode（CT，SK，x），它將密文 CT 作為輸入，私鑰 SK 與之相關聯 具有一組屬性 S 和一個來自 T 的節點 x。如果節點 x 是葉節點，那麼我們讓 i = att（x）並定義：

$$If\ i \in S,\ DecryptNode(CT, SK, x) = \frac{e(D_i,\ C_x)}{e(D_I',\ C_x')}$$

$$= \frac{e(g^r \cdot H(i)^{r_i}, h^{q_x(0)})}{e(g^{r_i}, H(i)^{q_x(0)})}$$

$$= e(g,g)^{rq_x(0)} \qquad (5)$$

如果節點 x 不是葉節點，對於作為 x 的子節點的所有節點 z，計算 DecryptNode（CT，SK，z）並將輸出存儲為 F_z，則 F_x 將會如下：

$$F_x = \prod_{z \in S_x} F_z^{\Delta_{i,S_x'}(0)}, \quad where \quad \begin{matrix} i = index(z) \\ S_x' = \{index(z) : z \in S_x\} \end{matrix}$$

$$= \prod_{z \in S_x} \left(e(g,g)^{r \cdot q_z(0)}\right)^{\Delta_{i,S_x'}(0)}$$

$$= \prod_{z \in S_x} \left(e(g,g)^{r \cdot q_{parent(z)}(index(z))}\right)^{\Delta_{i,S_x'}(0)}$$

$$= \prod_{z \in S_x} e(g,g)^{r \cdot q_x(i) \cdot \Delta_{i,S_x'}(0)}$$

$$= e(g,g)^{r \cdot q_x(0)} \qquad (6)$$

綜合上述步驟我們可以將密文策略的屬性加密簡化為圖 2.4 的示意圖，使大多數人更容易了解密文策略屬性加密。

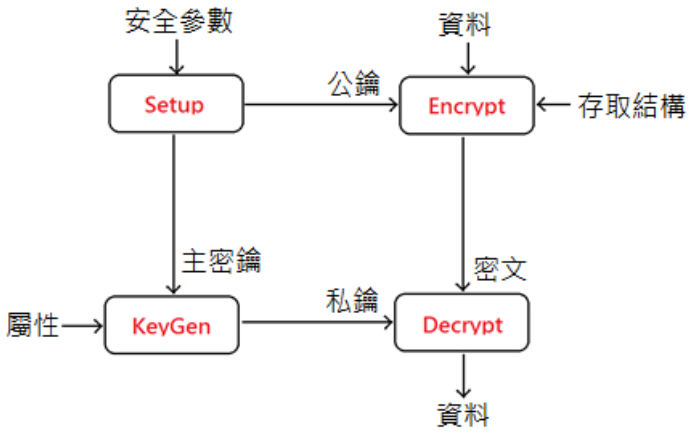


圖 2.4 密文策略屬性加密運作流程圖

2.2.2 金鑰策略屬性加密

金鑰策略屬性加密與密文策略相同，有 Setup、Encrypt、KeyGen、Decrypt 四個步驟，下面逐一介紹各個步驟的處理公式，所有公式均參考自文獻[10]。

I. Setup：產生公鑰(PK)以及主密鑰(MK)。定義一個屬性宇集 U = {1, 2,…, n}，替每個屬性 i ∈ U 隨機挑選一整數 t_i，最後隨機挑選一整數 y，產生如下公鑰與主密鑰：

$$PK = (T_1 = g^{t_1}, \dots, T_{|U|} = g^{t_{|U|}}, Y = e(g,g)^y) \quad (7)$$

$$MK = (t_1, \dots, t_{|U|}, y) \quad (8)$$

II. Encrypt：通過公鑰(PK)和屬性(γ)將資料(M)加密。選擇一隨機整數 s，產生密文(CT)如下：

$$CT = (\gamma, CT' = MY^s, \{CT_i = T_i^s\}_{i\in\gamma}) \quad (9)$$

III. KeyGen：利用主密鑰(MK)以及使用者的存取結構(T)來產生私鑰(SK)。替存取結構中的每個節點 x 設置一個多項式 q_x，根節點為 r。每個節點 x 的多項式都有一個維度 $d_x = k_x - 1$。對於根節點 r，設置 $q_r(0) = y$；對於其他節點 x，設置 $q_x(0) = q_{parent(x)}(index(x))$。多項式設定好後，對於每個葉節點 x 可產生如下密鑰：

$$SK_x = g^{\frac{q_x(0)}{t_i}} \text{ where } i = att(x) \quad (10)$$

各個節點的密鑰集合即是使用者的私鑰(SK)。

IV. Decrypt：利用私鑰(SK)將被加密的檔案(CT)解密。定義遞迴演算法 DecryptNode(CT, SK, x)，令 i = att(x)，則對於每個葉節點：

$$DecryptNode(E,D,x) = \begin{cases} e(D_x, E_i) = e\left(g^{\frac{q_x(0)}{t_i}}, g^{s\cdot t_i}\right) \\ = e(g,g)^{s\cdot q_x(0)} \quad if \;\; i \;\in\; \gamma \\ \\ \perp \; otherwise \end{cases} \tag{11}$$

非葉節點的部份，對於 x 的所有子節點 z，調用 DecryptNode(CT, SK, z) 並將輸出存儲為 F_z。設 S_x 是任意 k_x 大小的子節點 z 集合，使得 $F_z \neq \perp$。如果不存在這樣的集合，則節點不滿足並且函數返回⊥。否則，可以計算如下：

$$\begin{aligned} F_x &= \prod_{z\in S_x} F_z^{\Delta_{i,S'_x(0)}}, where \quad \begin{matrix} i = index(z) \\ S'_x = \{index(z) : z \in S_x\} \end{matrix} \\ &= \prod_{z\in S_x} \left(e(g,g)^{s\cdot q_z(0)}\right)^{\Delta_{i,S'_x(0)}} \\ &= \prod_{z\in S_x} \left(e(g,g)^{s\cdot q_{parent(z)}(index(z))}\right)^{\Delta_{i,S'_x(0)}} \quad \text{(by constr.)} \\ &= \prod_{z\in S_x} e(g,g)^{s\cdot q_x(i)\cdot \Delta_{i,S'_x(0)}} \\ &= e(g,g)^{s\cdot q_x(0)} \quad \text{(using polynomial interpolation)} \end{aligned} \tag{12}$$

