Electric Medical Record Exchange:Certificate approach

結合憑證之安全電子病歷傳輸系統

李正崙　朱慧媛　孫靖婷　楊滕清　廖晏辰　莊峻鴻　陳以德*　陳明德

高雄醫學大學 醫療資訊管理系

高雄、台灣

E-mail : u9514009; u9514030; u9514032; u9514036; u9514043; u9514049; itchen(@kmu.edu.tw); ecsemtchen@gmail.com

Abstract：

Electronic medical records (EMR) tend to be a part of health information system that allows storage, retrieval and manipulation of records. In addition, EMR also can save human resource, space and papers than traditional medical record. A referral patient has to fill in a form of personal information and examine again in another hospital if she/he do not use EMR. Hence, the Executive Yuan has approved the National Health Informatics Project (NHIP) proposed by the Department of Health (DOH). From 2006 to 2011, the NHIP has been promoting standards, laws, platforms, authentication and security of health information infrastructure.

In general, people who use EMR exchange feel relieved if the EMR exchange system is secure. But now, the DOH use USB flash disk for EMR exchange which exists many security faults such as USB virus. Furthermore, the responsibility of unintentionally medical records revelation is also a problem. In this study, we propose "EMR exchange: Certificate approach" which use cryptography mechanism such as OpenCA, symmetric/asymmetric encryption, hashing function, security socket layer (SSL) etc. Certificate approach protects EMR exchange from Trojans program and hacker invasion and SSL provides secure channel for EMR exchange. Finally, the technique of this study also could be used in telemedicine and elderly care applications.

***摘要*—傳統紙本病歷耗費許多人力、空間及大量紙本資源；且病患轉診時，除了要再次填寫相關個人資料，相關檢查也得重做一次，這不但浪費資源，更造成各醫療院所管理上不便。有鑑於此，衛生署陳報行政院「國民健康資訊建設計畫(NHIP)」通過自民國 96 年至 100 年，投入 23.59 億元，推動標準、法制、平台、儲存與使用、安全與認證等各項健康資訊基礎建設。電子病歷交換的安全性，是影響醫療院所及民眾使用意願的最大因素。目前我國是利用隨身碟，讓民眾攜帶自身的電子病歷走。然而這種方式，有許多安全性的問題存在。例如，隨身碟病毒、病歷資料外洩等責任歸屬問題等等。民眾在家或公開環境下，想從醫療院所下載電子病歷，如沒有安全傳輸保護，則電子病歷可能在網路傳輸過程中遭到駭客的攻擊或是攔截。因此，本篇著重在改善可攜式電子病歷傳輸的安全性，透過 OpenCA 憑證、金鑰加密等技術保護病歷實際內容，避免木馬程式與駭客的入侵，造成民眾個人資訊破壞或外洩，同時也可以提供安全的傳輸通道，除了讓各醫療院所間進行電子病歷交換時，能夠達到安全的傳輸與控管，亦可將相關技術用於遠距醫療與老人照護應用上，以期建立一個安全且有效率的可攜式電子病歷交換環境。**

***關鍵詞*—可攜性、電子病歷、HL7、國民健康資訊建設計畫、醫療保險責任法案、醫療資訊安全**

一、簡介

以往民眾想要取得自己的病歷或檢驗報告並不容易，在有轉診需求時，由於病歷屬於各醫

院的重要資產，彼此交換流通並不方便，同時也因為紙本病歷容易在民眾攜帶過程中遺失，因此攜帶個人病歷摘要來轉院就診情形並不多，然過往的就診紀錄能幫助醫生更了解病患狀況，並避免做重複的檢查以節省醫療資源的耗費。由於電子病歷的保密性及攜帶性都比紙本病歷佳，因此政府在 2007 年展開預計投入 20 億資金、為期五年的紙本病歷電子化計畫。

目前在各大醫療院所，隨著開業時間增長，病患人數的增加，紙本式病歷也會變得龐大，也不容易存放與管理，以五十多年的高雄醫學大學附設醫院為例，就需要好幾個房間來存放病歷紙本，而這些紙本的病歷，改為電子病歷，只需要幾張光碟片就可以儲存，所以電子病歷發展有其必要性與迫切性。再者，電子病歷可以運用在各醫療院所院間轉診使用，病患在 A 醫院做的檢驗報告等病歷資料，可以傳輸到 B 醫院用，藉此可以節省相關檢驗、人事費用，相對也節省了健保的支出。高雄醫學大學是 2007 年國內十間電子病歷試辦醫院之一，也是我們發展安全性電子病歷的一個重要資源。

