建構主計機構間電子認證環境之研究

研究報告

研究人員：黃采紅、蔡福隆、張良鵬

研究單位：行政院主計處電子處理資料中心

地　　址：台北市廣州街2號

研究期程：91年01月01日～91年12月31日

中華民國92年01月27日

目錄

第一章 自行研究計畫簡介

1.1 摘要

我國的電子簽章法於九十年十月通過立法，同年十一月十四日總統明令公布全文，隔年行政院正式宣布九十一年四月一日爲我國電子簽章法的施行日，這也等同於正式宣布我國開始進入安全的電子化、網路化政府時代，同時也爲電子文件取得合法的法律地位。將來我政府在施政上，不但可以藉由具有法律地位的電子文件或電子表單在網路上提供線上作業(例如網路申辦業務)，大量減少書面紙張的印製，同時也加速處理流程，提高政府效率。電子簽章不但可解決網路上身份辨識的問題，也可確保電子文件或電子表單的完整性與不可否認性，甚至未來在檔案管理上對於歸檔電子文件或電子表單也易於管理，同時能提供快速的線上搜尋與驗證。但是電子簽章的施行，有賴於電子認證環境的建立，在公開金鑰基礎建設 PKI(Public Key Infrastructure)標準和 X.509 認證架構下，數位簽章是目前技術較爲成熟且廣泛使用的電子簽章方法，其運作必須有一公正之第三者成立憑證機構，由憑證機構製作簽章用的公、私鑰，並提供電子文件存證、公證及時戳的服務。行政院主計處爲全國最高政府主計機構，掌理全國主計業務包括預算、會計、統計等，尤其是在會計事務處理與內部審核的作業流程中有太多的文件與表單(憑證)可以電子化，我們主計機構是政府的一環，自然不應自外於電子化政府的

推動，尤其是在金流與內部審核的電子化部分，因此如何建構主計機構間電子認證環境，主計業務如何導入電子簽章機制，將是一個迫切且值得研究的課題。

1.2 計畫目的與研究方法

在公開金鑰基礎建設 PKI 下，透過數位簽章及加解密程序的使用，可以達到確保電子文件和電子表單的完整性、機密性、來源鑑別、不可否認性等安全需求。無論是電子商務交易、電子化政府、企業電子化，數位簽章都是重要的工具之一。就像網路裝置之間必須遵循一定的協定規範才能彼此互通一樣，數位簽章也必須有舉世公認的標準存在，附加數位簽章的電子文件才能跨越國界、跨越作業系統平台、跨越應用程式，正確地通過驗證而暢行無阻。

本計畫之研究目的如下所述:

(1)研析國內政府電子認證環境的建置情形與現況，包括電子簽章的施行、公開金鑰基礎建設(PKI)、政府憑證總管理中心(GRCA)、政府憑證管理中心(GCA)、內政部憑證管理中心(MOICA)等。

(2)研析國內外採行的數位簽章、公開金鑰基礎建設等相關標準。

(3)研析主計業務是否納入電子簽章機制需求的必要性。

(4)研析如何建構主計機構間電子認證環境，並提供建言。

本計畫首先蒐集目前國內政府電子認證環境的建置情形與現況、同時蒐集世界各國採行的數位簽章、公開金鑰基礎建設等相關標準加以研析，最後結合主計機構業務上的需求與特性，提出建構主計機構間電子認證環境以及如何導入電子簽章機制的建言。

本研究計畫的進行將採取下列步驟：

(1)蒐集及整理分析國內政府電子認證環境的建置情形與現況。

(2)蒐集及整理分析世界各國採行的數位簽章、公開金鑰基礎建設等相關標準。

(3)研究如何結合主計機構業務上的需求與特性，導入電子簽章機制。

(4)撰寫研究報告。

(5)投稿相關研究領域的研討會或期刊。

第二章 、國內政府電子認證環境現況介紹

隨著網際網路盛行，許多使用者皆欲憑藉網際網路取得可用資源與服務，而電子憑證將可提供這些使用者相關的身分 識別及資料保護等特性。對於採用網路遞送資料的應用系統，若能提供完善的公開金鑰機制，將可以滿足實體間身分可鑑別性、交易資料的完整性與私密性，以及交易後的不可否認性等安全需求。

在公開金鑰基礎建設下安全的電子認證機制，主要是爲了確保資料在網路上傳輸與儲存的安全，其機制主要包括有下列四項：

(1)機密性(Confidentiality)：電子文件可以金鑰加解密，以達到機密性。

(2)完整性(Integrity)：電子文件接收者透過數位簽章之核對可確保此文件之完整性。

(3)來源確認(Authentication)：電子文件接收者可確認此文件之發送者身分。

(4)不可否認性(Non-repudiation)：因爲只有電子文件發送者知道自己的私密金鑰，而且電子文件具有發送者之數位簽章附據，使其無法否認發送之事實。

透過網路進行通信或交易時，其通信或交易的主體我們稱爲電子文件，就如同傳統之書面文件，只是網路是爲一個開放空間，電子文件很可

能遭第三者蓄意僞造，也可能在傳輸的過程中非法遭擷取或竄改，而使得事後交易一方亦可能否認其交易行爲。因此爲避免上述情形發生，電子文件必須與數位簽章進行電腦邏輯處理，以產生一個較完整之通信或交易主體，就如同已簽名或蓋章的書面文件。以一般電腦製作之電子文件，因容易經由網路被竄改及僞造，因此必須以符合特定要件之安全程序，所製作之電子文件，較能確保資料在儲存及傳輸過程中的完整性。因此凡是使用安全程序製作之電子文件，才足以驗證資料訊息內容是某一特定時間點至驗證時間點之間未經竄改者，方可稱爲安全電子文件。又由於安全的電子文件之安全性及可信賴度，皆可與書面文件相比擬，因此安全電子文件與書面文件將具有相等之法律效力。

在正式介紹國內政府電子認證環境現況之前，玆先整理相關術語簡述如下：

(1)電子簽章：指依附於電子文件上，用以辨識及確認電子文件簽署人身分及電子文件真僞者。

(2)數位簽章：指將電子文件以數學演算法或其他方式運算爲一定長度之數位資料，並以簽署人之私密金鑰對其加密，形成電子簽章者。以「非對稱型」加密技術爲基礎之「數位簽章」是目前市面上使用最普遍的一種電子簽章。

(3)電子憑證：是指憑證機構(例如 GCA)以數位簽章方式簽署的資料訊

息，用以確認憑證申請者之身份，並證明其確實擁有一組相對應之公開金鑰及私密金鑰之數位證明。

(4)公開金鑰：指一組具有配對關係之數位資料中，用以對電子訊息加密或驗證簽署者身份及數位簽章真偽之數位資料

(5)私密金鑰：指一組具有配對關係之數位資料中，作爲電子訊息解密或製作數位簽章之用。

總而言之，數位簽章是目前技術較爲成熟且被廣泛使用的一種電子簽章，它是使用數學演算法（或稱雜湊函數）將電子文件轉化爲固定長度之數位資料（訊息摘要），並用簽署者之私鑰（代表簽署者本人之數位資料）對其加密形成一簽體，使任何人可藉未轉化前之原始資料訊息、簽體及與私鑰相關連之公鑰（公開之數位資料），驗證該簽體是否使用與簽章公鑰相對應之私鑰所製作，以及簽體製作後，原始資料訊息是否遭受竄改。

在公開金鑰基礎建設下，數位簽章的運作必須有一公正之第三者成立憑證機構，由憑證機構製作簽章用的公、私鑰，並提供電子公文存證、公證及時戳的服務。私鑰就好比是私人的印鑑，公鑰則好比是印鑑證明，簽署者利用私鑰（印鑑）在電子公文（書面文件）上簽章，產生的電子公文稱爲簽體（已簽章之書面文件），收到簽體的一方則可以向憑證機構申請簽署者之公鑰（印鑑證明），以驗證簽體之真僞。其簽章的方法目前大多採用美國的 RSA 數位簽章系統，首先簽章者 A 將文件 m 複製一份，配合 A 的

私鑰，當做 RSA 系統的輸入，得到輸出結果，也就是 Signature。再來文件 m 與 Signature 一併送至驗證者 B。而驗證者 B 在接收到文件 m 及 Signature 後，將 Signature 及簽章者 A 的公鑰當做 RSA 系統的輸入，將輸出結果文件 m'與原訊息文件 m 做一比較。當 m' = m 時，即表簽章者 A 對於 m 這份電子公文表示同意，反之，若不相等，則表示 m'遭到竄改。

目前以公開金鑰基礎建設(Public Key Infrastructure, PKI)爲設計核心的應用系統正逐漸增多，若要達成這些系統間相容性則必須考量資料的標準格式。假設數個 PKI 應用程式架構在網際網路的環境下，欲使各系統間的傳遞資料能夠相互使用，則前提爲協定設計中須採用標準資料格式與意義。接著就我國政府機關公開金鑰基礎建設之現況分節詳述如后：

