第七章 數位簽章與數位憑證

您知道嗎？網路也可以發送存證信函！順便告訴您一聲，您昨天所收到升經理的公文，是志明兄整您的，您發現了嗎？

『網路身分證』; 夠新奇了吧！有了它便可以在網路上通行無阻嗎？它是否可偽造？又誰可以認證它的正確性呢？

7-1 數位簽章簡介

7-1-1 文件的印章/簽名

『數位簽章』(Digital Signature, DS) 是公開鑰匙系統中最重要的應用之一。它具有確定傳送訊息內容的『完整性』與傳送端的『不可否認性』功能；乍看之下似乎與『訊息確認碼』(MAC) 的功能非常類似，的確如此，只不過數位簽章的應用較 MAC 廣泛。MAC 主要以雙方共享的秘密鑰匙來確認彼此身份，所以通訊雙方事先已經約定好一把秘密鑰匙、或已協議完成一把共享鑰匙；至於數位簽章則應用在『陌生環境』之下與『陌生人』之間的通訊，除了期望能表明自己的身份之外，並且可以保證所攜帶的訊息確實是自己所傳送的。在與陌生人通訊的情況之下，雙方未能確定對方身份之前，是不可以貿然建立共享鑰匙的，因此，如何確定對方身份可說是數位簽章最主要的功能。

數位簽章是一連串的二進位數字，它的功能如傳統的『簽章』一般。傳統式的簽章，是在文件上加蓋『印章』或『簽名』並作為發文者對該文件負責之用。然而，以目前的科技而言，要仿冒印章或簽名似乎沒也什麼困難，因此，許多單位 (如銀行) 都會要求申請人親自辦理，及當面簽名蓋章以確保其真實性 (並且錄影存證)。但在虛擬網路上交易，親自辦理似乎不太可行，為了確定交易不至遭他人冒用，則必須仰賴數位簽章的輔助。首先，欲在網路上從事交易的人 (或組織單位)，必須擁有一對公開鑰匙系統所建構的公開鑰匙與私有鑰匙配對；對內而言，使用私有鑰匙的功能如同印章一樣，可在文件上簽署以證明它的合法性；對外而言，使用公開鑰匙則代表自己的身份，可用來證實自己所簽署文件的真偽。既然私有鑰匙如同印章一樣，則必須將它妥善收藏，免得被他人

拿去盜印文件；至於公開鑰匙則儘量公佈讓眾人知曉，以便他人來驗證簽章（私有鑰匙所簽署的）。也就是說，他人可利用發行者的公開鑰匙，來驗證發行者利用私有鑰匙所簽署的文件。

數位簽章的功能常被嵌入於電子郵件系統裡，亦即電子郵件軟體會針對寫好的信件（無論公文或私人信件），作數位簽章的處理，並將簽署訊息放在文件的後面一起傳送給收信者；收信者接收到之後，郵件軟體便由信箱內的鑰匙串列找出發信者的公開鑰匙，除了驗證該信件是否遭他人修改過之外，還需確定發信者身份未遭受他人冒名頂替。

7-1-2 數位簽章架構

從技術層面來看，數位簽章必須達到『完整性』與『不可否認性』的功能。圖 7-1 為數位簽章範例（RSA 簽章），其運作程序說明如下：首先傳送端將訊息經過雜湊演算法計算後得到一個雜湊值，再利用它的私有鑰匙向雜湊值加密成為一個數位簽章（DS），接著，再將數位簽章附加在訊息後面一併傳送出去；接收端收到訊息之後，以同樣的雜湊演算計算出雜湊值（H'），並利用傳送端的公開鑰匙將 DS 解密，得到另一端的雜湊值（H）。接著，比較兩個雜湊值，如果相同的話，則可以確定該訊息的『完整性』（雜湊值相同），此外也可以確定其『不可否認性』（私有鑰匙與公開鑰匙配對）。

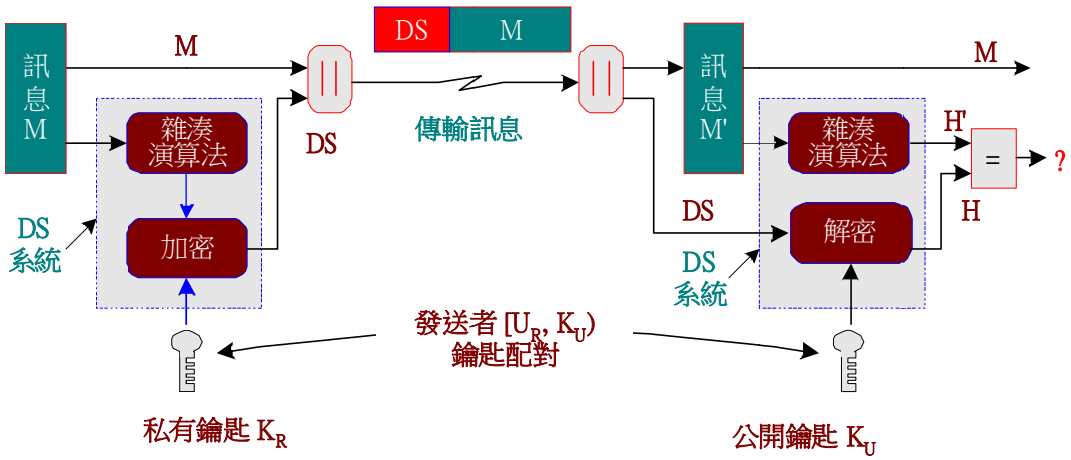

圖 7-1 數位簽章架構圖

由圖 7-1 可以看出數位簽章系統主要是由公開鑰匙系統與雜湊演算法兩者所構成，目前使用較普遍的有 RSA 數位簽章與 DSS 數位簽章，以下分別介紹這兩套系統。

7-2 RSA 數位簽章

利用 RSA 公開鑰匙系統製作出來的數位簽章機制，就稱為『RSA 數位簽章』(RSA Digital Signature)，其實 RSA 演算法一開始就是因為數位簽章的需求而發展出來的，有關 RSA 數位簽章的運作程序，請參考圖 7-1 所示。至於 RSA 演算法已在第四章詳細介紹過，在此僅就演算法植入數位簽章作簡單敘述。RSA 數位簽章的公開鑰匙與私有鑰匙製作如下：

- 選出兩個較大的質數 p 和 q。
- 計算兩個質數的乘積 $n = p \times q$。
- 計算出小於 n 且與 n 互質的整數個數$\psi(n) = (p-1)(q-1)$。
- 選出一個質數 e (一般可固定為 3)，$1 < e < \psi(n)$；使 e 與$\psi(n)$ 互質，亦即 $gcd(\psi(n), e) = 1$。
- 計算出 $d = e^{-1} \bmod \psi(n)$。
- 公開鑰匙 $K_U = \{e, n\}$。
- 私有鑰匙 $K_R = \{d, n\}$。

