SCIS 2006 The 2006 Symposium on
Cryptography and Information Security
Hiroshima, Japan, Jan. 17-20, 2006
The Institute of Electronics,
Information and Communication Engineers

Design and Implementation of a live CD for Time-Stamping Service

Sheng-Hsiung Su* Chung-Huang Yang†

Abstract— As the Internet is vigorous development, security measures are expected to become more and more important on the Internet. A Time-stamping authority (TSA) is a trusted authority which provides a proof that a datum existed before a particular time. In this research, we implemented a RFC-3161 compliant live CD for time-stamping service over the Internet. The TSA server software was implemented on Linux Live CD platform using perl language and with modified open source software from OpenTSA while the TSA client software was implemented on Windows platform using C++ Builder tool. Our TTS live CD combines newsgroup with database. The time-stamping tokens information can be download from newsgroup and verified time stamp information, and the TSA server can issue time stamp token information to newsgroup instantly. These mechanisms will improve time-stamping service.

Keywords: Time-stamping, live CD, digital signature, PKI

1 Introduction

A time-stamping authority (TSA) [1,2,3,4,9] is a usually operated as trusted third party which provides a proof that a datum existed before a particular time. The time-stamping service (TTS) become a part of public-key infrastructures (PKI) [5,6] As documents or hash values of documents are send to TSA for time-stamping, the TSA server will response with a time-stamping token in order to indicate that the (hashed) datum existed at a particular point in time. There are many situations where we need to certify the date and time that some data was created or modified [7].

In this paper, we present our effort in the design and implementation of CA/PKI on the PC Windows environment. Borland C++ Builder 6 [11] is used as a software tool to develop Internet-based TSA client. The open source software provided by the OpenTSA project [14] is used as a middleware to build a TSA server on the Linux live CD environment. IF time-stamping authority can publish time-stamping token information and provides user to confirm, there will be more helpful to PKI. Usenet is a network service that unify news server, time-stamping token message is disseminated by newsgroup. The communication protocol between TTS clients and TTS server is followed IETF RFC3161 standard [15].

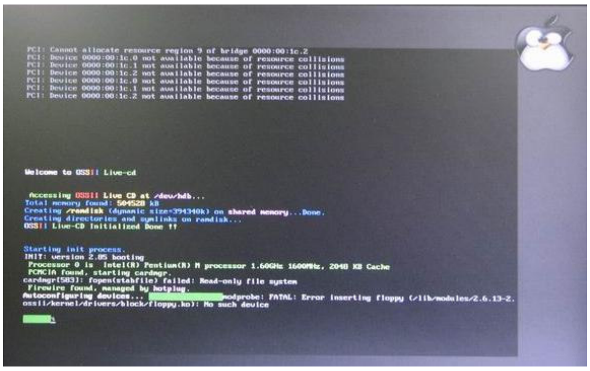
2 Implementation of TTS Server

OpenTSA [12,13,14] is an open-source implementation of SSL/TLS protocols for Linux and Windows platforms. It also contains basic cryptography functions, such as message encryption and digital signature. We installed and set up OpenTSA software on a Linux machine with extra C codes and perl [11] codes to provide RFC3161-compatible time-stamping format so that they will act as Time-stamping authority.

Fedora live CD is a convenient tool to help system administrator manage network service. The user can modify Fedora live CD and rebuild system to support service that you want easily. In this research, we integrate Opentsa to Fedora live CD and support time-stamping service. Figure 1 shows TTS live CD turn on.

*Institute of Information and Computer Education, National Kaohsiung Normal University (NKNU), 116, Ho Ping First Road, Kaohsiung 802, TAIWAN, shsu@icemail.nknu.edu.tw

†(NKNU) http://crypto.nknu.edu.tw/, chyang@computer.org



```
PCI: Cannot allocate resource region 9 of bridge 0000:00:1c.2
PCI: Device 0000:00:1c.0 not available because of resource collisions
PCI: Device 0000:00:1c.1 not available because of resource collisions
PCI: Device 0000:00:1c.2 not available because of resource collisions
PCI: Device 0000:00:1c.0 not available because of resource collisions
PCI: Device 0000:00:1c.1 not available because of resource collisions
PCI: Device 0000:00:1c.2 not available because of resource collisions



Welcome to OSS!! Live-cd

 Accessing OSS!! Live CD at /dev/hdb...
Total memory found: 504528 kB
Creating /ramdisk (dynamic size=394340k) on shared memory...Done.
Creating directories and symlinks on ramdisk...
OSS!! Live-CD Initialized Done !!

Starting init process.
INIT: version 2.85 booting
 Processor 0 is  Intel(R) Pentium(R) M processor 1.60GHz 1600MHz, 2048 KB Cache
 PCMCIA found, starting cardmgr.
cardmgr[583]: fopen(stabfile) failed: Read-only file system
 Firewire found, managed by hotplug.
Autoconfiguring devices...                    modprobe: FATAL: Error inserting floppy (/lib/modules/2.6.13-2.
ossii/kernel/drivers/block/floppy.ko): No such device
```

Figure 1: Screen of TTS live CD turn on

TTS live CD is based on Fedora Core 4 live CD which adds and modifies OpenTSA tools to integrate newsgroup and database [8,14,16,17,18]. Figure 2 shows external architecture of the TTS. The time is important for the TTS server, especially in precision. In this research, our time originates from the NTP (Network Time Protocol) server.