2.1 政府機關公開金鑰基礎建設

我國政府機關公開金鑰基礎建設(Government Public Key Infrastructure, GPKI)係依據電子化政府推動方案(九十至九十三年度)，爲健全電子化政府基礎環境建設，建立行政機關電子認證及安全制度而設立，請參閱圖 2.1。它是依照 ITU-T X.509 標準建置的階層式(Hierarchy)公開金鑰基礎建設，包含公開金鑰基礎建設的信賴起源(Trust Anchor)政府憑證總管理中心(Government Root Certification Authority, GRCA)，及各政府機關所設立的下屬憑證機構(Subordinate CA)所組成。加入政府機關公開金鑰基礎建設的憑

證機構爲各目的事業主管機關所建置的憑證機構，其簽發的憑證應用於電子化政府的各項應用，以提供更便捷的網路便民服務，提昇政府行政效率，促進電子商務的應用發展。

圖 2.1 政府機關公開金鑰基礎建設(GPKI)

2.2 電子化政府電子認證之推行組織

(1)行政機關電子憑證推行小組：行政院研究發展考核委員會爲建立電子化政府電子認證制度，推動電子化政府公鑰基礎建設，加速行政機關網路便民服務之應用發展，成立行政機關電子憑證推行小組。

(2)政府憑證總管理中心：政府憑證總管理中心(Government Root Certification Authority, GRCA)，爲政府機關公開金鑰基礎建設階層架構

的最頂層憑證機構，也是公開金鑰基礎建設的信賴起源，必須具備最高的公信度。政府憑證總管理中心負責政府機關公開金鑰基礎建設領域內外憑證機構間的交互認證，目前由行政院研究發展考核委員會以委外服務方式，委託中華電信數據分公司營運。

2.3 電子化政府電子認證體系之發展

我國電子化政府電子認證體系之發展分爲二個階段：第一階段以試辦性質建置各機關共用之憑證機構爲目標，以政府憑證管理中心(Government Certification Authority, GCA)涵蓋 GRCA 及其他相關憑證機構功能；第二階段(現階段)依應用狀況及技術成熟度，規劃 GPKI，建置 GRCA，並由各目的事業主管機關建置相關憑證機構。

八十九年八月十八日及十一月十三日行政院研考會邀集相關機關及學者專家，召開電子化政府電子認證服務研商會議。爲達到憑證應用之完整性及多樣性，決議將電子化政府電子認證體系架構分爲：

(1)公開金鑰基礎建設(Public Key Infrastructure, PKI)：提供公鑰憑證(身份憑證)服務。

(2)授權管理基礎建設(Privilege Management Infrastructure, PMI)：提供屬性憑證(資格憑證)服務。

2.4 電子化政府電子認證服務分工與建置方式

依據「電子化政府推動方案(九十至九十三年度)」具體措施分工如下：

(1)行政院研考會： 建置 GRCA、GCA、政府測試憑證管理中心 (TestCA)、法人及 非法人團體憑證管理中心(XCA，由各主管機關擔任憑證註冊窗口)。

(2)經濟部： 建置工商憑證管理中心(MOEACA)。

(3)內政部：建置自然人憑證管理中心。

至於建置方式目前規畫如下：

(1)首先建置 GRCA 及 GCA。

(2)當 GRCA 及 GCA 建置完成後，將原 GCA 簽發之政府機關(構)、單位及伺服器應用軟體憑證，改由新的 GCA 簽發。

(3)原 GCA 簽發的公司行號憑證改由經濟部之 MOECA 簽發。

(4)原 GCA 簽發的自然人憑證改由內政部之自然人憑證管理中心簽發。

(5)原 GCA 代管之各類憑證，代管至相關主管機關建置憑證機構及原 GCA 代簽發之憑證有效期限屆滿為止。

第三章、國內外相關技術與標準規範

政府機關要建構電子認證環境，首先要針對國內外相關技術與標準規範加以剖析，包括數位簽章的技術、數位簽章的標準與規範、公開金鑰基礎建設的標準與規範等。

3.1 數位簽章技術與標準規範

由於數位簽章是目前技術較爲成熟且廣泛被使用的電子簽章方法，因此本節所要探討的是國內外現存的數位簽章技術與標準規範。在數位簽章技術方面包括美國 National Institute of Standards and Technology (NIST)公佈的國家數位簽章標準 DSA(Digital Signature Algorithm)、蘇聯所公佈的數位簽章標準 GOST、基於離散對數問題的數位簽章法、日本的數位簽章標準 ESIGN、盲目簽章(blind signature)及代理簽章(proxy signature)，詳細剖析如附錄一。在數位簽章標準規範方面首先介紹美國 RSA 公司制定的 PKCS#1 與 PKCS#7，因爲公開金鑰密碼標準 PKCS (Public Key Cryptography Standards,)系列標準已經獲得普遍的認同與採用，國內許多業者與政府也大都引用此標準，所以包括 S/MIME、SSL、SET 等應用層協定都是以它們的基礎。接著我們會介紹美國 NIST 的 DSS(Digital Signature Standard)標準、ISO/IEC 14888 系列標準、IEEE P1363、XML 的數位簽章標準等標準規範，詳細剖析如附錄二。

在上述五種數位簽章標準中，美國 RSA 公司的公開金鑰密碼標準 PKCS

是較早被使用，也是目前最通用的標準。另外數位簽章所需的私密金鑰經常藉助 IC 卡來儲存，並且善用 IC 卡所具有之計算能力來提昇系統之安全度。

3.2 公開金鑰基礎建設

目前有關 PKI 的標準與規範大部份都是以由國際電信聯盟電信標準化部門 (International Telecommunication Union Telecom munication Standardization Sector, ITU-T) 在 1988 年所提出的 TU- T X.509 標準爲基礎，再加以擴充衍生而來的，例如 ISO (International Organization for Standardization)標準的 ISO 9594-8 認證標準、PKIX 等。

我國政府機關公開金鑰基礎建設也是依照 ITU-T X.509 標準來建置階層式(Hierarchy)的公開金鑰基礎建設。其實 X.509 只是 X.500 目錄服務(Directory Service)的一部份，而 X.509 的全名爲開放系統互連(Open System Interconnection, OSI)之目錄：認證架構(OSI-The Directory：Authentication Framework)，其主要目的也是爲了能在開放網路上互相認證。X.509 的內容主要包括認證的方式、公開金鑰憑證(Public Key Certificate)的格式、憑證路徑、金鑰與憑證的管理、憑證與憑證廢止清冊、擴充欄位等。

網際網路工程小組(The Internet Engineering Task Force, IETF)下另設的一個工作團隊–PKIX，根據 ITU–T X.509 制定 PKIX 公開金鑰基礎建設標準草案，有關 PKIX 目前的詳細進展都可以在 IETF 網站上找到。ITU-T 最早

於 1988 年公佈了第一版的 X.509，曾經被廣泛得使用，也是早期最常用的版本；此後不久，ITU-T 推出了第二版的 X.509，和第一版差別主要是多加入了使用主體單位唯一 ID(subject unique identifier)和簽發單位唯一 ID(issuer unique identifier)的概念，以避免將來使用人數大增時，使用主體或簽發單位名稱可能重複使用的問題，但是版本 2 並未得到廣泛使用。之後在 1995 年 X.509 針對對於憑證及憑證廢止清冊的定義不夠完整的問題提出修正與補充，接著更於 1997 年將新舊版本整合，即爲目前最新版的 X.509。X.509 三個板本欄位的差異，如圖 3.1 所示。

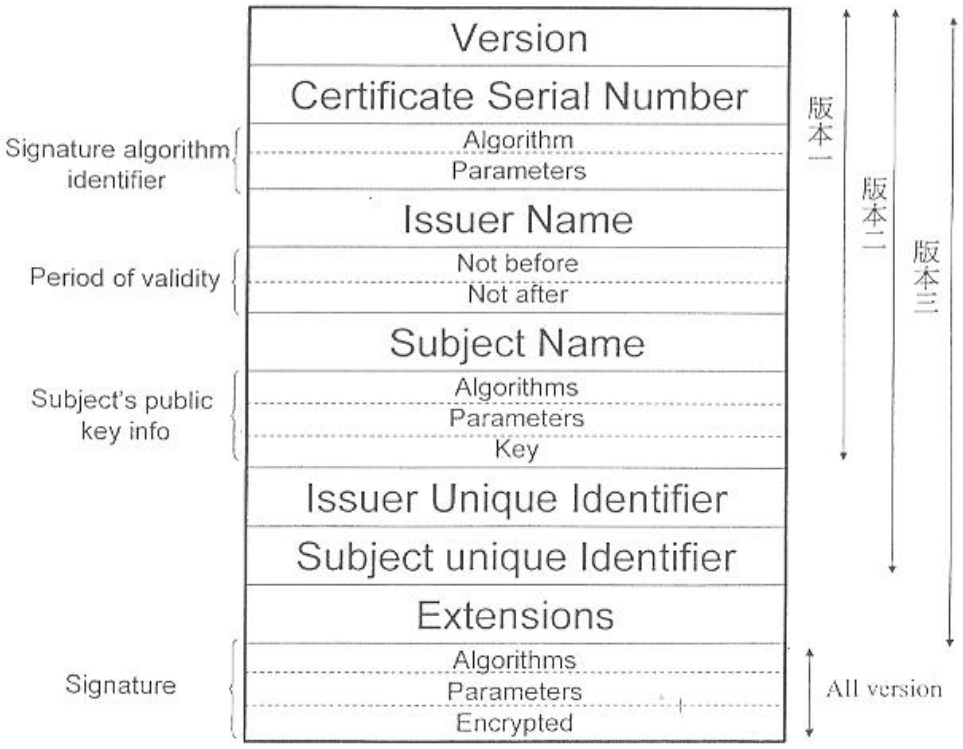


圖 3.1　X.509 三個版本欄位差異

而在這第三版的 X.509 中，主要是加入了擴充的概念，也就是任何人都可以在這新的欄位裡加入自己所需要的資訊，例如：IP 位址、電子郵件地址等，當然此版也是目前最廣泛使用的版本。

