SCIS 2002 The 2002 Symposium on
Cryptography and Information Security
Shirahama, Japan, Jan. 29 - Feb.1, 2002
The Institute of Electronics,
Information and Communication Engineers

Design and Implementation of CA/PKI on the Windows Environment

曾秀琦*1 楊中皇*2
Xiu-Gi Zeng Chung-Huang Yang

Abstract CA and PKI are crucial parts of many secure network applications. In this paper, we present our effort in the design and implementation of CA/PKI on the PC Windows environment. The Borland C++Builder 5 is used as a software tool to develop client and server software while the Microsoft SQL is used to manage X.509v3 certificates stored at CA. Our CA server is also equipped with a proprietary hardware device or an off-the-shelf -smart card and reader to store and retrieve private key of the CA. Both client and server software will be freely released to public in the near future.

Keyword Certification authority, CA, public-key infrastructure, PKI, time stamping, cryptography

1 Introduction

The increasing use of electronic means of data communications, coupled with the growth of computer usage, has extended the need to protect information. Considerable progress has been made in the techniques for encryption, decryption, and fending off attacks from intruders over the last decade.

Public-key infrastructures (PKI) [1] are comprised of supporting services that are needed for using public-key technologies on a large scale. A public-key certification system works by having a certification authority (CA) for the generation and management (application, storage, renewal, revocation, and inquiry) public-key certificates. Both CA and PKI are crucial parts of many secure network applications such as VPN, secure email, online stock trading, time-stamping, etc.

In this paper, we present our effort in the design and implementation of CA/PKI on the PC Windows environment. Borland C++Builder 5 [2] is used as a software tool to develop Internet-based client and server software on the Windows platform and Microsoft SQL database is used to manage X.509v3 certificates [3] at the CA server. Each certificate contains a public-key value and information about a particular person, agency, and other entity that holds the corresponding private-key. Certificates are digitally signed by the issuing CA, using CA's private-key. To safeguard private key of CA, our CA server is equipped with a proprietary hardware device based on the Motorola MC68HC908AB32 [4] microcontroller or an off-the-shelf smart card [5].

2 Design and Implementation of CA/PKI

PKI could have multiple CAs; each CA services a set of users and issues certificates for those users. Depending largely on how the trust relationship between CAs is arranged, the PKI provides a method for validating a complete certification path traversing multiple CAs from the CA that's certifying other party's public-key to a root CA whose public-key has already been held in each PKI client. Figure 1 shows system diagram of our PKI.

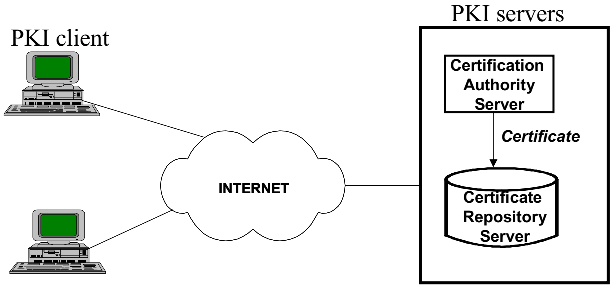


Figure 1: Client-server PKI over Internet

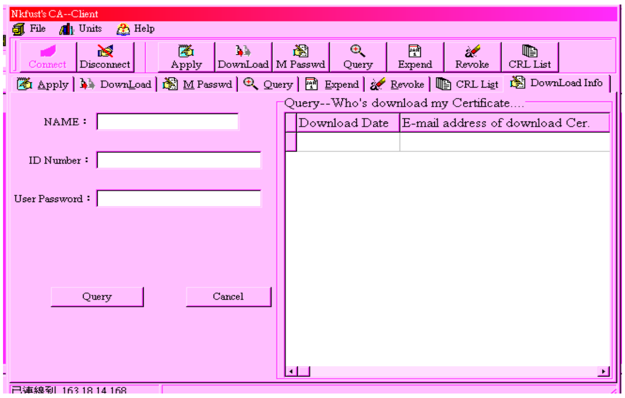
At present, our PKI involves a tree-structure of CA servers with each CA node certifying its children nodes while being certified by its parent node. This top-down

* 國立高雄第一科技大學 資訊管理系, 中華民國台灣〒824 高雄縣燕巢鄉大學路1號, National Kaohsiung First University of Science and Technology, 1 University Road, Yenchao, Kaohsiung 824, TAIWAN, R.O.C., http://www.crypto.nkfust.edu.tw
[1] Email: u8924808@cc.nkfust.edu.tw
[2] Email: chyang@computer.org

hierarchical structure allows easy implementation and each party only need to hold a copy of the top-level CA's public-key. This also reflects the hierarchical structure of our government.

Borland's C++Builder 5 [2] is an object-oriented, visual programming environment, it provides tools to develop, test, debug, and deploy applications, including a large library of reusable components, a suite of design tools, application and form templates, and programming wizards. Using C++Builder 5, we developed our PKI client and server software.