2009 年初香港政府公立醫院發生幾起「USB」隨身碟遺失事件，令人關注電子病歷安全性問題。所以香港動用 20 億人民幣為全民建立電子病歷，且預計十年後達成電子病歷交換。2009 年 8 月 11 日，行政院衛生署修正通過醫療機構電子病歷製作及管理辦法，解決電子病歷在實施上的技術障礙；同年台中榮民總醫院、台大醫院、岡山秀傳醫院、台北醫學大學附設醫院、已通過檢查，近期將宣告實施電子病歷；其中台中榮民總醫院自今年 6 月 1 日起，開始實施護理給藥簽用紀錄、護理紀錄及護理執行治療紀錄之電子病歷，此為我國電子病歷發展上之重要突破！電子病歷的安全性更是其能否成功推行的因素，所以本篇主要運用及修改自由軟體 Care2X、OpenCA、OpenSSL 等，來設計一套安全的電子病歷儲存、交換與傳輸系統。

二、相關文獻

對於安全電子病歷傳輸相關文獻，我們參考了相關的電子病歷文獻與密碼學元件，藉由這些相關技術文件，來建立安全的可攜式電子病歷傳輸系統，分別介紹如下。

2.1 電子病歷

根據 (Joint Commission on Accreditation of Healthcare Organization, JCAH) 文件，電子病歷分為個人資料與個人病歷兩大部分。基本資料可以包括身份證號、姓名、出生日期、出生地、婚姻狀況、住址、電話、家屬、父、母親姓名等等相關個人資料。另外在個人病歷方面，包含了病患號碼、入院/看診日、主治醫師、病歷(包含診斷、處置、醫囑…等等)、用藥處方、藥局、檢驗結果、圖像(包含心電圖、X-Ray、MRI…等等)、以及其他醫事人員執行業務所製作的紀錄等等相關資料，都包含在個人的電子病歷資料中如下表所示：

表 1 電子病歷格式

基本資料	病歷資料
● 身分證號 ● 姓名 ● 出生日期 ● 出生地 ● 婚姻狀況 ● 住址 ● 電話 ● 家屬父、母親姓名 (Optional) ● …	● 病患號碼 ● 入院/看診日 ● 主治醫師 ● 病歷(包含診斷、處置、醫囑…等等) ● 用藥處方、藥局 ● 檢驗結果 ● 圖像(包含心電圖、X-Ray、MRI…等等) ● 其他各類醫事人員執行業務所製作之紀錄 ● …

2007 年衛生署推動 NHIP 計畫，使得目前各大醫療院所也漸漸加入電子化病歷的行列，對於電子化病歷，病患病歷不再使用傳統紙本病歷，

而是電子化的版本，不但節省醫療資源，更可以節省人力開銷。對於政府所提的 NHIP 計畫，重點在於提供一個安全、健康資訊基礎建設，提供病患可攜式的電子化病歷，以利於病患轉診或是各大醫院間病歷交換，與避免醫療、用藥重覆等不必要的浪費。

NHIP 計畫重點在於推行可攜式電子病歷[1]，另外能夠讓各大醫療院所能夠透過電子病歷交換，來達到電子病歷的方便性與流通。可攜式電子病歷內有四項重大編碼，一為疾病分類 International Classification of Diseases(ICD)，一為臨床與檢驗代碼 Logical Observation Identifiers Names and Codes (LOINC, http://loinc.org)，LOINC；Systematized Nomenclature of Medicine-Clinical Terms (SNOMED CT, http://www.ihtsdo.org) 為臨床名詞(SNOMED CT)的表示法與檢索系統內編碼，與國內健保碼為四套重要之編碼標準。其中疾病分類依照世界衛生組織 (World Health Organization, WHO) 所訂定 International Classification of Diseases (ICD, http://www.who.int/classifications/icd/en)為主，ICD 每隔十年左右會更新一次疾病分類，目前 ICD 最新為 2006 年 1 月 26 日訂定的 ICD-10，此外在 2008 年 7 月 10 日版本更新為 2.24 版，且同時提供了 Regenstrief LOINC Mapping Assistant (RELMA 3.24) 應用程式，來提供醫療院所將院內碼對應到 LOINC。Systematized Nomenclature of Medicine-Clinical Terms (SNOMED CT, http://www.ihtsdo.org) 為臨床名詞(SNOMED CT)的表示法與檢索系統內編碼，健保碼則是國內醫療院所申報醫療費用的通用碼(http://www.nhi.gov.tw)。