ITU-T X.509 標準是世界各國政府及業界最常引用的依據，我國的政府機關公開金鑰基礎建設也是依照此標準來建置，因此在探討政府機關的電子認證環境時不可忽略這一套標準，有關 X.509 的認證方式、公開金鑰憑證(Certificate)的格式、憑證路徑、金鑰和憑證的管理、憑證及憑證廢止清冊擴充等詳如附錄三。

第四章、主計機構間電子認證環境之建置

行政院主計處爲全國最高政府主計機關，其轄下計有三仟六佰餘主計機構，形成一個完整獨立的政府主計體系，行政院主計處掌理全國主計業務包括預算、會計、統計等，尤其是在會計事務處理與內部審核的作業流程中有太多的文件與表單(憑證)可以電子化，主計機構也是政府的一環，自然不應自外於電子化政府的推動，因此應儘速思考如何建構主計機構間電子認證之環境以及主計業務如何導入電子簽章機制，以電子文件取代書面文件，同時加速處理流程，提高施政效率。

4.1 主計機構與人員之認證

行政院主計處掌理全國主計業務與主計機構，三仟六佰餘主計機構分散在各級政府機關或事業機構中，這些主計機構大都是各級政府機關或事業機構的內部單位(例如內政部會計處、台南縣政府主計室等)，因此在政府機關公開金鑰基礎建設 GPKI 下，爲求與其他政府機關(構)或單位一致的整體考量下，宜直接採用政府憑證管理中心 GCA 簽發之政府機關(構)與單位憑證。

至於一萬四仟餘主計人員的電子憑證問題，由於主計人員大都爲政府之公務人員，經本處評估考量除了自行建置 CA 與簽發主計人員電子憑證須附加的成本(包括 CA、RA、Middleware 相關軟體體及後續的維護管理等成

本)之外，亦須考量將來與其他政府公務人員之間的交互認證的問題，更何況主計人員無論在預決算、會計事務處理與內部審核的作業流程中必須經常與非主計人員接觸，因此建議全體政府公務人員應統一簽發電子憑證，主計人員不宜自行建置 CA 與簽發主計人員電子憑證。

可是在我國電子化政府電子認證體系下，似乎並未考慮簽發公務人員電子憑證，而是在公開金鑰基礎建設 PKI 下先由內政部建置自然人憑證管理中心，簽發自然人憑證給公務人員，然後再結合授權管理基礎建設 PMI 所提供的屬性憑證(資格憑證)以證明其公務人員身分。目前內政部自然人憑證管理中心仍在建置中，授權管理基礎建設 PMI 所提供的屬性憑證(資格憑證)也還沒有明確的作法，現行電子化政府以公開金鑰基礎建設 PKI 爲設計核心的應用系統中，有公務人員參與者，皆暫時以專案的方式由原有政府憑證管理中心 GCA 簽發職務專屬憑證，例如電子支付。這對將來的系統整合，以及各級政府機關在推動電子化作業時如何取得所屬公務人員之電子憑證形成困擾，如果負責推動電子化政府的主管機關不及早正視這個問題，恐將影響電子化政府之推動與成效。

4.2 建議事項

綜合上述所言，本研究計畫提出下列建議：

(1)各級主計機構應該統一向政府憑證管理中心 GCA 申請簽發政府機關

(構)或單位電子憑證：除了行政院主計處、行政院主計處電子處理資料中心等少數主計機構爲獨立機關外，其餘全國三仟六佰餘主計機構大都是各級政府機關或事業機構的內部單位，我國政府在公開金鑰基礎建設(Public Key Infrastructure, PKI)中已成立政府憑證管理中心(GCA)負責管理與核發各級政府機關之機關(構)憑證及單位憑證，全國三仟六佰餘主計機構皆爲各級政府之機關(構)或內部單位，自然應納入此認證體系，不另自行建置主計機構憑證管理中心。

(2)有關政府公務人員如何取得電子憑證的問題應及早因應，不論直接簽發公務人員電子憑證或先由內政部自然人憑證管理中心簽發自然人憑證再加上 PMI 提供的屬性憑證(資格憑證)，均應審慎考慮，方便全國一萬四仟餘主計人員順利取得電子憑證，避免主計人員因負責多個不同應用系統而以專案方式持有多張 IC 卡(因業務而暫時簽發電子憑證及私密金鑰)的情形發生，以利主計業務電子化之推動。

(3)儲存主計機構及主計人員之私密金鑰的媒體以使用 IC 卡爲宜：早期政府憑證管理中心(GCA)簽發的自然人憑證使用磁片來儲存電子憑證及私密金鑰，其安全等級不如使用 IC 卡，因爲使用 IC 卡時所輸入之 PIN 密碼錯誤超過設定次數，如 3 次，使用者之存取權限將被鎖住。此時發卡單位可利用 unblocking PIN 密碼解鎖。若輸入之 unblocking PIN 也錯誤超過設定次數，則此張 IC 卡就報廢無法再使用。因此，導入電子簽章

機制之後，未來主計機構及主計人員處理主計業務時宜以 IC 卡儲存機構或個人之私密金鑰。

(4)主計業務要導入電子簽章機制，可分爲二個階段來執行，第一階段以主計機構爲主，目前營運中之全國主計網(eBAS)已建立起各級主計機構共用之應用平台，可經調查彙整各級主計機構之業務需求，明列那些應用系統需要使用主計機構之電子簽章，需求確定後，各級主計機構可由行政院主計處統一向政府憑證管理中心 GCA 申請簽發機關(構)或單位電子憑證，同時導入電子簽章機制。第二階段以主計人員爲主，例如會計事務處理與內部審核作業的電子化，此部分尚待公務人員取得電子憑證的問題先行解決，不宜以專案的方式暫時簽發主計人員電子憑證。

第五章、總結

我國的電子簽章法於九十年十月通過立法，隔年年四月一日正式施行，這也等同於正式宣布我國政府開始進入安全的電子化、網路化時代，同時也爲電子文件或表單取得合法的法律地位。將來主計機構在主計業務上，不但可以藉由具有法律地位的電子文件或電子表單(例如會計憑證)在網路上提供線上作業，大量減少書面紙張的印製，同時也加速處理流程，提高施政效率。

行政院主計處爲全國最高政府主計機構，掌理全國主計業務包括預算、會計、統計等，尤其是在會計事務處理與內部審核的作業流程中有太多的文件與表單(憑證)可以電子化，主計機構也是政府的一環，自然不應自外於電子化政府的推動，尤其是在金流與內部審核的電子化部分，因此應儘速建構主計機構間電子認證之環境，主計業務也應儘早導入電子簽章機制。

以目前的加解密和電子簽章技術，透過數位簽章及加密程序的使用，當然可以達到確保電子文件和電子表單的完整性、機密性、來源鑑別、不可否認性等安全需求。但是資訊科技日新月異，難保今日認爲安全的技術明日不會被破解，因此本研究計畫認爲電子簽章不適合於國家重要事件，主計業務上若有悠關國家政策之重要文件仍應以書面爲之。

