適用於數位典藏之安全網路資訊資源整合

曹偉駿
大葉大學
資訊管理系
wjtsaur@yahoo.com.tw

林宜進
大葉大學
資訊管理系
kimobbb@yahoo.com.tw

劉經緯
大葉大學
資訊管理系
gin_we@yahoo.com.tw

摘要

本研究提出整合式存取控制機制，除了讓數位典藏機構可享有資料整合後所帶來的便利性，同時也能提供系統管理員權限控管。以提升現行機制的執行及通訊效率，並達到只有合法角色，才能依循安全政策，安全且有效率的存取整合後的網路資訊資源。

在本研究所提出之整合式存取控制機制中，首先系統會依照安全政策設定檔內所含的網路資訊資源的來源位置、格式對照表、角色及權限規則，將使用者的查詢條件依權限規則轉換成多種資訊資源查詢語法，使其能同步對多個資訊資源進行查詢。最後整合不同格式的結果成 XML 格式，並依照使用者權限萃取出查詢結果給使用者。

導入此機制可以讓典藏機構，原本存在不同資料庫或不同網站的內容，利用對照表的方式將原始資料轉換為符合各典藏機構之檔案格式，並依照使用者權限顯示其所能看到的資料範圍。

關鍵字：
數位典藏、存取控制、異質資料整合

1. 前言

隨著數位典藏的史料文件個別數位化後格式眾多，且網路各種資料格式如雨後春筍相繼出現，而運用 XML 的互通性來整合各項資料的議題也越來越受矚目。常見網路資訊資源包括 HTML、XML、Database 和 Web Services 等型態，且網站上所提供資訊資源的查詢方式與權限規則會隨著系統環境及安全政策的不同而有顯著差異，因此容易造成各項網路資源難以綜合運用。另外，整合後的 XML 資料被越權存取或遭有心人士的非法竄改，不只造成資料上的損失，還會使得典藏機構名譽受損。數位典藏環境是多個典藏機構所組成，且其格式與權限需求不一。因此，如何兼顧資料整合的便利性與存取控制是刻不容緩的需求。

隨著數位典藏的史料文件個別數位化後格式眾多，由於網站設計架構上的差異，造成網路資源具有分散、獨立且格式不統一的特性。各種資源的查詢方式、權限規則均有所不同，造成難以同時綜合各項網路資源。目前常見的查詢方式共有四種 [1, 3, 9]，分別為：在網頁查詢時採用表單來查詢所需的資訊，資料庫方面採用 SQL 語法來當做資料庫查詢語法以取得資料集，在 XML 檔案格式方面採用 XQuery 語法進行查詢，以及在 Web Services 採用 SOAP 當通訊格式。以上四種查詢方式不只資料格式相異，而且使用權限的規則會隨著系統環境及安全政策亦有顯著的不同。除此之外，在整合網路系統資源時，還必須同時考慮安全性的整合，達到只有合法角色，才能依安全政策存取整合後的網路資訊資源，讓使用者能安全且有效率的使用整合式網路資訊資源 。

本研究其餘的論文章節如下：第 2 章文獻探討；第 3 章整合式存取控制機制；第 4 章安全性與效能分析；第 5 章系統實作；第 6 章結論與未來發展方向。

2. 文獻探討

為了提出網路資訊資源的整合式存取控制機制，本章首先將探討資訊資源整合之演進過程及其架構，接著將探討現行存取控制機制，以尋求適合本研究之方法。其詳細內容如以下各節。

2.1 現行網路資訊資源的整合架構

網路資訊資源(Networked information resources) 為可經由網際網路上取得之資訊，常見資料格式包括 HTML, XML, Database Result [10, 11, 13]。目前新興之 Web Services 亦透過網際網路上取得之存取資料，所以本研究

增列 Web Services 為整合對象，其通訊協定通常附屬於 HTTP 上執行，並透過 SOAP 當做查詢及回傳資料方式。

網路資訊資源整合的議題，近期學者[2, 10, 12, 13]多採用對照表方式。對照表中記載資料來源位置、來源格式、個別來源格式和中介格式的對應關係。而在 2002 年 Lee 等人以具平台獨立性的 XML 為中介格式，提出 XML-based Mediation Framework(XMF)系統架構[10](如圖 1 所示)，其架構分為 Application Layer、Mediation Layer 及 Resource Layer 架構，後來 2003 年 Yoo 等人 [18] 將 XMF 架構改良成為較有彈性的設計。但是，在需要存取控制的場合，尚有四項不足之處以致無法有效運用，如下所示：

