結合 IC 卡強化時戳服務之設計與實現

葉志青[a]，楊中皇[b]，褚芳達[c]
[ab]高雄師範大學資訊教育研究所
802 高雄市苓雅區和平一路 116 號
TEL:(07)7172930; FAX:(07)7174884
[c]中華電信研究所前瞻技術研究室國家時間與頻率標準實驗室
326 桃園縣楊梅鎮民族路五段 551 巷 12 號
TEL: (03) 4244973
e-mail: [a] dexter@icemail.nknu.edu.tw；[b] chyang@computer.org；[c] cfonda@cht.com.tw

摘　要

因為網際網路每年快速的成長，安全措施已被視為網際網路上最重要的一環。時戳服務機構（time-stamping authority，TSA）是一個可信任的機構，它可以提供資料在某一特定時間前即已存在的證據。在這個研究中，我們透過網際網路實作依據 RFC-3161 標準的可信賴的時間服務系統（Trusted Time Service，TTS），伺服端是在 Linux 平台上安裝及修改開放源始碼軟體 OpenTSA，而客戶端則是在 Windows 平台上使用 C++ Builder 工具自行開發軟體。同時，伺服端配有 USB IC 卡讀卡機和現成的 IC 卡來儲存及擷取時戳伺服端的私鑰。客戶端亦配有 USB IC 卡讀卡機並利用 IC 卡來儲存通行碼，以便與伺服端進行密碼驗證，以防止阻斷服務攻擊。

關鍵字：時戳、數位簽章、公開金鑰基礎建設、IC 卡

壹、緒論

邁向廿一世紀，網際網路的應用快速發展，許多以網路為主的應用及服務模式逐漸成型，文件及各項交易趨向數位化，而在目前的網路架構下，資訊毫無保護的在公開的網路上傳遞，衍生了許多以前傳統社會從未有的安全問題，例如當電子公文在網路傳遞的同時，可能遭到擷取、竄改、複製、偽造等，而電子交易時，付款機制的安全性、認證系統的確認性、交易的不可否認性以及資料的完整性等安全問題，至使必須利用安全技術，加入一些安全機制，以確保在網路上資料的安全。

我國電子簽章法[27]於民國九十年十一月十四日公佈，並於九十一年四月一日起正式施行，內容除明定電子簽章、電子文件之法律效力外，亦規範了憑證機構的管理機制，以達到有效保護消費者的權益，同時也為憑證之建置與運作，即電子簽章的使用提供相當明確的法源依據，未來將由可信任的認證機構進行數位憑證的發放，藉此建立電子商務交易的信任機制。而其中依電子簽章法第十一條第二項所規定之「憑證實務作業基準應載明事項」中第三十條第五項：紀錄對於時戳之要求；可見時戳對於憑證作業的重要性。因此，假若能透過時戳服務系統提供精準的時戳服務，相對的也為電子文件、電子交易等提供更大的安全保障。

時戳（Time Stamp）就好比郵戳一般；郵局在處理郵件時所蓋用的郵戳，是目前最通用的時間證明，以「郵戳為憑」即可證明文件收寄的時間[28]；而時戳所扮演的角色即為數位化的郵戳，是資訊時代不可或缺的安全機制，時戳可以為任何電子文件或電子交易提供準確的時間證明，並且驗證文件或交易的內容自蓋上時戳後是否曾被人修改過。時戳服務機構（time-stamping authority，TSA）[1,9,16,17]通常扮演一個可信任的第三機構來提供資料在某一特定時間前即已存在的證據，而時戳服

務（time-stamping service）已成為公開金鑰基礎建設（PKI）的一部份[2,6]，並且是許多網路安全應用的關鍵部份，例如 Kerberos、網路交易等等。當文件或文件的雜湊值被送往 TSA 要求時戳時，TSA 伺服端將回應一個時戳記號（time-stamping token），其可用來指出資料或雜湊值在某一特定時間前即已存在。有諸多情況是我們必須去證明一些資料被建立或修改的日期和時間[8]，而 TSA 將會透過一目瞭然的聯結通道來聯結時戳記號，對任何人而言，要事前未被查覺地在聯結通道的中間插入一份文件，在計算上是不太可能的[14,19]。