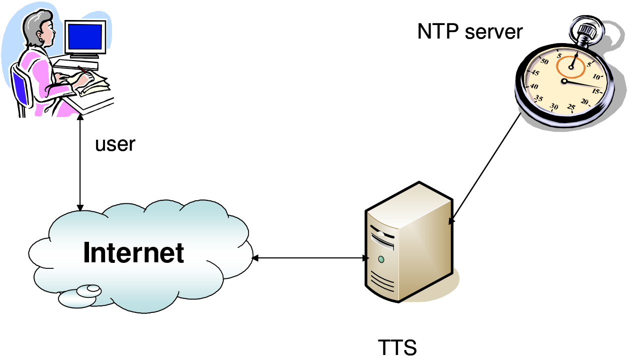


Figure 2: External Architecture of the TTS

Our TTS include Mysql [17] database which have two tables, respectively is token_id and token_date. When the client submits the time-stamping request (tsq), TTS will send time-stamping respond (tsr) and will store token_id and token_date in the database. Besidies Our TTS server use inn [18] tool and pnews [8] to install news server support Apache. Figure 3 shows internal architecture of the TTS.

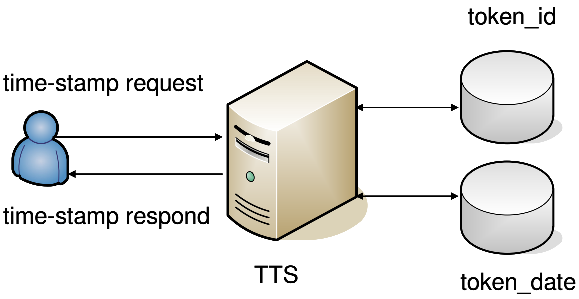
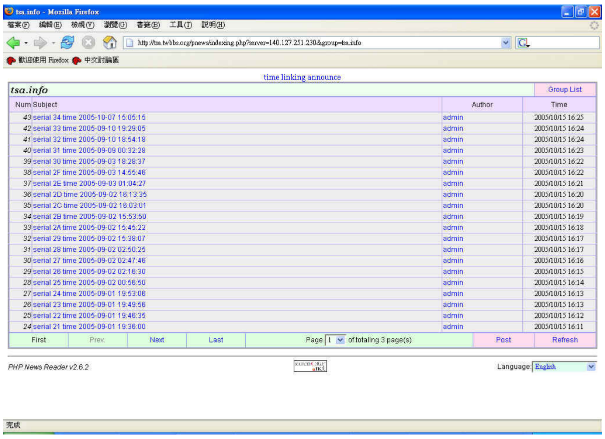


Figure 3: Internal Architecture of the TTS

If TTS creates time-stamping respond (tsr), time-stamping token message will issue on newsgroup, which can be demonstrated by HTTP (Hyper Text Transfer Protocol) and NNTP (Network News Transfer Protocol) protocol. Figure 4 shows time-stamping authority issue time-stamping token message on web [8,18].


tsa.info - Mozilla Firefox
檔案(F) 編輯(E) 檢視(V) 瀏覽(G) 書籤(B) 工具(T) 說明(H)
http://tsa.twbbs.org/pnews/indexing.php?server=140.127.251.230&group=tsa.info
歡迎使用 Firefox
中文討論區
time linking announce
tsa.info
Group List

Num	Subject	Author	Time
43	serial 34 time 2005-10-07 15:05:15	admin	2005/10/15 16:25
42	serial 33 time 2005-09-10 19:29:05	admin	2005/10/15 16:24
41	serial 32 time 2005-09-10 18:54:18	admin	2005/10/15 16:24
40	serial 31 time 2005-09-09 00:32:28	admin	2005/10/15 16:23
39	serial 30 time 2005-09-03 18:28:37	admin	2005/10/15 16:22
38	serial 2F time 2005-09-03 14:55:46	admin	2005/10/15 16:22
37	serial 2E time 2005-09-03 01:04:27	admin	2005/10/15 16:21
36	serial 2D time 2005-09-02 16:13:35	admin	2005/10/15 16:20
35	serial 2C time 2005-09-02 16:03:01	admin	2005/10/15 16:20
34	serial 2B time 2005-09-02 15:53:50	admin	2005/10/15 16:19
33	serial 2A time 2005-09-02 15:45:22	admin	2005/10/15 16:18
32	serial 29 time 2005-09-02 15:38:07	admin	2005/10/15 16:17
31	serial 28 time 2005-09-02 02:50:25	admin	2005/10/15 16:17
30	serial 27 time 2005-09-02 02:47:46	admin	2005/10/15 16:16
29	serial 26 time 2005-09-02 02:16:30	admin	2005/10/15 16:15
28	serial 25 time 2005-09-02 00:56:50	admin	2005/10/15 16:14
27	serial 24 time 2005-09-01 19:53:06	admin	2005/10/15 16:13
26	serial 23 time 2005-09-01 19:49:56	admin	2005/10/15 16:13
25	serial 22 time 2005-09-01 19:46:35	admin	2005/10/15 16:12
24	serial 21 time 2005-09-01 19:36:00	admin	2005/10/15 16:11

