應用XACML架構達到安全的雲端資料分享——以政府機關私有雲為例

陳昱仁[1] 廖耕億[1*] 吳孟哲[2]
[1]長庚大學資訊管理學系 [2]長庚大學商管專業學院 [*]通訊作者
cyr@mail.cgu.edu.tw gyliao@acm.org ache2019@gmail.com

摘要

隨著網路速度越來越快及資訊產業的蓬勃發展，造就了雲端運算 (Cloud Computing) 的風潮，因此許多資訊大廠如微軟、惠普、IBM及Google等開始提供雲端運算相關服務，並鼓吹雲端所帶來的效益。除了企業開始推廣雲端運算外，各國政府也紛紛投入資源，希望提升該國的雲端產業，因此我國政府也開始推動雲端運算產業，並積極推出不同功能的「政府雲」，提供對個人、企業及政府更便利的服務。雲端服務雖然帶給使用者方便性，但也衍生出雲端服務的安全疑慮，如何在使用雲端服務所帶來價值，並兼顧到資訊安全即是一個很重要的議題。本研究擬針對政府機關在異質私有雲資料交換安全問題，提出一安全雲端資料分享機制，該機制應用網路服務 (Web Services) 來達到異質私有雲間資料傳遞，且使用加密演算法加密資料來確保傳遞過程中資料的機密性，並透過可擴展存取控制標記語言 (eXtensible Access Control Markup Language, XACML) 其安全存取及權限控管的特性，針對網路服務在資料交換的過程中身分驗證的目的，以達到異質私有雲資料交換的即時性及安全性。

關鍵詞：雲端運算、網路服務、可擴展存取控制標記語言、私有雲

壹、緒論

近年來雲端運算 (Cloud Computing) 的崛起，全球的科技業者都陸陸續續發展出相關的創新應用，如中華電信的hicloud CaaS、Google的Apps for Business及Microsoft的Windows Azure等都是目前雲端運算下的當紅產物。雲端運算之所以受到重視，網路的普及、頻寬的提升、虛擬化、Web 2.0和即時通訊等技術成熟，都為雲端運算帶來極大的助力。除了科技業者外，各國政府也積極投入雲端運算應用與發展，我國政府也開始打造「政府雲」，希望透過雲端運算服務，提供對個人、企業及政府機關更便利的服務，並協助相關產業的轉型與發展。

雲端服務為人們帶來了便利的開端，但也衍生出一些問題，尤其在資訊安全部分更被重視，雲端安全聯盟 (Cloud Security Alliance, CSA) 在2010年3月初發布了一份研究報告，標題是 「Top Threats to Cloud Computing V1.0」，列出了目前雲端運算所遭遇的七大安全威脅。使用者必須依據自身所處的環境決定這些威脅的影響並採取適當的控制措施。這些威脅包括：1.濫用或利用雲端運算進行非法的行為；2.不安全的介面與API；3.惡意的內部人員；4.共享環境所造成的議題；5.資料遺失或外洩；6.帳號或服務被竊取；

7.未知的風險模型。經由上述七點威脅可以知道，越是便利的技術和服務，越有可能因為疏失或無意中造成資訊安全的隱憂及漏洞，因此，如何在雲端運算中提升資訊安全防護，又可不失雲端運算的便利性，即是本論文研究的重點。

中央健康保險署在2013年開始實施二代健保，這項新的措施不只單單影響個人，包含雇主也需要依規定繳納補充保險費，以政府機關實際作業狀況為例，新措施影響最大的就是薪資業務承辦人員，因為薪資承辦人員除了要在薪資系統內彙算雇主及員工個人應繳納之補充保險費外，還要將繳納明細資料轉換成中央健保署提供格式檔案，透過中央健保署提供網站將繳納明細上傳才算完成，這樣一來除了增加承辦人員工作負擔外，申報資料無法透過系統就源傳送至健保署也有可能導致資料遭到竄改的風險。若承辦人員只需透過機關「薪資雲」以安全方式將相關申報媒體資料傳送至中央健康保險署的「健保雲」，這樣一來便可透過雲端技術達到政府預算節流及節能減碳，也可藉由雲端應用帶動政府的流程精簡與再造。

在政府所打造的「政府雲」中，往往需要取得其他雲端服務資料整合，來達到雲端服務更高的效益，然而，雲端資料交換的過程中存在著許多安全上的風險，因此，本研究應用一種可擴展存取控制標記語言 (eXtensible Access Control Markup Language, XACML)，並且使用加密演算法將資料進行加密，針對不同的雲端服務在可信賴的前提下，安全的分享雲端中的資料，本研究主要目的如下：