7-2-1 RSA 簽章演算法

RSA 數位簽章可選擇不同的雜湊演算法來實現，譬如 MD4、MD5 或 SHA-1。將訊息經過雜湊函數計算後，再經過 RSA 演算法加密 (私有鑰匙)，便可得到數位簽章的簽署碼；接收端將此數位簽章碼解密後 (公開鑰匙)，與接收到的訊息所計算出的雜湊值比較，便可辨其『完整性』與『不可否認性』。RSA 數位簽章的演算如下：(如圖 7-1 所示)

- 傳送訊息： M
- 雜湊演算法：H
- 加密演算法：$E_K[M] = (M)^d \bmod n$
- 解密演算法：$D_K[C] = (C)^e \bmod n$
- 傳送端計算雜湊碼：H(M)
- 數位簽章：$Sig(M) = E_{KR}[H(M)] = (H(M))^d \bmod n$
- 接收訊息：M'

- 接收端計算雜湊碼：H(M')
- 驗證簽章：$Ver(Sig(M)) = D_{KU}[E_{KR}[H(M)]] = ((Sig(M))^e \bmod n = H(M)$

  如果 H(M) = H(M')，則確認成功；否則確認失敗。

7-2-2 RSA 安全性考量

數位簽章的安全性考量，與訊息確認碼（MAC）一樣，必須考慮到破解鑰匙的暴力攻擊法、以及偽造訊息的生日攻擊法。但對於數位簽章有一個更重要的特點，它的訊息主要以明文方式傳送（雜湊演算法也是公開的），而且簽署後的簽章碼（Sig(M)）必須保持一段長時間的使用（一般都是一年）；在這一段時間內，破解者可以利用已知的明文來測試各種可能發生的鑰匙（暴力攻擊法），並比對現有的簽章碼。

數位簽章的暴力攻擊法，亦如同於 MAC 的攻擊方式，如 6-6-1 節所示。攻擊者必須收集多筆明文與簽署碼的配對（{M, Sig(M)}，才可以嘗試出加密的私有鑰匙；然而，利用私有鑰匙簽署的訊息並不多，因此，也很難利用暴力攻擊法來搜尋出私有鑰匙。但話說回來，如果同一把鑰匙簽署多筆文件之後，他被破解的機率也就越高，這也是 CA 中心（7-6 節介紹或第九章介紹）必須擁有多組鑰匙配對的原因。另一方面，以密碼學的概念而言，並沒有無法破解的加密演算法，祇不過破解的效益是否合乎『計算上的成本』而已。由此可見，簽署數位簽章的鑰匙必須足夠長，以增加破解所需要的成本，進而減低破解者的意願。增加鑰匙長度最起碼的條件是，RSA 演算法中的質數 p 和 q 必須夠大，使得要利用他們的乘積 n，分解出相對應的 p 與 q 變得非常困難。一般就短時間（大約一年）的安全性考量，n 不得少於 1024 個位元；至於長時間的安全性考量，n 必須在 2048 位元以上。

對於生日攻擊法而言，攻擊者並不需要去破解加密鑰匙，只要能找出另一個偽造明文，而能得到出相同的簽署碼，就算破解成功（如 6-6-2 節說明）。因此，增加簽署碼的長度，即是抗拒生日攻擊法的不二法門；而延長簽署碼長度的方法，便是選用較長雜湊值的演算法。

另外，採用 RSA 數位簽章時必須注意下列事件，才不容易被詐騙或破解：

1. 執行數位簽章與文件加密的鑰匙，不可以使用同一對公鑰與私鑰配對。

2. 執行數位簽章時，必須針對文件的雜湊值加密，而不可以直接對文件明文加密。
3. 不可以對亂數做數位簽章。
4. 不同鑰匙配對之間不可以有相同的模數（n）。
5. 必須將明文填補（亂數或 0）到與模數（n）相同長度之後，再執行數位簽章。
6. 需加密的文件，必須先做數位簽章，再做訊息加密（第十二章說明）。

7-3 DSS 數位簽章

『數位簽章標準』（Digital Signature Standard, DSS）是美國 NIST（National Institute Standard and Technique）於 1994 年所制定的數位簽章標準，此標準制定了『數位簽章演算法』（Digital Signature Algorithm, DSA），並於 2000 年修改成 FIPS PUB 186-2，除了可以選用 RSA 演算法，還增加『橢圓曲線數位簽章演算法』（Elliptic Curve Digital Signature Algorithm, ECDSA）。本書限於篇幅，僅介紹 DSA 演算法；至於 ECDSA 演算法請參考其它文獻 [79, 89, 98]。在 DSS 標準上，也可分別選用不同的雜湊演算法，一般都建議使用 SHA-1 演算法（請參考 5-6 節介紹）。

7-3-1 DSA 演算法

『數位簽章演算法』（DSA）僅供數位簽章使用，無法應用於其他地方，這一點與 RSA 演算法迥然不同，雖然 DSA 演算法不能使用於資料加密或交換鑰匙方面，但亦屬公開鑰匙演算法之一。圖 7-2 為 DSA 的運作流程；傳送端將訊息經過雜湊演算法（如 SHA-1）計算後得到一個雜湊碼，作為 DSA 函數的輸入值；除了輸入雜湊碼與發送者的私有鑰匙（K_R）之外，還需要一個數值 k 與一個公開的公共鑰匙 K_G；數值 k 供簽章用所產生的亂數，每次簽署的 k 值都不相同；公共鑰匙 K_G 是一組通訊法則，由一些公開的參數所構成（容後介紹），並且是通訊雙方事先擁有。經過簽署函數計算後，得到一個由數值 s 與 r 所構成的簽署碼（Sig(M)），並將它附加在訊息後面一起傳送給接收端；接收端收到訊息之後，以同樣的雜湊函數計算出雜湊值（H'），同時將簽署碼（Sig(M) = {s, r}）、公共鑰匙（K_G）、以及發送者的公開鑰匙（K_U），使用驗證函數計算出發送端的雜湊值（M）；比較兩個雜湊值（H 與 H'）是否相同，相同表示確認成功（簽章與訊息相符）；否則可能訊息遭竄改或簽章鑰匙不對（訊息來源有問題）。

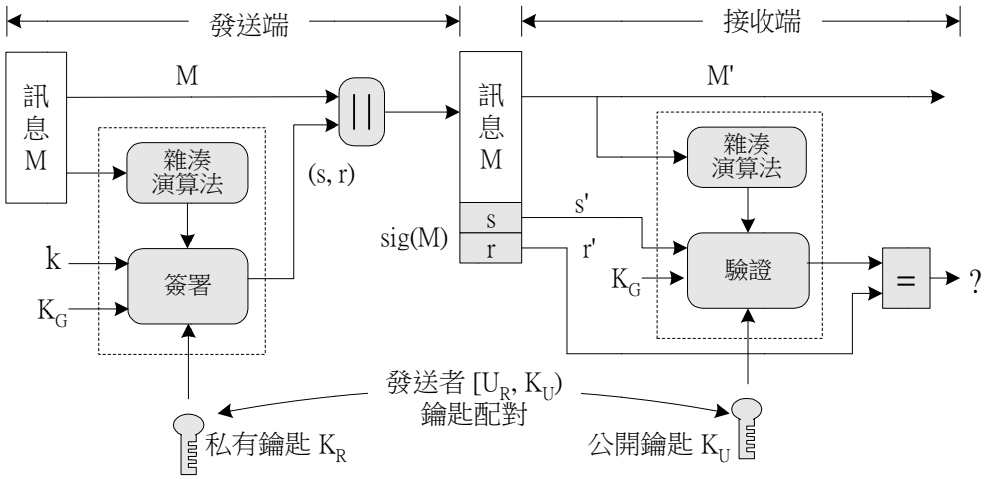

圖 7-2 DSA 演算法的運作流程

上述不難發現 DSA 的製作方式與 RSA 確有所不同。DSA 必須公佈一個運算規則 K_G（或稱公共鑰匙），它是由一組參數所構成，簽署與驗證時都必須使用到它；另外，簽署時還使用一個隨機亂數 k，它是被隱藏於簽署碼（Sig(M) = {s, r}）之內，以增加演算法的複雜度。

