植基於智慧卡之企業矩陣型組織的數位版權管理系統

李鴻璋 教授
淡江大學資訊管理學系
hclee@mail.im.tku.edu.tw

張釋心
淡江大學資訊管理學系
hsin76628@gmail.com

摘要

企業數位版權管理系統是用來管理企業電子文件檔案及保護敏感性資訊被外洩的一種機制。根據觀察，目前所存的企業數位版權管理系統大多適用於傳統型直線式組織型態的企業，對於現今的企業在文件控管上並不能發揮最大的功效。隨著時代的演進與科技的進步，企業規模越來越大，而有效控管企業的組織型態也由傳統型直線式組織轉變為較彈性的矩陣型組織。本文即針對矩陣型組織的企業提出一套有別於過往的企業數位版權管理系統。

除了針對矩陣型組織設計一套企業數位版權管理系統以外，以往的企業數位版權管理系統都將文件創立者(DC, Document Creater)與文件權限設定者(DRS, Document Right Setting)設定為同一人，但此種方式並不能完全確保資料的隱密性，本論文欲將文件創立者與文件權限設定者設定為不同的角色，並提出將 XrML 權限檔案設計為主從兩段式，可因不同使用者而產出較短之個人版 XrML 權限檔案，上面只記載個人之員工編號，相較起以往完整記錄部門資訊之 XrML 權限檔案，授權更為單純更安全，使得文件得到更妥善的保護，達到一個安全理想的企業數位版權管理系統。

最後將此系統應用到醫學層面，使得整套醫療系統符合矩陣型組織及其大部分功能，並設計符合 HIPAA 規範，提出文件權限設定者可達到 1 人以上達成共同簽署，使得電子病歷資料可以得到更安全的控管。

關鍵詞：矩陣型組織、智慧卡、企業數位版權管理系統、XrML

1. 緒論

1.1 研究背景

隨著資訊科技的發展，企業使用電腦記錄文件已經是很普遍的事，相對電子文件資料外洩的事件也越趨頻繁。根據美國資料竊盜資源中心（The Identity Theft Resource Center，ITRC）調查報告中顯示，2008 年資料外洩的事件共有 656 件，其中普通型態的文件資料外洩佔了 17.7%，但電子型態的資料外洩卻佔了高達 82.3%，資料外洩的原因又以人為疏失佔 35.2%最大比重。而專業於企業資料保護的 PGP Corporation 與隱私暨資訊管理研究機構 Ponemon Institute 於 2009 年發表的調查顯示，美國 2008 年因外聘人員或承包商導致資料外洩的比例佔了 44%，與內部疏失有關的則佔了 88%，每件資料外洩意外的平均成本為 665 萬美元，Ponemon 創辦人 Larry Ponemon 表示不只是資料外洩成本持續增高，企業還得面臨寶貴客戶流失的問題[9]。

而近期最重大的資安外洩事件莫過於 2010 年 7 月美軍 9 萬份機密文件外洩的事件，洩密過程為洩密者曾在美軍的基地工作，竊取到機密文件之後，將機密文件複製到光碟裡，再把光碟偽裝成 Lady Gaga 的唱片作為掩飾盜走資料，最後交予「維基解密」而曝光，而此事件影響國家關係的敏感[6]。

上述調查結果反映了文件外洩的氾濫以及所造成的重大損失，固電子文件安全控管上的議題在近年來已經備受高度關注，因而發展了數位版權管理(Digital Rights Management，DRM)，後將此應用於企業上稱為企業數位版權管理(Enterprise-oriented Digital Rights Management，EDRM)。

1.2 研究動機與目的

企業數位版權管理是一種可以將電子文件安全控管的方式，企業組織在近年來已經使用這種方式來控管電子文件使得公司資訊可以受到完整的保護。但是目前被提出的企業數位版權管理系統大多都以傳統型的企業組織形態去設計，較沒考量到現今以及未來企業組織架構的發展趨勢。

傳統直線型組織型態使得數位存取控制只繼承於上層工作人員，使得企業不能靈活規畫組織結構，而有效運用公司人力資源的矩陣型組織(Matrix Organization)已成為未來企業必然的組織型態[1]，本論文中，我們針對矩陣型組織企業提出一套不同於以往的企業數位版權管理架構，將文件創立者與文件權限設定者區別為不同的角色，並將 XrML 權限檔案設計為主從兩段式，可因不同使用者而產出較短之個人版 XrML 權限檔案，運用此架構有效運用人力資源管理企業文件的同時也可以防止文件外洩，使得企業文件達到機密性、驗證性、可用性、可追溯性以及機動性。另外再將此架構應用於醫學層面，使得整套醫療系統符合矩陣型組織及其大部

分功能，並設計符合 HIPAA 規範，將文件權限設定者可達到 1 人以上達成共同簽署，使得病歷資料得到安全的控管。

2. 文獻探討

企業數位版權管理系統近來有許多學者提出，此章我們將介紹幾種之前學者所提出的企業數位版權管理系統。

2.1 Lin 等人所提機制

2009 年 Lin 等人[10]提出這套數位版權管理系統，針對層級結構的組織所設計。層級結構(Hierarchical structure)的概念為：(1) 管理人直接繼承他下屬相關數位內容的存取權限。 (2) 下屬的數位內容的存取權限，只是他的直接管理人的一個子集 。但此種結構的缺點為每個數位控制的存取控制將更加複雜且更不能靈活規畫組織結構。

2.1.1 四個角色

這套機制有四個角色分別為使用者(User, U)、數位權限伺服器(Digital Rights Server, DRS) 、數位內容伺服器(Digital Content Server, DCS) 以及應用軟體(Monitor Software, MS)。而針對這四個角色有以下五點題要：

(1) 數位權限伺服器在整套系統中扮演憑證管理中心(Certificate Authority, CA)的角色，因此數位權限伺服器在組織中有對使用者發行及更新憑證的工作。
(2) 每一個角色都有專屬自己的憑證及一對公鑰和私鑰。 憑證上會記錄擁有者的公鑰。
(3) 在每方之間都有安全數據通信。而這個假定可以通過採取 SSL 協議或其他現有的密碼系統達到。
(4) 此套 eDRM 只提供服務給在組織中使用電腦的用戶 。
(5) 應用軟體已經裝置在組織裡的電腦中，以用來了解客戶有關下載的狀態。

2.1.2 三個階段

此套系統將使用者的步驟分成三階段，分別為上傳(Uploading)、下載(Downloading)及使用(Usage)。當使用者想要對數位內容進行上傳或下載時，一定要出示他的憑證，憑證是使用這套系統所必需的，它就像是一個使用者的身分辨別。

當一個使用者成為員工時，他會收到一個憑證，而這憑證上面記錄了使用者的員工編號及他的工作部門及層級，這些資訊都由數位權限伺服器所簽章包裝起來($Cert_U = Sig_{DRS}(U_{id}\|Dep(y)\|Lvl(z))$)。另外數位權限的部份定義如下所示，$Right_{sno} = Sig_U(sno \| U_{id} \| Content_{sno} \| R(i,j) \| P(i,j) \| M(i,j) \| D(i,j))$，其中 R. P. M. D 代表如下 read (R), print (P), modification (M)及 delete (D)。 i 代表核准在同一部門的相關工作者的存取控制次數。 j 代表核准上級的存取控制次數。0 跟-1 跟所有正數都可以放。舉例來說，R(-1, -1)代表真對相關工作者跟上級沒有限制的次數 Read，M(2, 0)就是對相關工作者的限制次數是 2 但上級是 0。 以下詳細說明三個階段。