在這個研究中，我們探討建立時戳服務系統之之相關應用技術，以掌握時戳服務系統的優缺點，進而提出 TTS（Trusted Time Service）系統設計，以強化現行的時戳服務系統。在伺服端，我們使用依據 OpenSSL 計劃[12,21]提供的開放原始碼軟體 OpenTSA 做為中介軟體，在 Linux 環境中建立 TTS 伺服器，且為了保護私鑰，TTS 伺服端配有 SLE 4428 IC 卡和 Omnikey[11]的 USB IC 卡讀卡機，時戳回應則由存在 IC 卡內的私鑰所簽發；在客戶端，Borland C++Builder 6 [4]為發展 TTS 客戶端的軟體工具，其能透過 IC 卡的密碼確認方式[7,18,22]與伺服端進行認證；此外，客戶端與伺服端的通訊協定即是依據 IETF RFC3161 標準[6]。

貳、文獻探討

本章主要係針對完成可信賴的時間服務系統（TTS）所需的相關理論及技術加以探討；其中包括相關的密碼學演算法、ASN.1、X.509 數位憑證、OpenTSA、IC 卡、阻斷服務攻擊等簡介。

2.1 相關的密碼學演算法

安全雜湊演算法（secure hash algorithm, SHA）[24,29]是由美國國家標準技術局（NIST）所發展出來的，並在 1993 年成為安全雜湊標準（FIPS PUB 180），而在 1995 年又將其修訂版本發佈為 FIPS PUB 180-1，即為通稱的 SHA-1。SHA-1 接受任何長度小於 2^{64} 的訊息並產生一個 160 位元的訊息摘要。

MD5 是 Ron Rivest 在 MIT 所發展出來的訊息摘要演算法[15]。MD5 可以輸入任意長度的明文位元，而其輸入的明文會被分成好幾個 512 位元的區段來處理，最後產生 128 位元的訊息摘要。

2.2 ASN.1

ASN.1（Abstract Syntax Notation One，定義在 X.208）[3]抽象語法符號，是一個 ITU-T(X.680)和 ISO(ISO 8824-1)標準。其目的是為了在不同機器間用統一的語法表達事物，這是描述資料結構的電腦語言。ASN.1 提出一套標準的方法，以敘述各種類型的資料結構，使資料的表示法有一種統一的方式。

ASN.1 可以用多種方式加以編碼，為了實際儲存或交換數據，資訊需要編碼成一個位元模式。從一個抽象語法轉換成位元模式最普遍的資料編碼規則稱作 BER（Basic Encoding Rules，定義在 X.209），其目的是將 ASN.1 所敘述的資料結構，以一定的方式編碼以便在網路上傳遞。而另外還有一種編碼方式稱為 DER （Distinguish Encoding Rules，定義在 X.690），它是 BER 的子集，對 ASN.1 提供唯一的編碼方式。

2.3 X.509 數位憑證

數位憑證是指一份經過數位簽署、包含了所有者名稱以及其公開數位金鑰的電子文件。數位憑證可以用多種形式呈現，但一般的數位憑證標準格式稱做 X.509，為 OSI 標準中的一部份。X.509 是由 ITU 所提出，X.509 是 X.500 系列中的一部份，X.500 定義了目錄服務（directory service），而 X.509 定義了 X.500 的目錄所提供給使用者的確認服務架構，這個目錄可以當成公鑰憑證的儲藏庫。每個憑證都包含了某位使用者的公鑰，並且都已經用可信

賴機構的私鑰簽署過。此外，X.509 也定義了數位憑證的結構以及利用數位憑證作認證的協定。X.509 數位憑證乃以 ASN.1 符號表示法定義，詳細記載了組成該數位憑證的二進位資料。而 ASN.1 可以用多種方式加以編碼，現今標準多使用簡單的 DER (Distinguished Encoding Rules)，可以產生二進位數位憑證。

X.509 用到了公開金鑰密碼學與數位簽章技術，其認證結構與確認協定被用在很多不同的地方，包括電子商務中用來作認證、SSL/TSL、SET、IPSec 及 S/MIME 等等，均採用 X.509 的數位憑證來增加其安全性[24]。