1. 提供跨雲端服務資料分享機制
2. 提供雲端資料安全存取機制
3. 提供雲端資料分享授權機制

貳、相關文獻與技術

一、網路服務 (Web Services)

Web Services是一種以XML為基礎，其主要目的是用於在異質的應用系統中，可以透過網際網路在不同平台及不同程式語言，來達到相互操作或溝通的服務。Web Services技術，解決了原本異質系統在整合上所面臨的問題。

吳信輝[11]說明了Web Services的執行模式：「Web Services的基礎包括：XML、WSDL、SOAP、UDDI，其底層運作架構模式是以XML格式為基準將資料轉變為Web Services的資料，利用WSDL描述將服務的對象做一個描述，使另一端可以透過這一個描述，解譯所得的資料。以SOAP通訊底層，進行傳送的動作，向UDDI進行搜尋或是註冊動作。」蔡煥麟[16]指出，將可延伸標記式語言 (eXtensible Markup Language, XML)、簡單物件存取協定 (Simple Object Access Protocol, SOAP)、網路服務描述語言 (Web Services Description Language, WSDL)，網路服務目錄註冊服務 (Universal Description Discovery and Integration, UDDI) 這些核心元素組合起來，就可以形成一個 Web Services 架構，架構中包含了三種主要的角色，分別是Web Services的服務提供者 (Service Provider)、服務消費者 (Service Consumer)，與介於兩者之間的服務中介者 (Service

Broker)。Web Services可以透過網際網路提供很多功能，但還是會遭遇到下列的問題：

1. 資料傳輸安全問題：透過安全超文字傳輸協定 (Secure Hypertext Transfer Protocol, HTTPS) 的加密連線功能，可以讓資料傳輸過程中受到保護，但如果不透過HTTPS時，資料在傳輸過程中就有可能會暴露在危險當中。
2. 身分驗證問題：使用Web Services無法透過身分驗證來確認資料來源的正確性，也無法依身分給予適當授權，容易造成資安上的隱憂。

二、雲端運算 (Cloud Computing)

雲端運算是一種透過網際網路來達到軟體及硬體資源共享的概念，早在1980年就已經有類似的作法，近年來因為網路頻寬越來越大及虛擬化技術快速成長的情形下，以及雲端運算的特性，讓雲端運算迅速的竄起。雲端運算也被稱為「網路運算」，雲端運算是一種概念、一種服務，並非是獨特的技術，其本質源自於「分散式運算」以及「網格運算」。而雲端服務主要的用意是在資源有限或有效利用的情形下，使用者可透過網際網路取得相關資料計算與資料儲存的服務。以使用者觀點來看則是可以隨時依個人需求在網路上使用雲端相關服務，而且無需考慮維護、管理這些成本及負擔。根據國際研究機構Gartner的定義:「雲端運算是指透過網路相關技術，對使用者提供具彈性的科技化服務」。

行政院於2010年4月開始推動雲端基礎建設、平台和服務，並請行政院研究發展考核委員會負責規劃政府雲，目標是希望透過雲端服務能達到「行動便民」，未來民眾不管到哪裡，透過單一窗口，隨時可獲得政府服務。依據張念慈、彭秀琴[15]的研究指出，我國政府目前也致力於雲端運算的發展，下面列出針對雲端運算，政府機關適合的相關應用：

1. 電子化e政府平台。
2. 政府機關網路設備共構機房。
3. 政府機關可開發雲端共用系統。

三、可擴展存取控制標記語言 (XACML)

可擴展存取控制標記語言 (XACML)[7]是一種以XML為基礎的安全性存取權限控制語言並應用於網際網路上，XACML可用於設定安全管理政策及控管存取權限。XACML是由結構化資訊標準推廣組織 (Organization for the Advancement of Structured Information Standards, OASIS)，整合相關研究成果所訂定的標準化安全存取控制規格。XACML是OASIS在2003年所提出的，因此國內外有許多學者開始針對XACML的優點及特性進行研究，通常研究的範圍大多在安全存取及權限控管的角度上，本研究整理成表一。