表 2-1 相關符號對照表

符號	定義
Content	數位內容
sno	使用者端產生的序號，用來連接作者創作的數位內容及相對應的數位權限。
$Content_{sno}$	有序號的數位內容。但只有用於上傳階段，接下來會使用$Content_{Cid}$替代。
$Content_{Cid}$	有公司認證序號的數位內容
C_{id}	由數位權限伺服器產生的公司認證序號
K_{Cid}	用來加密數位內容的對稱式鑰匙
EC_{id}	加密過後的數位內容
$Cert_x$	X 的憑證
$Right_{sno}$	有使用者端序號的數位權限
$Right_{Cid}$	有公司認證序號的數位權限
$EK_x()$	使用K_x執行加密的步驟
$DK_x()$	使用K_x執行解密的步驟
MS_{id}	應用軟體的序號
U_{id}	使用者的員工編號
h()	單向雜湊含數
TP_Right_{Cid}	使用者可以使用的權限所集合的一個子集，及給予相對應數位內容的公司認證序號
content_list	使用者可請求數位內容的公司認證序號清單

(1) 上傳階段

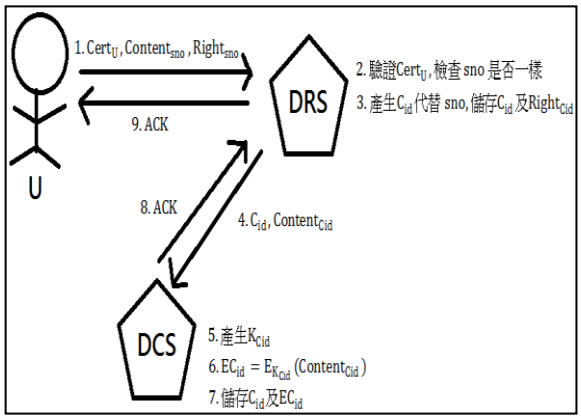

圖 2-2 上傳階段示意圖

此階段為使用者上傳所創的數位內容及相對應的數位權限。步驟說明如下：

(a) 使用者傳送他的憑證、編號過的創造的數位內容及相對應的數位權限給數 位權限伺服器。

(b) 數位權限伺服器收到傳送的資訊之後，驗證使用者的憑證，再檢查數位權限編號跟數位內容的序號(sno)是否一樣。

(c) 數位權限伺服器產生一個公司認證序號(C_{id})，此時公司認證序號開始代替之前的序號(sno)，並儲存公司認證序號及數位權限，再將公司認證序號及數位內容傳給數位內容伺服器。

(d) 數位內容伺服器收到資訊之後，自動產生一把對稱式鑰匙K_{Cid}，再用這把鎖匙對數位內容加密成EC_{id}，並儲存公司認證序號及EC_{id}。最後再傳一個成功回應給數位權限伺服器。

(e) 數位權限伺服器再傳送一個回應給使用者。

(2) 下載階段

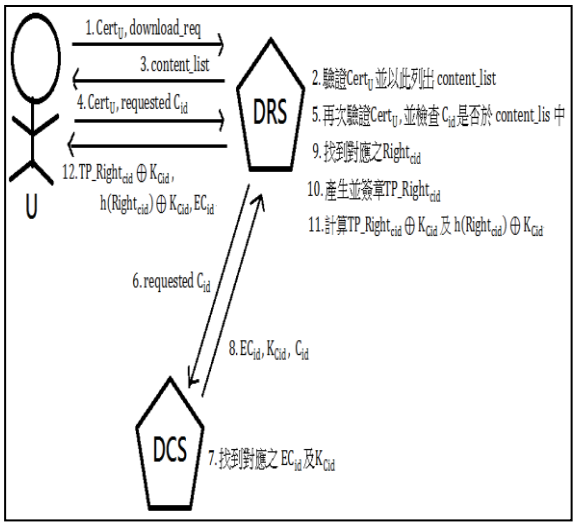

圖 2-3 下載階段示意圖

使用者下載欲存取的數位內容及相對應的數位權限。步驟說明如下：

(a)使用者傳送他的憑證跟下載請求給數位權限伺服器 。

(b)數位權限伺服器收到資訊之後,驗證憑證身分，並從憑證中取得使用者的工作部門及層級，列出該所屬的工作部門及層級可以下載檔案相對應的公司認證序號清單(content list)，再將此清單傳送給使用者。

(c)使用者收到清單之後，從中選擇了他要的公司認證序號(C_{id})，再傳送他的憑證跟公司認證序號給數位權限伺服器。

(d)數位權限伺服器收到憑證跟公司認證序號，再次驗證憑證，並驗證此公司認證序號是否於清單內。驗證都通過之後，數位權限伺服器傳送公司認證序號給數位內容伺服器。

(e)數位內容伺服器找到對應的加密後檔案(EC_{id})及加密鑰匙(K_{Cid})，並一起將公司認證序號(C_{id})傳回給數位權限伺服器。

(f)數位權限伺服器收到加密後檔案跟加密鑰匙後，找到相對應的數位權限，產生TP_Right_{Cid} 並對此簽章，最後再跟加密鑰匙做 XOR 運算，並對數位權限做雜湊並與跟加密鑰匙做 XOR 運算，再跟加密過後的檔案傳送回給使用者。

(3) 使用階段

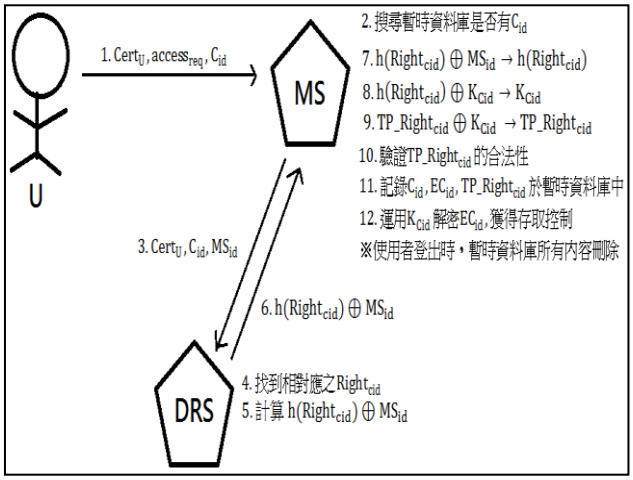

圖 2-4 使用階段示意圖

即使下載階段已經完成，使用者在此時還不能對他收到的數位內容做任何的動作，因為這些數位內容是加密過的檔案，所以使用者必須透過應用軟體向數位權限伺服器取得授權才能存取數位內容。說明步驟如下：

(a)使用者傳送憑證、存取請求、公司認證序號給應用軟體。

(b)當應用軟體收到資訊後，在暫時資料庫裡搜尋是否有公司認證序號，如果有，應用軟體傳送使用者憑證、公司認證序號跟應用軟體本身的序號(MS_{id})給數位權限伺服器。如果沒搜尋到，應用軟體會引導使用者回下載階段。

(c)數位權限伺服器收到資訊後，驗證應用軟體序號，找到相對應的數位權限之後做雜湊並跟應用軟體序號作 XOR 運算，傳回給應用軟體。

(d)應用軟體收到剛剛計算的值之後，運用自己的序號算出雜湊後的數位權限，再運用雜湊後的數位權限導出加密鑰匙，再導出TP_Right_{Cid}，最後再驗證TP_Right_{Cid}簽章的有效性。如果有效，應用軟體記錄公司認證序號、TP_Right_{Cid}跟加密過的檔案到暫時資料庫。

(e)最後，應用軟體檢查內容的公司認證序號是否跟TP_Right_{Cid}相符，如果通過，應用軟體運用剛剛導出的加密鑰匙解密數位內容讓使用者可以進行控制。當使用者登出時，暫時資料庫的所有內容會被刪除。

2.2 陳金鈴所提機制

2009 年，陳金鈴[4]提出一種植基於群體導向授權為基礎的企業數位產權管理系統，群體導向授權是指使用者在 n 個授權者中通過 t 個授權者的身分驗證同意後，方可獲得解密金鑰，進而存取數位內容。但此種授權方式在現實情況來說並不符合企業所需。此套系統包含以下五點概念：

(1)核心技術在於發佈者才可以決定誰是使用者以及他們的權限。

(2)「開啟文件必須先取得授權」，區隔了文件的使用者與發佈者。

(3)專案成員的組織是動態且時常變動的。

(4)企業內部數位內容存取的授權方式也應該是動態的授權。

(5)結合 DRM-AP(DRM-enable Application)表示使用者端具有可防弊端(Tamper-Proof)的數位產權應用軟體。

圖 2-5 為以群體為基礎的企業內部產權管理系統授權方式架構圖。並將此圖分為以下兩個階段說明，分別為"內容封裝及次密鑰產生階段(圖中步驟 1~3)"及"獲取解密金鑰階段(圖中步驟 4~7)"。