目前國內外醫療院所所用的傳輸標準是 XML-base 的 Health Layer 7 [7] / Clinical Document Architecture (HL7/CDA, http://www.hl7.org; http://healthinfo.med.dal.ca/hl7intro/ CDA_R2_NormativeWebEdition)。目前 HL7 版本由早期的 1987 年的第一版 Version 1，發展至 1994 年被 American National Standards Institute (ANSI)所接受採用。發展至今，HL7 最新版本為 Version 3，重要的功能除了減少了例外狀況及曖昧不明確的定義並且也採用物件導向(Object-Oriented)的概念；HL7 Version3 訊息建構於 Reference Information Model (RIM)上如下圖所示：

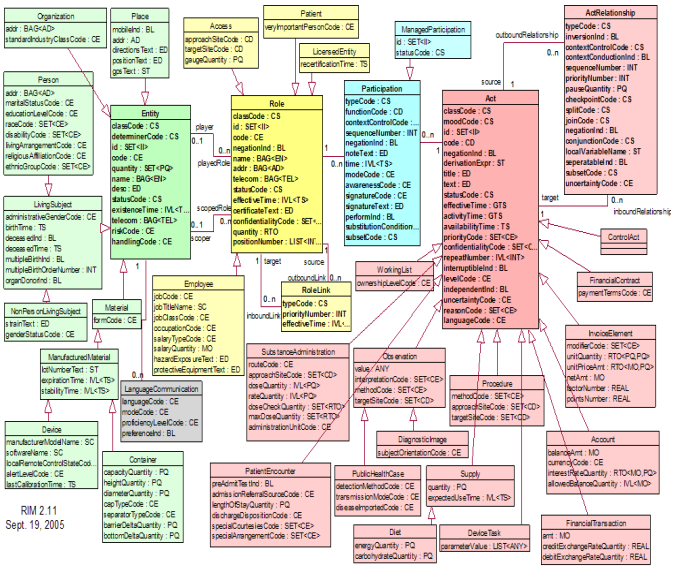

圖 1：RIM 2.11 (http://www.hl7.org)

另外一方面，CDA 是用來展現歷史醫療紀錄的文件格式，用來輔助 HL7 交換傳輸的不足地方。CDA 可以把分散的資料及片斷的病歷整合，也可以把含有 Header 及 Body 可用 MIME 編碼放入 HL7 message 中。

行政院衛生署促進全台整理醫療院所內的病歷電子化，也推廣了"網路健康服務推動計畫"用來整合全國醫療體系的資訊網公用性資料庫、建立藥品交互作用資料庫、推廣病歷電子化、設置及營運醫療憑證管理中心(HCA)、推廣醫療資訊標準，委託研究醫療資訊安全及隱私保護相關法規以及評選優良醫療網站等，以及落實持續照護及民眾健康管理等相關工作。

此外其中一項重要工作就是 Taiwan electronic Medical record Template (TMT) [2] 的建立(http://emr.doh.gov.tw)，TMT 收集全國 241 家醫院約 1 萬多張的病歷單張，再經過專家的整理、歸納及共同討論而制定出適合臺灣各醫療院所使用，植基於 XML 的電子病歷樣本。http://emr.doh.gov.tw 網站提供了 Viewer 及 Mapping Tools 供各界研究。

2.2 密碼學演算法

密碼學(Cryptography)是用來確保資料安全性及完整性之技巧總稱。密碼學中最基本的兩個步驟分別為加密和解密。加密(Encryption)即是利用加密金鑰(encryption key)，將明文(plaintext)轉換成密文(ciphertext)的過程；解密(decryption)則與上述相反。透過密碼學，可使發送端將訊息安全的送出，也保障在其傳輸過程中即使被他人所攔截，也無法得知其訊息的真正內容，最後接收端再將收到的訊息還原成本來的訊息。但僅能提供資料的保密性，在目前的網路環境中已經不敷使用，因此本篇除了保密性外，將整合公開金鑰建設(Public Key Infrastructure, PKI)系統中具備完整性、確認性、及不可否認性等密碼學技術，來達成安全性電子病歷傳輸系統。

2.2.1 對稱金鑰演算法

對稱金鑰演算法(Symmetric Key Algorithm)係利用同一把金鑰進行加密和解密的動作。由於是利用同一把金鑰加、解密，故這把金鑰是無法公開的，因此又稱為私密金鑰(private key or secret key)。所以對稱金鑰演算法也稱做私鑰演算法。這類的演算法常常用來對資料區塊或資料串流做加密。區塊演算法是先將訊息等分成大小相同的區塊，一次對一個區塊做加密。串流演算法則是一次對一個位元做加密。由於可能的金鑰總數非常大，除非拿到解密的金鑰，否則幾乎無法破解對稱金鑰演算法。此次研究中採用的對稱式加密演算法為 AES，茲介紹如下：

AES 演算法

AES (Advanced Encryption Standard 高級加密標準模組) [12] 係由美國聯邦政府採用的一種區塊加密標準。這個標準用來替代原先的 DES，已經被多方分析且廣為全世界所使用。AES 的區塊長度固定為 128 位元，密鑰長度則可以是 128、192 或 256 位元。其加密過程是在一個 4×4 的位元組矩陣上運作，對每一回合的 AES 加密迴圈，均利用 AddRoundKey、SubBytes、ShiftRows、MixColumns 此四種方法。

2.2.2 非對稱式金鑰演算法

非對稱式金鑰演算法(Asymmetric Key Algorithm)又稱做公鑰演算法。此種演算法有兩把金鑰，分別用來加、解密。公鑰(public key)用來對明文加密；私鑰(private key)則做為解密之用，公鑰是公開讓所有人知道的，並不會危害到整個資料的安全；因此所以每個人都可以利用公鑰來做加密，卻只有特定持有私鑰的人才能夠解密。此次所採用的非對稱式加密演算法為 RSA，茲介紹如下：

RSA 演算法

RSA 演算法是由 Ronald Rivest、Aid Shamir、Leonard Adleman 這三位教授所發展出。演算法中利用兩個大質數其乘積容易計算，但卻難以從乘積反推的特性，做為金鑰產生的方式。令 p、q 各為一大質數，$n=pq$, $\Phi(n)=(p-1)(q-1)$，選取 e 為公鑰，其中 e 與 $\Phi(n)$ 互質且 $ed \bmod \Phi(n) = 1$。而我們令 M 為明文，C 為密文，以下為解密時運算過程

$$(C^d \mod n)$$

$$= ((M^e \mod n)^d \mod n)$$

$$= M^{e*d} \mod n = M^{(k*(p-1)*(q-1))+1} \mod n$$

$= M^1 \bmod n = M$

2.2.3 雜湊函數

雜湊函數 [14] 是一種從資料中建立較小的訊息摘要(message digest)。該函數將資料打亂混合，重新建立一個雜湊值。雜湊值通常用具有以下兩個特性：

1. 抗拒事先描繪（Preimage Resistance）：當給定一特定的雜湊輸出值後，無法欲找出任何文件可以輸出此一特定的雜湊值。
2. 抗拒第二事先描繪（Second Preimage Resistance）：即使給定一份文件及其雜湊值後，亦無法找出第二份文件可以輸出此一特定的雜湊值。

雜湊函數可分為單向雜湊函數(one-way hash function)及後門雜湊數(trapdoor hash function)兩種：單向雜湊函數容易計算出結果，但其反函數卻難以運算得到原答案。而後門雜湊數為單向函數的變化型，經由後門(trapdoor)的幫助，可以提升求得反函數的速度。本篇中採用SHA-serial演算法，茲介紹如下：