表一：XACML運用/研究彙總表

年代	作者	應用/研究說明
2002	俞正宏 [12]	採用 XML/XACML 的規格來制定授權管制政策，並提出一套動態工作流程授權管制系統。
2005	薛承文 [17]	利用 XACML 來達成電子病歷精細度 (Fine-grant) 的存取控制。
2006	梁士杰 [14]	利用 XACML 標準來定義家用網路安全策略並達成家用網路存取控制之目的。
2010	Chou and Huang [3]	提出一個擴展 XACML 模型 (EXACML)，使其更適合用於 Web Services 上，確保 Web Services 存取安全。
2010	Hamlen, Kantarcioglu, Khan and Thuraisingham [5]	運用 XACML 應用於 Hadoop 來解決雲端安全的問題。
2010	Lakshminarayanan [6]	介紹 Web Services-Security 和雲端服務，並建立彈性的 Web Services 安全架構。
2012	Ayed and Teraoka [1]	擴展 XACML 模式，提出一個多網域存取控制基礎架構於雲端服務。
2012	Dinh, Wang and Datta [4]	運用 XACML 於雲端服務，並將 XACML 擴展靈活的存取控制決策和數據存取。
2012	朱凱弘 [10]	將 XACML 應用於工作流程的簽核，並使用 Web Services 來達到異質平台間的資料傳遞。
2013	Ayoubi, Mourad, Otrok and Shahin [2]	將 XACML 與 BPEL 整合並實行於 Web Services，透過新的整合方式來達到 Web Services 的安全問題。

XACML的優點除了擁有高擴充性外，也因為其一致性的決策描述語言，避免了在網際網路中面臨不同應用環境的問題。XACML可提供管理人員針對需求自行設定存取控制權限，以便確認取得所需要的資料。另外，XACML也提供請求授權功能，用於當資料於傳輸請求時，XACML將請求中的資訊與已確定的授權規則資訊進行比較，最後回覆允許或拒絕訊息。Tim Moses[8]以Data-Flow Model及Policy Language Model來描述XACML之架構。依據陳威仁[13]的研究指出，XACML通訊協定主要規範了請求訊息及回應訊息的規格，其說明如下：

1. XACML請求訊息主要規格
   (1) Subject：請求存取的個體。
   (2) Resource：欲存取的資料或資源。
   (3) Action：請求者對資源所要執行的動作。
2. XACML回應訊息主要規格
   (1) Permit：允許存取請求。
   (2) Permit with Obligations：允許存取請求，但需依照Policy訂定的義務進行後續作業。

(3) Not Applicable：不允許存取請求，因為PDP中無任何授權決策適用於本次授權請求。

(4) Indeterminate：不允許存取請求，因為其他例外狀況，例如有兩個互相矛盾的授權決策或其他原因，造成PDP無法評估授權請求。

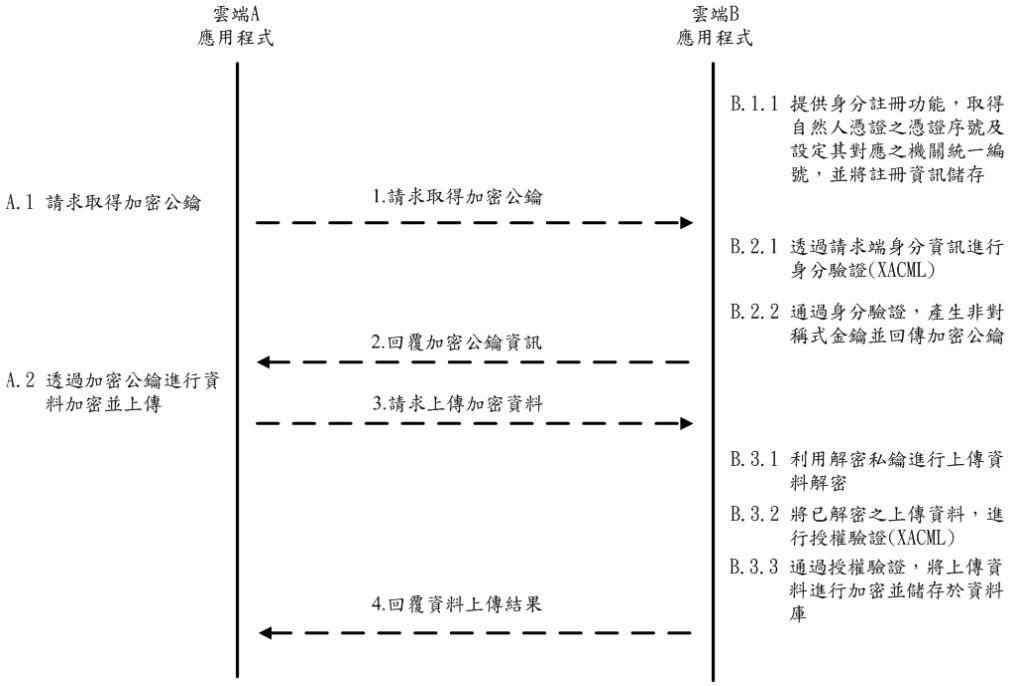
蔣仲翔[18]的研究指出，在使用XACML來做為安全存取及權限控管的應用之前，需先維護授權決策資料檔，XACML會透過授權決策資料檔來做為存取控管的驗證條件，並將驗證結果回傳回應訊息給請求者。XACML的Policy Language Model包括Rule、Policy及Policy Set等項目。