7-3-2 DSA 安全性考量

DSA 演算法是 ElGamal [59] 編碼系統的修正版，因為 ElGamal 是採用極複雜之離散對數的計算方法，倘若想藉由數學推導來破解它，時至今日尚未發現較有效的方法，因此唯有仰賴老方法：暴力攻擊法與生日攻擊法。如同 RSA 演算法一樣，應付這兩種攻擊法，即是增加鑰匙的長度或增加簽署碼的長度。一般情況，使用期間較短的簽署，大多使用 1024 個位元長度即可；但對於使用期間較長的簽署，可需要 2048 位元以上才算安全。另一種對策，便是減少簽署文件，或使用多把鑰匙分別簽署不同的文件，如此所產生訊息與簽署配對的數目就會減少，破解者不易從既有的訊息來破解；這種概念與傳統的『印章』非常類似，如常使用某一印章來發送文件，則被仿造的機會就會越大。

7-4 數位憑證

所謂『數位憑證』（Digital Certificate）是一份電子性的文件，被用來證明持有人的身分證明，其中至少包含著持有人的姓名、郵政地址、電子郵件地址、以及公開鑰匙，並且必須經過具權威的單位證明其有效期限。如果沒有公正單位可證明公開鑰匙的合法

性的話，任何人（譬如志明）皆可利用公開密碼演算法（如 RSA 演算法）產生一對公開鑰匙與私有鑰匙，並宣稱這把公開鑰匙可以代表自己的身份，且將之公佈在於網路上；同樣的道理，別人也可佯稱是志明，同樣將公開鑰匙公佈出來；由此可見，僅有公開鑰匙來證明身份是不夠的，而必須有一公正單位來證明這把公開鑰匙確是屬於該人所擁有。

數位憑證的環境裡，必須存在一個權威單位專門發給個人（或組織單位）一對公開鑰匙與私有鑰匙，並且可以對外宣稱某一把公開鑰匙確是代表某一個人（或法人）的身份。但對他人而言，為何這把鑰匙可以代表某一個人的身份？它的依據又在哪裡？譬如，為何這把公開鑰匙是志明先生的？而不是春嬌小姐所有？誰可以證明它？再說，全台灣有上千位志明先生，到底是哪一位志明？這也有爭執所在的地方。因此，我們需要一個具有權威的單位，專門發給個人公開鑰匙，並且必須詳盡敘述它的個人資料，這就是所謂『數位憑證』(Digital Certificate); 而此具有權威的單位，便稱為『憑證授權』(Certificate Authority, CA) 中心。以下將分別敘述其功能。

我們希望在 Internet 網路的虛擬環境下，也能接近於實體環境。在實體環境，每個人（或組織單位）也許擁有許多證件，譬如，汽車駕駛執照。當您在高速公路上駕駛汽車，遇到警察檢查您的駕照時，您只要亮出您的駕照即可，至於您的駕照是真是假？這可要警察自己去判斷（當然，如果是造假的，可能會吃上官司）。又譬如，當您進出海關時，只要秀出您的護照即可，至於護照真假與否，這可要看海關人員的功力了。至於警察如何判斷汽車駕照的真假？除了這個駕照是政府（權威單位）所發外，在駕照上必須有政府單位的戳記，方便警察人員辨識。倘若這樣還是無法嚇止他人使用假駕照的話，警察人員可以透過網路向電腦查詢，這個駕照的來源、以及使用期限，使造假者無法遁形。

『數位憑證』就是讓個人（或組織單位）在網路的虛擬環境下，可以表示個人身分的依據。當個人需要在網路上進行任何交易，只需秀出個人的數位憑證，足以表明自己的身份；至於此數位憑證是真是假？如警察人員一樣，可向有關單位詢問是否有發出此數位憑證。由此可見，CA 中心除了負責發行數位憑證外，也需隨時提供使用者查詢數位憑證真偽的責任；並且每一張數位憑證都有期限的限制，如何去判斷已過期或註銷的憑證，CA 中心有需要提供一套完整的系統來處理。

另一方面，數位憑證可以達到直接仲裁與第三者仲裁的功能；當接收者收到憑證之後，可自行或透過其它管道確認這張憑證的真偽；當對方否認交易行為時，接收者亦可將他的憑證及傳送訊息，呈送給法院當作證物使用（目前已具有法律地位）。因此，數位憑證是時下電子商務或電子化管理環境，不可或缺的主要工具。

7-4-1 數位憑證的種類

到底有哪些數位憑證？這種觀念和實際環境（Real Environment）非常類似。在實際環境裡，每個人可能擁有許多證照，譬如，身分證、汽車駕照、機車駕照、護照、學生證等等，每一種證照都有其特殊功能；譬如，到政府機關申請資料，必須要身份證、在學校圖書館借書，需要學生證等等。以公司行號而言，各行各業都需要營業執照，證明其營業項目是否許可。

虛擬空間也是一樣，存在許多數位憑證，每一種數位憑證具備其特殊功能，使用者可依照需要向不同的 CA 中心申請憑證，一般 Internet 網路上所使用的憑證可分為下列四種：

- 憑證授權（CA）中心憑證：雖然 CA 中心負責發行憑證給使用者，但 CA 中心也需要另一個較高權威單位授權給它，足以表示其信用度。
- 伺服器憑證：在網路的虛擬空間裡，無法確定通訊對方的身份，尤其存取某一伺服器時，也很難判斷這部伺服器是否偽裝的，因此，伺服器需向 CA 中心申請憑證，以確定自己真正的身份。這種情況非常類似實際環境裡的商家，有了商家的營利登記憑證，我們才能確定這家商店是否合法化。目前 Internet 網路上，從事於電子商務的伺服器，大多需要申請 SSL 數位憑證，以確定伺服器本身的名稱及公鑰。
- 個人憑證：這是由自然人或法人向 CA 中心所申請的憑證。在個人憑證裡會詳細描述當事人身分，譬如，E-Mail 位址、郵政地址、身份證字號等等。但隨著個人憑證的使用時機不同，許多 CA 中心也發行以別名取代的憑證，但這種憑證大多祇能使用於某一特殊用途，並無法廣泛使用。
- 軟體發行憑證：這是確定軟體身份的憑證。當您購買某一套軟體（或執行某一軟體）時，如何證明這套軟體的真偽，可由軟體發行憑證來驗證它的身份。