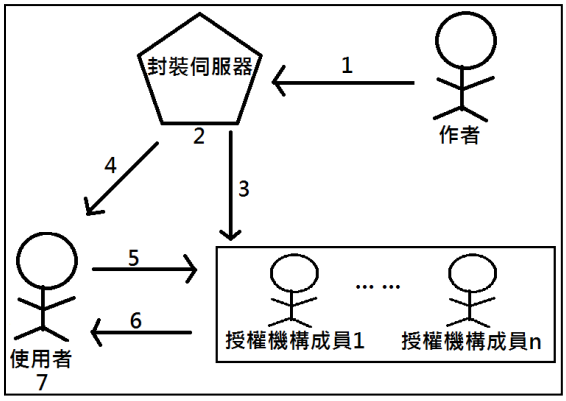

圖 2-5 以群體為基礎的企業內部產權管理系統授權方式架構圖

2.2.1 內容封裝及次密鑰產生階段

由封裝伺服器將檔案封裝加密。產生安全等級不一的秘密分享多項式，所需授權門檻 t 值可依企業需求動態調整，並將產生的次密鑰送給授權機構成員。說明步驟如下：

(1) 作者將所創數位內容上傳至封裝伺服器儲存。

(2) 封裝伺服器產生一把對稱式加密鑰匙(Key_{cid})，並用此鑰匙將編碼過數位內容加密成密文($C = E_{KEY_{CID}}(M)$)，另外將密文做雜湊後，以封裝伺服器自己私鑰簽章得簽章值($Sig_C = S_{SK_{PS}}(h(C))$)。最後結合內容表頭相關資訊及密文成為數位產權管理內容格式檔 DCF。

內容識別碼 (CID)	數位產權執行軟硬體類別(DRM-APtype)	群體授權識別碼 (ID_G)	數位內容說明
授權門檻值 t	屬性(Attributes)	加密後的內容簽章值 Sig_C	授權機構的 URL
加密過後的內容(C)			

圖 2-6 DCF 架構圖

(3) 依照企業安全需求設定安全等級 t 值，並將加密鎖匙封裝到下列秘密多項式中，$f(x) = KEY_{CID} + a_1x + \cdots\cdots + a_{t-1}x^{t-1} \mod P$ ，其中 P 是大於等於加密鎖匙的質數，$a_1, a_2 \cdots a_{t-1}$是[1,p-1]內任意整數。使得應用軟體只要取得 t 個授權者的次密鑰就可經由 Lagrange 多項式插入法解出加密鎖匙。

另外，封裝伺服器將數位內容識別碼 CID、授權者代碼ID_{AAi}及所屬群組碼ID_G做雜湊以產生一個訊息摘要為產生次密鑰SS_i的參數(i=1 to n)，$SS_i = f(h(CID, ID_{AAi}, ID_G))$，n 個次密鑰會被送到 n 個機構成員中。

2.2.2 獲取解密金鑰階段

授權成員驗證決定是否同意發出其本身所握有之次密鑰給使用者。再由 DRM-AP 重組這些次密鑰成解密金鑰，解開數位內容。

(4) 使用者下載 DCF 檔，下載對應的 DRM-AP，將其嵌入到使用者的電腦中，負責解密被加密過的數位內容。

(5) 使用者透過 DRM-AP 提出憑證向授權機構成員請求獲取解密金鑰，要求授權者賦予存取權利。DRM-AP 由個人憑證中擷取資訊，當作使用者的密鑰SK_U，以此密鑰再對數位內容識別碼(CID)、使用者識別碼(ID_U)、需求資訊(REQ)做簽章 ，得到Sig_U，$Sig_U = S_{SK_U}(CID, ID_U, REQ)$。並將$Sig_U$、CID、$ID_U$、REQ 及個人憑證 Certificate 送給所隸屬的授權成員去驗證。

(6)$V_{PK_U}(SG_U) \stackrel{?}{=} (CID, ID_U, REQ)$，授權機構成員以使用者的公鑰驗證$Sig_U$，並將次密鑰($SS_i$) 、數位內容識別碼(CID)、群組識別碼($ID_G$)、授權機構的成員識別碼($ID_{AAi}$) 及該使用者能存取之權限(RIGHT)等包裝成簽章值SG_{AAi} ，

$SG_{AAi} = S_{SK_{AAi}}(CID, ID_G, ID_{AAi}, RIGHT, SS_i)$，最後將$SG_{AAi}$、CID、$ID_G$、$ID_{AAi}$、RIGHT、$SS_i$送回給 DRM-AP。

(7) DRM-AP 收到上述資訊後以同樣方法驗章，最後得到 t 個有效次密鑰SS_i後，才能藉由以下 Lagrange 多項式插入法重建組成解密金鑰KEY_{CID}，

$$f(x) = \sum_{i=1}^{t} SS_i \prod_{j=1, j\neq i}^{t} \frac{x - h(CID, ID_{AAj}, ID_G)}{h(CID, ID_{AAi}, ID_G) - h(CID, ID_{AAj}, ID_G)} \mod P$$

$f(0) \mod P$
$= KEY_{CID} + a_1 \cdot 0 + \cdots\cdots + a_{t-1} \cdot 0^{t-1} \mod P$
$= KEY_{CID}$

$M = D_{KEY_{CID}}(C)$

2.3 呂安邦所提機制

呂安邦[2]所提機制以智慧卡為核心，除了以智慧卡驗證使用者身分以外，還利用智慧卡進行公鑰及私鑰的運算處理，並將數位內容檔案都存於智慧卡受保護的區域中。

2.3.1 四個角色

(1) 文件伺服器(File Server, FS)

主要功能有(a)儲存所有加密文件資料。(b)新進員工向此註冊及申請卡片。(c)認證要求檔案的員工或主管。(d)將離職員工編號加入移除名單 revocation list。(e)產生公司種子 seed 及與各部門主管協議產生秘密資訊 B。

(2) 授權伺服器(License Server, LS)

每部門擁有一部專屬該部門的授權伺服器，由主管管理。主管將產生好的文件檔案授權利用授權伺服器的公鑰加密並儲存。授權伺服器利用部門祕密資訊 B 驗證要求檔案的員工是否為合法的部門人員。

(3) 部門主管(Manager)

專屬的智慧卡稱為主管卡，內含唯一的職員識別碼、公司種子 seed 及該部門秘密資訊 B。主管製作好一個空的文件檔案後，利用智慧卡產生隨機值 A 並與卡片中的公司種子做運算為文件加密金鑰。另外製作相對應的授權。

(4) 員工(Employee)

專屬的智慧卡稱為員工卡，只能用來計算文件檔案的金鑰及對加密後的文件進行解密的功能。

2.3.2 五個階段

(1) 註冊階段

新進職員將自己基本資料傳給文件伺服器檢驗並登錄。文件伺服器將公司種子、所屬部門秘密資訊 B 及唯一員工識別碼置入智慧卡受保護區域。資料確認無誤後將發行智慧卡給員工。

(2) 建立文件階段

a. 主管建立空的文件檔案(file)及使用權限(Usage_Right)並給定文件編號(f_id)
b. 透過智慧卡產生隨機值 A，用卡中的公司種子計算加密金鑰($sk = seed \oplus A$)
c. 透過智慧卡對 file 加密成$E_{sk}(file)$，並產生簽章$Sign_M(E_{sk}(file))$
d. 將$E_{sk}(file)$、$Sign_M(E_{sk}(file))$及$PK_{FS}(B)$傳至文件伺服器
e. 文件伺服器以自己私鑰解密$PK_{FS}(B)$得到 B，透過 B 驗證使用者所屬部門。
f. 利用主管公鑰驗證$Sign_M(E_{sk}(file))$的簽章值。
g. 儲存$E_{sk}(file)$ 。
h. 將A、f_id 及 Usage_Right製成license = {A，f_id，Usage_Right}，並對license簽章再與license一起用授權伺服器的公鑰加密成$M1 = PK_{LS}(Sigh_M(license), license)$並傳送給授權伺服器。
i. 授權伺服器以私鑰解M1驗證$Sigh_M(license)$及得到license並儲存之。