1. 對於不同使用者，無法針對其角色權限，輸出不同的查詢結果。
2. Mediation Layer 在連結 Resource Layer 過程中，常常需要通過身份驗證，而 XMF 架構中並未清楚描述運作流程。
3. 使用者無法驗證整合結果是否來自合法的 Mediation Layer。
4. XMF 提出網路資訊資源整合架構，但並未針對每個步驟提出具體演算法。

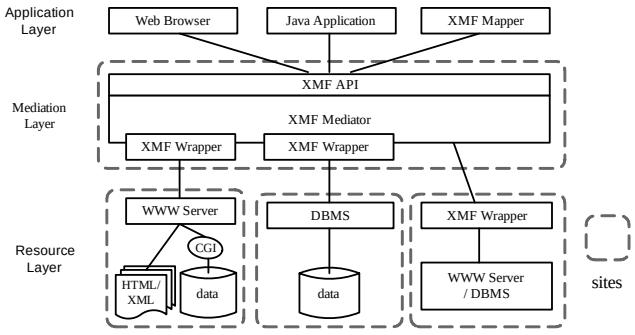


圖 1 XML-based Mediation Framework(XMF)系統架構

2.2 存取控制

Ferraiolo 和 Richard.Kuhn[4]等學者在 1992 年首次提出以角色為基礎的存取控制(Role-Based Access Control, RBAC)，該存取權限控管模組，又經由 Sandhu 等學者加以改進[5]，最後由 National Institute of Standards and Technology (NIST)組職加以整理成為標準，稱為 NIST RBAC[6]，此標準中又以 Limited Hierarchical RBAC 能同時定義角色繼承關係又能限制繼承範圍。另外，2004 年 Jeon [8]利用 XPath 定義出使用者擁有 XML 文件中那些標籤的存取權限。然而，上述研究中權限設定的儲存格式並非採用 XML 表示，所以不只程式解讀不易，而且無法彈性的儲存使用者、角色與權限的關係。另外，Lu [19]提出適用於 XML 環境之 multisignature scheme，以解決多重簽章的問題。再者 Christian [20]亦探討 Web Services 下的安全標準，在 XML 方面的安全性有所貢獻。

綜合上述，首先探討資訊資源整合之演進及其架構，接著探討存取控制。然而，單純合併兩項現有技術，無法有效地針對不同角色權限輸出不同查詢結果，因此本研究提出整合式存取控制機制。

3. 整合式存取控制機制

本研究設計「整合式存取控制機制」，其機制中安全政策設定檔是整個運作流程的安全控管中心，查詢前處理機制是依安全政策設定檔進行權限過濾，其可提升系統運作效能。系統運作時主要依照安全政策設定檔、使用者角色權限及查詢條件，回傳合乎權限的查詢結果。具體作法如本章各小節所示。

3.1 建立安全政策設定檔

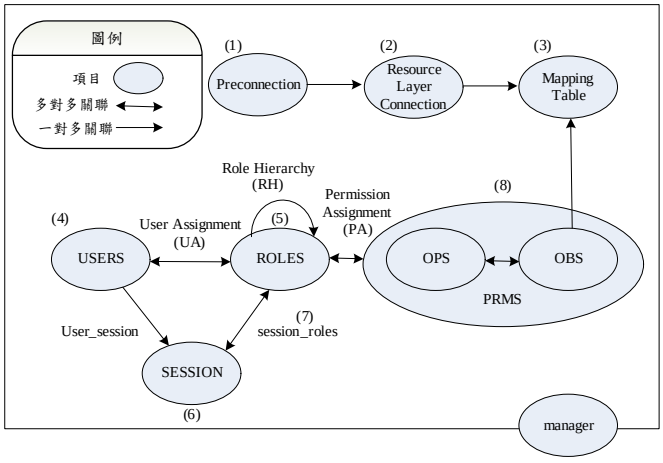


圖 2 安全政策設定檔架構

安全政策設定檔是整個運作流程的安全控管中心，其記載系統運作流程與角色權限關係，以XML檔案格式儲存在Mediation Layer 伺服器。其架構整合Limited Hierarchical RBAC 與對照表觀念，並提出預先連線，以

符合整合式存取控制的XML環境所需。如圖2、表1所示。

```
<policy>
  <preconnection_list>…</preconnection_list>
<connection_list>…</connection_list>
  <map_list>…</map_list>
  <user_list>…</user_list>
  <roles_list>…</roles_list>
  <session_list>…</session_list>
  <session_roles>…</session_roles>
  <PRMS>…</PRMS>
  <manager>…</manager>
</policy>
```