參、安全雲端資料分享機制設計

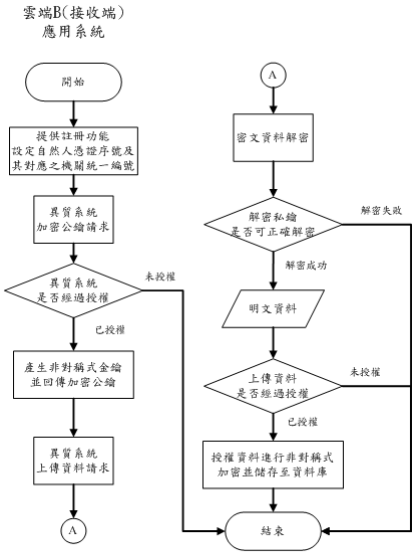
本研究擬針對雲端運算的安全性與機密性，提出一個完善的雲端運算保護機制，該機制涵蓋整個傳輸與接收過程，並確保資料是安全且資料機密是受到保護的。本研究所提出架構中的所有角色說明如下：

1. 傳送端資料庫
   - 由傳送端應用程式從傳送端資料庫讀取出欲傳送資料。
2. 傳送端應用程式
   - 向接送端應用程式取得加密公鑰。
   - 將傳送端資料庫讀取出的資料進行加密加以保護。
3. Web Services
   - 提供傳送端應用程式及接收端應用程式傳輸加密公鑰及資料管道。
4. 接收端應用程式
   - 提供傳送端應用程式使用者於接收端應用程式利用自然人憑證註冊功能，並設定其對應之機關統一編號。
   - 驗證欲取得加密公鑰的傳送端應用程式是否經過授權，並回傳加密公鑰給被授權的傳送端應用程式。
   - 將傳送端應用程式所傳送的資料利用解密私鑰進行解密，並驗證欲存取資料的使用者資訊。
5. 接收端資料庫
   - 將接收端應用程式已解密及通過驗證的資料內容，透過加密方式儲存至資料庫。

本研究所設計的安全雲端資料分享機制可以區分成幾個不同步驟，依序分別是身分註冊、身分驗證、資料加密及資料傳送、資料接收及資料解密、權限驗證、資料儲存與回覆訊息傳送，圖一是在異質雲端系統中，安全雲端資料分享機制訊息流程圖；而圖二則是接收端系統流程圖。



圖一：安全雲端資料分享機制訊息流程圖



圖二：安全雲端資料分享機制接收端系統流程圖

一、 身分註冊

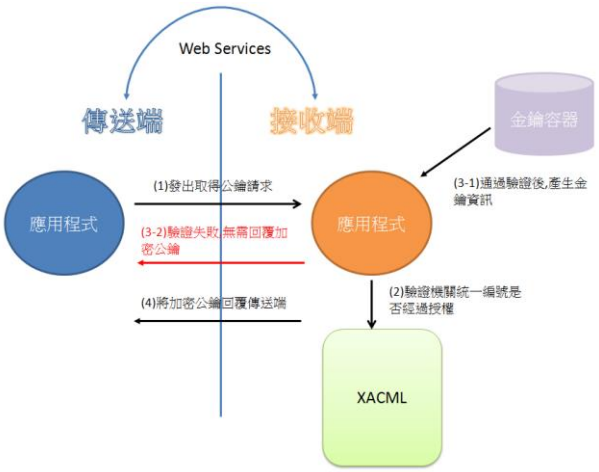
為了使傳送端應用程式可以透過特定資訊來達到身分辨識的目的，所以使用者必須在接收端應用程式進行身分註冊，身分註冊需透過自然人憑證取得憑證序號，並設定其對應之機關統一編號，透過上述設定，來達到後續身分驗證及權限驗證等安全機制。

二、身分驗證

在將傳送資料加密前，首先要取得接收端應用程式所提供的加密公鑰，因此，當接收端應用程式收到傳送端應用程式取得加密公鑰請求時，會進一步驗證傳送端應用程式使用者是否經過授權。步驟如圖三所示，步驟說明如下：

1. 傳送端應用程式呼叫接收端應用程式的 Web Services，確認接收端應用程式的 Web Services 存在後，開始發出取得加密公鑰請求並傳送機關統一編號至接收端應用程式。
2. 接收端應用程式接收到請求後，透過 XACML 來驗證傳送端的機關統一編號是否經過授權。
   (1) 接收端應用程式接收到請求後，需轉換成 XACML 的請求格式，XACML 透過已維護的授權決策資料檔與請求內容進行驗證，最後回覆授權結果。XACML 在接收到請求內容後，會先取得其對應的 Policy，並依請求內容找到其對應的驗證項目「加密公鑰授權驗證 (Resource)」，最後驗證「機關統一編號 (Subject)」及「取得 (Action)」的值是否相符，藉此達到身分驗證的作用。
   (2) 經過授權的機關統一編號，接收端應用程式產生加解密金鑰及保存方式如下：
   - 在安全保護加解密金鑰前提下，接收端應用程式產生加密公鑰及解密私鑰，並將金鑰資訊儲存至金鑰容器中。
   - 傳送端應用程式回覆加密公鑰資訊給接收端應用程式。

   (3) 未經過授權的機關統一編號，傳送端應用程式回覆空值給接收端應用程式。



圖三：身分驗證流程

三、資料加密及資料傳送

在資料加密的過程中，最主要的工作是將傳送端應用程式欲傳送的資料利用接收端所提供的加密公鑰將資料進行加密，最後透過 Web Services 將資料傳送到接收端應用程式，完成雲端與雲端間的資料傳遞，步驟說明如下：

1. 產生欲傳送的原始資料。
2. 取得從接收端應用程式提供的加密公鑰。
3. 產生機關統一編號及自然人憑證序號等驗證資訊。
4. 為了避免在傳送的過程中資料遭到竊取造成資料外流，所以在資料傳送前，透過已收到的加密公鑰先將欲傳送資料及驗證資訊進行加密保護，其中，加密資料未包含機關統一編號。
5. 傳送端應用程式呼叫接收端應用程式的 Web Services，並確認接收端應用程式的 Web Services 存在，開始將傳送端應用程式資料及加密公鑰傳送至接收端應用程式。

四、資料接收及資料解密

當接收端應用程式接收到資料時，先使用解密私鑰將受保護的資料解密，步驟說明如下：

1. 將從傳送端應用程式回傳的加密資料，依機關統一編號找出其對應之金鑰容器，並利用解密私鑰進行解密動作。
2. 當加密資料已完成解密，接收端應用系統便將機關統一編號對應之金鑰容器資訊清除，因此，每次資料上傳，皆需重新申請加密公鑰。
3. 當加密資料無法正確解密時，表示該傳送端的請求可能有問題，此時接收端應用程式便回覆驗證失敗訊息給傳送端應用程式。

五、權限驗證

當接收端應用程式已將資料完成解密之後，便開始進行權限驗證，透過 XACML 來確認該請求是否經過授權，甚至是否有存取特定功能的權限。步驟說明如下：

1. 將已解密的識別資料利用 XACML 來驗證，再將請求資訊轉換成 XACML 格式，最後 XACML 依接收端應用程式事前準備好的 Policy 檔案，來進行驗證動作。XACML 在接收到請求內容後，會先取得其對應的 Policy，並依請求內容找到其對應的驗證項目「資料授權驗證 (Resource)」，最後驗證「自然人憑證序號 (Subject)」及「資料上傳 (Action)」的值是否相符，判斷請求者身分是否有執行權限，達到權限控管效果。
2. 透過 XACML 來驗證請求是否經過授權及存取權限，其結果為以下兩種：
   (1) 通過接收端應用程式驗證，表示資料來源可靠。
   (2) 未通過接收端應用程式驗證，表示可能接收到來源不明資料，並傳送驗證失敗訊息回覆給傳送端應用程式。

六、資料儲存

接收端應用程式將已授權資料進行加密，完成後的密文資料儲存於接收端資料庫。步驟說明如下：

1. 接收端應用程式將已授權資料進行加密並將密文傳送至接收端資料庫。
2. 接收端資料庫進行資料儲存，並回覆接收端應用程式資料儲存結果。

七、回覆訊息傳送

接收端應用程式將資料是否完成儲存之訊息，透過 Web Services 回覆傳送端應用程式。步驟說明如下：

1. 接收端應用程式回覆訊息傳送至傳送端應用程式。
2. 傳送端應用程式顯示接收到的回覆訊息。

肆、個案系統實作

中央健康保險署[9]為穩固健保財源使健保永續經營，以確保國人健保就醫權益，立法院審議通過的二代健保法，業經總統 2011 年 1 月 26 日公布，行政院發布自 2013 年 1 月 1 日實施。由於二代健保與一代健保申報制度差異甚大，增加了企業、政府機關等業務承辦人員負擔，例如二代健保申報媒體檔資料需每月（或每年）透過中央健康保險署網站上傳，業務承辦人員無法透過一套系統就源完成申報媒體檔傳送，除了業務承辦人員需額外學習一套系統操作方式造成困擾外，也有可能因為申報媒體檔案內容遭到竄改而上傳了錯誤資料，也因為申報媒體檔案內容夾帶個資相關資料，若資料不小心遭竊取，那後果將不堪設想。因此本研究將假設「健保雲」可接收二代健保申報媒體資料，經由政府機關私有雲「薪資管理系統」將二代健保申報明細資料，透過本研究方法，把申報資料安全的傳送至「健保雲」來達到媒體申報的目的，以減輕系統操作人員負擔並確保資料正確性。

由於本研究無足夠的硬體設備可實際建構出兩個私有雲環境，因此利用兩台不同主機來模擬成私有雲環境，在傳送端及接收端主機上安裝資料庫系統，傳送端可將欲傳送資料從傳送端資料庫讀出，而接收端主機可將驗證正確的接收資料儲存於接收端資料庫。在接收端及傳送端應用程式部分，將於兩台主機上各建置一個網頁應用程式來模擬資料的傳送端及接收端，傳送端的網頁應用程式會將欲傳遞的資料透過接收端的加密公鑰加密後進行傳送，再由傳送端的網頁應用程式將透過 Web Services 把資訊傳遞至接收端的網頁應用程式，由接收端的網頁應用程式在接收資料時利用解密私鑰解密並透過 XACML 驗證權限，且將完成驗證資料儲存至資料庫。

本研究針對第參章所提出的安全雲端資料分享機制來進行個案系統實作，模擬政府機關私有雲「薪資管理系統」（傳送端）將傳送某年月二代健保申報資料至健保雲（接收端）。首先使用者於登入薪資管理系統後，查詢出欲傳送的二代健保申報資料，查詢結果畫面如圖四所示。

圖四：二代健保資料查詢結果

當使用者完成勾選欲上傳資料後，點選資料上傳按鈕，薪資系統即透過 Web Services 發出取得加密公鑰請求至健保雲系統，接收到請求的健保雲系統便將傳送過來的統一編號來產生 XACML Request（如圖五），來確認該統一編號是否允許取得加密公鑰。其中定義了請求個體為統一編號 (12345678, Subject)，請求資源為取得公鑰 (RequestPublicKey, Resource)，而存取的指令則是取得 (Get, Action)。

```
<Request>
  <Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
    DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>12345678</AttributeValue>
    </Attribute>
  </Subject>
  <Resource>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
    DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>RequestPublicKey</AttributeValue>
    </Attribute>
  </Resource>
  <Action>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
    DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>Get</AttributeValue>
    </Attribute>
  </Action>
  <Environment>
  </Environment>
</Request>
```

圖五：請求取得加密公鑰 XACML Request

在 XACML 接收到請求時，便由 XACML Policy 進行權限評估的動作，圖六為 XACML Policy 的檔案內容，其中定義了該 Policy 是用於取得加密公鑰 RequestPublicKey (Resource)，且識別代號必須為 12345678、A2345678、A9345678 或 D8345678 (Subject)，以及執行的動作為 Get (Action)，全部的條件皆符合時，才會回覆驗證成功訊息。

圖六：驗證取得加密公鑰 XACML Policy

圖七是 XACML 驗證後所回覆的內容，其中若 Decision 的值為 Permit，代表驗證結果為允許，若驗證結果為不允許時，Decision 的值即為 Deny。

```
<Response>
  <Result ResourceId="RequestPublicKey">
    <Decision>Permit</Decision>
    <Status>
        <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
```