SHA-serial 演算法

SHA-1 (Secure Hash Algorithm-1) 為 SHA-serial 演算法中的一員，比原始的對稱金鑰演算法快速，且仍具有高安全性。能將輸入資料打散，使亂度平均分配到函數輸出值的每個位元上。以 SHA-1 為例，在運算的過程中首先添加「墊塞位元」，當不滿 512bits 時，在最後的區塊，先加入一個位元「1」，其餘補「0」，直到總長與 448 mod 512 同餘；再來添加「記錄長度」，原始資料長度以 64bits 整數表示，加到計算結果為 1.xxx 之後，成為 512 的倍數之後；再用「初始化記錄器」，SHA-1 用 160bits 的紀錄器，由 5 個 32bits 暫存器（ABCDE）表示。接下來以 512bits 為單位處理訊息，共有四回合，每回合 20 個步驟，將明文 512bit 分為 80 份 32bit 的子區塊 W_t：

$W_t = M_t$，當 $0 \leqq t \leqq 15$

$W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) << 1$，當 $16 \leqq t \leqq 79$

其中⊕表示 XOR，<< 表示 Shift，經過一些邏輯運算，將每回合輸出為下回合初始值，最後一回合再與初始值做 XOR 即為答案[14] 。SHA-1 (FIPS 180-1)在 August, 2002 FIPS-2 發布後同時停用，目前 National Institute of Standards and Technology (NIST)正在公開徵求 SHA-3 演算法。在第一回合 SHA-3 選拔會議中，選出了 51 個演算法候選人；在 July, 2009 剛結束的第二回合會議中，討論出 14 個演算法候選人，分別為 BLAKE、Blue Midnight Wish、CubeHash、ECHO、Fugue、Grøstl、Hamsi、JH、Keccak、Luffa、Shabal、SHAvite-3、SIMD 及 Skein。第三回合會議在 August, 2010 舉行，藉時將選出最安全的演算法成為 SHA-3。

2.3 公開金鑰基礎建設

公開金鑰建設(PKI)，採用公開金鑰演算法系統來為網路傳輸的訊息提供保護，藉由著憑證管理中心（Certificate Authority, CA）將使用者的個人身分跟公開金鑰鏈結在一起，CA 基本架構如圖 2。每個憑證中心使用者的身分必須是唯一的，同時在人為監督下，可以合併使用分散於各地的其他協同軟體。對每個使用者，憑證中心發行的公開金鑰憑證含有不可偽造的個人身分、公鑰、有效條件與其他資料等。

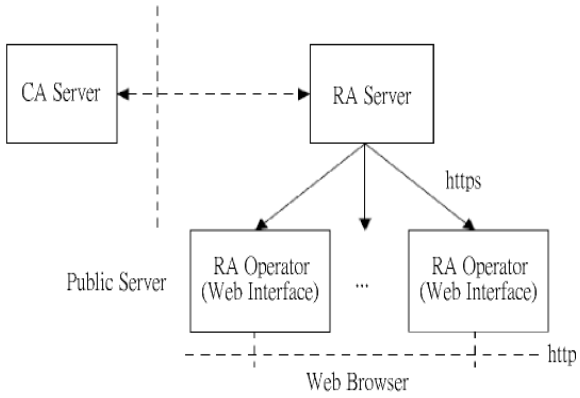

圖 2 CA 基本架構

2.3.1 憑證

憑證（Certificate）是由具有公信力的可信賴第三者（Trusted third party，TTP）所簽發，憑證機構將一把公開金鑰與申請者的身分做確認之後，將其相關訊息做數位簽章後轉換成一張憑證，該憑證可以在網路上公開流通。一般來說憑證採用 X.509 V3 的標準制定，內容包含版本、序號、簽章演算法、發布單位、憑證有效期限、使用者名稱、公開金鑰資料、簽發者識別碼、使用者識別碼、簽章演算法等內容。

2.3.2 憑證管理中心

憑證管理中心（CA）一般為具有公信力的第三方機構，主要負責的工作內容包含使用者憑證的認證與簽發，管理已發布的相關憑證及公開金鑰等，在 PKI 的架構當中屬於核心的關鍵組織。

2.4 OpenCA

在建置完整的 PKI 系統需要耗費高額的成本，同時在配置的過程也有一定的難度。利用開放原始碼的 OpenCA [16]可以建構一個功能完整的 CA 系統。OpenCA 由 Perl 作為 CGI 腳本開發，配置方面，可以使用搭配下面開放原始碼套件：OpenSSL [3]、OpenLDAP [17]、Apache [18]和 Apache Mod_SSL，來達到節省成本、易於配置、及方便管理等優點。