Figure 4 : Issue time-stamping token on web

3 Implementation of TTS Client

We developed a TTS client software on Windows platforms which interfaces with TTS server through HTTP and NNTP protocol over the Internet. The client software, developed using C++ Builder version 6.0 [11] tool from Borland programming environment. A TSA client allows creating a time-stamping request by selecting a file and choosing a message-digest algorithm , then sends the time-stamping request to the TSA [[3]. The main interface of TSA client is illustrated in Figure 5.

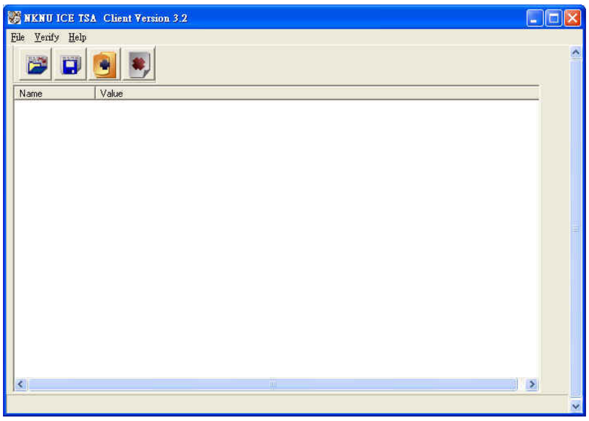
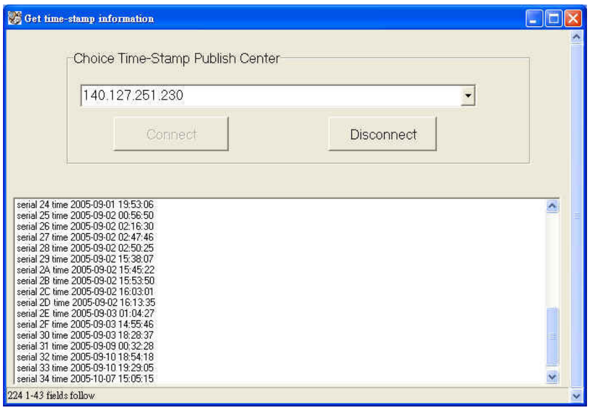


Figure 5 : TTS client main interface

Upon receiving time-stamping response from the TTS server, our TTS client software could parse and display the response in a meaningful manner. The TTS client software could also save time-stamping response and verify it against presentation of the original (hashed) datum. on which the time-stamping request had been sent to TTS. Besides the time-stamping tokens can be download from newsgroup. The outcome of TTS client download newsgroup with time-stamping token is illustrated in Figure 6.



140.127.251.230

```
serial 24 time 2005-09-01 19:53:06
serial 25 time 2005-09-02 00:56:50
serial 26 time 2005-09-02 02:16:30
serial 27 time 2005-09-02 02:47:46
serial 28 time 2005-09-02 02:50:25
serial 29 time 2005-09-02 15:38:07
serial 2A time 2005-09-02 15:45:22
serial 2B time 2005-09-02 15:53:50
serial 2C time 2005-09-02 16:03:01
serial 2D time 2005-09-02 16:13:35
serial 2E time 2005-09-03 01:04:27
serial 2F time 2005-09-03 14:55:46
serial 30 time 2005-09-03 18:28:37
serial 31 time 2005-09-09 00:32:28
serial 32 time 2005-09-10 18:54:18
serial 33 time 2005-09-10 19:29:05
serial 34 time 2005-10-07 15:05:15
224 1-43 fields follow
```

Figure 6: Download time-stamping tokens

4 Conclusions

PKI will become more and more prevalent in the near future while time-stamping service (TTS) is a part of PKI. A secure and convenient TTS solution will be useful in e-commerce applications. In this paper, we have described our preliminary results on implementing RFC3161-compatible TTS client and TTS live CD combines newsgroup with database for time stamping service.

References

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

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

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

[4] H. Massias and J.J. Quisquater, "Time and Cryptography," *TIMESEC Technical Report*, 1997.

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

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

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

[8] pnews, http://sourceforge.net/projects/pnews/

[9] Chung-Huang Yang, Chih-Ching Yeh, and Li-Ling Hsu, "On the design and Implementation of a Secure Time-Stamping Service," *The 2004 Symposium on Cryptography and Information Security* Sendai, Japan, Jan.27-30, 2004.

[10] Larry Wall, Tom Cbristiansen and Jon Orwant, Programming Perl, 3rd Ed, O'RELLY, 2000.

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

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

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

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

[15] C. Adams, et al, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol," IETF RFC 3161, August 2001.

[16] Fedora Core4 liveCD, http://knoppix.tnc.edu.tw/

[17] Mysql project, http://www.mysql.com/

[18] Inn project, http://www.isc.org/