參考資料

1. 政府機關公開金鑰基礎建設，http://grca.nat.gov.tw/cindex.htm
2. 政府憑證管理中心，http://www.pki.gov.tw
3. 吳宗成，資通安全技術資源簡介，國家科學委員會科學技術資料中心，2002 年 12 月。
4. 黃景彰，資訊安全—電子商務之基礎，華泰文化事業公司，2001 年 6 月。
5. 陳彥學，資訊安全理論與實務，文魁資訊股份有限公司，2000 年 12 月 初版
6. 邱榮輝，認識 IC 卡，資訊安全通訊，第四卷第三期，1998 年 6 月
7. 賴溪松、韓亮、張真誠，近代密碼學及其應用，松崗電腦圖書資料股份有限公司，1996
8. 陳彥學、高銘智，淺談 PKCS#7 及其他相關安全協定，資訊安全通訊，第五卷第三期，88 年 6 月
9. 陳彥學，數位簽章現行標準簡介，資訊安全通訊，第七卷第三期，90 年 7 月
10. 朱建達，建立於公開金鑰基礎建設的單一簽入系統論文
11. 賴溪松、葉育斌 編著，資訊安全入門
12. 樊國楨，資訊安全理論與實務，第二章密碼學演算法介紹、第四章公開金鑰憑證管理中心 X.509 簡介，2000 年 10 月
13. PKCS#1, http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/
14. PKCS#1 v2.1: RSA Cryptography Standard, June 14, 2002, ftp://ftp.rsasecurity.com/rsalabs/pkcs/pkcs-1v2-1.pdf
15. PKCS#7, http://www.rsasecurity.com/rsalabs/pkcs/pkcs-7/
16. RFC 3075, XML-Signature Syntax and Processing, http://www.ietf.org/rfc/rfc3075.txt
17. RFC 3706, Canonical XML Version 1, http://www.ietf.org/rfc/rfc3706.txt
18. RSA Security Inc. http://www.rsasecurity.com/rsalabs/pkcs/
19. IETF Secretariat http://www.ietf.org/html.charters/pkix-charter.html
20. Liang-Peng Chang(張良鵬), Sheng-Dong Liu(劉勝東), Ching-Fen Tsai, and Tzong-Chen Wu, "The Implementation of Hierarchical Authorization Using PKI-enabled Smart Card and Threshold Proxy Signature for Internet Networks", Proceeding of SCI2002, p238, ORLANDO, FL., USA 2002

附錄一、數位簽章技術

一、美國國家數位簽章標準 DSA：

1991 年，美國 National Institute of Standards and Technology (NIST)公佈 DSA 爲國家數位簽章標準。DSA 公佈後，雖引發以下爭議，但業界及學界仍是接受此一標準：

(1) DSA 不能用來做加密或金鑰分配之用，只能用來做數位簽章。

(2) DSA 是由美國國家安全局(NSA)所發展出來的一種 ElGamal 數位簽章法的變形，普遍上使用者仍是存有疑慮而擔心 NSA 藏有暗門(trapdoor)設計，並不像 RSA 或 ElGamal 數位簽章法是由學術界人士所設計出來的而較能信任。

(3) DSA 的計算速度比 RSA 要來得慢。DSA 所需簽署時間與 RSA 大約相同，但所需驗證簽章的時間要比 RSA 慢約 10 至 40 倍。

(4) RSA 雖然不是政府頒布的一項標準(牽涉到專利的問題)，但是全世界的使用者早已將之視爲一項重要的數位簽章標準來使用。

系統公開參數：

p: 512 至 1024 位元的大質數

q: 160 位元的 $p-1$之質因數

g: $g = w^{(p-1)/q} \bmod p$ ，其中 $w < p-1$且 $w^{(p-1)/q} \bmod p > 1$

h: 一個單向雜湊函數(one-way hash function)，輸出值爲 160 位元

註：搭配 DSA 的單向雜湊函數標準爲 SHA-1(Secure Hash Algorithm)

金鑰產生：

每一個使用者任選一個整數 $x \in Z_q$ 爲私鑰，並計算公鑰

$y = g^x \bmod p$ 。

簽署程序：(欲簽署訊息爲 m)

(1) 任選一數 $k < q$ 。

(2) 計算 $r = (g^k \bmod p) \bmod q$ 。

(3) 計算 $s = k^{-1}(h(m) + xr) \bmod q$ 。

(4) (r,s) 爲 m 的數位簽章。

驗證程序：

(1) 計算

$$a = s^{-1} \bmod q$$
$$b = ah(m) \bmod q$$
$$c = ra \bmod q$$
$$d = (g^b \times y^c \bmod p) \bmod q$$

(2) 若 $d = r$ ，則 (r,s) 通過驗證。

二、蘇聯數位簽章標準 GOST：

GOST 爲蘇聯於 1995 年所公佈的數位簽章標準，其做法與 DSA 很類似，但較具使用彈性，安全度也較高；但其公開的文件報告或設計原理則較不

爲人所知。

系統公開參數：

p: 介於 509 至 512 位元或 1020 至 1024 位元的一個大質數

q: 254 至 256 位元的 $p-1$ 之質因數

g: 任意小於 $p-1$ 的整數，滿足 $g^q \bmod p = 1$

h: 一個單向雜湊函數(one-way hash function)

金鑰產生：

每一個使用者任選一個整數 $x \in Z_q$ 爲私鑰，並計算公鑰

$y = g^x \bmod p$ 。

簽署程序：(欲簽署訊息爲 m)

(1) 任選一數 $k < q$ 。

(2) 計算

$$r = (g^k \bmod p) \bmod q$$

$$s = (xr + kh(m)) \bmod q$$

若 $h(m) \bmod q = 0$ ，則令 $h(m) = 1$ 。

若 $r = 0$ ，則重複回步驟(1)執行。

(3) (r,s) 爲 m 的數位簽章。

驗證程序：

(1) 計算以下參數：

$$a = h(m)^{q-2} \bmod q$$
$$b = sa \bmod q$$
$$c = a(q-r) \bmod q$$
$$d = (g^b \times y^c \bmod p) \bmod q$$

(2) 若 $d = r$，則 (r,s) 通過驗證。

三、基於離散對數問題的數位簽章法

基於離散對數問題的數位簽章法(Discrete Logarithm Signature Schemes)是以 ElGamal 方法爲主要代表。以下介紹一個一般化的離散對數數位簽章法及其六種變形。

系統公開參數：

p: 一個大質數

q: 爲 $p-1$ 或 $p-1$ 的一個大質因數

g: $1 < g < q$，滿足 $g^q = 1 \bmod p$

h: 一個單向雜湊函數

金鑰產生：

每一個使用者任選一個整數 $x \in Z_q$ 爲私鑰，並計算公鑰

$y = g^x \bmod p$。

簽署及驗證簽章：

簽署一個訊息 m 時(通常 m 需經過 h 轉換以防止選擇密文攻擊)，

簽署者首先任選一個亂數 $k \in Z_q$ ，滿足 $\gcd(k,q)=1$ ；接下來，

計算 $r = g^k \bmod p$ 與 $r' = r \bmod q$ 。

數位簽章 (r,s) 產生與驗證公式如下表所列：

簽章產生式	簽章驗證式
$r'k = s + mx \pmod q$	$r^{r'} = g^s \times y^m \pmod p$
$r'k = m + sx \pmod q$	$r^{r'} = g^m \times y^s \pmod p$
$sk = r' + mx \pmod q$	$r^s = g^{r'} \times y^m \pmod p$
$sk = m + r'x \pmod q$	$r^s = g^m \times y^{r'} \pmod p$
$mk = s + r'x \pmod q$	$r^m = g^s \times y^{r'} \pmod p$
$mk = r' + sx \pmod q$	$r^m = g^{r'} \times y^s \pmod p$

四、日本的數位簽章標準 ESIGN

ESIGN 是由日本 NTT 的 T. Okamoto 於 1990 年所發明的方法，可以視爲是日本的數位簽章標準。1999 年，ESIGN 也正式被列入 ISO 國際標準。在相同的金鑰長度與簽章大小的條件之下，ESIGN 簽署與驗證程序比 RSA 或 DSA 都要來得快速。

金鑰產生：

(1) 任選一組大質數 p 與 q (至少爲 192 位元)，並計算 $n = p^2 \times q$ ，

其中，n 爲公鑰，p 與 q 爲私鑰。

(2) 公佈一個單向雜湊函數 h 與一個整數 k，其中 $h(m)$ 的輸出值介於 0 與 $m-1$ 之間，$k \in \{8,16,32,64,128,256,512,1024\}$。

簽署程序：(欲簽署訊息爲 m)

(1) 任選一數 k。(視安全需求而定)

(2) 計算

$$w = \left\lceil \frac{(h(m) - x^k) \bmod n}{p \times q} \right\rceil$$

$$s = x + (w \times (k \times x^{k-1})^{-1} \bmod p) \times p \times q$$

(3) s 爲 m 的數位簽章。

驗證程序：

若 $h(m) \le s^k \bmod n < h(m) + 2^{2|n|/3}$，則 s 通過驗證，其中 $|n|$ 爲 n 的位元數。

五、盲目簽章(blind signature)

Chaum 於 1982 年所提出的方法，被大舉應用於電子投票、電子現金、電子支票等系統的設計，也是電子商務應用中必備的數位簽章技術。

盲目簽章主要應用於電子投票系統中選票的簽署，以防止發票者知道投票者的投票內容。另外，盲目簽章亦可應用於當簽署者的計算能力很弱時必須藉助於一個具強大計算能力的第三者或伺服器來協助簽署，但第三者或伺服器並不知道被簽署的訊息爲何。

系統參數：如同 RSA 所定義的參數，簽署者的公鑰爲(e,n)，私鑰爲 d。

簽署程序：假設 A 欲讓 B 簽署一個訊息 m，但不讓 B 知道 m

(1) A 任選一亂數 k，$1<k<n$，並計算$t=m\times k^e \bmod n$，

隨後，A 將 t 送給 B 簽署。

(2) B 簽署 t，亦即$t^d=(m\times k^e)^d \bmod n$。

(3) A 計算 m的簽章如下：$s=t^d\times k^{-1} \bmod n$，亦即

$s=m^d \bmod n$。

驗證程序：$m=S^e \bmod n$

六、代理簽章(proxy signature)