圖 3 安全政策設定檔基本標籤結構

表 1 安全政策設定檔的構成項目

項目	說明
1 Preconnection	XML 標籤名稱：preconnection_list 記載 Resource Layer 的身份認證訊息，以供 Mediation Layer 能通過身份驗證，取得 Resource Layer 的資料。
2 Resource Layer Connection	XML 標籤名稱：connection_list 記載 Resource Layer 的資料型態及取得資料的方法，以供 Mediation Layer 取得 Resource Layer 的資料。
3 Mapping Table	XML 標籤名稱：map_list 記載 Resource Layer 所提供的資料格式及整合後的 XML 資料格式的對照表。
4 USERS	XML 標籤名稱：user_list 記載合法的使用者屬性、身份驗證方式及其具有何種角色。
5 ROLES	XML 標籤名稱：roles_list 記載角色、賦予的權限及 ROLES 間的繼承關係。
6 SESSION	XML 標籤名稱：session_list 記載當使用者登入時，判斷使用者的屬性規則。
7 session_roles	XML 標籤名稱：session_roles 記載使用者正式登入時，如何依 SESSION 賦予不同的角色。
8 PRMS	XML 標籤名稱：permission_list PRMS 記載權限設定，OPS 以 XPath 表示 Mapping Table 中整合後的 XML 資料位置，OBS 記載是否可存取。
9 manager	XML 標籤名稱：manager
	Mediation Layer 記載管理者的身份及設定檔有效期限及 XML 數位簽章，以達到安全政策設定的的完整性及不可否認性。

本研究所支援的網路資訊資源種類如表 2 所示，具體的連線方式將會定義在 preconnection_list 供 Meditation Layer 與 Resource Layer 身份驗證與資料存取。

表 2、本研究之網路資訊資源

資源編號	資訊資源	通訊協定	查詢方式	資料格式
Type 1	WWW	HTTP	Query String	HTML
Type 2	WWW	HTTP	Query String	XML
Type 3	DBMS	JDBC	SQL	Database Result
Type 4	DBMS	JDBC	SQL	XML
Type 5	Web Services	HTTP	SOAP Request	SOAP Response

3.2 查詢前處理機制

查詢前處理目地在事先依照角色計算出其權限有效範圍，未來使用者登入成功並得知所屬角色後，直接得知查詢結果的有效範圍，以加速查詢時驗證標籤權限時間。

一般權限的設定為「全部權限都禁止除非政策所允許、同時允許和禁止的標籤則視為禁止」，假設一份 XML 文件中全部標籤為 ϕ 集合，有權限標籤為 Θ 集合，無權限標籤為 $\triangle$ 集合，其中 $\Theta \in \phi$ 且 $\triangle \in \phi$，則在一般權限定義下使用者真正有效的權限為 $\alpha = \Theta - \triangle$，如圖 4a 所示。

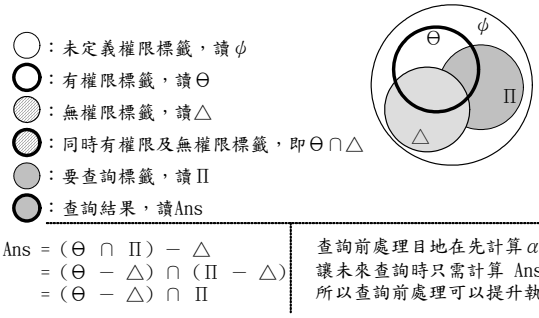


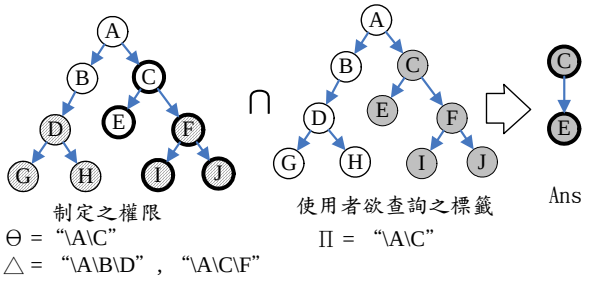
圖 4a 查詢前處理的原理

在安全政策設定檔之 map_list 中，記載 Resource

Layer 資料如何對應給使用者的查詢結果，例「<map dest="gs:/目地位置" source="ls2:/資料來源位置" />」。另外，在安全政策設定檔中 role_list\role 中，記載系統所有的角色及對應的授權，並記載角色間的繼承關係。

然而，在 XML 內容為樹狀結構，直接以 Jeon 的方式儲存權限，無法適用於未知的 XML 資料內容的。本研究改用安全政策設定檔中 map_list\map@desc 求得預期的查詢結果，由預期的查詢結果計算出 α 值，以 XML 查詢前處理的運作概念，如圖 4b 所示。查詢前處理可分四個步驟說明如下：