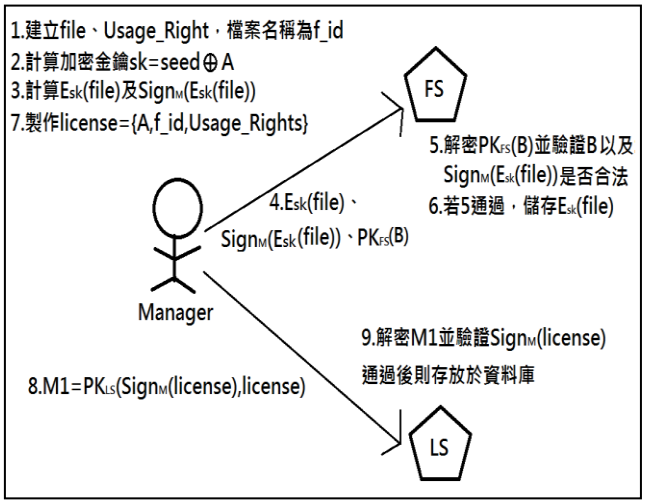

圖 2-7 建立文件階段示意圖

(3) 使用文件階段

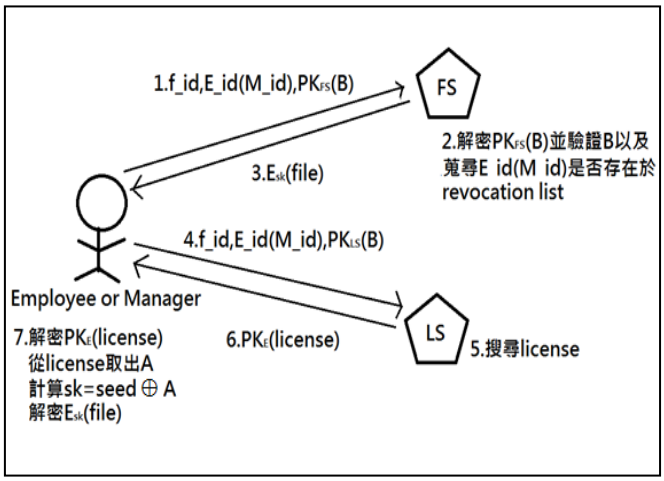

圖 2-8 使用文件階段示意圖

a. 職員向文件伺服器搜尋 f_id 檔案，傳送 E_id(M_id)及 PKFS(B)給文件伺服器
b. 文件伺服器利用私鑰對$PK_{FS}(B)$解密的到 B 並以此驗證職員合法性，並核對職員是否存在於 revocation list 中，如果存在則拒絕服務。
c. 文件伺服器將$E_{sk}(file)$傳給職員，並記錄要求。
d. 職員接著傳送編號 f_id、E_id(M_id)及$PK_{FS}(B)$給授權伺服器要求 license。

e. 授權伺服器找到相對應的 license 並用職員公鑰加密成PK_E(license)給職員
f. 職員透過智慧卡利用私鑰解密PK_E(license)從 license 得到 A，再與卡片中的 seed 做運算得到 sk = seed ⊕ A，再解密出文件檔案 file。

(4) 其它部門使用文件階段

a. 當部門 B'的職員傳送文件編號 f_id 及存取要求給部門 B 的主管
b. 主管產生一個暫時秘密資訊 C，計算M2 = $PK_{E'}(Sign_M(request, f_id, E'id, C, T))$並回傳給該職員。計算license' = {A, f_id, Usage_Right'}，並將$PK_{LS}(E'_id, license', C)$傳給部門 B 的授權伺服器。計算$PK_{FS}(E'_id, f_id, C, T)$給文件伺服器為驗證資訊。
c. 職員收到 M2 後利用私鑰進行解密，再傳送 M3 = $PK_{FS}((Sign_M(request, f_id, E'id, C, T), request, E'_id, f_id)$給文件伺服器要求檔案。
d. 文件伺服器收到M3後利用私鑰解密並核對職員編號是否存在移除名單中，檢查是否有暫時秘密資訊 C 及簽章值，再將E_{sk}(file)傳給職員並記錄。
e. 職員傳送 M4 = $PK_{LS}((Sign_M(request, f_id, E'id, C, T), request, E'_id, f_id)$給授權伺服器要求相對應的 license'。
f. 授權伺服器收到M4後利用私鑰解密並驗證簽章值及對照E'id是否合法。將license'加密為PK_E(license')給職員。
g. 職員利用私鑰解密PK_E(license')後從中取出 A 進而計算出加密金鑰sk，再解密加密檔案。

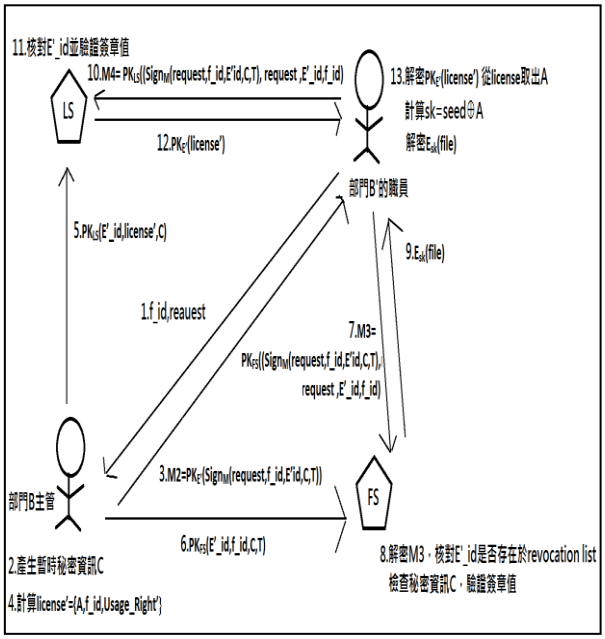

圖 2-9 其它部門使用文件階段示意圖

(5) 移除名單階段

為了防範離職職員利用舊的智慧卡或取公私機密文件，文件伺服器建立一個移除名單 revocation list，當職員離職時卡片編號加入此名單中，每當職員要求檔案或認證時都會以此名單驗證身分的合法性。

3. 所提機制：MOSE 機制

此章節將介紹本篇論文所提的機制，而我們將此機制稱為 MOSE。其原因取機制中的關鍵因素矩陣型組織(Matrix organization)、智慧卡(Smart card)及企業數位版權管理系統(EDRM)的各個英文單字字首。

3.1 三個角色及關鍵因素的應用

我們將 MOSE 機制分為三個角色，各角色說明如下：
(1)員工 (employee)為三種身分分別為文件創立者(DC, Document Creater)、文件權限設定者(DRS, Document Right Setting)、文件使用者(DU, Document User)。
(2)權限伺服器(RS, Right Sever)為兩部分分別為人事資料庫(Personnel DataBase, PD)及權限資料庫(Right DataBase, RD)。其中人事資料庫記載所有員工的所屬部門及工作層級。
(3)文件伺服器(DS, Document Sever)

3.1.1 智慧卡(Smart card)的應用

智慧卡由版權控管單位發行給新進職員，每個員工都有專屬的智慧卡，在系統中就如同員工(employee)的個人憑證，用來辨別使用者的身分，而在進行機制中任一階段時都必須出示個人識別憑證。智慧卡上記錄著員工識別碼(E_{id})、員工專屬的公開金鑰($EK_{E_{id}}$)、員工專屬的私密金鑰($DK_{E_{id}}$)。而權限伺服器的人事資料庫中記錄了每個員工的工作部門及層級，因此智慧卡上不用特別記錄，此設計可省去每次更新職位就必須將智慧卡拿去更新的動作。