2.4 OpenTSA

OpenTSA 計劃的目標是要發展一個穩定的、安全的，並且符合 RFC-3161 協定標準的開放原始碼時戳服務（time stamping authority，TSA）之伺服端與客戶端應用程式；該計劃是由 Zoltan Glozik 所發展，而相關之軟體或加密軟體是由發展 OpenSSL[21]函式庫的 Eric Young 和 Tim Hudson 所編寫，其第一個釋出的時戳測試套件是在 2002 年的 5 月 10 日。

OpenTSA 為 OpenSSL 的一個工具套件，因之與 OpenSSL 相同，其在某些簡單許可條件下可以被自由的使用在商業或非商業的用途上。在官方網站可以下載最新的 OpenTSA 發展套件和支援 Apache 的時戳模組。

2.4 IC 卡

IC 卡在外觀上跟一般磁條卡很類似，也就是在符合國際標準的塑膠卡片上封裝積體電路（IC），而卡上有八個金屬接點可作為讀寫裝置提供電源、控制信號和資料傳輸的介面。IC 卡內的積體電路可包含微處理器（MCU）和記憶體，若封裝在卡片內的積體電路只含記憶體而不具備微處理器，則稱為記憶卡（Memory Card），大多僅作資料保存使用，如電話卡；若封裝在卡片內的積體電路包含微處理器和記憶體，則又稱為智慧卡（Smart Card）[28]，因其具有微處理器，使得其如同一部電腦般的具有邏輯運算、資料儲存等功能，因此可以利用此功能發展出不同的密碼學演算法來代替現有的系統[25]。

傳統的磁條卡（如信用卡、提款卡）容量小，且資料容易讀取、安全措施薄弱、容易偽造，因此產生很多的犯罪行為。IC 卡[23]較磁條卡難以偽造，此外，IC 卡讀卡機的介面目前有 RS-232、USB、PCMCIA、3.5 吋磁片等，而其價格也日益便宜；隨著 Windows2000 提供內建的 PC/SC 讀卡機驅動程式庫，更使得 IC 卡越來越適合網路安全應用[26]。

2.6 阻斷服務攻擊

所謂阻斷服務攻擊（Denial-of-Service, DoS），通常是指攻擊者發送大量的網路封包給目標伺服器，致使目標主機的 CPU、網路頻寬、記憶體等資源耗盡，因而使得某些服務遺失或者是一些功能失效，暫時無法提供正常服務；第一個著名的 DoS 攻擊就是 Morris Worm，它可以在數小時內征服五千台電腦。

而另一個較常見的攻擊方式稱為分散式阻斷服務攻擊（Distributed Denial-of-Service, DDoS），其對系統所造成的傷害程度遠超過傳統的 DoS 攻擊，它是一種多對一的阻斷攻擊方式（如圖 1），攻擊者透過許多具有安全漏洞的電腦主機作為跳板，對受控主機下達攻擊指令，便可命令更多的代理攻擊主機對攻擊目標主機傳送大量的封包，致使目標主機因無法處理大量封包而導致系統癱瘓。2000 年 2 月，Yahoo、eBay、CNN、Amazon 等大型企業網站，就因無法處理一時大量湧進的資訊封包而造成系統癱瘓，致使無法提供正常服務給合法的使用者。