代理簽章 Mambo 等人首先於 1996 年所提出，代理簽章方法參與角色包含原始簽署者(original signer)、代理簽署者(proxy signer)與驗證者，代理簽署者在原始簽署者的授權下，可以代表原始簽署者執行簽署任務。授權可區分爲完全授權(full delegation)、部分授權(partial delegation)與授權書授權(delegation by warrant)三種類型：

(1) 完全授權：代理簽署者握有與原始簽署者相同的私鑰，其產生的代理簽章與原始簽署者的簽章相同。

(2) 部份授權：代理簽署者所持有的私鑰是由原始簽署者依自己的私鑰計算而得，代理簽署者的私鑰並不可以推導出原始簽署者的私鑰，故代理簽

署者與原始簽署者所簽署的簽章是有所不同。此類簽章又區分成兩種類型，分別爲無保護代理的代理簽章(proxy-unprotected proxy signature)與有保護代理的代理簽章(proxy-protected proxy signature)。前者意指除原始簽署者及代理簽署者以外，無任何第三者可以產生出代理簽署者的有效代理簽章；後者則意指只有代理簽署者能產生出自己的有效代理簽章。

(3) 授權書授權：由原始簽署者簽署一份授權書以證明該代理簽署者確實是在原始簽署者的授權之下執行簽署工作。

系統公開參數：

p: 大質數

g: mod p的一個原根(primitive root)

原始簽署者的私鑰及公鑰 (x, y)：$x \in Z_p$，$y = g^x \bmod p$

代理金鑰產生程序：原始簽署者執行以下步驟：

(1) 任選一亂數$k \in Z_{p-1}$，計算$K = g^k \bmod p$與$\sigma = x + kK \bmod p-1$。

(2) 將代理金鑰(σ, K)秘密傳送給代理簽署者。

代理驗證程序：代理簽署者檢驗代理金鑰的有效性：$g^{\sigma} = yK^K \bmod p$。若上式成立，則代理簽署者接受$(\sigma, K)$爲有效代理金鑰；否則退回$(\sigma, K)$並要求原始簽署者另外代理金鑰，或終止執行以下程序。

簽署程序：令m爲欲簽署訊息。代理簽署者使用σ作爲簽署金鑰，利用基

於離散對數 mod p 的數位簽章技術產生簽章 $Sig_{\sigma}(m)$，並將 $(m,\ Sig_{\sigma}(m),\ K)$ 傳送給驗證者。

驗證簽章程序：驗證者計 $y' = yK^{K} \bmod p$，將 y' 視爲原始簽署者新的公鑰，並執行簽章驗證程序以驗證代理簽章。

附錄二、數位簽章標準規範

一、美國公開金鑰密碼標準 PKCS

美國 RSA 公司制定的公開金鑰密碼標準 PKCS，在密碼技術領域中佔有舉足輕重的地位，他們所制定的一系列公開金鑰密碼標準定義與規範出各種資訊安全服務所需要的處理程序與訊息格式，由於這些標準十分完整且可行，所以受到業界與其他標準單位的採納，例如 IETF、ANSI、WAP Forum 等組織制定的標準都採用 PKCS 當作基礎，而 S/MIME (Secure/ Multipurpose Internet Mail Extensions)、SET (Secure Electronic Transaction)、SSL (Secure Socket Layer)等協定更是直接採用 PKCS 的規範。

PKCS 系列標準中，與數位簽章直接相關的標準有兩個：PKCS#1 與 PKCS#7。前者是定義底層 RSA 數位簽章或加密時的編碼程序，而後者是定義上層訊息加密或簽章的資料封裝格式。

PKCS#1 的內容主要在闡釋 RSA 演算法的基本原理、RSA 的金鑰、RSA 加密的機制、RSA 數位簽章的機制，以及數位簽章的訊息編碼方式等等。PKCS#1 的 1.0 版是在 1991 年 2 月誕生的，後續的 1.5 版被 IETF 列爲 RFC 2313，2.0 版則被列爲 RFC 2437，目前 PKCS#1 最新的版本是 2002 年六月發表的 v2.1 版。其前後版本在整體架構上並沒有太大的變化，主要都是文件編排上的調整、內容的增修，以及增加新的定義與方法。

PKCS#7 標準主要是定義密碼訊息的封裝標準，針對各種安全服務的特性，應該執行哪些密碼動作以及應該放置哪些相關資訊，規範了相對的資訊封裝格式。透過 PKCS#7 標準，發送者就能依據標準封裝經過密碼處理的資料，而接受者也可以依據格式取出所需要的資訊進行解密或驗證的動作。PKCS#7 定義了六種可能的意義：

- Data
- SignedData
- EnvelopedData
- SignedAndEnvelopedData
- DigestedData
- encryptedData

其中與數位簽章有關的兩個是 signedData 與 signedAndEnvelopedData，我們將分別說明如下。

(1) 簽章資料(Signed-Data)

當 contentType 的值爲 signedData OBJECT IDENTIFIER::={PKCS-7 2} 就代表 content 的內容是以 Signed-Data 的資料格式來封裝。這種封裝格式會包含兩部份資料，一是任何類型的「內容」，以及若干份對內容摘要進行的「數位簽章」。PKCS#7 對於「內容」本身到底是什麼並沒有限制，甚至可以是利用其他格式封裝起來的加密資料(PKCS#7 是允許巢

狀結構)。而且一份內容可以包含零個或數個數位簽章，也就是多重簽章的應用方式。如果是零個數位簽章，那麼採用這種封裝格式的用意通常是爲了傳遞數位憑證或憑證註銷列表(Certificate Revocation List, CRL)。

這個類型是採用 ASN.1 的 SignedData 資料結構來進行封裝，其詳細定義如下：

```
SignedData ::= SEQUENCE {
    version Version,
    digestAlgorithms DigestAlgorithmIdentifiers,
    contentInfo ContentInfo,
    certificates
        [0] IMPLICIT ExtendedCertificatesAndCertificates
        OPTIONAL,
  crls
        [1] IMPLICIT CertificateRevocationLists OPTIONAL,
        signerInfos SignerInfos }
```

其中 version 是表示版本編號，目前是 1；digestAlgorithms 是若干個雜湊函數的 OID，代表簽章者可以選用的雜湊函數；contentInfo 是將要被簽章的內容；certificates 是放置簽章者的數位憑證，可以是

憑證串鏈(chain)，如果是多重簽章，則可以放置多組憑證串鏈。crls 是用來放置 CRL；signerInfos 是是若干個 signerInfo 的集合，用來放置所有簽章者的相關資訊，其定義如下：

SignerInfos ::= SET OF SignerInfo

而每個 SignerInfo 可以用來存放一個簽章者的資訊。

(2) 簽章且彌封資料(Signed-and-enveloped-data)

以這種格式處理的資料會被加密並且簽章，因此不僅可以達到保密性，也同時可以達到完整性與不可否認性的功能。這種封裝格式是 Enveloped-data 與 Signed-data 兩種格式的融合。

由以上的可以了解 PKCS#1 與 PKCS#7 之間的關係，前者是將需要簽章的訊息進行編碼與實際的簽章動作，得到的結果可以依據 PKCS#7 進行包裝。而收到訊息的接收者則可以依據 PKCS#7 定義的格式取出訊息內容、簽章、簽章者的數位憑證(包含公開金鑰)等資訊，然後依據 PKCS#1 定義的方法進行驗證。當然，PKCS#7 並不限定數位簽章的方法，因此您也可以使用其他數位簽章演算法，得到的結果再依循 PKCS#7 的規範進行封裝。

二、美國 NIST 的 DSS 標準

除了 RSA 的公開金鑰密碼標準 PKCS 之外，美國 NIST 在 FIPS PUB 186 所制定的數位簽章標準(Digital Signature Standard, DSS)是另一個目前

較廣被採用的標準，日韓等國的數位簽章國家標準也都是參考 DSS 制定出來的。相較於 RSA 可以用在加密與數位簽章，DSS 只能用在數位簽章的用途上。

DSS 所使用的演算法是 DSA (Digital Signature Algorithm)，這個演算法主要是結合 Elgamal 與 Schnorr 兩種演算法而來，主要是植基於解離散對數(discrete logarithm)的數學難題，而 RSA 則是植基於因數分解(factorization)。

除此之外，DSA 和 RSA 還有一個很大的不同，那就是 DSA 的簽章結果是一組數對 (r, s)。其中 r 爲先前簽章(pre-signature)，這部分的輸入值只有一個隨機產生的亂數，因此與要被簽署的訊息內容無關，因此這部分的計算是可以事先完成並且將結果儲存起來。而 s 的計算，輸入值則包括 r 以及訊息之雜湊值。包含先前簽章是 DSA 一個非常重要的特性，因爲 DSA 演算法的計算量偏重在 r 的計算(高指數模數運算)，而這部分的計算可以事先準備好，因此在真正在執行簽章動作的時候，節省許多運算量。