1. 用 XQuery 的方式找出安全政策設定檔裡來自"policy\map_list\map" 的 XML 標籤，從該標籤屬性 desc 中推斷出預期的查詢結果。
2. 用 XQuery 的方式找出安全政策設定檔裡來自 "policy\role_list\role" 的 XML 標籤，可得知所有現存的角色 *role*。另外 *role.role_id* 為角色編號，*role.father* 為父層角色。
3. 從現存的角色 *role*，依角色的繼承的關係 *role.father* 進行遞迴追蹤，將其有效權限加入該角色中。
4. 由步驟 1 所得知預期的查詢結果及步驟 3 所得知每個角色的權限，再依照圖 4a, 4b的觀念，推斷出每個角色真正具有權限的標籤 α_{role_id}。



查詢前處理目地在先計算
α = (Θ − △) = {A, B, C, E}

讓未來查詢時只需計算
Ans = (α ∩ Π)
= {A, B, C, E} ∩{C, E, F, I, J}
= {C, E}

所以查詢前處理可以提升執行效率

圖 4b 查詢前處理的概念

使用者依照公司的職級的不同，而有不同的存取權限，其常見的角色階層組織如圖 5。在查詢前處理已經會先根據不同角色，計算出每個角色應有之權限，登入後亦得知使用者所屬的角色，所以最後只需根據每個角色的權限，過濾出其能閱讀 XML 標籤，該使用者即可得到權限內所能的查詢結果。

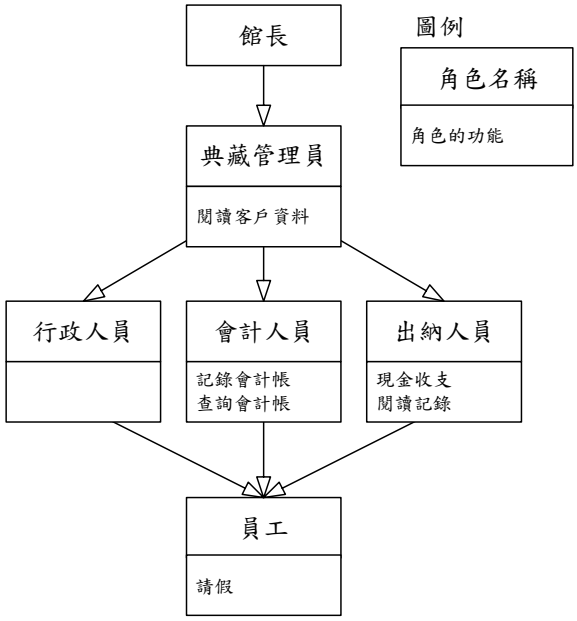


圖 5 角色及查詢權限架構

待建立安全政策設定檔及執行查詢前處理均執行完畢後，系統即開始運作。

3.4 系統運作過程

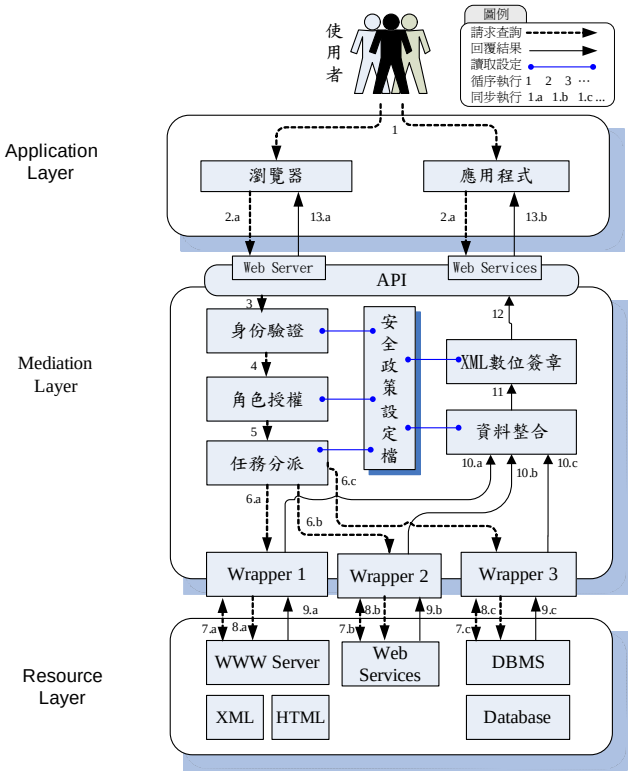


圖 6 整合式存取控制機制

系統運作時主要依照安全政策設定檔、使用者角色權限及查詢條件，回傳合乎權限的查詢結果，Mediation_API 函式負責管理系統運作流程，包括整合式存取控制機制中(參考圖 6)的下列步驟。其演算法如圖 7 所示。