無論何種憑證都擁有一把公開鑰匙、以及持有該憑證的身份資料。如果您對它的憑證有所疑問時，可向發行該憑證的 CA 中心查詢。至於數位憑證的格式為何？接下來我們介紹它。

7-4-2 數位憑證的格式

數位憑證是一個電子資料，將所有資料紀錄在某一個檔案上，以供其他應用系統查詢。為了使這個檔案能廣泛的被其他系統查詢，需要建立一個標準格式。目前 Internet 網路上最廣泛的憑證格式標準為 X.509 v3（version 3），又稱為『X.509 數位憑證』（X.509 Digital Certificate）。表 7-1 為 X.509 的標準規範，每一張數位憑證至少包含有：憑證版本、序號、數位簽章演算法、憑證發行者、有效期間、主體、公鑰、數位簽章等等。其中數位簽章即是利用 CA 中心的私有鑰匙向上述資料簽署所得的值，其功能如同一般證照的鋼印一樣，表示發行者保證這張憑證的正確性；接收到憑證者之後，可利用 CA 的公開鑰匙，來驗證該憑證的真偽。

表 7-1 X.509 數位憑證的格式

版本	憑證格式的版本，如 X.509 Version 3。
序號	用戶的唯一識別碼，同一 CA 所發行憑證的序號不可重複。
演算法識別碼	用來計算此憑證的數位簽章演算法。
發行者	發行此憑證的 CA 單位。
有效期限	憑證的有效期間。
主體	憑證持有人的相關資料，可能包含有：姓名、郵政地址、E-Mail 地址等等。
公鑰資料	持有人的公開鑰匙、以及其演算法。
數位簽章	CA 的數位簽章，CA 將上述資料經過雜湊演算法計算過後，再經過 CA 的私鑰加密。

然而，表 7-1 並沒有詳細列出各個項目的資料結構，本書將其歸類於第九章 PKIX

系統再詳細介紹。

7-5 憑證與私鑰的儲存

7-5-1 憑證與私鑰的儲存元件

既然數位憑證是由一串列的電子資料所表示，如何儲存這些電子資料並且能隨時攜帶於身上，正是儲存媒體（或元件）所考量的地方。儲存電子資料的媒體可由兩個方向來思考，一者僅具儲存資料的功能，此類大多屬於被動元件，必須透過讀取設備程式的管理及操控；另一者需具有管理功能的主動元件，大多有特殊的作業系統管理儲存媒體與讀取設備之間的通訊。我們用一個簡單的範例來說明兩者之間的差異點。譬如，磁卡（如信用卡）僅具有儲存卡號及簡單的數據，讀卡機讀取資料之後，再與主機通訊來判斷該磁卡上的訊息是否正確。由此可見，磁卡內所儲存的資料可任意被讀取，甚至偽造另一片磁卡。IC 卡（如 Smart Card）具有管理功能，當讀取設備需要取資料時，IC 卡會與讀取設備之間執行某些關鍵性的詢問與確認（如通行碼），可以確定讀取設備的身份之後，IC 卡才會將自己的資料傳送給讀取設備，如此一來，IC 卡上的資料便較不容易被詐騙取得，也較不容易被竄改偽造。另一方面，儲存媒體大多不僅存放數位憑證而已，通常會將私有鑰匙一併存放在裡面，否則私有鑰匙僅是一連串無意義的電子資料，無論要記憶或書寫它都有困難，因此，還是主動式的儲存媒體較為安全一點。

簡單的說，具有管理功能的儲存媒體才能達到持卡者與讀取設備之間的相互認證身份（容後介紹），然而僅具有儲存功能的媒體大多不具有這些功能。這方面對於數位憑證而言是非常重要的，我們希望在虛擬環境之下交易，至少必須能單向或相互認證對方身份，才不至於被冒名詐騙得逞。無論如何，我們還是將可能儲存數位憑證的媒體歸納如下：

- ◈ 磁碟片：這是早期所使用的儲存媒體，其具有較高的容量（1.44 MByte），但不具有管理功能是屬於被動式元件。磁碟片可以任意拷貝並讀取，因此容易被偽造竄改，目前已很少使用。
- ◈ 隨身碟：其功能大都與磁碟片相同，只是具有更大的儲存空間（32 Mbyte 以上）。目前電腦大多配備有 USB 介面，不需特殊軟體（Windows XP 以上）便可以讀取隨身碟上的資料，雖然方便但他的安全性是可慮的。

◈ USB Token：其裝置大多與隨身碟相似，但在隨身碟上安裝有特殊管理功能(如 COS 作業系統，容後介紹)，可以和讀取設備之間作符記（Token）協定的交談，如此便具有較高的安全效果。

◈ IC 卡：具有管理功能的 IC 卡又稱之為『智慧卡』(Smart Card)，卡片上的 IC（Integrated Circuit）可包含著 CPU (如 8051) 及相關記憶體單元。基本上，IC 卡就具有簡單電腦系統的功能，不但有獨立的處理單元、作業系統（COS）、輸入/輸出介面，它與讀取設備（讀卡機）之間可從於較嚴謹的通訊協定（如 PKCS #11），來管理並認證對方身份的工作（第八章介紹），因此他的安全性可以說是最高的，也是目前最普遍採用的儲存元件。

7-5-2 智慧卡儲存 – IC 卡

IC 卡 (或智慧卡，存放**私有鑰匙**與**數位憑證**) 如同一般名片大小，裡面主要包含著一個 IC 晶片。晶片裡的主要模組包含中央處理單元（如 8051 CPU）、隨機存取記憶體（RAM）、快閃記憶體(FLASH)、以及一般輸入/輸出介面。為了加速 IC 卡的處理能力，除了嵌入（硬體或軟體）各種對稱式演算法（如 DES、Triple DES 與 AES）外，也具備了 RSA 或 DSA 數位簽章演算法。整體上，IC 卡是一個整合型的『單晶片系統電路』（System-On-Chip Circuit, SOC），並由『智慧卡作業系統』(Card Operating System, COS) 負責管理整個卡片的運作程序。CA 中心將申請人的『**私有鑰匙**』及『**數位憑證**』燒錄於 IC 卡上，並保證其內容不至於遭受竄改或複製。持有人必須透過通行碼的認證才可以取出私有鑰匙；另外，IC 卡與讀卡機之間也會經由『盤問/回應』協定確定對方身分；至於亂數或憑證內訊息的加密，或者處理數位簽章的簽署與驗證工作，亦可以在 IC 卡內完成即可，而不需仰賴外部的輔助設備（如個人電腦）。

7-5-3 智慧卡運作

既然數位憑證儲存於某一媒體上，而此媒體是一個獨立的元件 (譬如 IC 卡)，也許他人可以盜取此儲存媒體來從事網路交易。因此，最起碼必須有一個通行碼來作第一道關卡的保護；也就是說，當使用者想要由儲存媒體上讀出資料（如私有鑰匙）時，必須經過通行碼的認證，才可以讀出資料並從事網路交易，而且驗證通行碼的訊息必須儲存

於媒體上（如 IC 卡），其運作如圖 7-8 所示。

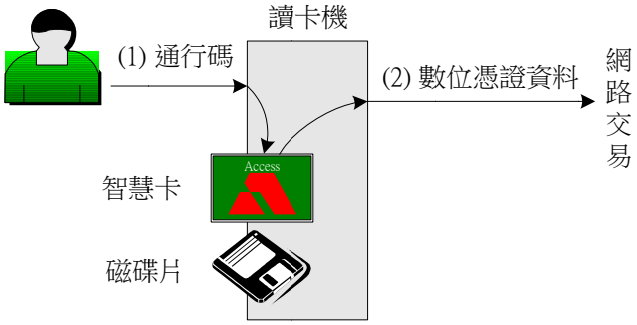

圖 7-8 儲存媒體驗證通行碼

7-6 數位憑證的運作

利用 IC 卡儲存數位憑證（或稱為 Smart Card）是目前最普遍採用的元件，然而，當我們由 IC 卡取出憑證資料（X.509 憑證，主要是公開鑰匙），並傳送給對方時，對方如何判斷此憑證並非他人所冒用。記得前面討論過，我們希望數位憑證能像真實環境裡的各種證照的功能，並且能在網路上自由通行。譬如，當您遇到交通警察要求您出示駕駛執照時，警察最起碼由駕照的相片確認是您本人，或者是他人所偽造；另一方面，您也可以從攔截您停車之警察的警徽或相關警察證件，辨別其真偽。很不幸的，在虛擬環境裡並無法正確地觀察到出示憑證的本人面貌，因此必須有特殊的處理功能始可行，我們就單向認證與雙向的相互認證兩方面來探討。

7-6-1 憑證單向認證

圖 7-9 為數位憑證單向認證的運作程序，事先持卡者（User A）已將私有鑰匙與數位憑證（公開鑰匙）儲存於 IC 卡上（鑰匙配對 $\{K_{Ra}, K_{Ua}\}$），其運作說明如下：當持卡者欲傳送憑證（如公開鑰匙 K_{Ua}）之前，先產生一個亂數（N_1）並輸入密碼（訊號 (1)）取得 IC 卡上的私有鑰匙（K_{Ra}），再向該亂數簽署（Sig 函數）之後，連同亂數及憑證（$Cert_A$）包裝成一個訊息（訊號 (2)），一併傳給伺服端，訊息如下：

$$Sig_{KRa}[N_1] \,||\, N_1 \,||\, Cert_A$$

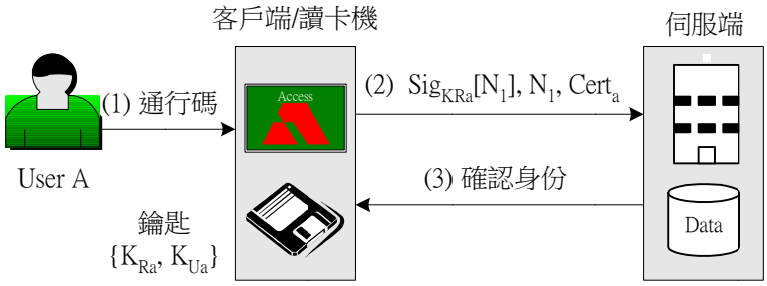

圖 7-9 數位憑證的單向認證

伺服端收到訊息之後，利用憑證（$Cert_A$）內的公鑰（K_{Ua}）及所收到的亂數（N_1），驗證所收到的簽署碼是否正確，如果正確的話，則確定發送憑證者是公鑰擁有者本身。因為，私鑰只有持有人自己擁有，利用它的公鑰可以驗證其私鑰所簽署的數位簽章，則應該是他本人沒錯，除非持卡者自己的私鑰已被破解或冒用；或者持卡人取得 IC 卡裡私鑰的通行碼洩漏，而且 IC 卡也被他人盜用才有可能被冒名發送憑證。

7-6-2 憑證相互認證

許多情況下必須達到雙方都能確認對方的身份（雙方 IC 卡之間的確認），才能確保通訊交易的安全，其運作程序如圖 7-10 所示；可區分為兩個時相（Phase），我們還是以所標示的訊號次序來說明，並簡略使用者輸入通行碼部分。說明如下：（Alice 為發起者，Bob 為回應者）

- 訊號 (1)：Alice 發送自己的憑證（$Cert_A$，包含公開鑰匙）與一個亂數（N_1）給 Bob。
- 訊號 (2)：Bob 收到對方的憑證之後，先不去認證憑證的正確性，而另外產生一個亂數（N_2），並與 Alice 的亂數一起利用自己的私鑰簽署；之後，將亂數（N_2）與簽署碼（$Sig_{KRb}[N_1 || N_2]$）一併傳送給 Alice。
- 訊號 (3)：Bob 將自己的憑證（$Cert_B$，包含公開鑰匙）與產生一個亂數 N_3 一併傳送給 Alice。接下來，Alice 由該憑證上取得 Bob 的公開鑰匙，再利用訊號 (2) 所收到的亂數與簽章碼（$Sig_{KRb}[N_1 || N_2], N_2$），驗證該簽章碼是否正確，便可確定 $Cert_B$ 是否是 Bob 的憑證。
- 訊號 (4)：Alice 確定對方身份無誤之後，則另外產生一個亂數 N_4，與所收到的亂數 N_3，一併利用自己的私鑰簽署，再將亂數與簽署碼（$Sig_{KRa}[N_3 || N_4], N_4$）傳送給 Bob。

接著，Bob 收到該訊之後，則取出 Alice 憑證內的公開鑰匙（$Cert_A$，訊號 (1)），並驗證所收到的簽署碼是否正確。如此一來，Bob 也可以確認 Alice 的身份是否正確。

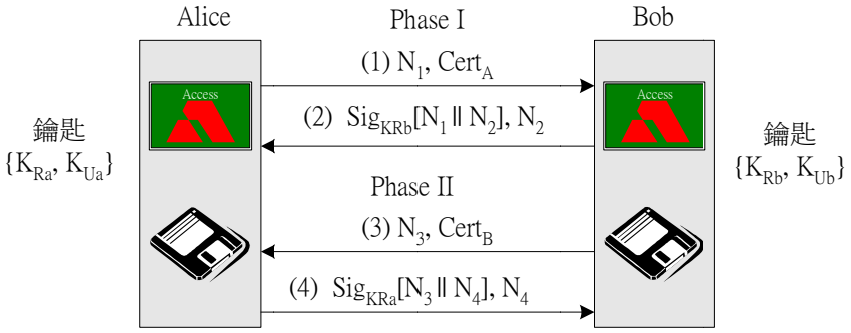

圖 7-10 數位憑證的相互認證

在圖 7-10 中省略了使用者輸入通行碼部分；同樣的，持卡者想要由 IC 卡內取得自己的私鑰，簽署交換的亂碼時，都必須經過通行碼的確認才行。

值得特別注意的是，採用 IC 卡（智慧卡）為儲存媒體的話，無論簽署/驗證簽章演算法、加密/解密演算法、亂數產生等等關鍵性的處理，都是由 IC 卡內部的作業系統（如 COS）所完成，不需仰賴外部設備（如個人電腦或讀卡機），如此才能確保整個認證程序的安全性。

7-7 憑證授權中心

7-7-1 憑證授權中心簡介

發出憑證的單位（或公司），便稱為『憑證授權』（Certification Authority, CA）中心。CA 中心必須是個可信賴的公正單位（私人或政府），它會依據合法申請者的請求，發出數位憑證。數位憑證裡包含了申請人的辨識資料（姓名、地址、或身份字號）、申請人的公開鑰匙、序號與其他資料，並保證不會造假；並且 CA 中心利用自己的私有鑰匙向上述資料簽署，所得到的數位簽章亦存放於憑證裡。也就是說，每一張數位憑證必須有發行 CA 的數位簽章，而此數位簽章是利用 CA 中心的私有鑰匙簽署保證。任何人，如果懷疑某一張數位憑證的真實性，便可以利用 CA 的公開鑰匙來認證它的正確性。這裡又出現另一個問題，欲取得 CA 的公開鑰匙，必須先拿到該 CA 的數位憑證，再由它的數位憑證取得它的公開鑰匙，但這張數位憑證的真實性如何？亦無法保證；因此，需仰賴

另一個較高權威的 CA 來證實它，這就是 CA 中心也需要憑證的理由。

目前台灣較具權威的認證中心有：

(1) 政府憑證管理中心 (www.pki.gov.tw)：這是官方發行單位，可以針對個人（自然人）或公司行號（法人）發行數位憑證，所發行的憑證較具有權威性，它的用途也較為特殊。譬如，透過網路向政府單位承標各種工程或器材，便需要此 CA 中心所發行的憑證來證明自己的身份。

(2) 內政部憑證管理中心（moica.nat.gov.tw）：官方發行單位，這是針對個人所發行的憑證，功能就如同個人身分證一樣，又稱為『電子身分證 IC 卡』。個人（或稱自然人）欲透過網路向政府機關申辦任何事項，便以此憑證來確認身份。

(3) 工商憑證管理中心（moeaca.nat.gov.tw）：官方發行單位，其建置工商憑證管理中心核發事業主體的數位憑證，同時提供可靠、安全及快速的網路申辦系統，提高作業效率及服務品質。

(4) 台灣網路認證中心 (www.taica.com.tw)：這是民間發行單位（主要是台灣證券交易所），主要用途是使用於透過網路來執行股票交易（下單買賣）時，認證下單者的真正身份、也發行電子錢包；但目前也有許多網路銀行，利用此 CA 所發行的憑證來確認客戶的身份。

(5) 儲匯局電子證書認證中心 (ca.prsb.gov.tw)：亦屬政府單位所發行的，可透過它來發送數位『存證信函』。

(6) 網際威信 (www.hitrust.com.tw)：這是民間發行公司，接受個人、公司行號、以及 SSL 伺服器憑證申請，主要是從事一般電子商務上的身份證明。

7-7-2 CA 的服務項目

當 CA 中心核准申請並產生鑰匙配對之後，便將私鑰與數位憑證儲存於 IC 卡上，再交付給申請者。持有者可能會遺失 IC 卡、忘記通行碼、或者對鑰匙配對的安全性感到懷疑時，則可向 CA 中心提出相關的服務請求。CA 中心的服務項目如圖 7-11 所示，說明如下：

- ◈ 憑證簽發、更新與終止：CA 中心負責用戶身份的審核及憑證簽發、更新、以及終止用戶的憑證。
- ◈ 憑證保存：CA 中心必須紀錄所有的有效憑證、過期憑證、以及註銷憑證的清單，並提供相關憑證變動資料給用戶。
- ◈ 憑證查詢與分送：除了保存各種憑證資料外，還必須供應客戶的查詢，或者分送憑證給客戶。
- ◈ 公正仲裁：當交易發生爭執時，CA 必須提供如用戶的身份及公鑰等公正資料，給相關仲裁單位（如法院）。

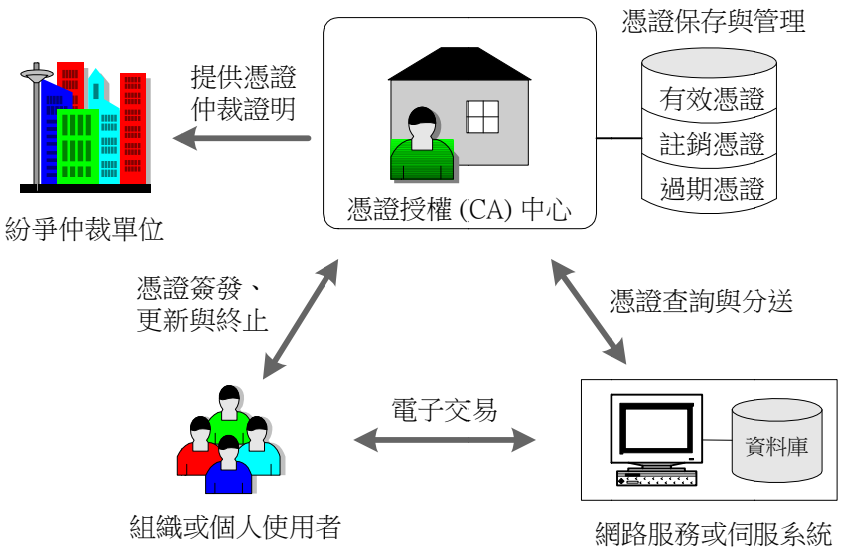

圖 7-11 CA 的服務項目

7-7-3 憑證的產生與認證

如果想要在 Internet 網路上從事各種交易（或電子商務），那就必須向一家較具有權威的 CA 中心申請數位憑證。但隨著交易行為不同，或許會有不同的數位憑證，這種現象就如同一個人（或組織）持有各種不同的證照一樣。譬如，志明想要透過網路申報綜合所得稅，因此他可以選擇向台灣網路認證中心（taica）申請數位憑證，其步驟如下：（如圖 7-12 所示）

(1) 志明向 taica 提出申請數位憑證的請求，並將個人相關資料傳送給（或親自辦理）taica。

(2) taica 查核志明的相關資料後，便製造志明個人的公鑰與私鑰配對；也製作志明的數位憑證，其中包含志明的個人資料(姓名、地址、E-mail 位址等等)、公鑰、以及 taica 公司的數位簽章；其中數位簽章是表示 taica 對這張憑證的證實資料(如同印章一樣)。並將志明的數位憑證登錄到資料庫系統內 (LDAP 系統，第九章介紹) 以供查詢。

(3) taica 公司將數位憑證與私有鑰匙分別傳送給志明，其可能儲存於不同的磁片上，或者一起燒錄於 IC 卡上。

(4) 志明便可利用 taica 公司所發行的數位憑證與私鑰，向稅捐處申報綜合所得稅。

(5) 稅捐處收到志明的數位憑證，由數位憑證的標頭上可以觀察出是出自 taica 公司所發行，便利用 taica 公司的公開鑰匙來認證此憑證的真偽。認證方法是，首先利用 taica 的公鑰來向憑證內的數位簽章解密，再利用憑證上所敘述的雜湊演算法來向憑證內的資料做計算，如果兩者結果相同的話，則表示憑證是正確的，而且內容也沒有被竄改過。

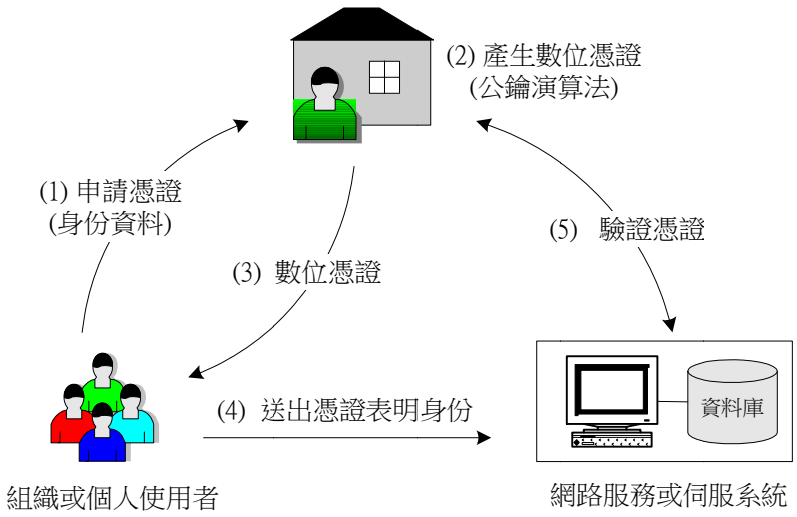

圖 7-12 憑證產生與驗證的程序

當 CA 中心收到客戶端的申請時，如何去過濾及查核申請人的資料，這真是一件繁複的工作。如果權威性較高的 CA，它期望所發出的憑證不會重複，亦是，針對每一個人 (或法人) 只能發給一張憑證；另外，也許會有冒名或相同名字的申請者，當然必須詳加調查清楚，通常查核的工作可能要 2 天到一個星期的時間。

7-7-4 憑證的註銷

每一張數位憑證都有一個有效期間的限制，即使有效期間內，憑證也有可能被註銷，註銷原因有：

- ◈ 用戶的私鑰遺失、或公鑰被破解，而需要修改私鑰與公鑰時。
- ◈ 用戶的私鑰被盜竊使用，而需要重新修改私鑰與公鑰。
- ◈ CA 中心發現憑證發給錯誤的個人或團體、或發現原申請資料不正確。
- ◈ 用戶不再使用此憑證，而申請註銷。

CA 中心必須隨時備有這些已註銷的資料，讓一般客戶查詢。一般 CA 都會使用『憑證註銷序列』(Certificate Revocation List, CRL) 來記錄所有已被註銷的憑證。當客戶收到某一張憑證之後，除了觀察憑證上所紀錄的有效期間是否過期外，也可以透過網路向 CA 中心查詢，該憑證是否過期。然而，當 CA 的申請者愈多時，可能的註銷憑證也會快速的成長；另一方面，為了安全起見，CA 中心還提供各種憑證資料的查詢，因此，CA 除了需要一個快速的資料庫系統來儲存，也需要有一個共通的通訊協定來制定查詢的運作程序與語法，這些都是 PKI 系統所需制定的標準。目前 PKI 大多採用 LDAP 協定來制定客戶端與伺服端之間的查詢動作，並以 X.509 標準來制定數位憑證的格式，我們將在第九章再詳細介紹相關的技術。

7-8 數位簽章的仲裁機制

數位簽章必須配合良好的確認機制，才能達到真正『不可否認性』的功能。譬如，世雄發送一個股票買賣的信件給志明，並於信件上附加數位簽章。後來股票賠錢了，世雄否認他曾經發過此信，並仿稱它的私有鑰匙已被盜用、或他的鑰匙已被破解、甚至表示其內容不符，且是被仿造的。如此一來，志明如何提出一個有力的證據，證明此信確由世雄所發。由此可見，我們希望數位簽章能和書寫簽名一樣的功能，所以必須具備下列特性：

- ◈ 能夠驗證簽章的所有者、日期與時間。
- ◈ 簽章的同時，能夠確認文件的內容。
- ◈ 發生糾紛時，能由第三者驗證此簽章。

如此說來，數位簽章除了必須具有高複雜的演算功能之外，更需要取得合法的法律地位。美國已正式公佈數位簽章如同一般簽名一樣的功效，台灣也於 2001 年正式公佈它的法律地位。全世界大部分的國家更陸續承認數位簽章具有簽署『自然人』與『法人』文件的合法性。為了配合它的合法性，我們必須有更適當的處理，才能讓它走上法律之途，如此便牽涉到糾紛發生時的仲裁方面的問題。基本上，可分為直接仲裁與第三者仲裁兩種方式。簡單的說，萬一走上法院，接收端如何提出一個有力的證據足以取信法院，並證明該文件確由傳送端所發，達到真正『不可否認性』的功能。

還未介紹各種仲裁機制之前，先定義兩個會使用到的系統：

- 加密系統 E：利用鑰匙 K_A 向訊息 M 加密後的密文為 $E_{Ka}[M]$。
- 簽署系統 Sig：利用鑰匙 K_A 向訊息 M 簽署後的簽署碼為 $Sig_{Ka}[M]$。

值得注意的是，這裡僅介紹仲裁機制的基本構想，在實作上並非完全如此；本書於第十二章裡介紹許多安全性電子郵件的製作規範，從中可以瞭解較實際製作方法。

7-6-1 直接仲裁

直接仲裁表示接收者必須自己去判斷訊息來源的正確性，最起碼的條件是接收者必須透過其它管道取得傳送者的公開鑰匙，並且能肯定該公開鑰匙是正確的。其操作方式可分為訊息不經加密保護（非隱密式）與加密保護（隱密性）兩種。

【(A) 非隱密式直接仲裁】

非隱密式直接仲裁表示訊息未經過加密處理，並且由接收端自行判斷簽署鑰匙的正確性，其運作程序如圖 7-4 所示。發送者利用自己的私有鑰匙（K_{Rx}）向訊息簽署後，將訊息與簽署碼（$M, Sig_{KRx}[H(M)]$）一併傳送給接收者；接收者利用傳送端的公開鑰匙與所收到的訊息，確認簽署碼是否正確，如果正確的話，則表示訊息確實來自傳送端並未遭受竄改。接收者可將對方的訊息與簽章一起儲存起來，萬一對方不承認時，再取出作為證物。

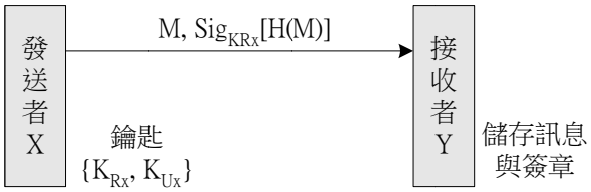

圖 7-4 非隱密式直接仲裁

【(B) 隱密式直接仲裁】

某些情況下，所傳輸的訊息必須特殊保護，以防被他人窺視其訊息內容，這正是隱密式直接仲裁的主要原因。如圖 7-5 所示，訊息與數位簽章傳輸之前，先利用對方的公開鑰匙加密（E_{KUy} [M || Sig_{KRx} [H(M)]]）後再傳送給接收者。接收者收到訊息之後，利用自己的私有鑰匙（K_{Ry}）解密以後，再以發送者的公開鑰匙（K_{Ux}）來驗證數位簽章，同時儲存一份副本，作為爾後發生爭執的證明使用。