圖七：驗證取得加密公鑰權限的 XACML Response

確認驗證結果為允許後，健保雲系統即產生加密公鑰及解密私鑰，並將金鑰資訊儲存於金鑰容器中，完成後便將加密公鑰透過 Web Services 回傳至請求端的薪資系統。當請求端的薪資系統接收到健保雲系統回傳的加密公鑰時，便開始將上傳資料加密，包括機關代碼、上傳者資訊及二代健保申報媒體資料，加密完成後再透過 Web Services 將資料送至健保雲系統。

健保雲系統接收到資料後，首先利用解密私鑰將加密資料進行解密，若解密失敗代表上傳資料有問題，並回覆解密失敗訊息，若解密成功，健保雲系統需將金鑰容器資料清除，達到金鑰一次性的功能，因此，當薪資系統需要上傳二代健保申報資料時，皆需重新取得公鑰。完成解密後，將機關代碼、上傳者資訊等請求驗證上傳者使用權限資料來產生 XACML Request（如圖八），透過 Policy（如圖九）檢核完成後，回傳 XACML Response（如圖十），若最後回覆的結果為 Permit，健保雲系統即可將已解密的二代健保申報資料再進行非對稱式加密，並將密文資料儲存至資料庫（如圖十一），最後回傳申報成功訊息給薪資系統。

```xml
<Request>
  <Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
    DataType="http://www.w3.org/2001/XMLSchema#string">
        <AttributeValue>A23456789</AttributeValue>
    </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
    DataType="http://www.w3.org/2001/XMLSchema#string">
        <AttributeValue>H223456789</AttributeValue>
    </Attribute>
  </Subject>
  <Resource>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
    DataType="http://www.w3.org/2001/XMLSchema#string">
        <AttributeValue>4Bonus</AttributeValue>
    </Attribute>
  </Resource>
  <Action>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
    DataType="http://www.w3.org/2001/XMLSchema#string">
        <AttributeValue>Upload</AttributeValue>
    </Attribute>
  </Action>
  <Environment>
  </Environment>
</Request>
```

圖八：請求驗證上傳者使用權限 XACML Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Policy xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os" PolicyId="4BonusDataPolicy"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable">
  <Description>4BonusDataPolicy</Description>
  <Target>
    <Subjects>
      <Subject>
        <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">A23456789</AttributeValue>
          <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
          DataType="http://www.w3.org/2001/XMLSchema#string" />
        </SubjectMatch>
      </Subject>
    </Subjects>
    <Resources>
      <AnyResource />
    </Resources>
    <Actions>
      <AnyAction />
    </Actions>
  </Target>
  <Rule Effect="Permit" RuleId="4BonusRule">
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">H223456789</AttributeValue>
            <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
            DataType="http://www.w3.org/2001/XMLSchema#string" />
          </SubjectMatch>
        </Subject>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">F223456789</AttributeValue>
            <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
            DataType="http://www.w3.org/2001/XMLSchema#string" />
          </SubjectMatch>
        </Subject>
      </Subjects>
      <Resources>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">4Bonus</AttributeValue>
            <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
            DataType="http://www.w3.org/2001/XMLSchema#string" />
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Upload</AttributeValue>
            <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
            DataType="http://www.w3.org/2001/XMLSchema#string" />
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
  </Rule>
  <Rule Effect="Deny" RuleId="EndRule" />
</Policy>
```

圖九：驗證上傳者使用權限 XACML Policy