本研究中，為了防止類似阻斷服務攻擊的手法癱瘓系統，所以在 TTS 客戶端配載 USB IC 卡讀卡機並利用 IC 卡來儲存通行碼，以便與伺服端進行密碼驗證。

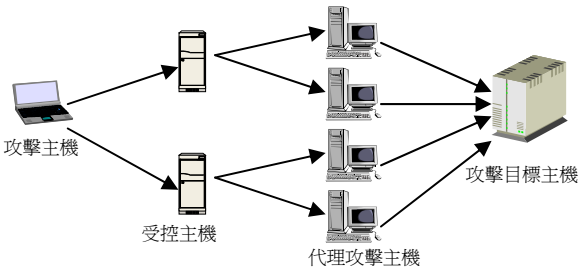

圖 1 DDoS 攻擊模式

參、系統架構

本研究中所建立的 TTS 可信賴的時間服務系統主要分成兩個部份，其系統架構圖如下所示：

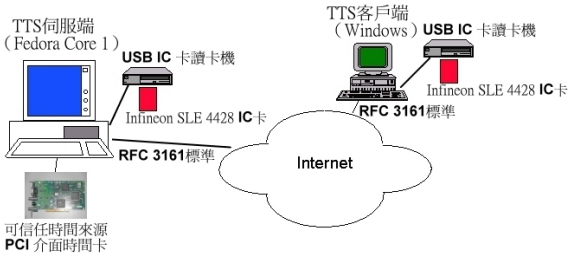

圖 2 TTS 系統架構圖

一、在 Linux 平台下使用 OpenTSA[13] 開放源始碼所架設的 TTS 伺服器，與基本時戳服務運作模式不同的是，在本系統中另外加入一 TrueTime 公司（2002 年併為 Symmetricom 公司）的 PCI 介面時間卡以取得精確且可任賴的時間來源。此外，伺服端亦加入 IC 卡和 Omnikey 的 USB IC 卡讀卡機以加強保護及管理私鑰。

二、採用 Borland 公司的 C++ Builder 6.0 做為開發客戶端軟體程式工具；客戶端軟體具有時戳請求、時戳驗證、時戳接收、儲存及顯示時戳憑證等功能。此外，客戶器端亦使用 IC 卡和 USB IC 卡讀卡機和伺服端進行認證。

3.1 TTS 伺服端

本研究中 TTS 伺服端是在 Fedora Core 1 平台上安裝和修改開放原始碼 OpenSSL 的工具套件 OpenTSA[13]，並以 C 語言軟體作為研發的工具，同時，TTS 伺服端配有 USB IC 卡讀卡機和 SLE4428 IC 卡來儲存及擷取 TSA 的私鑰。

OpenTSA 係使用 gettimeofday()函數取得伺服器系統的時間，但在時戳服務上如果僅以此表示時戳的產出時間，則可能影響其服務的公正性，因此，為了能讓充當 TSA 的 TTS 伺服端可以取得精確且可任賴的時間來源，本研究中於伺服端加上一 PCI 介面時間卡（如圖 3），伺服端透過此介面卡不但可以取得精確且可任賴的時間，且時間透過硬體取得亦較不易被人所更改。

TSA 的 CA 基本上產生的時戳憑證是由 TSA 的私鑰所簽發；目前，我們使用 Infineon SLE 4428 的 IC 卡記憶晶片來儲存 TSA 的私鑰，這個 IC 卡擁有 1024 位元的 EEPROM 記憶體，並且其內容被 4 個可改寫安全編碼（PSC）的 16 進制位元所保護；同時 USB IC 卡讀卡機則是具 Linux 驅動程式的 Omnikey CardMan 讀卡機。

茲將 TTS 伺服端規格整理如下所示：

表 1 TTS 伺服端規格

規格名稱	規格說明
作業系統	Unix-like
訊息摘要演算法	SHA1、MD5
數位簽章	sha1WithRSAEncryption、md5WithRSAEncryption
通訊協定	HTTP、HTTPS
私鑰保護	USB IC 卡讀卡機、IC 卡
時間來源	硬體時間卡取得

3.2 TTS 客戶端

表 2 TTS 客戶端規格

規格名稱	規格說明
CPU	Pentium III 500GHz
記體體	192MB
作業系統	Windows 2000 Professional
程式語言	C++ Builder 6.0
時戳請求格式	符合 RFC-3161 標準
訊息摘要	SHA1、MD5

如表 2 所示，我們在 Windows 平台上使用 Borland C++ Builder 6 做為開發 TTS 客戶端軟體，並透過網際網路使用 HTTP 通訊協定連結 TTS 伺服端。

當 TTS 客戶端要求進行時戳服務時，必須先使用 IC 卡通行碼進行認證（圖 3）。因此，假設有人企圖對伺服端以類似 DoS 攻擊手法，大量發出時戳服務請求時，則伺服器可先根據通行碼過濾無法通過認證的請求服務。