綜合上述步驟我們可以將金鑰策略的屬性加密簡化為圖 2.5 的示意圖，使大多數人更容易了解金鑰策略屬性加密。

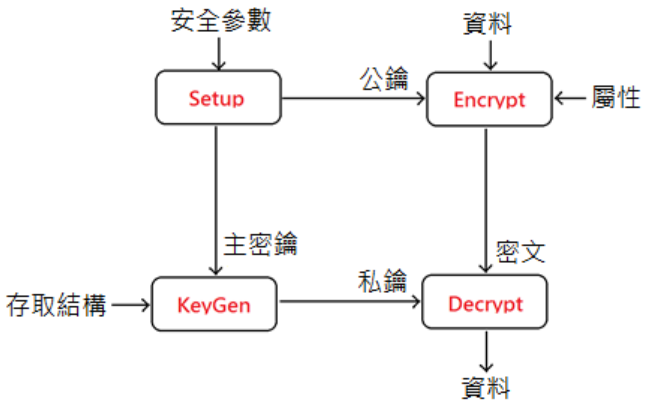


圖 2.5 金鑰策略屬性加密運作流程圖

第三章　研究方法

3.1　設計理念與文獻參考

簡其弘先生提出了一個跨組織合作系統(cross organizations cooperation)的雛型[15]，概念上類似於建立一套獨立於合作組織之間的存取控制系統，本文深受啟發。然而，該篇文章所著重的點在於工作流程的整合，本篇文章參考了其將角色基存取控制用作跨組織整合的概念，並針對系統的雲端化設計資訊安全方案。

Yong Wang 等人提出了一套新的雲端存取控制方案[16]，將角色基存取控制與屬性加密的概念結合，角色基存取控制系統的所有授權流程改以屬性驗證來取代。每個角色都有一個存取結構，每個用戶都有一組初始屬性，用戶的屬性滿足角色的存取結構即可以獲得該角色的屬性；通過使用者的初始屬性以及所獲得的角色屬性產生用戶的私鑰，所有文件均會先進行過屬性加密才上傳至伺服器，用戶的私鑰滿足密文的存取結構才能解密密文。這套系統將角色基存取控制與屬性加密融合，並精簡了系統的流程，在效率上的表現很優異，然而對於企業組織來說，卻不便於管理。原本的角色基存取控制要將角色授予給使用者，只需管理員一個步驟即可完成，但在上述系統中卻需針對欲授予給使用者哪些角色來調整使用者的初始屬性。因此本論文並不打算採用上述方案，然而，這篇文章在實作過程中提到了一個概念，設置兩個伺服器，屬性管理伺服器與金鑰管理伺服器，屬性管理伺服器數據儲存、加密、解密、角色和用戶的管理，金鑰管理伺服器則負責產生公鑰與主密鑰以及替用戶及角色生成相關密鑰，通過兩個伺服器分權分工的方式來提升資料的安全度，本文參考了此做法並加以改良。

Jing-Jang Hwang 等人於 2011 年便已提出了將加解密的作業獨立出伺服器的概念[17]，避免雲端伺服器以任何形式接觸到資料明文。本文決定將此概念套用至系統中。

參考了上述資料的一些作法與內容，本論文設計了一套專門針對雲端跨組織合作使用的存取控制系統，我們在雲端伺服器上建立角色基存取控制系統，作為中樞系統，用戶資訊、資料密文均儲存在此。在所有資料上傳至雲端伺服器前均需通過屬性加密來加密資料，而加解密金鑰則由另一個金鑰管理授權伺服器來負責處理，包含金鑰的生成與授予。最後，關於資料的加解密作業則都在用戶的電腦上執行，避免兩個伺服器以任何形式接觸到資料明文，盡可能減少資料明文被竊取的可能。

而關於在組織合作時，不同組織內部的職位權限不同，在合作時該如何指派權限的問題。本文中的做法是針對合作專案重新定義一份角色職位的權限表，所有參與專案的人員在系統中會被重新授予一套新的角色，權限則依新的職位權限表來給予。若在實務中有需求也可參考文獻[15]中整合工作流程的做法。

屬性加密的部分，本文採用的是密文策略屬性加密。由於用戶被授予的角色可能不只一個，且被授予的角色隨時都有可能變動，因此難以在用戶身上建立存取結構，故不採用金鑰策略。

3.2 系統設計

圖 3.1 為本論文所提系統的模型。分別建置雲端伺服器以及金鑰管理授權伺服器，雲端伺服器是本系統的中樞，角色基存取控制系統建立在此伺服器中，負責角色建立、角色權限指派以及所有參與人員的角色授予，同時，所有加密後的資料也都存放在此。金鑰管理授權伺服器顧名思義負責金鑰的授權，每個雲端伺服器建立的角色均有其各自的屬性，使用者在被授予所需角色後會向金鑰管理授

權伺服器申請個人私鑰，使用者所具備之屬性根據其申請私鑰時所具備之角色而定。

每當一筆新的資料產生，上傳至雲端伺服器前上傳者都必須先將檔案進行一次屬性加密並設定其存取結構，加密完成後再上傳至雲端伺服器。每當一個新的使用者加入，雲端伺服器的管理者會根據其在計畫中的職位給予其相對應的角色。使用者登入雲端伺服器後僅可查找及下載符合其角色權限的資料，將資料密文從雲端伺服器下載後再利用被授予的私鑰來解密獲得資料明文。下面將詳細介紹系統的結構。

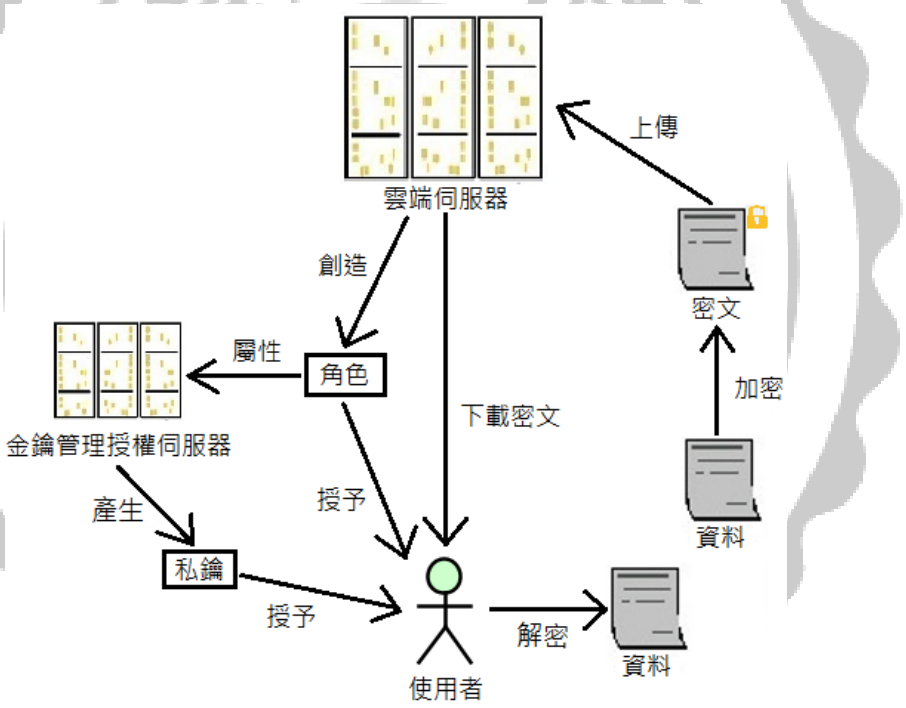


圖 3.1 本論文之系統模型

3.2.1 伺服器建置

分別建立兩台伺服器，一台作為雲端伺服器，另一台則作為金鑰管理授權伺服器。在雲端伺服器中架設角色基存取控制系統，雲端伺服器中包含了密文資料庫、使用者資料庫、角色資料庫以及權限資料庫，如圖 3.2 所示，在其中設置一個管理者帳戶，管理者帳戶具備新增、修改以及刪除使用者、角色、權限資料庫