三、ISO/IEC 14888 系列標準

ISO/IEC 14888 (1998 年)的主旨在定義數個具附件之數位簽章(signature with appendix)機制，它也被我國中央標準局列爲 CNS 標準(CNS_ISO 14888-x)中。ISO/IEC 14888 主要包含三個部分：

- ISO/IEC 14888-1：總則，包含具附件之數位簽章機制的原則與要求，

以及其他兩個部分常用到的符號與定義

- ISO/IEC 14888-2：身分基底機制(identity-based mechanisms)
- ISO/IEC 14888-3：憑證基底機制(certificate-based mechanisms)

所謂具附件的意思是執行簽署後的結果必須和原始訊息附在一起，接收者才能帶入適當的函數進行驗證(另一種形式是 message-recovery 的數位簽章機制)。而身分基底與憑證基底的差異在於公開金鑰的處理，前者的公開金鑰可以依據身分識別碼導出，而後者是採用數位憑證的方式來傳遞與檢驗簽章者的公開金鑰。目前在實務的應用上，主要是採用憑證基礎的方式。

四、IEEE P1363

IEEE P1363 標準的主旨在規範通用的公開金鑰密碼學原理，以及植基於這些原理所建構出來的密碼學機制。因此這份標準的走向比較類似於 PKCS#1，不同之處在於 PKCS#1 是專門針對 RSA 加密與數位簽章機制，而 IEEE P1363 還包含了「植基於離散對數」與「植基於橢圓曲線」兩類密碼學原理，而且除了加密與數位簽章，P1363 裡還包括金鑰協議機制。

五、XML 的數位簽章標準

XML(eXtensible Markup Language)是一種可延伸的標記式語言，且XML文件是屬於文字檔，因此無法使用先前介紹過的標準在XML文件內加入數位簽章資訊。因此IETF(Internet Engineering Task Force)在2001年三月制定了RFC 3075作爲XML使用數位簽章的標準，在一份XML文件中，數位簽章以及相關資訊應該放置在Signature元素中。

在上述五種數位簽章標準中，美國RSA公司的公開金鑰密碼標準PKCS是較早被使用，也是目前最通用的標準。

附錄三、公開金鑰基礎建設標準與規範

我國政府機關公開金鑰基礎建設也是依照 ITU-T X.509 標準來建置階層式(Hierarchy)的公開金鑰基礎建設。其實 X.509 只是 X.500 目錄服務(Directory Service)的一部份，而 X.509 的全名爲開放系統互連(Open System Interconnection, OSI)之目錄：認證架構(OSI-The Directory：Authentication Framework)，其主要目的也是爲了能在開放網路上互相認證。X.509 的內容主要包括認證的方式、公開金鑰憑證(Public Key Certificate)的格式、憑證路徑、金鑰與憑證的管理、憑證與憑證廢止清冊、擴充欄位等。

一、X.509 認證的方式

X.509 標準提出了簡單型和增強型兩種認證的方式，簡介如下：

(1) 簡單型認證(simple authentication)

這是一個安全度較低的認證的方式，互相溝通的雙方僅僅靠一組使用者名稱和密碼來完成之間的認證，而這認證也是存取控制唯一的依據，在 X.509 中提出了三種簡單型認證的實作方式：

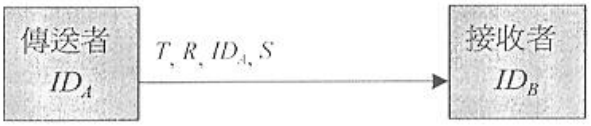
(a) 使用者直接把使用者名稱和密碼，以明文的方式傳送到接收方。

(b) 先將使用者的密碼及代號資料加上一隨機亂數(random number)或時戳後，再經由單向雜湊函數(one-way hash function)保護後送給對方，如圖一所示。

(c) 將(b)方法的結果加上另一組隨機數或時戳資料，經由雜湊函數運算後，再送給對方。

由於傳送者的密碼是是經過單向函數保護後才傳送，因此在傳輸的過程中是安全的；而加入隨機亂數或時戳的機制，即使傳送相同資料亦會有不同的輸出結果，因此可有效減少重送攻擊(replay attack)的威脅。

當接收者收到資料後的執行以下步驟驗證：



ID_A, ID_B: 使用者代號
PW_A: 使用者密碼
T: 時戳
R 隨機亂數(random number)
S: $h(ID_A, T, R, PW_A)$，保護資料

圖一 簡單型認證

(i) 找出使用者 ID_A 的相對密碼

(ii) 將 ID_A、T、R 及密碼經過單向函數運算

(iii) 比較運算後的數值和收到的保護資料，若是相同，則認證通過；反之，則認證失敗。

在實作上，方式(b)和(c)是大同小異，只是(c)多做一次的單向函數運算。另外在 X.509 中有提到(c)的兩次單向函數相不相同皆可，並沒有

硬性規定。

(2) 增強型認證(strong authentication)

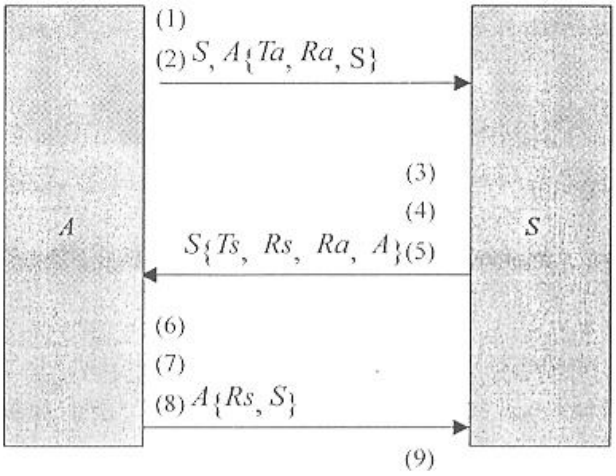
若是在一個封閉的環境中，且對安全的要求不高，則簡單型認證以符合基本安全需求；但若在開放式的網際網路上，簡單型認證似乎無法滿足安全需求。增強型認證主要是靠公開金鑰密碼系統的觀念，因此在討論增強型認證之前，必須要有一個假設：PKI 架構的完成，如此就能靠憑證的驗證來確認對方的身份。增強型強認證程序又分爲「單向」、「雙向」與「三向」三種認證方式，其差別在於提供不同安全程度的認證。

(a) 單向認證 (如圖二的步驟 1~3)

步驟 1：使用者 A 先產生一個不重複的 Ra (在 X.509 中建議此 Ra 可以包含兩個部份，一是循序累加的部份，二是隨機選取的部份。)

步驟 2：使用者 A 傳送簽章 $S, A\{Ta, Ra, S\}$ 給接收者 S。($A\{x\}$ 表示 A 用私鑰對 x 做簽章；Ta 可以包含一至二個時間，一是資料產生的時間，二是資料逾期時間。)

步驟 3：當 S 收到簽章後，驗證 A 的憑證是否有效、簽章的完整性、Ta 是否逾期、Ra 是否重複。若驗證成功，則表示 A 爲一個合法的使用者，但是 A 無法確認 S 的身分。



圖二　增強型認證

(b) 雙向認證 (如圖二的步驟 1~6)

雙向認證可同時確認接送雙方的身分，而整個流程除了前三個步驟和單向認證相同外，還多出三個步驟：

步驟 4：接收方 S 先產生一個不重複的隨機亂數 Rs

步驟 5：S 傳送簽章 $S\{Ts, Rs, Ra, A\}$ 給 A

步驟 6：當 A 收到簽章後，驗證 S 的憑證是否有效、簽章的完整性、Ts 是否逾期、R_S 是否重複等。若驗證成功，則表示 S 爲一個合法的使用者。

c. 三向認證 (圖二的步驟 1~9)

在 Ta、Tb 裡包含了逾期的時間，但在傳送的過程可能因爲網路塞車、網路電腦時間不同步等問題造成資料延遲，而使得認證失效，因此

三向認證的目的就是要解決這樣的問題，即收、送雙方不需檢查時間戳記，只要檢驗隨機數的正確與否即可。此方法雖然多了一些步驟，但技術上卻是比較可行的。

三向認證包括了九個步驟，步驟 1~6 和雙向認證相同，但可是不用檢查時間戳記，甚至不需要時間戳記，而剩下的步驟如下：

步驟 7：送方 A 收到資料後，檢查 Ra 是和自己在步驟一所產生的隨機數相同。

步驟 8：A 傳送簽章 $A\{Rs, S\}$ 給 S。

步驟九：當 S 收到簽章後，檢查 Rb 是否和步驟 5 所產生的隨機亂數相同。

二、公開金鑰憑證(Public Key Certificate)的格式

在一個公開的、不安全的網路環境中，如何能達到秘密通訊與身分鑑別目的，在 X.509 標準以憑證的概念，解決此一問題。首先，系統中須存在一個公開公正可信賴的第三者(Trust Third Party, TTP)，一般稱之爲憑證中心或認證中心(Certification Authority, CA)，負責產生、管理並維護通訊個體的公鑰憑證。首先由 CA 核對申請者身份、資料後，再用自己的私鑰對這位申請者的身份資料和相對應的公開金鑰做簽章的動作，產生一個數位簽章。所謂的相關資料包括了使用者名稱、使用者 ID、使用者對應的公