(1) 使用者將身份驗證資訊及查詢需求，輸入至 Application Layer。

(2) Application Layer 呼叫負責資訊資源整合的 Mediation Layer。

(3) Mediation Layer 由呼叫 Authentication 使用者的識別資料確認使用者是否具有合法身份，若為不合法使用者則中止流程。

(4) 呼叫 Authorization 以依照使用者屬性動態賦予角色及角色的授權。

(5) 呼叫 PreProcess 函式以得知使用者查詢時，需要啟動 Resource Layer 中那些資料連線，並依照使用者權限及使用者查詢條件，產生對 Resource Layer 的資料查詢語法。

(6) 依步驟 5 之資料連線及查詢語法用 thread 方式建立個別 XMF Wrapper，讓每個 XMF Wapper 能夠同步執行，不需彼此互相等候以提升效率。

(7) 個別 XMF Wapper 呼叫 PreConnection 函式，同時啟動多個身份驗證的網路連線，以取得每個 Resource Layer 的讀取權限。

(8) 個別 XMF Wapper 同步呼叫 ResourceQuery 函式，將已取得權限的網路連線，發出存取資料的同步請求。

(9) 個別 XMF Wapper 同步接收 Resource Layer 的回傳結果，並且同步呼叫 TransformXML 函數，將 Resource Layer 的執行結果直接轉換成未整合的 XML。

(10) 先執行 join 等待每個由 thread 所構成的 XMF Wapper 都執行完步驟 7,8,9，再呼叫 IntegrationXML，將步驟 9 中每個未整合的 XML ，以對照表方式進行資料整合，回傳為單一 XML 結果。

(11) 呼叫 FilterXML，將無法在步驟 3 中過濾使用者權限或查詢條件的部份，進行再次過濾，以生成合乎權限規則的單一 XML 結果。

(12) Mediation_API 接收步驟 11 的「合乎權限規則的單一 XML 結果」，並以 Mediation Layer 的私鑰對待回傳的內容做 XML 數位簽章

(13) 依使用者所呼叫的方式，分為網站的方式或 Web Services 的方式，將「合乎權限規則的單一 XML 結果」回傳給使用者。

Mediation Layer 的主要函式是 Mediation_API，其負責控制 Mediation Layer 系統運作流程，可分派任務給 Mediation Layer 中其他函式，亦控制系統流程中每個參數的傳輸。

變數定義

user_info：使用者基本屬性，及身份認證訊息

query_fields：使用者查詢所要顯示的欄位

query_where：使用者之查詢的限制條件

XMFWrapper：一個能獨立運作的 XMFWrapper 類別，具有 ResourceLayer 連線能力。其包括三項屬性：*conn* 為 Resource Layer 的連線資訊；*sourceData* 為未整合的資料內容；*sourceXML* 為未整合的 XML

XML-based Algorithm: **Mediation_API**

input:

Access request: (*user_info*, *query_fields*, *query_where*)

output:

targetXML

/*(3)身份驗證，由 *user_id*, *user_info* 識別使用者是否具有合法身份 */

if(**Authentication**(*user_id*, *user_info*) == Reject) Reject access

/*(4)角色授權，依照使用者的屬性，動態取得正確的 role_id。*/

role_id = **Authorization**(*user_id*, *user_info*)

/*(5)工作分派，由 *query_fields*, *query_where* 找出所需之資料連線，並將查詢語法轉換成適合 Resource Layer 的查詢語法(例 SQL 語法)，依照所需之資料連線，產生 XMFWrapper 陣列。*/

wrapper = new **XMFWrapper** Array

wrapper = **PreProcess**(*query_fields*, *query_where*)

index = 0

```
foreach(item in wrapper)
  /*(6)同步執行 thread 區段內的演算法*/
  thread
    item.conn = PreConnection()   /*(7)預先連線*/
    item.sourceData = ResourceQuery(item.conn)   /*(8)資料請求*/
    item.sourceXML = TransformXML(item.sourceData) /*(9)轉換
成 XML*/
  end_thread
endfor
join thread   /*等待全部 thread 區段，每個查詢均取得結果。*/
targetXML = IntegrationXML(wrapper)
  /*(9)整合每個 wrapper 的 sourceXML*/
targetXML = FilterXML(targetXML)   /*(11)確認資料格式*/
targetXML = XMLDigitalSignature (targetXML)   /*(12)XML 數位簽
章*/
return targetXML   /*(13) 回傳結果給使用者*/
```

圖 7 演算法 Mediation_API

4. 安全性與效能分析

我們將針對安全性分析、計算效率、通訊效率、資料存取情況與其他優勢，最後將本機制與先前研究逐一比較，詳細內容如本章各小節。

4.1 安全性分析

本研究經文獻的整理[7,17]，歸納網路常見攻擊手法，並提出其防範與對策，以使本系統能提供安全的操作環境，詳細內容如下：

1. 惡意使用者利用作業系統或網站的漏洞而入侵 Mediation Layer 伺服器，透過竄改安全政策設定檔提升使用者權限，可針對安全政策設定檔嘗試下列手法。
   (1) 竄改使用者資料，賦予使用者具有較高權限的角色，以取得較高權限。
   (2) 竄改使用者屬性，使該使用者未來登入時能動態取得較高授權的角色。
   (3) 新增具有較高權限的使用者，再用新增的使用者登入，以取得較高權限。
   (4) 刪除安全政策設定檔，使得全部使用者均無法正常登入或存取權限。