的權限，無論是使用者帳戶的新增，還是角色權限的異動，均只能通過管理者帳戶來執行，但此管理者帳戶不具備存取密文資料庫以及連結金鑰管理授權伺服器的權限，避免管理者私自濫用權限。

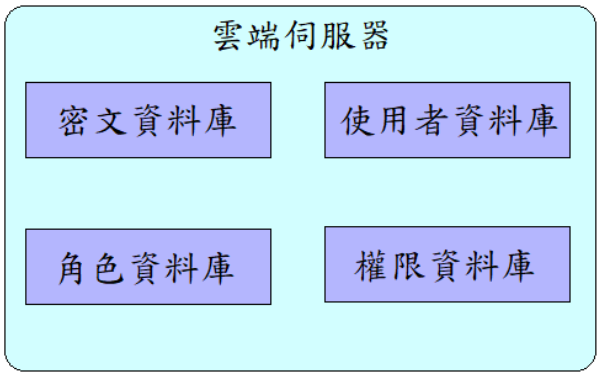


圖 3.2 雲端伺服器

接著是金鑰管理授權伺服器，同樣建立一個管理者帳戶，負責定義每個角色所具備的屬性。在系統開始運作時伺服器會產生一組公鑰跟主密鑰(0)，使用者向金鑰管理授權伺服器提出金鑰申請時會同時附帶使用者所具備的角色(1~2)，金鑰管理授權伺服器根據自身的資料庫對應出該使用者所具備的所有屬性並將這些屬性集合產生一個屬性集(3)，由此屬性集與主密鑰通(4)過第二章提及的 KeyGen 函式產生私鑰(5)並與公鑰一同傳回給使用者(6)。圖 3.3 為金鑰管理授權伺服器的運作流程圖。

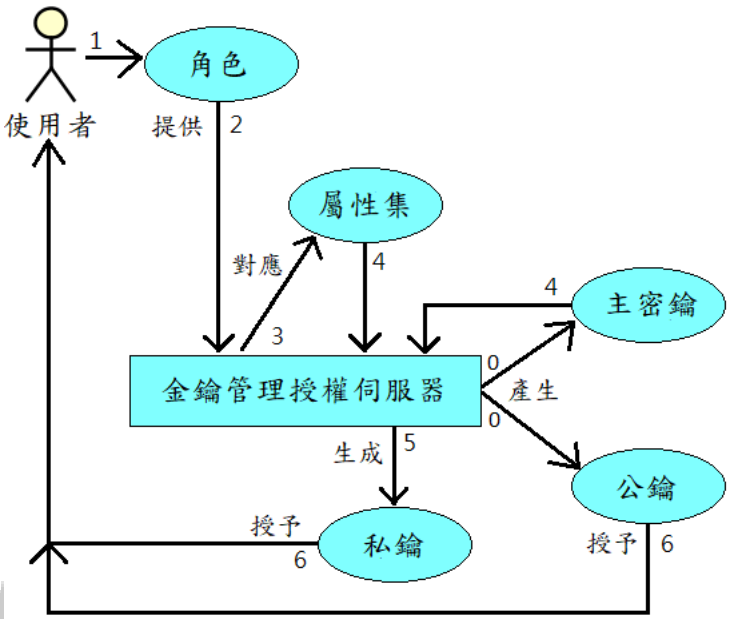


圖 3.3 金鑰管理授權伺服器

3.2.2 客戶端系統

上個段落介紹了伺服器的設計與功能，此段落將著重介紹客戶端的設計與功能。

客戶端系統主要包含三大功能：登入/登出、檔案查找與下載、檔案上傳。使用者進入系統後首先必須登入系統，當使用者登入系統後，系統將會自動連結至雲端伺服器並更新使用者的角色授權狀態與權限(系統可根據需求決定多久時間進行一次更新，故圖中以虛線圈出，並非每次登入都須執行一次)。接著，系統將自動向金鑰管理授權伺服器提出金鑰申請，並同時傳遞使用者所持角色身分至金鑰管理授權伺服器。金鑰管理授權伺服器會根據所持有角色的屬性來產生私鑰並將私鑰與公鑰傳送至客戶端系統。在使用者更新完角色與私鑰後即可通過雲端伺服器來查找資料。找尋到所需資料後即可提出下載申請，雲端伺服器在確認其角色權限後，若權限符合要求便會開始下載資料密文，下載完成後使用者可通過私鑰來解密密文以獲得明文。若使用者欲上傳資料，則需先設定資料的存取條件