2.5 Java Card

Java 卡 [4] 是智慧卡的一種，包括了內嵌的中央處理器以及多種記憶體。Java 卡與一般 IC 卡不同之處在於卡中有 Java 虛擬機(Java Virtual Machine, JVM)和 Java 卡運行環境(Java Card Runtime Environment, JCRE)，可讓開發人員使用 Java 程式語言來撰寫卡內的應用程式(applet)，並可讓此應用程式在卡內獨立運行。

三、系統實做

3.1 系統架構

本研究系統可分為憑證管理中心、HCA 發卡中心和醫院用戶端三部分，如圖 3 所示。憑證管理中心，其中 HCA 角色，我們使用開放原始碼的 OpenCA 套件，建構在 Linux Ubuntu 8.04 平台上，建置為符合 PKI 規範的憑證管理中心。HCA 發卡中心透過卡片 Applet 程式，在發卡初期，預先將對應到該民眾的憑證資料利用雜湊函數轉換成 Hash 值，並將該 Hash 值透過 Card Reader 存放於 Java Card 當中。在醫院用戶端的部分，我們利用 Java SDK 撰寫了包含 AES、RSA 等密碼學演算法，針對欲加密的資料類型採用不同的演算法。

3.2 OpenCA 的安裝

由於 OpenCA 並非能獨立建置的完整系統，使用的過程中需要其他開放原始碼軟體配合，安裝 OpenCA 前需先安裝 PHP、Apache、MySQL、OpenSSL、Perl5、mod-ssl 及 OpenLDAP 等相關套件並針對 OpenCA 所建議的版本做安裝。

在安裝完成會依照功能分成 Pub、RA、CA、Node 等介面，分別處理申請、審核、發放、管理憑證等動作。

3.3 系統概念圖

在實做的過程中，我們假定有病患(Patient)、醫療院所(Hospital)、HCA發卡中心 (醫療憑證中心) 、第三方公正機關這四個角色，如圖 3所示。HCA會將一張專門提供給該病患的$Cert_{Patient}$ 透過 API 存放在Java Card中，醫療院所也會向HCA取得醫療憑證$Cert_{Hospital}$。中央資料庫可以由HCA或是中央健保局擔任。

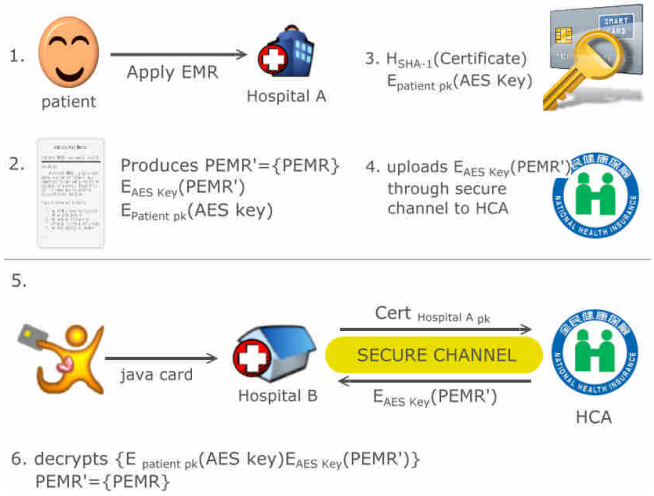

圖 3　可攜式安全病歷傳輸協定

假定某病人要從A醫院轉診至B醫院，他向A醫院提出申請動作後。A醫院將此名病患病歷以AES加密法加密成密文，詳細加密過程請參照3.5節。接著上傳加密後的病歷至中央資料庫。