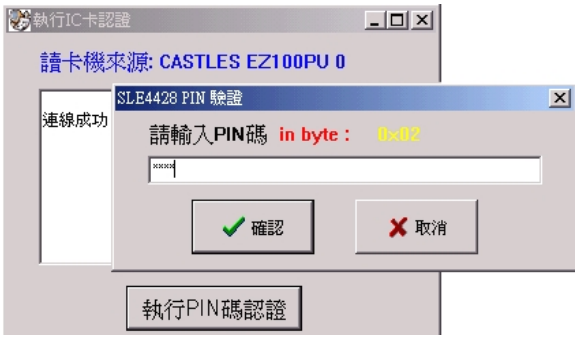

圖 3 IC 卡通行碼認證

TTS 客戶端允許藉由選擇一個檔案和選擇一個訊息摘要演算法（如 MD5 或 SHA-1）來建立時戳請求，並且傳送時戳請求給 TTS 伺服端。圖 4 即表示 TTS 客戶端的基本操作。

圖 4 TTS 客戶端軟體建立時戳請求服務

當從 TTS 伺服端接到時戳回應時，TTS 客戶端能夠使用有用的方法剖析和展示回應結果，同時，TTS 客戶端軟體可儲存時戳回應，且可根據最初傳送給伺服端的時戳要求時所提交的資料（雜湊值）來做驗證。圖 5 即表示客戶端軟體驗證的過程。

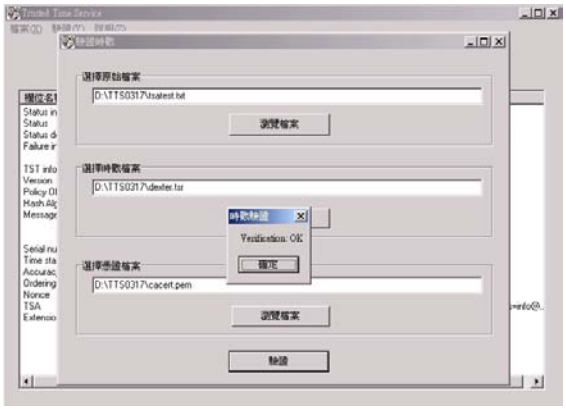

圖 5 時戳驗證

3.3 比較與分析

茲將本研究開發的 TTS 系統與原來 OpenTSA[13]做一比較，如表 3 所示，TTS 系統改善了 OpenTSA 諸多缺點，例如時間來源的取得、強化私鑰的保護、伺服端與客戶端的相互認證，進而防範 DoS 或 DDoS 攻擊等。

表 3 TTS 與 OpenTSA 比較表

差異性	TTS	OpenTSA
時間來源	硬體時間卡取得	gettimeofday() 函數取得
客戶端認證	有（使用 IC 卡、讀卡機）	無
伺服端認證	有	無
私鑰保護	有（使用 IC 卡、讀卡機）	無
防止 DoS 攻擊	有	無

肆、結語

如上所述，本研究中系統的特點如下：

（1）符合 RFC-3161 標準：本系統為 Client/Server 架構，並且符合 RFC-3161 標準，於 Linux 平台上架設 TTS 伺服端，並於 Windows 平台上開發客戶端應用軟體。

（2）整合時間卡：時戳服務的運作必須擁有精確的時間來源，本研究中於 TTS 伺服端加上 PCI 介面時間卡以取得精確且可任賴的時間。

（3）強化私鑰保護：TTS 伺服端加入 IC 卡和 USB 介面的 IC 卡讀卡機以加強保護及管理私密金鑰。

（4）防止阻斷服務攻擊：TTS 客服端加入 IC 卡和 USB 介面的 IC 卡讀卡機以作為使用者的通行認證，並藉以防範 DoS 或 DDoS 攻擊。

PKIs 在未來將會越來越盛行，而時戳服務（TTS）即為 PKI 的一部份；一個安全的、可驗證的且具稽核性的 TTS 服務將被有效的應用在電子化政府及電子商務中。在這個研究中，我們描述了我們依照 RFC-3161 標準來實作 TTS 客戶端及伺服端的初步成果。

參考文獻

[1] A. Kakura and S. Naito, A Secure and Trusted Time Stamping Authority, Internet Workshop, 88-93. 1999.