與存取結構，接著利用公鑰與存取結構將資料進行屬性加密，再將加密完成的密文上傳至雲端伺服器。圖 3.4 為客戶端系的使用流程圖。

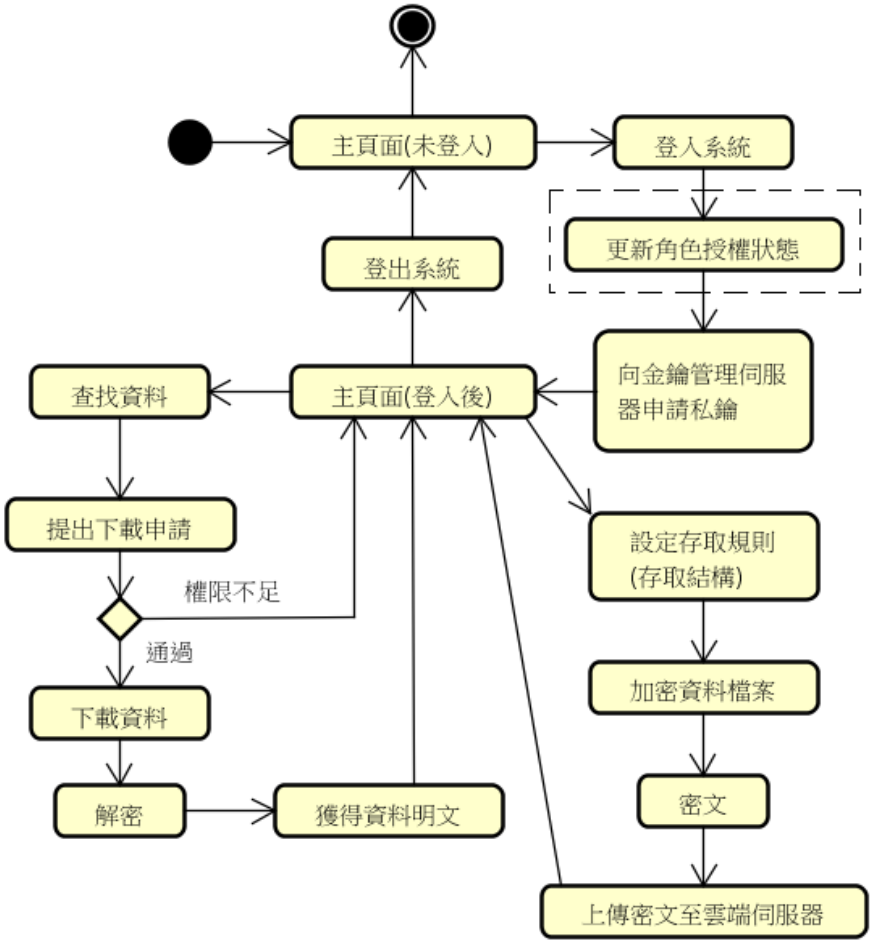


圖 3.4 客戶端使用流程圖

第四章　研究成果

本章將根據上一章所提之系統設計與架構，實際建置一套跨組織合作系統的雛型，下面將從系統的建置到實測依序介紹。

4.1 系統環境

本研究的開發環境如表 4.1 所示，本論文的目標是開發一套系統雛形，為求開發便利，本論文採用在單機環境上建置兩個虛擬伺服器的方式來取代兩台實際的伺服器。

表 4.1 開發環境

CPU	Intel Core I7-7700HQ
RAM	12GB
OS	Windows10

在本系統中，我們採用 MySQL 系統來建置我們資料庫，並使用 PHP 與 HTML 語言來編寫客戶端系統與介面，而屬性加密系統的部分，我們使用 JAVA 語言來撰寫，並引用了 Junwei Wang 於 Github 上所提供的 cpabe 函式庫來進行開發[18]。

4.2 資料庫建置

首先建立雲端伺服器的資料庫，在此伺服器中存在 6 個資料表，分別是用戶資料表、用戶-角色資料表、角色資料表、角色-權限資料表、權限資料表以及文件資

料表，圖 4.1 為雲端伺服器中的資料表關係圖，顯示了各資料表中包含的哪些數據以及資料表之間的相互對應關係。

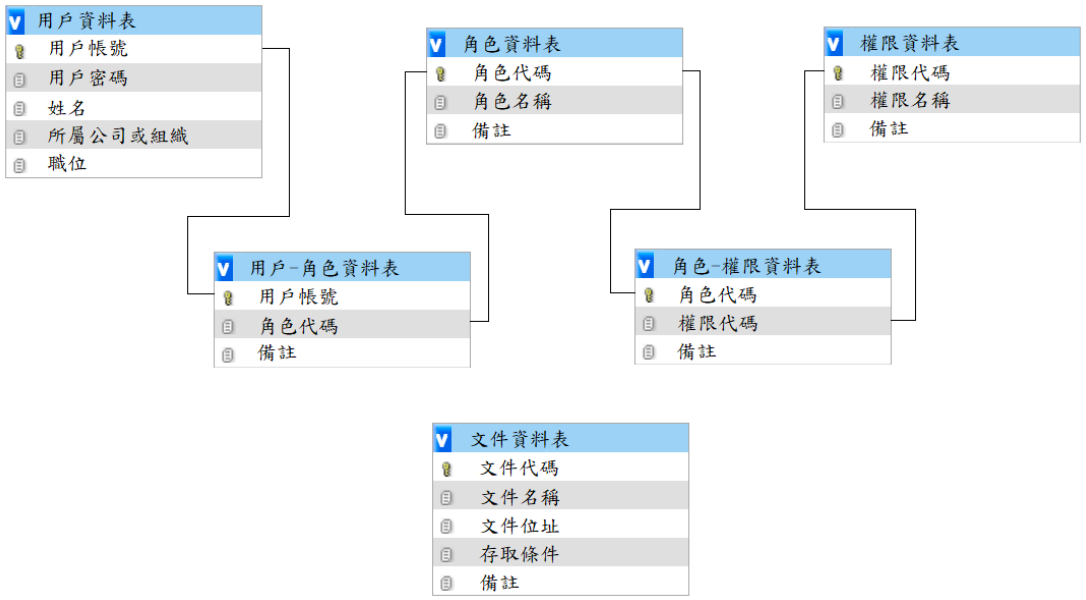


圖 4.1 雲端伺服器資料庫

用戶資料表即為我們的用戶資料庫，當中包含了用戶的帳號、密碼、姓名、所屬公司或組織以及在計畫中的職位等個人資訊；角色資料表即為角色資料庫，當中包含計畫中所有將會被授予的角色，資料表上有角色代碼、角色名稱、備註三個欄位；權限資料表為權限資料庫，包含計畫中所有將會被分派給角色的權限，資料表上有權限代碼、權限名稱、備註三個欄位。各資料表中的備註欄位用來記載特殊事項，方便管理員管理，在實際應用中可有可無。

用戶-角色資料表與角色-權限資表這兩個資料表較為特殊，是用來記載用戶與角色、角色與權限之間的對應關係，用戶-角色資料表中有用戶帳號、角色代碼、備註三個欄位，資料表中的用戶帳號與用戶資料表的用戶帳號相呼應，角色代碼則與角色資料庫中的角色代碼相呼應，用戶與角色之間是多對多的關係，一個用戶可對擁有多個角色，一個角色也能對應給多個用戶；角色-權限資料表中有角色代碼、權限代碼、備註三個欄位，資料表中的角色代碼與角色資料表的角色代碼相呼應，權限代碼則與權限資料表的權限代碼相呼應，角色與權限之間亦

是多對多的關係，一個角色可被授予多個權限，一個權限也能被對應給多個角色。每當人事或角色權限異動時，管理者僅須在這兩個資料表上進行更動即可完成角色與權限的重新分配。

文件資料表即為密文資料庫，每份檔案的基本資訊均集中於此，包含文件代碼、文件名稱、文件位址、存取條件、備註 5 個欄位，文件位址記錄文件在伺服器中的檔案位址，以便用戶下載時進行連結，存取條件則記錄下載此文件所需具備的權限。