乍看之下，直接仲裁已可以解決簽署證明的問題，然而發送者還是可以佯稱訊息並非它所發送，並且聲稱訊息及簽章都是偽造的，即使在訊息上加入時間戳記(Timestamp) 來註明傳輸日期也不見得讓他人取信。主要的原因是，提出證據的人是告訴人(接收者) 自己，的確讓法官很難相信他的真偽。所以最好還是由一個公正單位來提出證據，其說服力可能會高一點，這就是所謂的『第三者仲裁』。

7-6-2 第三者仲裁

『第三者仲裁』表示透過一個權威單位來證明訊息的合法性，這種情況非常類似目前郵政系統的『存證信函』。發信者傳送信件之前，先複製一份副本給郵局，並經郵局確認內容與原文無誤，並加蓋郵戳以確定傳送日期之後，再寄送給對方。如果雙方發生爭執時，可要求郵局顯示副本，其功能有二：一者可確認發信人；另一者也可確認信件內容。我們期望數位簽章能達到如『存證信函』一樣的功能。

如果發送者認為訊息不需要保密的話，可以採用圖 7-6 的傳輸方式。第三者（A）必須是一個具有權威性的單位 (最起碼發送者與接收者都能相信它)，它必須將公開鑰匙（K_{Ua}）公佈出來，並保存好自己的私有鑰匙（K_{Ra}）。發送者（X）當然擁有自己的鑰匙配對 {K_{Rx}, K_{Ux}}，其中 K_{Rx} 為私有鑰匙、K_{Ux} 為公開鑰匙。假設發送者（X）欲採用第三者仲裁方式，將一個訊息傳送給接收者（Y）的話，則必須經由第三者轉送，仲裁者需保

留一份副本，以備爾後發生爭執時，作為仲裁使用。

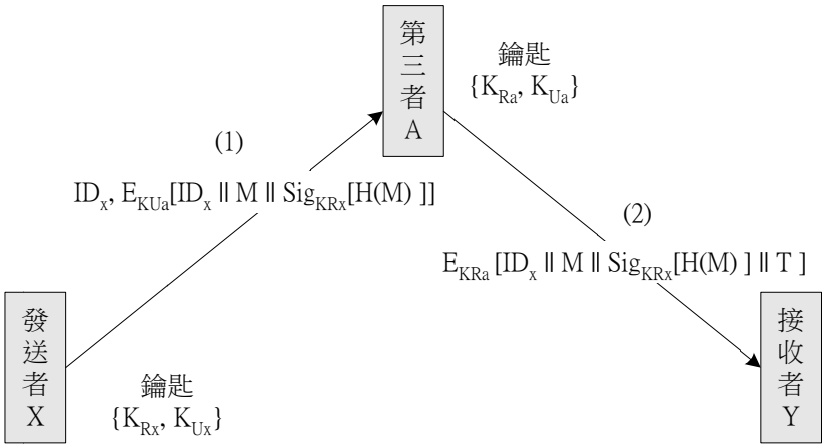

圖 7-6 訊息不保密的第三者仲裁

圖 7-6 的運作程序如下：首先發送者將訊息處理過數位簽章後，再將訊息（M）、簽章（$Sig_{KRx}[H(M)]$）、身份識別碼（ID_x）一起以仲裁者的公開鑰匙（K_{Ua}）加密後，再結合身份識別碼（ID_x）傳送給仲裁者，訊息格式為：

$$ID_x, E_{KUa}[ID_x \| M \| Sig_{KRx}[M \| H(M)]]$$

其中識別碼（ID_x）包含著發送者的身分證明，也許會攜帶著接收者的身分證明。仲裁者收到信件請求時，由識別碼內的訊息了解發送者與接收者的身份，首先將該訊息儲存起來以備爾後仲裁使用，再利用自己的私有鑰匙（K_{Ra}）解開信件內的訊息；接著，在訊息後面加入一個時間戳記（類似郵戳的功能），之後再利用自己的私有鑰匙（K_{Ra}）向訊息加密，成為另一個信件：

$$E_{KRa}[ID_x \| M \| Sig_{KRx}[H(M)] \| T]$$

再將此信件傳送給接收者（Y）。接收者收到訊息之後，首先將該訊息儲存起來，以備爾後解決紛爭使用。接著，再利用仲裁者的公開鑰匙（K_{Ua}）將訊息解密，也透過數位簽章認證功能，除了證實訊息沒有遭受竄改，也可確認發送者的身份（利用發送者的公開鑰匙）。

他日若發生爭執，接收者除了將保留一份的數位簽章呈上法院作證，當然也可要求仲裁者提出信件傳送的證據，致使發信者啞口無言，難以否認發信的事實，如此便可以達成『不可否認性』的功能。由以上的運作程序可以看出，所謂不保密的意思是仲裁者可以看到訊息的內容，然而信件在網路上傳輸還是保密的（利用仲裁者公開鑰匙加密）。

7-9 範例：電子投標系統

電子投標系統運作程序

- 杜絕綁標
- 杜絕圍標

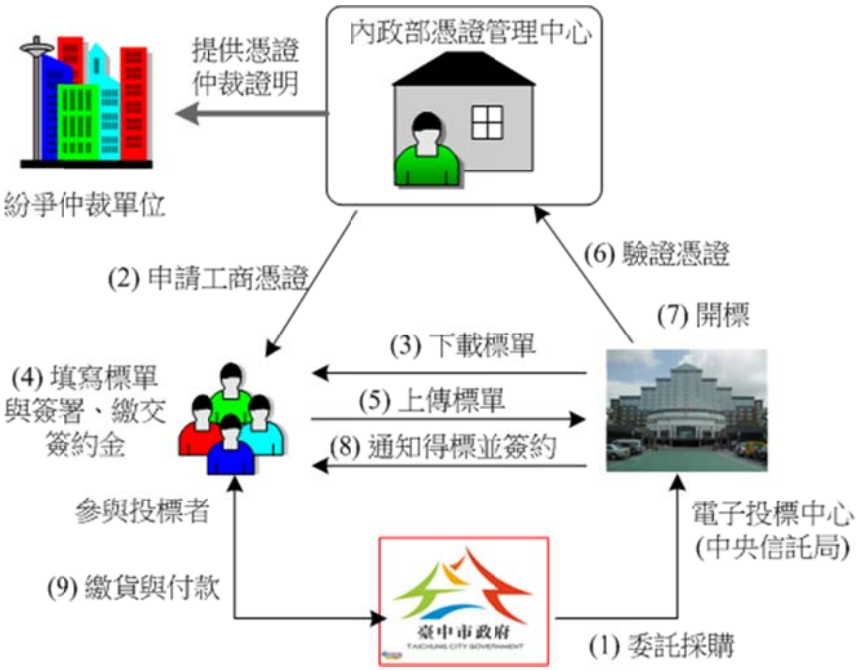

圖 7-7 電子投標系統運作程序