病人攜帶卡片前往B醫院，B醫院上傳此卡片中的憑證至中央資料庫，申請該病患的電子病歷，驗證過後取得傳回的病歷密文，再使用此卡片對病歷解密，以取得病歷明文。

3.4 HCA 端

HCA 利用憑證管理中心產生新的憑證後，利用雜湊函數演算法取得該憑證的雜湊值，並將其存放在 Java Card 內然後將卡片發給民眾，如圖 4 所示。

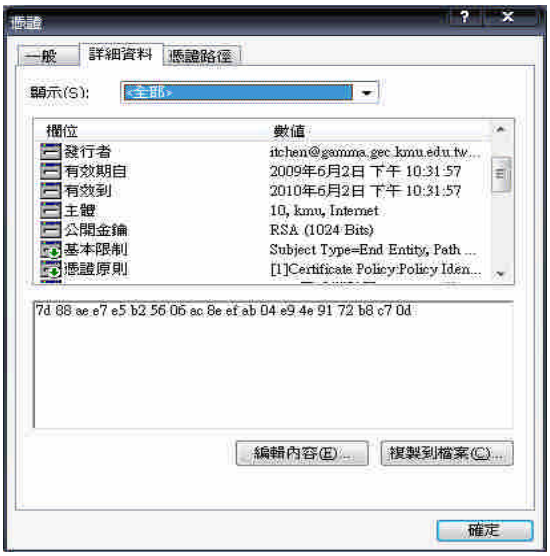

圖 4　CA 所發行憑證

3.5 使用者到達 Hospital A

當病患有跨院就醫需求時，可以向曾經就診過的醫院，申請個人的電子病歷，假設病患向 A 醫院就診後並申請電子病歷，A 醫院在接獲申請之後，先利用隨機產生的 AES 金鑰對該病患的病歷摘要 (病歷摘要 3.xml) 做加密，如圖 5 所示，此時會產生加密過後的病歷摘要密文(病歷摘要 3.xml.aes)如圖 6 所示。

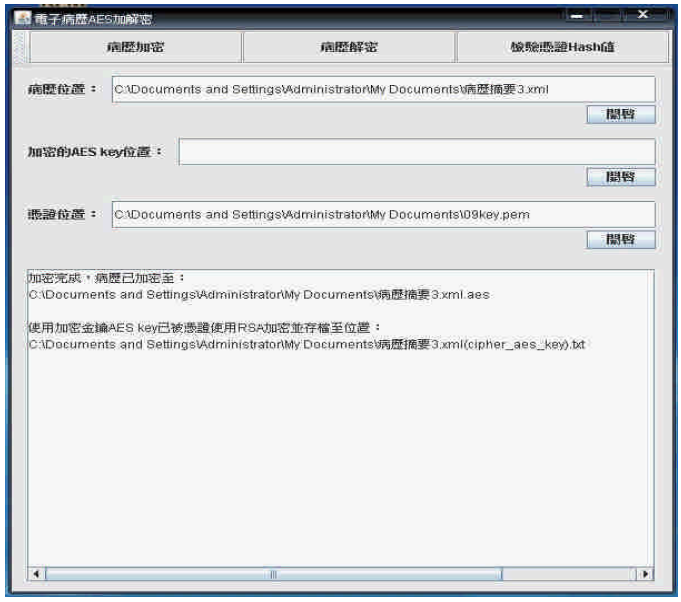

圖 5　利用 AES 加密電子病歷

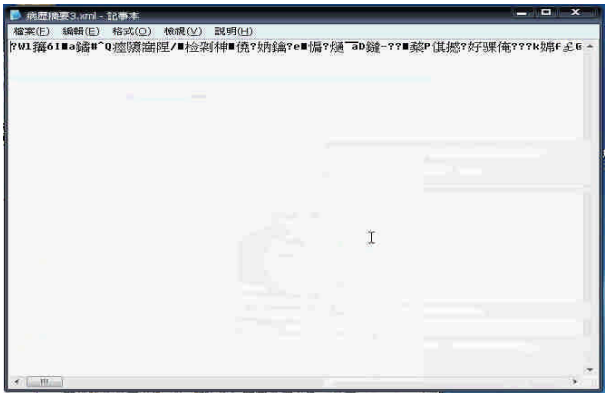

圖 6　加密之後的密文

接著 A 醫院為了提升金鑰的安全性，將會讀取病患卡片中憑證($Cert_{Patient}$)的 PublicKey 利用 RSA 演算法來對 AES 金鑰做加密，產生病歷摘要.xml (cipher_aes_key) 如圖 7 所示，並將這把