接著介紹金鑰管理授權伺服器的資料庫，金鑰管理授權伺服器的主要功能為金鑰生成與授予，資料庫中僅有角色資料表、屬性資料表、角色-屬性資料表三個資料表，如圖 4.2 所示。

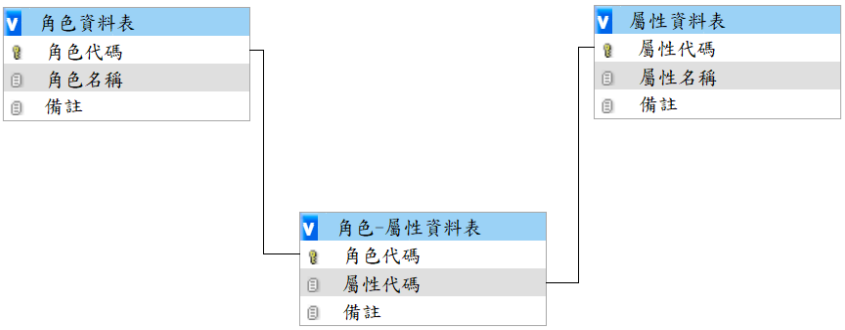


圖 4.2 金鑰管理授權資料庫

角色資料表同雲端伺服器上的一樣，具有角色代碼、角色名稱、備註三個欄位，主要用來對應雲端伺服器上的資料表；屬性資料表即為屬性資料庫，裡面包含所有會使用到的屬性，具有屬性代碼、屬性名稱、備註 3 個欄位。

角色-屬性資料表用來記載角色與屬性之間的對應關係，角色代碼對應到角色資料表中的相同欄位，屬性代碼對應到屬性資料表中的相同欄位。

4.3 角色、權限、屬性設計

本論文在參考了文獻[19]以及網路上提供的部分企業組織的職位分配結構圖後，依據實際軟體開發時公司可能需求的人力結構設計了一張職位結構圖，並以此為依據設計了一套角色系統以及權限系統。

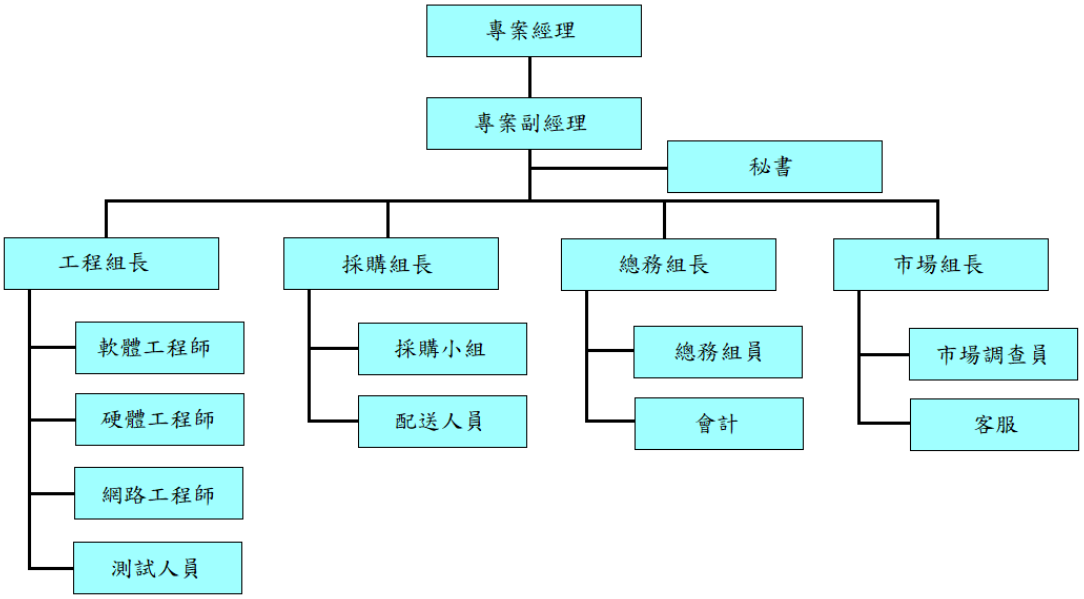


圖 4.3 角色結構圖

圖 4.3 為本論文系統的角色結構圖，系統中由上至下有專案經理、專案副經理、秘書、工程組長、採購組長、總務組長、市場組長、軟體工程師、硬體工程師、網路工程師、測試人員、採購小組、配送人員、總務組員、會計、市場調查員、客服 17 個角色，每個角色都有各自的權限，同時上級角色會繼承下級角色所具有的權限，表 4.2 為系統中各角色所具備的權限一覽表，本論文為系統設計了 17 種權限，分別為一級主管、二級主管、秘書、組長、工程部門、採購部門、總務部門、市場部門、軟體工程師、硬體工程師、網路工程師、測試員、配送員、會計師、市場調查員、客服人員、員工。

表 4.2 角色權限一覽表

角色	權限
專案經理	一級主管、二級主管、組長、工程部門、總務部門、市場部門、員工

專案副經理	二級主管、組長、工程部門、採購部門、員工
秘書	秘書、組長、員工
工程組長	組長、工程部門、軟體工程師、硬體工程師、網路工程師、測試員、員工
採購組長	組長、採購部門、配送員、員工
總務組長	組長、總務部門、會計師、員工
市場組長	組長、市場部門、市場調查員、客服、員工
軟體工程師	工程部門、軟體工程師、員工
硬體工程師	工程部門、硬體工程師、員工
網路工程師	工程部門、網路工程師、員工
測試人員	工程部門、測試員、員工
採購組員	採購部門、員工
配送人員	採購部門、配送員、員工
總務組員	總務部門、員工
會計	總務部門、會計師、員工
市場調查員	市場部門、市場調查員、員工
客服	市場部門、客服人員、員工

屬性設計的部分，在本論文中我們所設計的屬性種類與權限相同，上段所述的 17 種權限分別對應到屬性資料庫中的 17 種屬性，一級主管屬性、二級主管屬性等等，依此類推，而角色與屬性的對應關係也如同表 4.2 一般。本專案將權限與屬性做相同的設置，然而，在實際的應用之中是可以將權限與屬性分別設計成兩種系統的，也可以不使用角色來對應屬性，改成用權限來對應屬性，一切都可應實際需求來設計。

4.4 客戶端系統

本論文通過 PHP 與 HTML 語言設計了一套網頁客戶端系統，本節將逐步介紹客戶端系統的功能。進入系統後首先進行登入的動作(圖 4.4)，輸入完帳號密碼並點擊登入，系統會搜尋用戶資料庫中是否有帳號密碼相符合的用戶，帳號密碼正確則成功登入，畫面會顯示用戶的基本資訊(圖 4.5)，若帳號密碼有誤則顯