   此四項提升權限的手法均需修改安全政策設定檔內容。然而 Mediation Layer 的管理者，可透過以下兩個作法中任何一項，即可有效保護安全政策設定檔。亦可同時使用兩個作法，達到多重保護的目地。
   (1) 由於安全政策設定檔，包含 Mediation Layer 管理者對整合安全政策設定檔的數位簽章，而一般使用者無法取得管理者的私鑰，即使修改安全政策設定檔，系統亦可透過比對 XML 數位簽章，得知安全政策設定檔是否遭竄改。另外，安全政策設定檔都有其有效期限，且該有效期限亦被數位簽章保護。如果發現設定檔遭竄改或遭受無效期限內的設定檔覆寫時，均不信任此設定。並由管理者用正確的設定檔更正之，以使系統回復正常運作系統，能有效達到預防入侵的目的。設定方式如圖 8 所示。

```
<policy>
  …安全政策設定檔內容…
  <manager>
    <managerName>王大立</managerName >
    <limtetime>有效期限</limte>
    <sign>Mediation Layer 管理者對整合安全
政策設定檔之數位簽章</sign>
  <manager>
</policy>
```

圖 8 安全政策設定檔之管理者數位簽章

   (2) 將安全政策設定檔放置於安全且唯讀的設備內，其中安全的部份能夠靠加密及設定使用者讀取權限所完成，而唯讀則指無法修改內容的設備如唯讀光碟片。
2. 網路惡意使用者在使用者傳輸資料時，竊聽網路封包，欲得知使用者傳輸內容。

   在 Application 與 Mediation Layer 傳輸資料時，採用加密的方式進行傳輸，例如 Secure sockets layer (SSL)協定進行資料傳輸，亦或在回傳的 *targetXML* 直接用 XML 數位加密，達到傳輸資料的機密性。

3. 網路惡意使用者竄改傳輸資料，讓使用者得到錯誤的結果。或者合法使用者要確認收到結果是否真的來自 Mediation Layer。

由於 Mediation Layer 會針對回傳的 *targetXML*，使用 Mediation Layer 管理者的數位簽章，使用者收到 *targetXML* 時亦可檢驗該數位簽章的正確性，所以能確保回傳資料的完整性與不可否認性，如圖 6 的第 12 步驟所示。

4.2 計算效率之分析

以下列出兩個假設條件：

(1)假設 XMF 機制將兩個異質資料整合成一個 XML 文件需要時間為 t1。

(2)假設 XPath 權限控制將一份 XML 文件，依權限過濾結果需要時間為 t2。

依據 Yoo[18]的作法，若有 n 份文件要整合，而使用者擁有 m 份的權限，其中 $0 \le m \le n$，則單純整合 XML 後使用 Jeon 的方法[8]進行處理權限，需要兩兩合併資料來源，最後再執行一次權限控管機制，故需要時間

$$timeA = \sum_{i=0}^{\lceil \log_2 n-1 \rceil} 2^i \times t1 + t2 = \left(2^{\lceil \log_2 n \rceil} - 1\right) \times t1 + t2 .$$

本機制的前處理像是一個過濾器，將沒有權限的查詢事先過濾，可以節省資料合併的時間，故本機制需要時間

$$timeB = \sum_{i=0}^{\lceil \log_2 m-1 \rceil} 2^i \times t1 + t2 = \left(2^{\lceil \log_2 m \rceil} - 1\right) \times t1 + t2 .$$

由此可知，本機制所提升的效能為

$$timeC = timeA - timeB$$

$$= \left(\left(2^{\lceil \log_2 n \rceil} - 1\right) \times t1 + t2\right) - \left(\left(2^{\lceil \log_2 m \rceil} - 1\right) \times t1 + t2\right)$$

$$= \left(2^{\lceil \log_2 n \rceil} - 2^{\lceil \log_2 m \rceil}\right) \times t1 .$$