開金鑰、發行者資料等，而把這些相關資料和剛剛的數位簽合起來，就是一張憑證了。X.509 中提及憑證須符合兩點特性：

(1) 所有可以取得認證中心公鑰的使用者，可以找到任何已經認證過的公開金鑰。除認證中心之外，任何人皆無法擅自修改憑證內容。因此，當憑證經 CA 的私密金鑰簽章後，任何人皆可用 CA 的公鑰來驗證憑證的有效性。在 X.509 中所定義的憑證內容爲：(可參考圖 3.1)

CA<<A>>=CA { V,SN,AI,CA,UCA,A,UA,Ap,TA }

而各個符號所代表的意義如下：

(a) 版本(Version, V)： 憑證格式的版本。

(b) 序號(Serial Number, SN)： Issuer CA 對每一憑證所發給的唯一序號。

(c) 演算法(Algorithm Identifier, AI)：簽發此憑證所採用的演算法。

(d) 發證者(Issuer, CA)：簽發此憑證的 CA 名稱。

(e) 發證者識別號(Issuer Unique Identifier, UCA)： 此發證 CA 的唯一識別碼。

(f) 使用者(Subject, A)：此公鑰擁有者的名稱。

(g) 使用者識別號(Subject Unique Identifier, UA)： 此公鑰擁有者唯一的識別碼。

(h) 公鑰資料(Subject's Public Key Information, Ap)： 此憑證中的公鑰

以及公開金鑰演算法名稱。

(i) 憑證有效期限(Period of Validity, TA)：此憑證在啓用日與逾期日期之間有效。

在 X.509 中，除了定義了上面的資料欄位外，還使用了 ASN.1 的規範描述了這些憑證的資料格式。所有的內容如下：

```
Certificate  ::=  SEQUENCE  {
    tbsCertificate          TBSCertificate,
    signatureAlgorithm      AlgorithmIdentifier,
    signatureValue          BIT STRING  }
TBSCertificate  ::=  SEQUENCE  {
        version             [0]  EXPLICIT Version DEFAULT v1,
        serialNumber             CertificateSerialNumber,
        signature                AlgorithmIdentifier,
        issuer                   Name,
        validity                 Validity,
        subject                  Name,
    subjectPublicKeyInfo SubjectPublicKeyInfo,
    issuerUniqueID  [1]  IMPLICIT UniqueIdentifier OPTIONAL, -- If present, version MUST be v2 or v3
    subjectUniqueID  [2]  IMPLICIT UniqueIdentifier OPTIONAL, -- If present, version MUST be v2 or v3
    extensions      [3]  EXPLICIT Extensions OPTIONAL, -- If present, version MUST be v3  }
Version  ::=  INTEGER  {  v1(0), v2(1), v3(2)  }
```

```
CertificateSerialNumber  ::=  INTEGER
Validity ::= SEQUENCE {
    notBefore        Time,
    notAfter         Time }
Time ::= CHOICE {
    utcTime          UTCTime,
    generalTime      GeneralizedTime }
UniqueIdentifier  ::=  BIT STRING
SubjectPublicKeyInfo  ::=  SEQUENCE  {
    algorithm              AlgorithmIdentifier,
    subjectPublicKey       BIT STRING  }
Extensions  ::=  SEQUENCE SIZE (1..MAX) OF Extension
Extension  ::=  SEQUENCE  {
    extnID         OBJECT IDENTIFIER,
    critical       BOOLEAN DEFAULT FALSE,
        extnValue      OCTET STRING  }
```

若將上述的憑證格式和 ASN.1 的描述互相比對一下，就更容易了解 X.509 中的憑證內容了。

三、憑證路徑

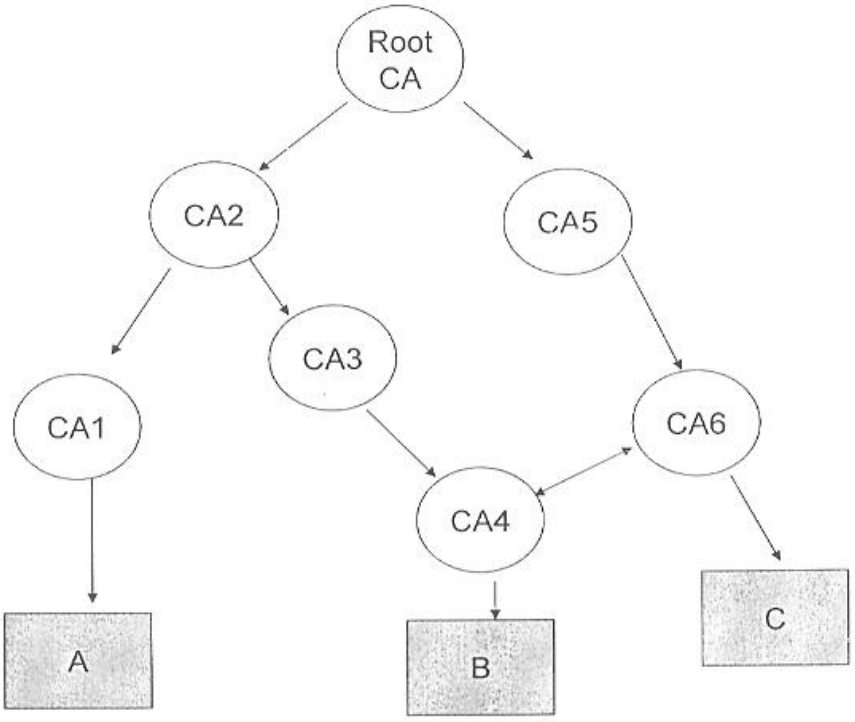
由於 CA 不可能只有一個，假設通訊個體 A、B 要通訊，但兩個人的憑證並不是由同一個 CA 所發，也就是說 B 的 CA 並非 A 所信任的，那 A 該不該 B 憑證呢？

在 X.509 中提到了"憑證路徑(certification path)"這樣的概念，即憑證的接收方可以藉由驗證 CA 的電子憑證方式，一步一步的重複驗驗追蹤，直到一個接收方可以信賴的 CA 爲止，即 Root CA。則 CA 之間宛如是一串鍊子所串成，即爲一個憑證鍊(Certificate Chain)。如此一來，即使溝通雙方分屬不同 CA，也能用這樣的方式來驗證該憑證的正確性與否。

以圖三所示，若使用者 A 的信任 CA 爲 CA1，而使用者 B 的憑證爲 CA4 所簽發，則當使用者 A 由目錄服務中獲得以下的憑證路徑時，CA1 <<CA2 >>,CA2<<CA3 >>,CA3 <<CA4>>,CA4 <<B>>，則使用者 A 就可憑這一串憑證路徑，利用使用者 A 所熟知的 CA1 公鑰，驗證 CA1 對 CA2 所發的公鑰憑證之簽章是否正確(CA1 <<CA2 >>)，若是正確，即可驗證得知 CA2 之公鑰爲可信賴的公鑰；接著以 CA2 的公鑰驗證 CA3 公鑰之正確性，如此不斷的重複，最後可驗證得到 CA4 公鑰的正確性，然後再以 CA4 的公鑰驗證 B 的憑證(CA4<<B>>)，便可知道 B 的公鑰是否可以信賴。

另一個情況爲，當使用者 B、C 要通訊，那是否意謂著使用者 B 必須依序透過 CA4、CA3、CA2、Root CA、CA5、CA6 的驗證，要是 CA 的整個架構比圖三複雜許多，如此的驗證是否太沒有效率可言？爲此，在 X.509 中有提到交互簽證的概念，也就是各個 CA 之間是可以互相簽證了，如此一來，若是 CA4、CA6 有互相簽證，那使用者 B 所得到的憑證路徑爲 CA4 <<CA6>>,CA6 <<C>>，縮短了許多。換句話說，使用者或是目錄服務也都

因此而減少了許多負擔、提升了效率。



圖三　憑證路徑

當然此驗證模式並非只適用如圖三般的階層式架構，在 X.509 中除了有階層式認證中心架構的描述，也提到了非階層式認證中心的架構，而兩者架構的運作模式其實也是大同小異，再看看第二個例子中的 CA4、CA6 互相簽證，其實就算是非階層式認證中心的架構了。除了交互簽證外，爲了降低使用者或是目錄服務的負擔，X.509 還提到了幾項原則：

(1) 若使用者 A 與 B 的憑證同屬一個 CA，則 A 只要對 CA<<B>>直接驗證即可，B 亦只要驗證 CA<<A>>。

(2) 若爲階層式認證中心架構，使用者可以事先儲存本身到 Root CA 的順/反向憑證，則後來就只要取得對方到共同信任 CA 路徑即可。(如圖三所示，當使用者 A 欲與 B 通訊，則只要在取得 CA3、CA4 之憑證即可。)