示錯誤訊息，並返回登入系統。登入成功後系統會自動連結金鑰管理授權伺服器，申請與下載私鑰與公鑰(圖 4.6)。

圖 4.4 登入介面

圖 4.5 登入成功

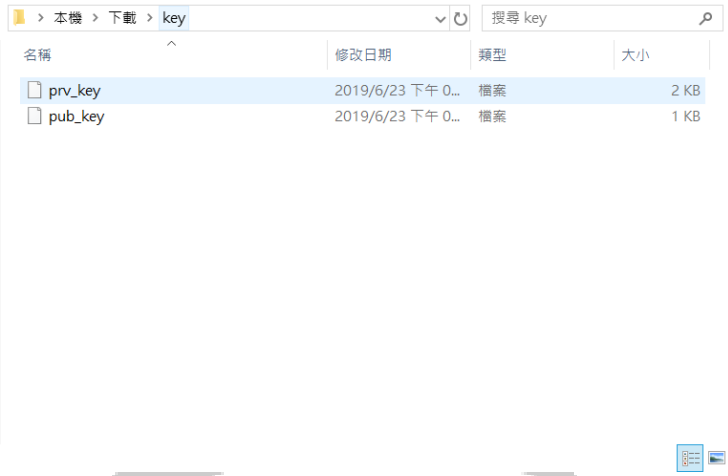
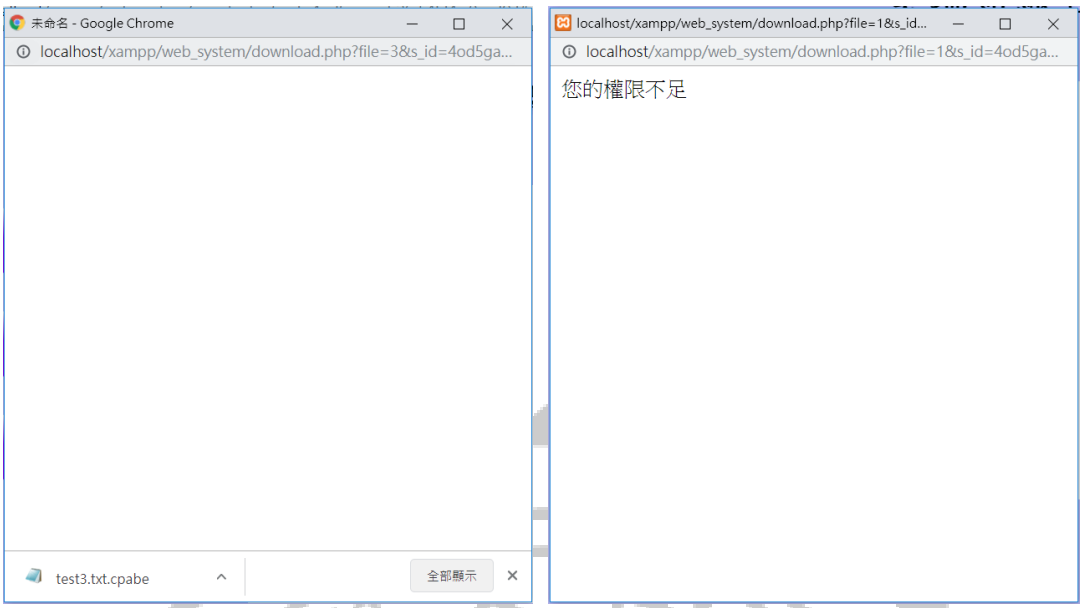


圖 4.6 私鑰與公鑰

點選系統左邊的檔案查詢，則會進入檔案查詢系統，由於我們伺服器上的檔案並不多，因此我們這邊並未建立檔案的分類或搜尋系統，而是直接顯示出所有伺服器上的檔案，如圖 4.7 所示。

圖 4.7 檔案查詢系統

找到欲下載檔案後點擊檔案名稱即會進入下載驗證系統，若使用者所具備的權限符合資料庫中記載的條件，將可成功下載檔案(圖 4.8-a)，若使用者所具備的權限不符合資料庫中記載的條件，則會顯示用戶權限不足的提示(圖 4.8-b)。



(a) (b)

圖 4.8 檔案下載系統 (a)驗證成功(b)驗證失敗

若欲上傳檔案則點選左邊的檔案上傳，會先進入下載權限設定的頁面，本論文所採用的權限設定方式，是通過讓使用者選擇哪些職位的人可以存取，系統再根據被選取的職位有哪些來產生相對應的存取權限設定，圖 4.9 為本系統的存取權限設定畫面。

圖 4.9 存取權限設定

點選送出後進入檔案上傳頁面，選取欲上傳檔案並上傳(圖 4.10)。

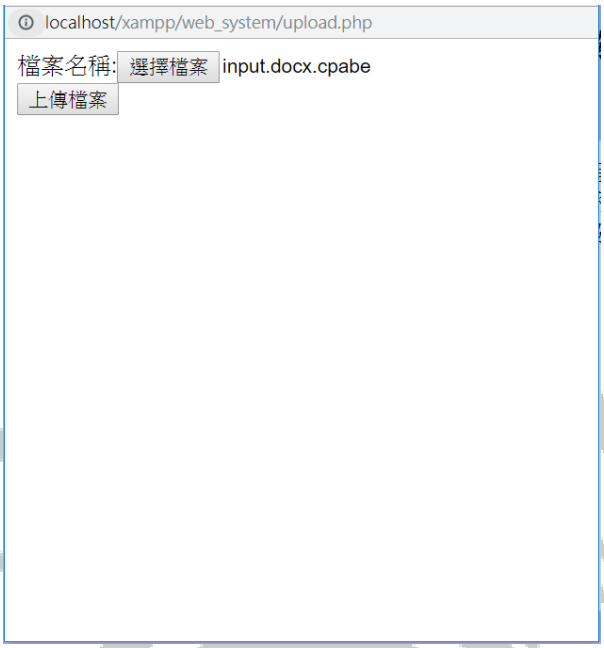


圖 4.10 檔案上傳

檔案上傳成功後即可在檔案查詢系統中看到所上傳的檔案(圖 4.11)。

圖 4.11 查看上傳的檔案

4.5 屬性加密系統

屬性加密系統共有 4 個階段，Setup、Encrypt、KeyGen、Decrypt，為了避免來自於傳輸過程與伺服器的資安風險，我們將 Setup 與 KeyGen 這兩個與金鑰生成相關的動作保留在金鑰授權伺服器中，而 Encrypt 與 Decrypt 這兩個加解密的動作則轉移至客戶的電腦上。

由伺服器執行 Setup 產生主密鑰與公鑰，當使用者登入客戶端系統後伺服器會收到使用者的角色集並將之與主密鑰結合，通過 KeyGen 產生私鑰，最後將公私鑰回傳給使用者(圖 4.6)。

加解密系統的部分我們用 JAVA 另外開發了一套 APP 軟體，如圖 4.12。

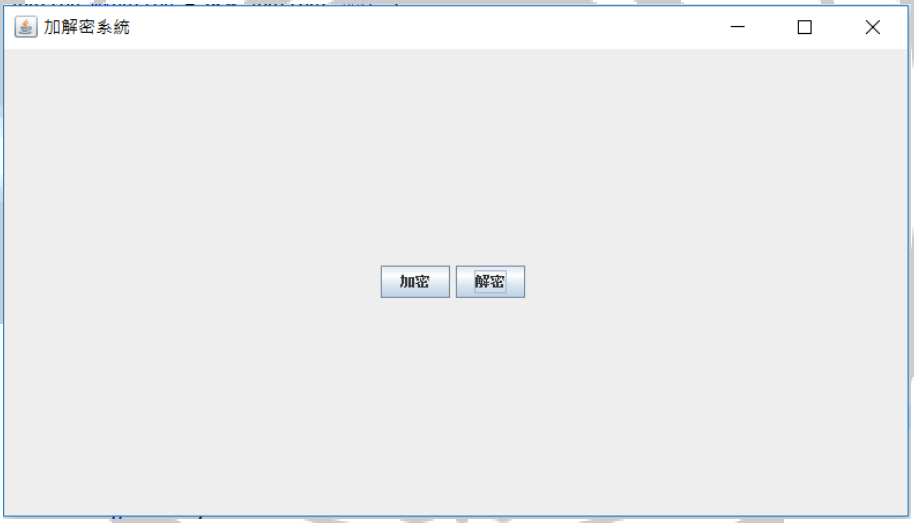


圖 4.12 加解密系統

首先介紹加密的部份，所有檔案上傳至雲端伺服器前均需先進行加密，點擊加密後進入加密系統(圖 4.13)，點擊選擇檔案與公鑰位置，確認欲加密檔案與公鑰的位置，中間的部份是存取權限的設定，設定方式與上傳檔案時的權限設定方式一樣，選擇可以閱覽的角色，系統會根據所選角色自動生成一套存取結構，點選提交即會開始加密，加密完成後會在原始檔案所在資料夾中產生一個副檔名為 cpabe 的密文(圖 4.14)。

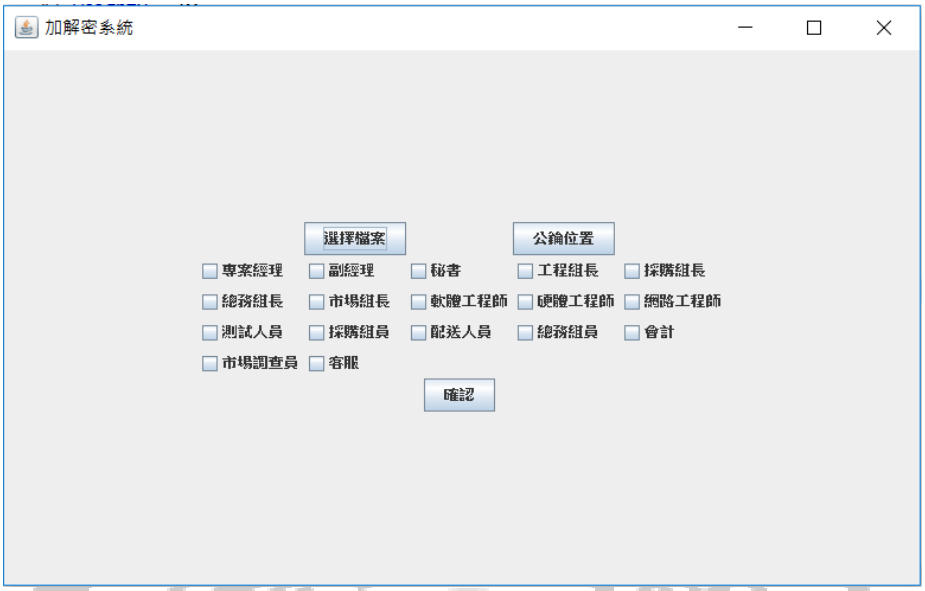


圖 4.13 檔案加密

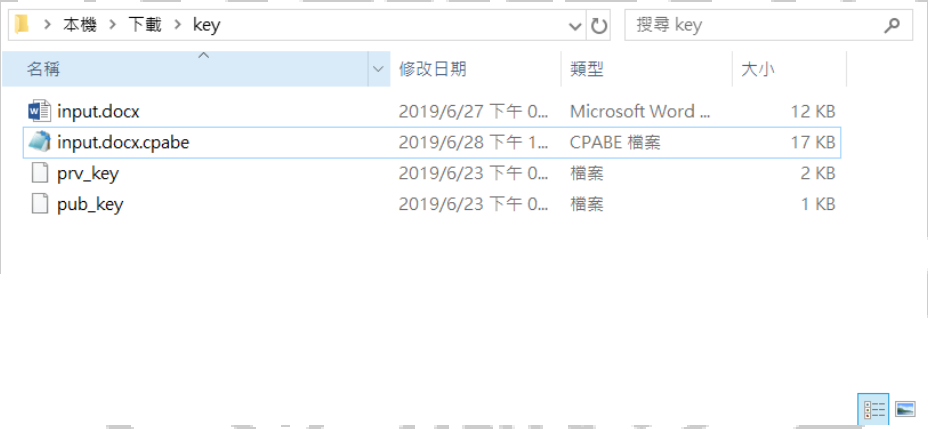


圖 4.14 生成密文

接著介紹解密的部份，點選圖 4.12 中的解密後進入解密系統(圖 4.15)，點選密文位置與私要位置，選取從雲端伺服器下載回來的密文以及存放在電腦中的私鑰，接著點選確認進行解密，解密完成後資料明文會被存放在與密文相同的資料夾中，如圖 4.16，為了方便與加密時所使用的原始檔案進行區別，我們在解密完成的檔案後加了一個 new 副檔名，只需將其去除即可正常開啟檔案。