[2] A. Nash, W. Duan, C. Joseph, and D. Brink, PKI: Implementing and Managing E-Security, McGraw-Hill, 2001.

[3] An RSA Laboratories Technical Note Burton S. Kaliski Jr., A Layman's Guide to a Subset of ASN.1, BER, and DER, Revised November 1, 1993

[4] Borland Software Corp. C++Builder version 6.0, http://www.borland.com/cbuilder/

[5] C. Adams and S. Lloyd, Understanding Public-Key Infrastructure, Macmillan Technical Publishing, 1999.

[6] C. Adams, et al, Internet X.509 Public Key Infrastructure Time-Stamp Protocol, IETF RFC 3161, August 2001. http://www.ietf.org/rfc/rfc3161.txt

[7] C. C. Chang and S. J. Hwang, Using smart cards to authenticate remote passwords, Computers and Mathematics with Applications, Vol. 26, No. 7, 19-27, 1993.

[8] Datum.com, The Importance of Time, http://www.trusted-time.com/

[9] H. Massias and J.J. Quisquater, Time and Cryptography, TIMESEC Technical Report, 1997

[10] Infineon Technologies AG, SLE 4428 Chip, http://www.infineon.com/cgi/ecrm.dll/ecrm/scripts/prod_ov.jsp?oid=15066&cat_oid=-9520

[11] Omnikey CardMan Desktop USB2020, Omnikey AG, http://www.omnikey.com/en/produkt_details.php3?produkt=1&variante=3

[12] OpenSSL Project, http://www.openssl.org

[13] OpenTSA Project, http://www.opentsa.org

[14] P.A.S. Ward and D.J. Tayler, A Hierarchical Cluster Algorithm for Dynamic, Centralize Timestamps, International Conference on Distributed Computing Systems, 585 -593, 2001

[15] R. Rivest, The MD5 Message-Digest Algorithm, IETF RFC 1321, April 1992, http://www.ietf.org/rfc/rfc1321

[16] S. Haber and W.S. Stornetta, How to Time-Stamp a Digital Document, Journal of Cryptology, Vol. 3, No. 2, 99-111, 1991

[17] S. Haber, B. Kaliski, and W. Stornetta, How Do Digital Timestamps Support Digital Signatures?, Cryptobytes, Vol. 1, No. 3, 14-15, RSA Laboratories, Autumn 1995

[18] S. J. Wang and J. F. Chang, Smart card based secure password authentication scheme, Computers and Security, Vol. 15, No. 3, 231-237, 1996.

[19] Surety.com, Digital Notary Service Technical Overview, http://www.surety.com/

[20] Symmetricom Inc., PCI Time and Frequency Processor, http://www.symmetricom.com/products/product.php/107

[21] Viega, M. Messier, and P. Chandra, Network Security with OpenSSL, O'Reilly, 2002.

[22] W. H. Yang and S. P. Shieh. Password authentication schemes with smart cards. Computers & Security, Vol. 18, No. 8, 727-733, 1999.

[23] W. Rankl and W. Effing, Smart Card

Handbook, 2nd edition, John Wiley & Sons, 2000.

[24] William Stallings, Cryptography and Network Security: Principles and Practice, 3rd edition, Prentice-Hall, Inc. 2003.

[25] 王濟民（民 87），智慧卡（Smart Card）之推廣與應用分析，台灣大學商學研究所碩士論文。

[26] 楊中皇（民 91），密碼學演算法於 IC 卡上的具體實現，資訊安全通訊，第八卷第三期，頁 8~17。

[27] 經濟部商業司，電子簽章法。http://www.moea.gov.tw/~meco/doc/ndoc/s5_p05.htm

[28] 樊國楨（民 86），電子商務高階安全防護--公開金鑰密碼資訊系統安全原理，資訊與電腦出版社

[29] 賴溪松、韓亮、張真誠（民 92），近代密碼學及其應用，初版，旗標出版股份有限公司。