加密後的 AES 金鑰存入病患卡片中，交由病患帶往欲就診的 B 醫院，同時將加密過後的病歷摘要密文，透過網路傳送至第三方公正機關做存放。

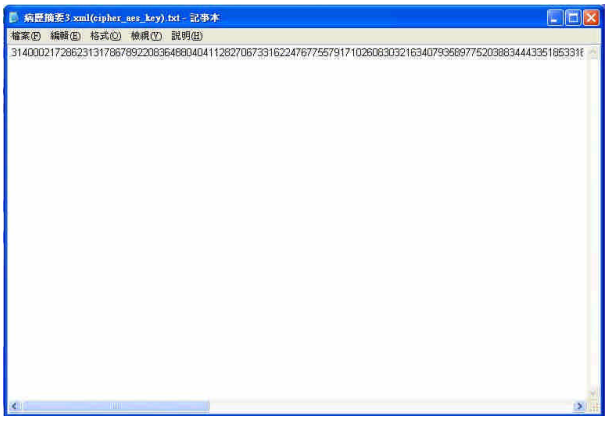

圖 7　AES 金鑰

3.6 使用者到達 Hospital B

病患到達 B 醫院之後，B 醫院讀取卡片中的憑證，用以做為身分辨識，並向第三方公正機關申請該病患的病歷摘要密文，同時利用卡片中憑證的 PrivateKey 將加密後的 AES 金鑰解密取得(病歷摘要 3.xml(cipher_aes_key).txt(已解密))，並利用該金鑰將由第三方公正單位所傳送過來的病歷摘要密文解密，得到(病歷摘要.xml.aes(已解密)如圖 8 所示。

圖 8 解密後之病歷摘要

四、結論

在此本篇論文中，我們著重在實做 PKI 及相關密碼學技術來改善可攜式電子病歷傳輸的安全性。在病歷傳輸過程中避免木馬程式與駭客的入侵，造成民眾個人資訊破壞或外洩，同時也可以提供安全的傳輸通道 (SSL)，除了讓各醫療院所間進行電子病歷交換時，能夠達到安全的傳輸與控管，本論文所用到的相關技術亦可用於遠距醫療與老人照護應用上，以期許建立一個安全且有效率的可攜式電子病歷交換環境。

五、參考文獻

[1] 陳以德、陳明德、簡文山。可攜性電子病歷保護。資通安全資訊網。2009 年 8 月 25 日，取自：http://ics.stpi.org.tw/Treatise/doc/80.pdf

[2] 臺灣電子病歷交換基本格式 TMT，取自：http://emr.doh.gov.tw

[3] OpenSSL 與網路信息安全 一基礎、結構和指令，取自：http://www.OpenSSL.org

[4] JAVA 卡系列之相關技術發展，取自：http://www.ibm.com/developerworks

[5] 郭育郎（2007）。結合 JAVA 卡之個人檔案保密防偽系統的設計與實作。國立高雄師範大學資訊教育研究所碩士論文，未出版，高雄市。

[6] 蔡育儒（2007）。Live-CA：結合 Java Card 與 PKI 的憑證管理系統之設計與實現。國立高雄師範大學資訊教育研究所碩士論文，未出版，高雄市。

[7] HL7 Web site, from the World Wide Web: http://www.hl7.org

[8] 台灣健康資訊交換標準相關資訊，取自：http://www.hl7.org.tw

[9] HL 7，取自：http://groups.google.com/group/hl7-taiwan

[10] 行政院醫事憑證管理中心，取自：http://hca.doh.gov.tw

[11] Zip AES 加密系統相關資訊，取自：http://www.winzip.com/aes_info.htm

[12] NIST, Advanced Encryption Standard (AES). November 11, 2008, from the World Wide Web: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[13] 自然人憑證相關資訊，取自：MOICA 內政部憑證管理 http://moica.nat.gov.tw

[14] SHA-serials, 2009 年 9 月 1 日，from Web: http://csrc.nist.gov/groups/ST/hash/sha-3/index.html

[15] PKCS, from Web: http://www.rsa.com/rsalabs/node.asp?id=2124

[16] OpenCA, from Web: http://www.openca.org

[17] OpenLDAP, from Web: http://www.openldap.org

[18] Apache, from Web: http://www.apache.org