$\because n \ge m, t1 > 0$

$\therefore timeC \ge 0$

故本機制效率較原機制佳約 *timeC*。

4.3 通訊效率之分析

若有n份文件要整合，使用者具其中m份的權限，且 $n \ge 2, 0 \le m \le n$，假設 XMF機制中查詢來源資料需傳輸資料量d_1，其中$d_1 > 0$。若查詢經權限限制的內容，其傳輸資料量為d_2，其中$d_2 \le d_1$。若未採用本研究所提之前處理機制，則整合的傳輸量 $dataA = n \times d_1$，但若採用前處理機制則傳輸量 $dataB = m \times d_2$。因為 $n \ge m$ and $d_1 \ge d_2$，所以 $dataB \le dataA$，故與「直接合併Yoo與Jeon的方法」相比時，本機制之傳輸效率較高。

4.4 資料整合與存取控制功能分析

要同時達到資料整合與存取控制的目標，若單純利用 Yoo 的方法 [18]完成資料整合，再使用 Jeon 的機制 [8]來完成存取控制，則會面臨執行效率不彰且網路通訊量大的問題，而本研究提出以查詢前處理的方式以解決前述問題。

Jeon [8]用 XPath 來表示使用者權限時，只提供使用者為基礎的權限設定，不適用大量使用者的管理。因此，本機制將其延伸為角色為基礎的存取控制，以方便使用者管理及權限分配。

Resource Layer 的資訊資源擴充，乃指當日後有新資料格式被制定出來，系統可經過擴充模組即與新資料格式相容。由於本機制之資料格式以 XML 為基礎，所以不論其資料來源的作業平台為何，只要建立傳輸通道，均能很容易解讀，亦即本機制具備跨平台獨立性。

綜合上述，本機制與現有機制作一比較，如表 3 所示。

表 3 本機制與現有機制比較分析

比較項目	Yoo 方法 [18]	Jeon 方法 [8]	直接合併 Yoo[18] 與 Jeon [8]方法	本機制
1.資料整合	○	╳	○	○
2.存取控制	╳	○	○	○
3.計算成本	中	中	中	低
4.通訊成本	中	無通訊	中	低
5.角色存取控制	╳	╳	╳	○

6.資訊資源擴充	○	╳	○	○
7.跨平台獨立性	○	○	○	○
8.支援 Web Services	╳	╳	╳	○
9.資料完整性	╳	╳	╳	○
10.不可否任性	╳	╳	╳	○
11.安全政策設定檔之防竄改	╳	╳	╳	○

註 1：○表示有該項功能，╳表示無該項功能。

註 2：直接合併 Yoo [18]與 Jeon [8]方法，乃指先用 Yoo 所改良之 XMF 方法進行資料整合，再交由 Jeon 所提之存取控制機制進行權限處理。

由表 3 得知，Yoo 方法 [18]雖資料整合，但未考慮安全性，而 Jeon 方法 [8]雖考慮存取控制，但未考慮整合。即使直接合併 Yoo[18]與 Jeon [8]方法只能達到低效率的資料整合及存取控制，但本機制卻可做到高效率的資料整合及存取控制

5. 系統實作

圖 9 系統管理者後台介面

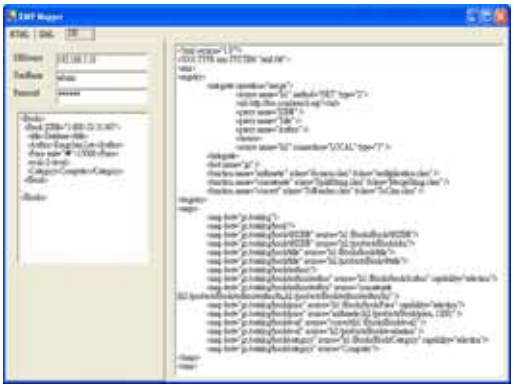


圖 10 系統管理員設定安全政策設定檔

圖 11 使用者查詢後呈現之畫面

在系統實作方面，共包含 Application Layer、Mediation Layer 及 Resource Layer，其中安全政策設定檔貫穿整個系統，主要採用 Microsoft Visual .NET 及 SQL Server ，

Tomcat 網站伺服器共同進行開發，圖 9 為 Mediation Layer 管理程式，圖 10 為編輯安全政策設定檔所內含的網路資訊資源之來源位置、格式對照表、角色及權限設定。由使用者輸入查詢條件，經由角色權限計算求得所「合乎權限規則的單一 XML 結果」如圖 11 所示。

6. 結論與未來發展方向

本研究提出以基於 XML 之網路資訊資源的整合式存取控制機制，達到兼顧資料整合的便利性與存取控制，並具體設計四項成果。首先設計出安全政策設定檔以記載系統運作流程與安全政策，再者設計出查詢前處理機制提升其運作效率，進而設計出整合式存取控制機制的運作流程，最後對本機制之安全性及效能進行分析。達到只有合法角色，才能依循安全政策來存取整合後的資訊，讓使用者能安全且有效率的利用網路資訊資源。