The PKI client could be integrated into PKI-enabled applications, such as secure email system or IPSec/VPN [6]. The client software, illustrated in Figure 2, will be responsible for the preparation of public-key information and connects to the CA server for getting X.509v3 certificate. It is also capable of revoking old certificates, renew own certificates, querying certificates, or obtaining certificate revocation lists (CRLs).



Query--Who's download my Certificate....

NAME :

ID Number :

User Password :

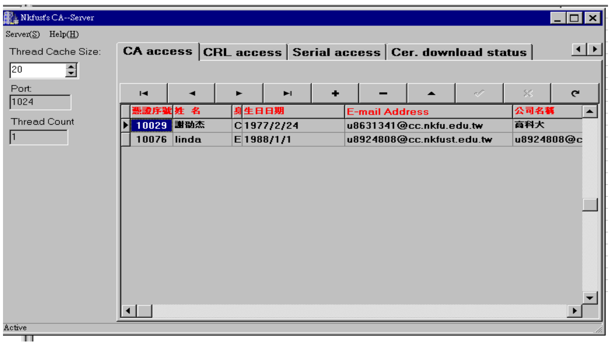
Download Date	E-mail address of download Cer.

已連線到 163.18.14.168

Figure 2: PKI client software

PKI servers could provide security services, including confidentiality, authentication, and non-repudiation. The core components of PKI servers are servers for certification authority (CA), certificate repository, time stamping, and certificate revocation. At present, the functions of our CA and certificate repository are implemented using Microsoft's SQL database to manage certificates issued by the CA.

Figure 3 shows the user interface of the implemented CA server. CA fundamentally performs the generation and management (application, storage, renewal, revocation, and inquiry) public-key certificates that are digitally signed by the CA's private key.



Thread Cache Size: 20

Port: 1024

Thread Count 1

憑證序號	姓 名	身	生日日期	E-mail Address	公司名稱
10029	謝劭杰	C	1977/2/24	u8631341@cc.nkfu.edu.tw	高科大
10076	linda	E	1988/1/1	u8924808@cc.nkfust.edu.tw	u8924808@c

Figure 3: Certification authority server

3 Safeguarding Private Keys of CAs

In order to protect the private key of CA, each CA server in our system is equipped with a proprietary hardware device or an off-the-shelf smart card and reader to store and retrieve private key, as shown in Figure 4.

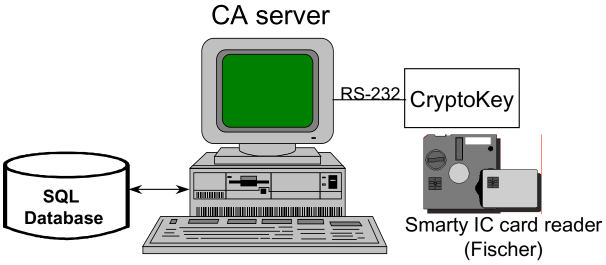


Figure 4: Private key of CA is stored in hardware devices

We developed a proprietary hardware device, CryptoKey, based on the Motorola's MC68HC908AB32 [4] microcontroller which offers an 8-bit 6808 CPU, 32K-byte flash ROM, 512-byte EEPROM, and 1K-byte RAM and a serial communication interface (SCI). It can be operated at maximum internal clock rate of 8 MHz and the EEPROM area is where we store cryptographic keys. The private key is stored in an encrypted form using the AES-128 algorithm [7]. Table 1 shows the performance of implemented AES-128 on the MC68HC908AB32.

Table 1: Performance of the AES-128 on MC68HC908AB32 Microcontrollers

Key Schedule	Encryption	Throughput
0.22 ms 1759 cycles	0.9 ms 7258 cycles	141 Kbits/s

4 Conclusions

Public-key infrastructure is essential for large-scale secure network applications, where communication confidentiality, authentication, and non-repudiation services are made an integrated part of the systems. We have presented a design and implementation of CA/PKI on the Windows environment. The developed software will be freely released to public in the near future.

References:

1. C. Adams and S. Lloyd, *Understanding Public-Key Infrastructure*, Macmillan Technical Publishing, 1999.
2. Borland C++Builder, http://www.borland.com/bcppbuilder/.
3. ITU-T Recommendation X.509 “Information Technology - Open Systems Interconnection - The Directory: Authentication Framework,” 1998.
4. *MC68HC908AB32 HCMOS Microcontroller Unit*, Rev. 1, Motorola Ltd., August 2000. See http://www.mcu.motsps.com/ or http://e-www.motorola.com/brdata/PDFDB/docs/MC68HC908AB32.pdf.
5. W. Rankl and W. Effing, *Smart Card Handbook*, 2nd edition, John Wiley & Sons, 2000.
6. N. Doraswamy and D. Harkins, *IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks*, Prentice Hall, 1999.
7. National Institute of Standards and Technology, “Advanced Encryption Standard (AES),” Federal Information Processing Standard, FIPS PUB 197, November 26, 2001.