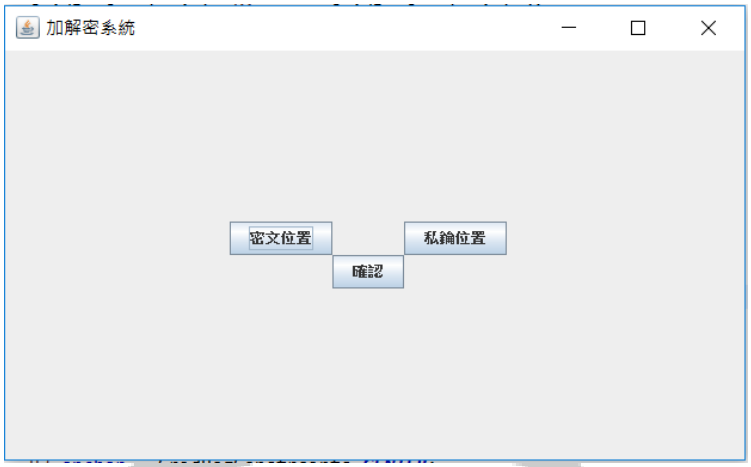


圖 4.15 解密系統

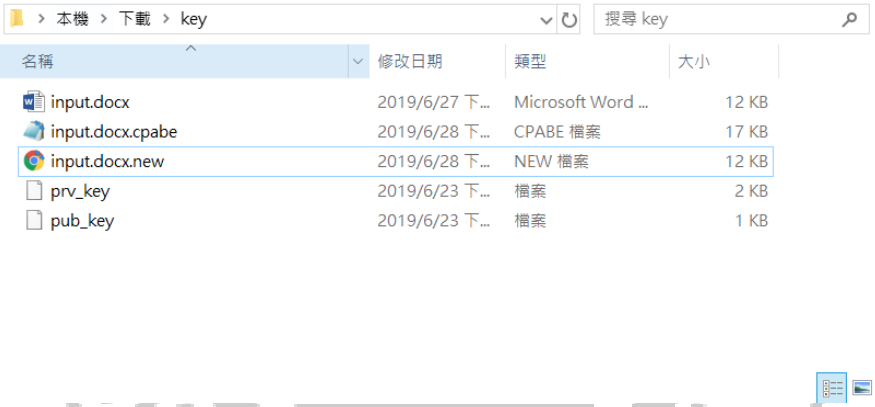


圖 4.16 獲得明文

第五章　結論

現代有許多的產品不再是依靠單一一家企業的力量來獨自完成，而是多家企業的合作成果，尤其對於緊握有少數關鍵技術的小公司來說，與其他企業的合作更顯重要。現在是一個網路與雲端皆越來越發達的時代，對於短期合作來說，租借伺服器來架設合作伺服器不失為一個可行的方案。

本文以實用以及資料安全為核心重新設計了一套供組織、企業合作時可以使用的雲端存取控制系統，並最終提出了一套系統雛形。我們利用角色基存取控制系統來架設我們的雲端伺服器，角色基存取控制系統提供了便利的管理功能，在實務上早已被廣泛應用。而為了避免傳輸過程以及雲端服務供應商可能出現的資料竊取風險，我們通過屬性加密來為所有即將被上傳至伺服器上的資料進行加密，所有的加解密作業均在客戶的電腦上完成，降低檔案被暴露在網路上的風險。這套系統在合作組織之間兼具公平性、便利性與安全性，相信在實際應用中具備一定程度的實用價值。

本文所提出之系統尚有許多可以改進、增強的地方，也可跟許多現有的系統或工具進行整合，譬如可搜尋加密(searchable encryption)。我們為來將持續進行相關研究，以期能挖掘出這套系統更大的價值，以及運用空間。

參考文獻

[1] B. R. Kandukuri, R. Paturi V, A. Rakshit, “Cloud Security Issues,” 2009 IEEE International Conference on Services Computing, pp. 517-520, 2009.

[2] C. Weinhardt, A. Anandasivam, B. Blau, N. Borissov, T. Meinl, W. Michalk, J. Stößer, “Cloud Computing – A Classification, Business Models, and Research Directions,” Business & Information Systems Engineering, vol. 1, no. 5, pp. 391-399, 2019.

[3] R. S. Sandhu, P. Samarati, “Access control: principle and practice,” IEEE Communications Magazine (Volume: 32 , Issue: 9), pp. 40-48, 1994.

[4] S. Yu, C. Wang, K. Ren, W. Lou, “Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing,” 2010 Proceedings IEEE INFOCOM, pp. 534-542, 2010.

[5] A. R. Khan, “Access control in cloud computing environment,” ARPN Journal of Engineering and Applied Sciences (Vol. 7, No. 5), pp. 613-615, 2012.

[6] David F. Ferraiolo, D. Richard Kuhn, “Role-Based Access Controls,” 15th National Computer Security Conference, Baltimore. pp. 554-563, Oct 13-16, 1992.

[7] R. S. Sandhu, E. J. Coyne, H.L. Feinstein, C. E. Youman, “Role-based Access Control Models,” IEEE Computer (IEEE Press), 29 (2): 38-47, August 1996.

[8] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, R. Chandramouli, “Proposed NIST Standard for Role-based Access Control,” ACM Transactions on Information and System Security (TISSEC), 4(3):224-274, 2001.

[9] A. Sahai, B. Waters, “Fuzzy Identity-Based Encryption,” Advances in Cryptology – EUROCRYPT 2005, pp 457-473, 2005.

[10] V. Goyal, O. Pandey, A. Sahai, B. Waters, “Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data,” 13th ACM conference on Computer and communications security, pp. 89-98, 2006.

[11] R. Ostrovsky, A. Sahai, B. Waters, “Attribute-Based Encryption with Non-Monotonic Access Structures,” 14th ACM conference on Computer and communications security, pp. 195-203, 2007.

[12] J. Bethencourt, A. Sahai, B. Waters, “Ciphertext-Policy Attribute-Based Encryption,” 2007 IEEE Symposium on Security and Privacy(SP'07), pp. 321-334, 2007.

[13] B. Waters, “Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization,” Public Key Cryptography – PKC 2011, pp. 53-70, 2011.

[14] Y. Ren, S. Wang, X. Zhang, Z. Qian, “Fully Secure Ciphertext-Policy Attribute-Based Encryption with Constant Size Ciphertext,” 2011 Third International Conference on Multimedia Information Networking and Security, pp. 380-384, 2011.

[15] 吳美玉，簡其弘，“以角色為基礎的存取控制於跨組織工作流程之研究，”中華大學-資訊管理學系(所), 2007.

[16] Yong Wang, Yuan Ma, Keyu Xiang, Zhenyan Liu, Ming Li, “A Role-Based Access Control System Using Attribute-Based Encryption”, 2018 International Conference on Big Data and Artificial Intelligence (BDAI), pp. 128-133, 2018.

[17] J. J. Hwang, H. K. Chuang, Y. C. Hsu, C. H. Wu, “A Business Model for Cloud Computing Based on a Separate Encryption and Decryption Service,” 2011 International Conference on Information Science and Applications, pp. 1-7, 2011.

[18] Junwei Wang, “Java Realization for Ciphertext-Policy Attribute-Based Encryption,” https://github.com/junwei-wang/cpabe.

[19] R. A. Botha, J. H. P. Eloff, “Designing role hierarchies for access control in workflow systems,” 25th International Computer Software and Applications Conference on Invigorating Software Development, pp. 117-122, 2001.