本機制解決原始 XMF 機制的四項存取控制缺失，首先對於不同使用者，能針對其角色權限，輸出不同的查詢結果。再者 Mediation Layer 在連結 Resource Layer 過程中，加入通過身份驗證功能，進而使用者可以驗證整合結果是否來自合法的 Mediation Layer，最後提出網路資訊資源整合架構，並且以演算法描述執行步驟。

在數位典藏中有眾多不同系統使用者及各種不同格式的資料，且員工因職位不同而有階層關係，存取權限亦不相同。本機制除了讓使用者可享有資料整合後所帶來的便利性，同時也提供系統管理員權限控管，因此本機制兼顧資料整合的便利性與存取控制，本機制可於跨組織間的資訊資源整合、網路資料搜尋、聯合資料庫的建立等議題上提供安全且有效率的存取控制機制。

未來本機制更可結合其他安全機制，例如：先進行單一登入[7]及加密簽章 [14-16]，之後再執行本機制以提升更高的安全性。另外，本整合式架構也可於執行本機制之後，加入線上付款機制，讓本研究可更進一步應用於電子商務環境中，以提供數位典藏更完整的解決方案。

致謝

本研究接受國科會研究計畫案 NSC 94-2422-H-212 -001, NSC 93-2622-E-212-005-CC3 資助，特此致謝。

參考文獻

[1] 林玉凡，“EB 前瞻產品-企業應用程式整合(EAI)產品之探討”，資策會電子商務應用推廣中心-FIND 研究群， 2001。

[2] 陳光明，“企業資訊入口網站設計之研究”，國立交通大學資訊管理所碩士論文，2002。

[3] F. Casati, M. Fugini and I. Mirbel , “An environment for designing exceptions in workflows,” *Information Systems*, Vol. 24, pp. 255-273, 1999.

[4] D. Ferraiolo and R. Kuhn, “Role-based access control”, *15th NIST-NCSC National Computer Security Conference*, 1992.

[5] D. Ferraiolo, J. Barkley, and D. Richard Kuhn, “A role-based access control model and reference implementation within a corporate Intranet”, *ACM Transactions on Information and System Security*, Vol. 2, pp. 34-64, 1999.

[6] D. Ferraiolo, R. Sandhu, S. Gavrila, D. Richard Kuhn and Ramaswamy Chandramouli,"Proposed NIST standard for role-based access control”, *ACM Transactions on Information and System Security*, Vol. 4, pp. 224-274, 2001.

[7] A. Harbitter and D. Menasce, “Performance of public-key-enabled Kerberos authentication in large networks,” *Proceedings of the IEEE Int. IEEE Symposium on Security and Privacy*, pp. 170-183, 2001.

[8] J. Jeon , “Filtering XPath expressions for XML access control,” *Computers & Security* , Vol. 23, pp. 591-605, 2004.

[9] P. Johannesson, B. Wangler and P. Jayaweera, “Application and process integration -concepts, issues, and research directions,” *Swedish National Board for Industrial and Technical Development*, 2001.

[10] K. Lee, J. Min, K. Park, “A design and implementation of XML-based mediation framework(XMF) for integration of internet information resources,” *Proceedings of the 35th Hawaii International Conference on System Sciences*, pp. 20-33, 2002.

[11] S. Osborn, “Integrating role graphs: a tool for security integration,” *Data & Knowledge Engineering*, Vol. 43, pp. 317-333, 2002.

[12] H. Theo, S. Gunter and T. Joachim, “The intrinsic problems of structural heterogeneity and an approach to their solution,” *The VLDB Journal*, Vol. 8, pp. 25-43, 1999.

[13] A. Wohrer, P. Brezany and I. Janciak, “Virtualization of heterogeneous data sources for grid information systems,” *Institute for Software Science University of Vienna*, 2004.

[14] W3C Extensible Markup Language (XML) ,<http://www.w3.org/XML/>, 2004.

[15] OASIS Security Services (SAML), <http://www.oasis-open.org/ committees/security>, 2004.

[16] OASIS Security Services (SAML), <http://www.oasis-open.org/ committees/ download.php/1371/oasis-sstc-saml-ocre-1.0.pdf>, 2001.

[17] XML Key Management Specification (XKMS), <http://www.w3.org/ TR/xkms/>, 2001.

[18] S. Yoo, K. Lee, and K. Lee, ”An XML-based mediation framework for seamless access to heterogeneous internet resources,” *Lecture Notes in Computer Science*, Vol. 797, pp.396-405, 2003.

[19] E. J. Lu, R. Chen, “An XML multisignature scheme,” *Applied Mathematics and Computation*, Vol 149, pp. 1-14, 2004.

[20] G. P. Christian, C. Joris, “Web services and web service security standards,” *Information Security Technical Report*, Vol 10, pp. 15-24, 2005.