3.1.2 矩陣型組織(Matrix organization)的應用

在此以矩陣型組織的概念落實文件上的控管，我們將文件分為三個種類分別是"部門層級文件"、"專案式文件"及"部門層級及專案式文件"，定義如下：
(1)部門層級文件：此文件屬於部門文件，依照所屬部門及工作層級判斷使用者是否具有權限來存取文件。
(2)專案式文件：此文件屬於專案文件，參與此專案的員工即有權限存取文件。

(3)部門層級及專案式文件：此文件屬於部門層級文件與專案式文件的綜合版，有些文件是需要某幾個部門及另外其他部門的某幾位員工才可以完成，因此設計了此混和文件使文件控管達到更好的彈性。

3.1.3 檔案權限(Document Right)

由文件權限設定者創造一完整 XrML 檔案權限，而檔案權限紀錄哪些使用者可進行哪些使用權限(編輯、列印或加註)。文件權限設定者上傳檔案權限之後，會與其他資訊封裝成權限使用資訊(Document Usage License)儲存於權限伺服器中的權限資料庫(RB)，爾後伺服器再針對不同的文件使用者所要求文件下之權限使用資訊產出較短之 XrML 權限描述檔案，文件使用者再依據此權限描述檔案的規範使用檔案。

文件權限設定者依照文件類別設定檔案權限，部門層級文件的設定工作部門及層級，專案式文件的設定員工識別碼，部門層級及專案式文件設定工作部門及層級或員工識別碼。

```
<license>
  <grant>
        <cx:write>
        <Group>財務部</Group>  <Level>3</Level>
        <Group>資訊部</Group>  <Level>5</Level>
        <Group>會計部</Group>  <Level>1</Level>
        <Group>法務部</Group>  <Level>3</Level>
        <Group>程式部</Group>  <Level>2</Level>
        <EmployeeID>E00005,E00006</EmployeeID>
        </cx:write>
        <cx:print>
        <Group>財務部</Group>  <Level>3</Level>
        <Group>資訊部</Group>  <Level>5</Level>
        <Group>法務部</Group>  <Level>3</Level>
        <Group>程式部</Group>  <Level>2</Level>
        <EmployeeID>E00006</EmployeeID>
        </cx:print>
        <comment>
        <Group>資訊部</Group>  <Level>5</Level>
        <Group>會計部</Group>  <Level>1</Level>
        <Group>程式部</Group>  <Level>2</Level>
        <EmployeeID>E00005,E00006</EmployeeID>
        </comment>
        <offline>yes</offline>
        <validityInterval>2011-01-22</validityInterval>
        <digitalResource>
          <fileID>P0001</fileID>
        </digitalResource>
  </grant>

  <issuer>
        <dsig:Signature>
              <!--權限設定者的簽章值 -->
        </dsig:Signature>
  </issuer>
</license>
```

圖 3-1 檔案權限可能表示例子

檔案權限可能表示例子(圖 3-1)意義如下：(1)財務部第 3 層以上、資訊部第 5 層以上、會計部第一層、法務部第三層以上、程式部第 2 層以上及員工編號 E00005、E00006 之員工可以編輯文件。(2)財務部第 3 層以上、資訊部第 5 層以上、法務部第三層以上、程式部第 2 層以上及員工編號 E00006 之員工可以列印文件。(3)資訊部第 5 層以上、會計部第一層、程式部第 2 層以上及員工編號 E00005、E00006 之員工可以於文件上加註。

3.2 三個階段

我們將 MOSE 機制分為三個階段，分別為文件使用權限上傳與建立、文件與檔案權限的下載及文件使用階段。

3.2.1 文件使用權限上傳與建立(Document upload and Right declaration)

此階段主要由文件創立者創立檔案及文件權限設定者設立權限後，上傳至權限伺服器及文件伺服器。步驟說明如下：

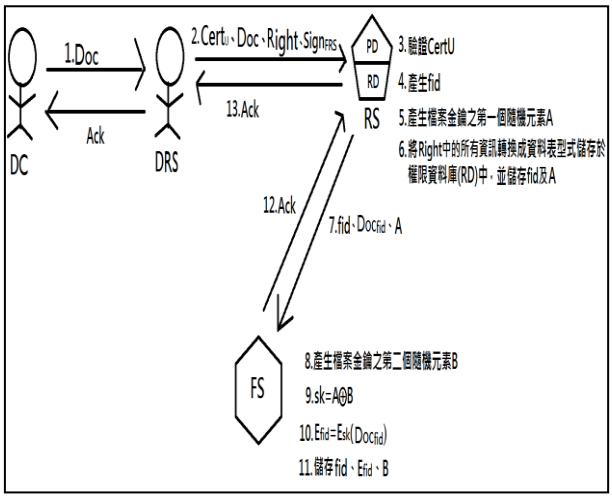

圖 3-2 文件使用權限上傳與建立階段示意圖

(1) 文件創立者創立一個新的文件檔案(Docutment, Doc)，並將此傳送至文件權限設定者。

(2) 文件權限設定者收到該檔案後，設立相對應的授權，並產出一完整的 XrML 權限檔案(此權限檔記錄有關此文件檔案之所有員工的相關權限，圖 3-3)，並將自己的憑證、文件檔案、相對應之檔案權限及自己的簽章值傳送給權限伺服器。

(3) 權限伺服器透過人事資料庫(Personnel DataBase, PD)中的資料驗證該文件權限設定者。

(4) 並產生相對應之文件檔案序號(fid)。

(5) 產生第一個隨機元素 A，此元素是之後產生檔案金鑰的重要元素。

(6) 權限伺服器讀取剛剛收到的 XrML 權限檔，將裡面的內容轉變成資料表型式(圖 3-4)儲存於權限資料庫(Right DataBase, RD)中，並儲存檔案編號及第一個隨機元素 A。

(7) 權限伺服器將文件檔案、相對應之檔案序號及隨機元素 A 傳送至文件伺服器(DS, Document Sever)。

(8) 文件伺服器收到上述資訊後，產生檔案金鑰之第二個隨機元素 B。

(9) 計算檔案加密金鑰sk = A ⊕ B

(10) 並以此金鑰加密Doc_{fid}成加密過後的檔案E_{fid}。

(11) 儲存隨機元素 B、加密過後的檔案E_{fid}及相對應之檔案序號於文件伺服器中。

```
<license>
  <grant>
          <cx:write>
          <Group>財務部</Group>  <Level>3</Level>
          <Group>資訊部</Group>  <Level>5</Level>
          <Group>會計部</Group>  <Level>1</Level>
          <Group>法務部</Group>  <Level>3</Level>
          <Group>程式部</Group>  <Level>2</Level>
          <EmployeeID>E00005,E00006</EmployeeID>
          </cx:write>
          <cx:print>
          <Group>財務部</Group>  <Level>3</Level>
          <Group>資訊部</Group>  <Level>5</Level>
          <Group>法務部</Group>  <Level>3</Level>
          <Group>程式部</Group>  <Level>2</Level>
          <EmployeeID>E00006</EmployeeID>
          </cx:print>
          <comment>
          <Group>資訊部</Group>  <Level>5</Level>
          <Group>會計部</Group>  <Level>1</Level>
          <Group>程式部</Group>  <Level>2</Level>
          <EmployeeID>E00005,E00006</EmployeeID>
          </comment>
          <offline>yes</offline>
          <validityInterval>2011-01-22</validityInterval>
          <digitalResource>
            <fileID>P0001</fileID>
          </digitalResource>
  </grant>

  <issuer>
          <dsig:Signature>
                  <!--權限設定者的簽章值 -->
          </dsig:Signature>
  </issuer>
</license>
```

圖 3-3 檔案相對應之完整 XrML 權限描述檔

員工資料表

員工編號	部門	層級	擔任文件創作者之專案	擔任權限設定者之專案	工作專案序號	...(個人資料等)
E00001	財務部	2	P001,		P003(3), P004(3), P005(7)	
E00002	會計部	3	P002, P003	P001,P004	P005(6)	
E00003	資訊部	3				
E00004	法務部	5				
E00005	程式部	2		P002,P003	P001(2), P004(2)	

權限-部門資料表

檔案序號	部門	層級	權限
P001	財務部	3	W,P,0
P001	資訊部	5	W,P,C
P001	會計部	1	W,0,C
P001	法務部	3	W,P,0
P002	程式部	2	W,P,C

編輯(W)　列印(P)　註解(C)

專案文件資料表

專案序號	專案名稱	是否離線作業	文件創作者	文件權限設定者	專案員工
P001	高鐵無線網路	是	E00001	E00002	E00005 (5), E00006(7)
P002	南北學生優惠案	是	E00002	E00005,E00006	E00006(6), E00007(7)
P003	手機無線上網	否	E00002	E00005	E00001(3)
P004	遠距教學	否	E00007	E00006, E00002	E00001(3), E00005 (2)
P005	選課系統	是	E00006	E00007	E00001(7), E00002(6)

專案文件分層權限：第一層-編輯; 第二層-列印; 第三層-加註; 第四層-編輯.列印; 第五層-編輯.加註; 第六層-列印.加註; 第七層-編輯.列印.加註;
※可離線作業即可存檔

圖 3-4 權限描述之資料表型式

3.2.2 文件與檔案權限的下載(Document download and Usage License)

此階段主要為文件使用者申請下載所欲存取的檔案及相對應之檔案權限。

步驟說明如下：

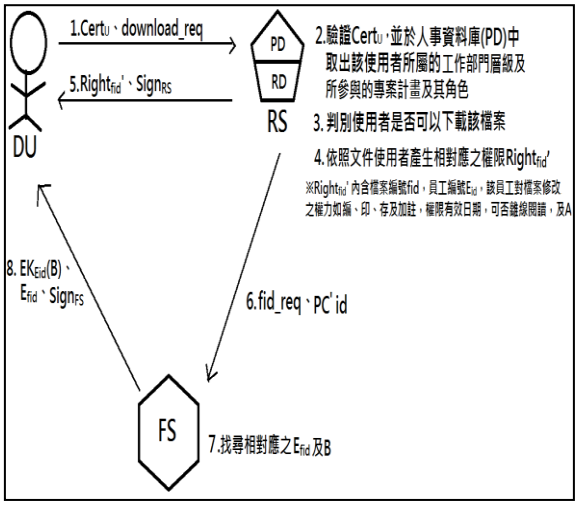

圖 3-5 文件與檔案權限的下載階段示意圖

(1) 文件使用者(DU, Document User)傳送憑證及下載的要求給權限伺服器，其中下載要求包含了檔案序號(fid)。

(2) 權限伺服器驗證此憑證並透過人事資料庫取出使用者的工作部門層級及該員工所參與的專案計畫及在其中之角色。

(3) 依照上述的資料驗證使用者的合法性。

(4) 上述步驟都驗證通過之後，針對該員工在此文件檔案中之權限，產生一個較短之 XrML 權限檔($Right'_{fid}$，圖 3-6)，此權限檔只記錄該員工在相對應文件檔案之可進行的權限內容。而此權限檔內含檔案編號、該員工之編號、該員工對文件檔案使用之權利如編輯列印存檔及加註、權限有效日期、可否離線閱讀及第一個隨機元素 A。

(5) 將此 XrML 權限檔($Right'_{fid}$)及權限伺服器之簽章值傳送給文件使用者。

(6) 權限伺服器將要求下載的請求及文件使用者所使用電腦之位址給文件伺服器。

(7) 文件伺服器依照檔案編號搜尋到相對應之加密檔案及檔案金鑰之第二個隨機元素 B。

(8) 將檔案金鑰之第二個隨機元素 B 以使用者的公鑰加密過後，與加密檔案及文件伺服器之簽章值給使用者。

```
<license>
  <grant>
          <keyHolder>E00005</keyHolder>
          <cx:write/>
          <cx:print/>
          <comment />
          <offline>yes</offline>
          <RandomA>0101001</RandomA>
          <validityInterval>2011/06/30</validityInterval>
          <digitalResource>
            <fileID>P0001</fileID>
          </digitalResource>

  </grant>

  <issuer>
          <dsig:Signature>
          <!--權限伺服器的簽章值 -->
          </dsig:Signature>
  </issuer>
</license>
```

圖 3-6 針對該位員工所產之較短 XrML 權限描述檔

3.2.3 文件使用階段(Document Execute)

此階段是文件使用者欲存取文件檔案的過程。

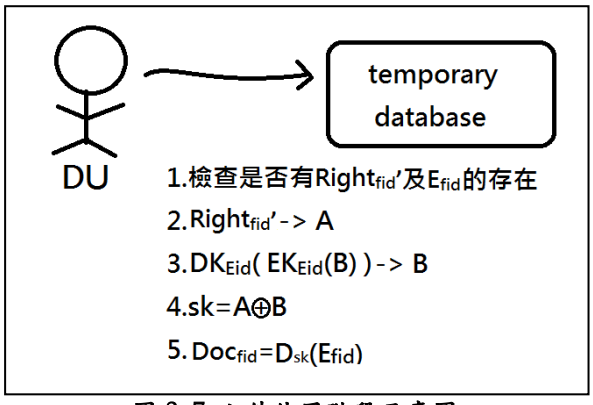

圖 3-7 文件使用階段示意圖

(1) 使用者端電腦檢查在暫時資料庫中是否有加密後之文件檔案E_{fid}及相對應之權限檔案$Right'_{fid}$的存在，如果沒有則引導使用者回下載階段。
(2) 從$Right'_{fid}$中取出檔案金鑰之第一個隨機元素A。
(3) 以自己的私鑰解出檔案金鑰之第二個隨機元素B。
(4) 算出檔案加密金鑰sk = A ⊕ B。
(5) 利用此加密金鑰即可解開加密後之文件檔案，並可開始依照權限內容對檔案進行使用。

※此階段另外有離線作業的功能，文件權限設定者評鑑文件的風險，設定該文件檔案是否可進行離線作業，所有 online 的檔案在離線之後一律刪除，可離線作業的文件檔案在離線時可備份於文件使用者的電腦中。離線時，透過使用者端應用程式認證使用者身分並依照$Right_{fid}'$檢查權限有效日期確認使用權以供作業。

4. 將 MOSE 機制應用於醫學層面

電子病歷的推廣也意味電子病歷的安全控管一天比一天重要，知名藝人 Selina 於 2010 年因拍爆破戲而嚴重灼傷送往林口長庚醫院，隨後傳出新聞指出其他科醫師為了滿足自己的好奇心，利用職務特權隨意瀏覽 Selina 的病歷[8]。這新聞也代表在電子病歷控管上尚未到達有效控制權限的部分，因此，在此章節我們參考 HIPPA 規範中之特性如病人的理解(Patient's understanding)、病人的控制(Patient's control)、機密性(Confidentiality)、資料完整性(Data integrity)、同意例外(Consent exception)[5][13]，將 MOSE 機制應用於醫學層面，將電子病歷視為每一份文件並利用權限設定達到有效的安全控管，以防止未經過授權的人隨意使用電子病歷。

4.1 角色對應關係及階段

我們將 MOSE 套用至醫學層面，而每一個角色之對應關係如下所示。
(1)三種身分分別為文件創立者相當於掛號部(Registration Department, RD)的角色，文件權限設定者將當於病患(Patient)及主治醫師(Attending Doctor, AD)的角色，文件使用者即欲使用病歷檔案的人(Document User, DU)。
(2)權限伺服器(Right Sever, RS)一樣分為人事資料庫(Personnel DataBase, PD)及權限資料庫(Right DataBase, RD)兩種。其中人事資料庫記載所有病患及醫療人員所屬科別及工作層級資料，而權限資料庫存放所有電子病歷檔案之權限。
(3)文件伺服器(Document Sever, DS) ：存放所有加密過後的電子病歷檔案。

4.1.1 智慧卡(Smart card)的應用

智慧卡由版權控管單位發行給新進醫療人員及病患，每個醫療人員及病患都有專屬的智慧卡，醫療人員的智慧卡稱為醫事人員卡，病患的智慧卡稱為全民健保卡。智慧卡在系統中就如同使用者之個人憑證，用來辨別使用者的身分，而在進行機制中任一階段時都必須出示智慧卡。醫事人員卡上記錄著醫療人員識別碼(D_{id})、醫療人員專屬的公開金鑰(EK_{Did})及醫療人員專屬的私密金鑰(DK_{Did})，全民健保卡中記錄了病患識別碼(P_{id}) 、病患專屬的公開金鑰(EK_{Pid})及病患專屬的私密金鑰(DK_{Pid})。而每個醫療人員的所屬科別及層級由權限伺服器的人事資料庫中記錄，因此醫事人員卡上不用特別記錄，此設計還可省去每次醫療人員更新職位就必須將智慧卡拿去更新的動作。

4.1.2 病歷資料的分類

我們將每一次的病程視為每一份病歷資料檔案，又視為一份文件，而文件控管上的分類對應關係如下：
(1)部門層級文件就如同一般科別看診，是指該病程只由某一專科負責。
(2)專案式文件就如同跨科看診，是指該病程由各個不同科的醫療人員所負責。
(3)部門層級及專案式文件就如同一般科別與跨科別的綜合版，是指由某一專科以及其他科醫療人員所組成。

4.1.3 檔案權限(Document Right)

每一份電子病歷檔案都有相對應之檔案權限，檔案權限紀錄哪些使用者可進行哪些使用權限。權限設定者由病人及主治醫師組成，病人有權控制所有授權，主治醫師有權掌控有關該科之所有病歷資料。主治醫師依患者情況授權下層醫療人員之權限，而此授權內容須經過病人本身了解並同意，由病人及主治醫師雙方共同簽署使授權生效。

病歷資料只能加註，無法編輯或刪除以防過往病歷資料被修改，因此授權權限之項目為讀取、列印或加註。另外每一次加註的內容都可標籤出該次內容之關鍵器官，以供未來方便查詢有關該器官之所有病歷資料。

而在病人無意識時、因法律因素或其他可能出現的例外情況，醫師將在病人無授權之情況下使用該病人之電子病歷，此需依照醫師的經驗及醫德並衡量當時之情況，判斷出最好的選擇，而每次只用此例外情況之功能時都會記錄使用者，以供未來查詢哪些使用者在未經授權情況下使用過該病歷資料。

4.2 三個階段

MOSE 機制應用至醫學層面仍為以下三階段表示，分別為文件使用權限上傳與建立、文件與檔案權限的下載及文件使用階段。

4.2.1 文件使用權限上傳與建立(Document upload and Right declaration)

病人於掛號部掛號即創立一個病歷資料檔，由病人及主治醫師共同簽署權限之後，上傳至權限伺服器及文件伺服器。步驟說明如下：

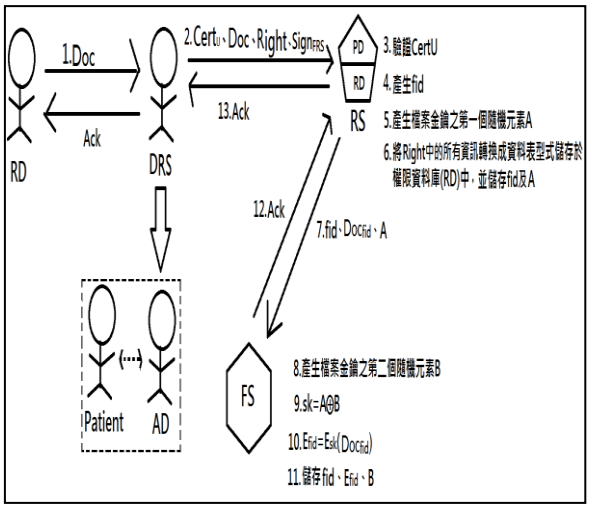

圖 4-1 醫學層面-文件使用權限上傳與建立階段示意圖

(1) 病人於掛號部掛號之後，如果為該病程之初次掛號，掛號部就創立該病程之病歷資料檔案(Docutment, Doc)，再由病人及主治醫師共同設定相對應之權限。
(2) 病人及主治醫師共同設定相對應之權限，主治醫師依造情況授權給其它醫療人員相關權限並告知病人本身，病人本身及主治醫師雙方都同意授權內容，設定完成後由使用者端應用程式產生一完整的 XrML 權限檔(此權限檔記錄有關此病歷資料之所有醫療人員的相關權限)，並共同簽署後，由主治醫師將憑證、病歷資料檔案、相對應之檔案權限及簽章值上傳至權限伺服器。
(3) 權限伺服器透過人事資料庫(Personnel DataBase, PD)中的資料驗證該主治醫師之憑證。
(4) 產生相對應之病歷資料檔案的編號(fid)。
(5) 產生第一個隨機元素 A，此元素是之後產生檔案金鑰的重要元素。
(6) 權限伺服器讀取剛剛收到的 XrML 權限檔，將裡面的內容轉變成資料表型式儲存於權限資料庫中，並儲存檔案編號及第一個隨機元素 A。
(7) 權限伺服器將隨機元素 A、病歷資料檔案及檔案之編號傳送至文件伺服器。
(8) 文件伺服器收到上述資訊後，產生檔案金鑰之第二個隨機元素 B。
(9) 計算病歷資料檔案加密金鑰$sk = A \oplus B$。
(10) 並以此加密金鑰將病歷資料檔案Doc_{fid}加密成加密過後的檔案E_{fid}。
(11) 儲存隨機元素 B、加密過後的檔案E_{fid}及相對應之檔案編號於文件伺服器中。

4.2.2 文件與檔案權限的下載(Document download and Usage License)

此階段為醫療人員對權限伺服器申請下載所欲使用的病歷資料檔案及相對應之權限。步驟說明如下：

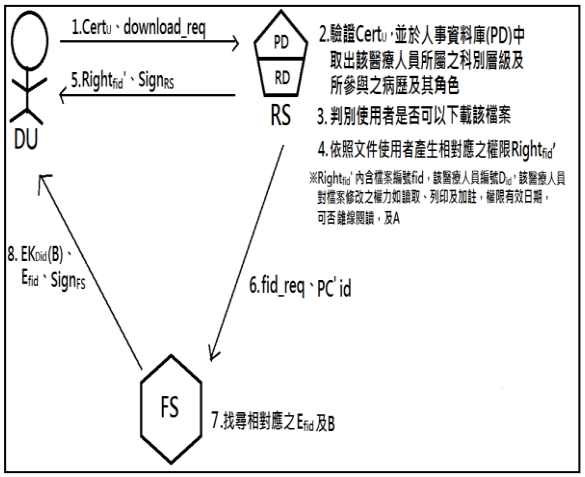

圖 4-2 文件與檔案權限的下載階段示意圖

(1) 該醫療人員傳送自己的憑證及下載要求給權限伺服器，其中下載要求包含病歷資料檔案序號(fid)。

(2) 權限伺服器驗證此憑證，並透過人事資料庫取出該醫療人員所屬的科別層級及有參與的病歷資料。

(3) 依照上述的資料驗證使用者的合法性。

(4) 上述步驟都驗證通過之後，針對該醫療人員在此病歷中之權限，產生一個較短之 XrML 權限檔($Right'_{fid}$)，此權限檔只記錄該醫療人員在相對應病歷資料檔案之可進行的權限內容。而此權限檔內含檔案編號、該醫療人員之編號、該醫療人員對病歷資料檔案使用之權利如讀取列印及加註、權限有效日期、可否離線閱讀及第一個隨機元素 A。

(5) 將較短之 XrML 權限檔($Right'_{fid}$)及權限伺服器之簽章值傳送給該使用者。

(6) 權限伺服器將要求下載檔案的請求及使用者所在電腦之位址給文件伺服器。

(7) 文件伺服器依照檔案編號搜尋到相對應之加密過後的病歷資料檔案及檔案金鑰之第二個隨機元素 B。

(8) 將檔案金鑰之第二個隨機元素 B 以使用者的公鑰加密過後，與加密病歷資料檔案及文件伺服器之簽章值給使用者。

4.2.3 文件使用階段(Document Execute)

此階段是醫療人員使用病歷資料檔案的過程。步驟說明如下：

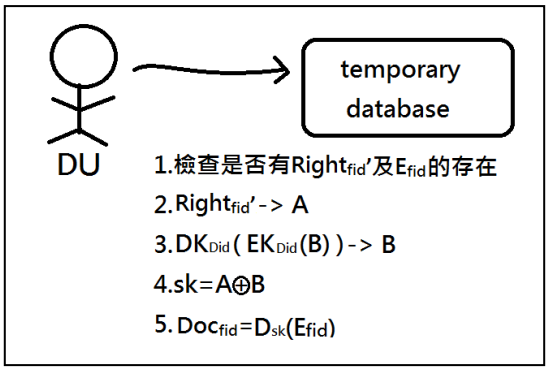

圖 4-3 文件使用階段示意圖

(6) 使用者端電腦檢查暫時資料庫中是否有加密後之病歷資料檔案E_{fid}及相對應之權限檔案$Right'_{fid}$的存在，如果沒有則引導使用者回下載階段。

(7) 從$Right'_{fid}$中取出檔案金鑰之第一個隨機元素 A。

(8) 以自己的私鑰解出檔案金鑰之第二個隨機元素 B。

(9) 算出檔案加密金鑰sk = A ⊕ B。

(10) 利用此加密金鑰即可解開加密後之病歷資料檔案，並可開始依照權限內容對檔案進行使用。

※此階段離線作業的功能：病人或主治醫師依照情況判定該病歷資料檔案是否可進行離線作業，所有 online 的病歷資料檔案在離線之後一律刪除，可離線作業的病歷資料檔案在離線時可備份於文件使用者的電腦中。離線時，透過使用者端應用程式認證使用者身分並依照$Right'_{fid}$檢查權限有效日期確認使用權以供作業。

5. 結論

5.1 比較與分析

現存大部份的企業數位版權管理系統大多步驟繁雜或是不符合真實情況，在此我們針對先前學者所提出的機制與 MOSE 機制做一比較及分析。以系統架構基礎來說，我們以智慧卡為基礎解決傳統被裝置綁住的問題。而機制中角色我們也較以往將各個角色界定得更清楚。而在 MOSE 機制與其他機制最大的不同在於我們將文件創立者與文件權限設定者分開，並且文件權限設定者可以達到一人以上，此種設計可讓文件控管更為彈性也更具有隱密性。以企業組織型態來說，傳統型的層級型組織或以群組授權為基礎的組織已不適用於現今的企業，較為彈性的矩陣型組織是現今及未來企業的必然發展趨勢。而下載檔案的位置如欲將資料都存在智慧卡中，智慧卡恐會面臨容量不足的問題，於是我們選擇使用者端電腦的暫時資料庫裡。 在金鑰產生的部分只有一把對稱式加密金鑰是很容易被破解的，而如果使用公司種子 seed 與隨機值做運算，

一旦公司種子被破，整間公司的所有文件都匯存在外洩的危機，於是我們使用兩個隨機值產生金鑰，並將兩個隨機值置於不同的地方達到更安全的理想。權限描述檔案部分，以往系統在權限設定者設立一個權限檔之後，所有的文件使用者都是以同一份權限檔案對檔案進行存取動作， MOSE 機制將權限設定者產出一完整之 XrML 權限描述檔案之後，伺服器再針對不同文件使用者產出所屬之較短 XrML 權限描述檔案，不同於以往將各員工及各部門資訊完整的記錄在上面，使得授權更為單純及安全。以往的企業數位版權管理系統都無離線作業的部分，MOSE 則是提供部分的離線作業於使用階段。

表 5-1 相關研究比較表

系統名稱	Lin 等人所提機制[10]	陳金鈴所提機制[4]	呂安邦等人所提機制[2]	MOSE 機制
系統架構基礎	Device-based	Device-based	Smart Card-based	Smart Card-based
機制所含角色	員工、文件伺服器、權限伺服器	員工、文件伺服器、權限伺服器	部門主管、員工、文件伺服器、權限伺服器	文件創立者、文件權限設定者、文件使用者、人事資料庫、權限資料庫、文件伺服器
文件創立者與文件權限設定者為同一人	是	是	是	否
權限設定者人數	1 人	1 人	1 人	1 人或 1 人以上
適用組織型態	層級型組織	群體授權為基礎的組織	層級型組織	矩陣型組織
下載文件檔案儲存位置	暫時資料庫	暫時資料庫	Smart Card	暫時資料庫
金鑰產生	對稱式加密金鑰	對稱式加密金鑰	公司種子 Seed ⊕檔案	檔案金鑰之第一個隨
			金鑰之隨機元素 A	機元素 A⊕檔案金鑰之第二個隨機元素 B
權限檔案	單一完整版	單一完整版	單一完整版	主從兩段式
支援離線作業	否	否	否	可設定

5.2 結論

本論文改善過往以傳統型直線式組織設計的企業數位版權管理系統不符合真實情況之缺點，將未來企業發展必然趨向的矩陣型組織型態融和入系統中，將權限檔案提出主從兩段的新概念，並將此架構應用於醫學層面提出權限設定者不只一人的新觀點，希望能讓各種企業有效使用此套機制，將公司所有的文件都可以達到完善的控管，減少機密文件外洩的事件。

參考文獻

[1] 何天立，《組織變遷過程中的組織文化及傳播型態》，碩士論文，世新大學傳播研究所，2008。

[2] 呂安邦，《植基於智慧卡的數位版權管理系統之研究》，碩士論文，大同大學資訊工程學系，2010。

[3] 范傑翔，《以 XrML 為基礎實現企業數位版權管理系統之權限控管》，碩士論文，大同大學資訊工程學系，2010。

[4] 陳金鈴，〈一種植基於群體導向授權為基礎的企業數位產權管理系統〉，電子商務研究，第 7 卷，第 2 期，頁 133-150，2009 年夏季。

[5] 財團法人台灣醫療改革基金會，〈美國病歷與病人醫療資訊的管理方式-HIPPA 的規定〉，會訊，第 17 期，2006 年 12 月。

[6] 鉅亨網，<美軍 9 萬份文件外洩揭濫殺平民隱瞞情報>，網址：http://news.cnyes.com/Content/20100727/kcajevmab9zra_3.shtml，上網日期：2010 年 12 月 09 日。

[7] 楊佳泰，《以 XrML 為基礎之多媒體數位版權管理機制之研究》，碩士論文，國立中正大學資訊工程研究所，2005。

[8] 蘋果日報，<Selina 病榻慶生，長庚病歷任翻？>，網址：http://tw.nextmedia.com/applenews/article/art_i

d/32926156/IssueID/20101031，上網日期：2010 年 11 月 02 日。

[9] iThome，<調查：2008 年每筆資料外洩成本為 202 美元>，網址：http://www.ithome.com.tw/itadm/article.php?c=53235，上網日期：2010 年 12 月 0 8 日。

[10] Chia-Chen Lin, Shih-Chi Wu, and Po-Hsuan Chiang " Enterprise-oriented Digital Rights Management Mechanism: eDRM," *2009 International Conference on Availability, Reliability and Security*, Fukuoka, 2009.

[11] Condric, L., Dech, D., and Galic, D. “The Importance of Project Office in Matrix Organization,” *8th International Conference on Telecommunications - ConTEL 2005*, Zagreb, 2005.

[12] ContentGuard, XrML... eXtensible rights Markup Language. http://www.xrml.org/index.asp , accessed 2010/12/22.

[13] Wei-Bin Lee and Chien-Ding Lee, *A Cryptographic Key Management Solution For HIPAA Privacy/Security Regulations*, Master’s thesis, Feng Chia University, 2008.

[14] Wikipedia, MOSE Project. http://en.wikipedia.org/wiki/MOSE_Project , accessed 2010/12/12.

[15] Xin Wang, Guillermo Lao, Thomas DeMartini, Hari Reddy, Mai Nguyen, and Edgar Valenzuela "XrML – eXtensible rights Markup Language," *ACM Conference on Computer and Communications Security ,* Washington, DC, USA, 2002.