```xml
<Response>
  <Result ResourceId="4Bonus">
    <Decision>Permit</Decision>
    <Status>
        <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
```

圖十：驗證上傳者使用權限的 XACML Response

	sno	orgno	idno	cname	addr
1	1	123456789	0xF67518839DCCA42D2FD6D8DBA912202B20A2C945E68FCC...	0xFD05A0DED3CA9756239AA3E733AC98C51326315EE5C2E7...	0x5B462C7FDAD7F4690D56F(
2	2	123456789	0x2B55A7C2381F8F976DCC1D0DF6E8014A00DC59A91A95377...	0xDC1E64449281199E9C1ACDA61A1B9CF22A1AC348C2C0717...	0xFD5547339CF0AAA606F0C2

圖十一：使用非對稱式加密後儲存於資料庫查詢結果

伍、結論與未來研究方向

在雲端服務火紅的時代，雲端服務為企業節省了資金成本及提高服務品質、為使用者帶來使用方便性等優點，但對於雲端資料安全卻是企業及使用者擔心的部分，如何讓企業或使用者對雲端資料安全部分由疑慮轉變為信任，是國內外研究學者努力的方向。由於我國政府機關性質皆不太相同，因此當某些業務需透過其他政府機關資料才能完成時，通常都以媒體檔案匯入匯出的方式來達到解決目的，但也造成人力成本的提高及拉長作業時間之情形。有鑒於近年我國政府致力於政府雲的催生，提供更便民的服務，因此本研究參考過去以XACML來做為存取權限控管機制以及非對稱式加密來保護機敏性及個資資料的模式，並以Web Services來為異質雲端傳送資料，期望可以達到安全的異質雲端資料分享機制。

本研究針對雲端資料安全部分利用XACML及RSA加密演算法提出一安全雲端資料分享機制，惟該機制仍有改善及其深入研究的部分，在未來可朝下列兩點繼續研究：

1. 本研究僅針對私有雲資料分享機制進行研究，由於公有雲的系統會有更高的資訊安全考量，建議可以朝公有雲資料分享機制加以探討。
2. 本研究僅運用XACML於安全的雲端資料分享，建議未來可以再跟其他學者提出的EXACML來比較兩者之間的優缺點。

參考文獻

[1] S. B. Ayed and F. Teraoka, “Collaborative Access Control for Multi-Domain Cloud Computing,” *IEICE Transactions on Information and Systems*, E95-D(10), pp. 2401-2414, 2012.

[2] S. Ayoubi, A. Mourad, H. Otrok and A. Shahin, “New XACML-AspectBPEL Approach for Composite Web Services Security,” *International Journal of Web and Grid Services*, 9(2), pp. 127-145, 2013.

[3] S. C. Chou, and C.H. Huang, “An Extended XACML Model to Ensure Secure Information Access for Web Services,” *Journal of Systems and Software*, 83(1), pp. 77-84, 2013.

[4] T. T. A. Dinh, W. Wang, and A. Datta, “City on the Sky: Extending XACML for Flexible, Secure Data Sharing on the Cloud,” *Journal of Grid Computing*, 10(1), pp. 151-172, 2012.

[5] K. Hamlen, M. Kantarcioglu, L. Khan and B. Thuraisingham, “Security Issues for Cloud Computing,” *International Journal of Information Security and Privacy*, 4(2), pp. 36-48, 2010.

[6] S. Lakshminarayanan, “Interoperable Security Standards for Web Services,” *IT Professional*, 12(5), pp. 42-47, 2010.

[7] OASIS, "eXtensible Access Control Markup Language (XACML) Version 1.1," http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf, Accessed on November 2013.

[8] Sun, "Sun's XACML Implementation, " http://sunxacml.sourceforge.net/, Accessed on November 2013.

[9] 中央健康保險署電子報，「二代健保補充保險費簡介」，http://www.nhi.gov.tw/epaper/ItemDetail.aspx?DataID=2862&IsWebData=0&ItemTypeID=7&PapersID=244&PicID=，2013 年 11 月。

[10] 朱凱弘，「以 XACML 與 WebServices 為基礎之簽核系統」，*東華大學資訊工程學系研究所碩士論文*，2012。

[11] 吳信輝，「Web Services 技術介紹」，中央研究院計算中心通訊，第 20 卷， 第 23 期，2004。

[12] 俞正宏，「應用 XML/XACML 於工作流程管理系統之授權管制研究」，*中央大學資訊管理學系碩士論文*，2002。

[13] 陳威仁，「以數位學習為例之分散式動態權限管理框架」，*成功大學資訊工程學系碩士論文*，2005。

[14] 梁士杰，「應用 XACML 於家用網路安全策略之研究」，*輔仁大學資訊工程研究所碩士論文*，2006。

[15] 張念慈、彭秀琴，「雲端運算下資訊安全之探討」，*經建會管制考核處技術手冊*，2010。

[16] 蔡煥麟，「Web Services 入門」， http://sun.cis.scu.edu.tw/~nms9115/articles/delphi/WebServices/WebServices1.htm，2013 年 11 月。

[17] 薛承文，「應用 XACML 於電子病歷權限控管之研究」，*臺北大學資訊管理研究所碩士論文*，2005。

[18] 蔣仲翔，「在服務導向架構下的動態存取控制」，*臺灣師範大學資訊工程研究所碩士論文*，2010。