(3) 若使用者 A 經常和某一 CA 所簽發的使用者通訊，則 A 可事先儲存本身與該 CA 之間的順/反向憑證。

(4) 若使用者 A 與 B 經常通訊，則可將對方的憑證儲存下來，即可免向目錄服務要求服務。

四、金鑰和憑證的管理

(1) 金鑰管理

在向憑證中心申請憑證之前，當然先要有金鑰，在 X.509 的本文中提到了三種金鑰的產生方式：

(a) 由使用者自行產生：此種方式的優點就是使用者的私密金鑰只有自己知道，但使用者產生金鑰的能力可能有限。

(b) 由 TTP 產生：在 TTP 產生金鑰後，透過安全的管道將私密金鑰傳送給使用者，然後將和產生金鑰有關的資料都銷毀。

(c) 由 CA 產生: 此種方式是(2)的特例，優點就是不用額外再找一個第三者，因爲 CA 本來就是使用者所相信。

由於金鑰大都是一串很長的數字，不易爲人類所記憶，因此 X.509 也建議使用智慧卡(smart card)等儲存媒體，並用通行碼保護資料的安全；此外，如何避免產生安全度就低的弱金鑰也是相當重要的課題。

(2) 憑證管理

在整個 X.509 的描述中，是以憑證爲核心，而憑證的管理，舉凡憑證的簽發、使用者資料的保密、攻擊的防範、到憑證的廢止等，每一個環節都是非常重要的。以下條例出 X.509 中幾項憑證管理的內容：

(a) 憑證中心在替使用者簽發憑證之前，必須確實核對使用者的身分，而在整個 PKI 架構中，審查身份通常是由另一個機構，註冊中心(Regulatory Authority, RA)來負責的，以降低憑證中心的負擔。

(b) 憑證中心不可以簽發相同的憑證給不同的使用者。

(c) 憑證中心以離線的運作方式簽發憑證，以減少被干預，甚至被攻擊的可能。

(d) 使用者將公鑰以及一些相關資料交給憑證中心簽證時，其傳送的管道必須要確保認證性與完整性，若是資料遭到竄改，將會嚴重的影響使用者本身的權益。

(e) 憑證中心將簽完後的憑證送到目錄服務的過程中，不需要特別的保護，因爲簽發憑證的同時，即是運用了公開金鑰密碼系統的原理，若是傳送過程有誤或是資料遭到竄改，就會被發現。

(f) 憑證內包含了有期限標示，憑證中心可以在過期之前便產生新接續的憑證，以提高效率。而針對第 6 點的更換新憑證，在 X.509 中也提出了兩項的建議：

- 憑證中心可視情況不同，而訂定各自的策略。至於新舊憑證的期限是否要重疊，亦無硬性規定，若是新舊憑證的期限不能重疊，那可能會造成同一天內失效後，必須要重新簽發相當多的憑證，增加憑證中心的困擾。
- 已經過期的憑證，必須要從目錄服務中移除。若是憑證中心有提供不可否認的服務，則這些過期的憑證仍需保留一陣子，以防將來有糾紛時，可以檢查。

再者，雖然每一個憑證內都包含有效期限，到期後便會自動失去效力；但若在有效期限內遇到一些特殊狀況，而可能會導至憑證的提早廢止失效。所以在 X.509 中也特別提到了，當以下的情形發生時，該憑證必須提前做廢止的動作，也就是所謂的憑證廢止(Certificate Revocation)：

(a) 系統參與者遺失或洩漏其私秘金鑰。

(b) 使用者不再由原來的憑證中心爲其作簽證時。

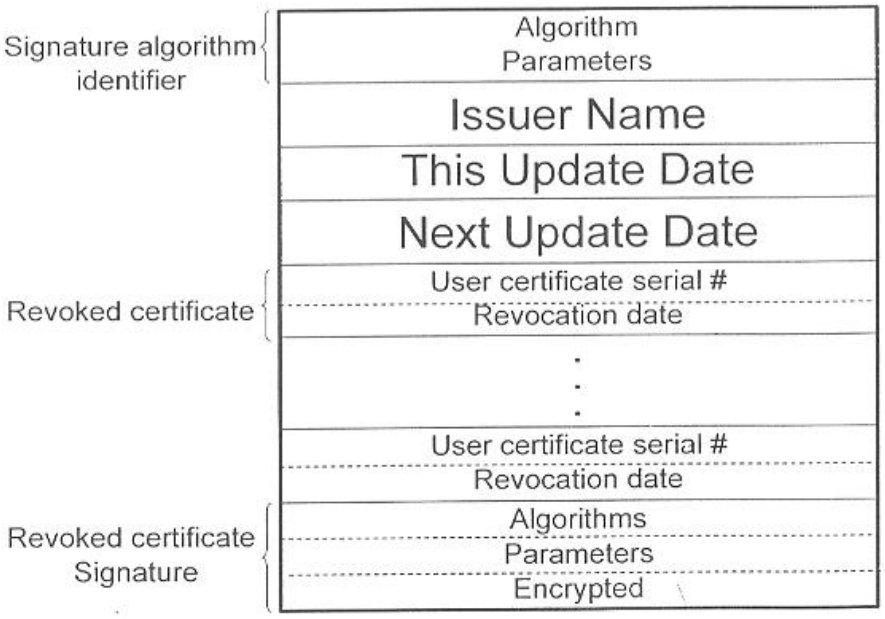
其中若是情況(a)發生時，則可能對系統、使用者造成很大的威脅、破壞，因此在 X.509 本文中建議了兩種廢止憑證的方法：

- 即刻廢止法(immediate revocation)：直接通知所有的系統使用者，

並告訴所有使用者他的新憑證。

- 緩慢廢止法(slow revocation)：憑證中心須額外管理兩個廢止清單，第一個記錄了由此憑證中心所簽發，未過期而被廢止的憑證，並要加上時間戳記，此清單需經過憑證中心的簽名，稱之爲 RCL (Revoked Certificates List)；第二個清單則是記錄了所有與該憑證中心相關，並被廢止憑證，此清單同樣要經此憑證中心簽名，而稱爲 ARL (Authority Revoke List)。不管 RCL 與 ARL 兩個廢止清單是否爲空，都必須存在。而該憑證中心亦將 RCL 與 ARL 兩個廢止清單直接放在目錄服務上供取用。

在 X.509 中所定義的廢止憑證內容如圖四所示。



圖四　廢止憑證內容

而各欄位的意義分別爲：

Issuer Name：憑證中心名稱。

This Update Date：此次發佈資料的時間。

Next Update Date：下次要發佈的時間。

Revoke certificate：廢止的憑證序號、時間。

Signature：憑證中心用自己的私密金鑰對這個廢止憑證所作的運算結果，包含所用的演算法。

而使用 ASN.1 規範描述的這個廢止憑證清單資料格式如下：

```
CertificateList  ::=  SEQUENCE  {
    tbsCertList              TBSCertList,
    signatureAlgorithm       AlgorithmIdentifier,
    signatureValue           BIT STRING  }
TBSCertList  ::=  SEQUENCE  {
    version                      Version OPTIONAL,
                                  -- if present, MUST be v2
    signature                    AlgorithmIdentifier,
    issuer                       Name,
    thisUpdate                   Time,
    nextUpdate                    Time OPTIONAL,
    revokedCertificates      SEQUENCE OF SEQUENCE  {
            userCertificate          CertificateSerialNumber,
            revocationDate            Time,
            crlEntryExtensions        Extensions OPTIONAL
```

```
                                    -- if present, MUST be v2
                                }   OPTIONAL,
    crlExtensions           [0]   EXPLICIT Extensions OPTIONAL
                                -- if present, MUST be v2 }
```

同樣的，同時比對圖四和上面的 ASN.1 描述，可以對這些憑證欄位的用法更清楚。

五、憑證及憑證廢止清冊擴充

由於憑證的使用逐漸廣泛，加上很多網路上的商業行爲都必須要憑證架構的支援，而 ITU 最早於 1988 年所制定的 X.509 似乎不敷使用，且憑證及憑證廢止清單的定義並不完整，同時爲了讓憑證的使用更有彈性，因此在 1997 提出了新版的文件。

憑證廢止清單的主要目的是爲了讓使用者可以查驗未到期但已失效的憑證，並由憑證中心定期產生，憑證中心亦有義務維護憑證廢止清單，而在基本憑證廢止清單擴充部分，主要新增四個欄位：憑證廢止清單編號(CRL number)、註銷原因(Reason code)、註銷指令(Hold instruction code)、不合法日期(Invalidity date)。而各新增欄位的目的、功能如下：

(a) 憑證廢止清單編號(CRL number) ：讓使用者可以清楚的知道是否有遺漏任何 CRL。

(b) 註銷原因(Reason code)：讓使用者能進一步的了解該憑證註銷的原

因。

(c) 註銷指令(Hold instruction code)：讓使用者了解後，做出相對應的動作。

(d) 不合法日期(Invalidity date)：記錄私鑰可能洩露的時間。

ITU-T X.509 標準是世界各國政府及業界最常引用的依據，我國的政府機關公開金鑰基礎建設也是依照此標準來建置，因此在探討政府機關的電子認證環境時不可忽略這一套標準。