Design and Implementation of CA/PKI on the Windows Environment

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
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).

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.

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.

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>
<thead>
<tr>
<th>Key Schedule</th>
<th>Encryption</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.22 ms</td>
<td>0.9 ms</td>
<td>141 Kbits/s</td>
</tr>
<tr>
<td>1759 cycles</td>
<td>7258 cycles</td>
<td></td>
</tr>
</tbody>
</table>

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: