

Classic Client 6.1 for Linux

User Guide

All information herein is either public information or is the property of and owned solely by Gemalto NV. and/or its subsidiaries who shall have and keep the sole right to file patent applications or any other kind of intellectual property protection in connection with such information.

Nothing herein shall be construed as implying or granting to you any rights, by license, grant or otherwise, under any intellectual and/or industrial property rights of or concerning any of Gemalto's information.

This document can be used for informational, non-commercial, internal and personal use only provided that:

- The copyright notice below, the confidentiality and proprietary legend and this full warning notice appear in all copies.
- This document shall not be posted on any network computer or broadcast in any media and no modification of any part of this document shall be made.

Use for any other purpose is expressly prohibited and may result in severe civil and criminal liabilities.

The information contained in this document is provided "AS IS" without any warranty of any kind. Unless otherwise expressly agreed in writing, Gemalto makes no warranty as to the value or accuracy of information contained herein.

The document could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Furthermore, Gemalto reserves the right to make any change or improvement in the specifications data, information, and the like described herein, at any time.

Gemalto hereby disclaims all warranties and conditions with regard to the information contained herein, including all implied warranties of merchantability, fitness for a particular purpose, title and non-infringement. In no event shall Gemalto be liable, whether in contract, tort or otherwise, for any indirect, special or consequential damages or any damages whatsoever including but not limited to damages resulting from loss of use, data, profits, revenues, or customers, arising out of or in connection with the use or performance of information contained in this document.

Gemalto does not and shall not warrant that this product will be resistant to all possible attacks and shall not incur, and disclaims, any liability in this respect. Even if each product is compliant with current security standards in force on the date of their design, security mechanisms' resistance necessarily evolves according to the state of the art in security and notably under the emergence of new attacks. Under no circumstances, shall Gemalto be held liable for any third party actions and in particular in case of any successful attack against systems or equipment incorporating Gemalto products. Gemalto disclaims any liability with respect to security for direct, indirect, incidental or consequential damages that result from any use of its products. It is further stressed that independent testing and verification by the person using the product is particularly encouraged, especially in any application in which defective, incorrect or insecure functioning could result in damage to persons or property, denial of service or loss of privacy.

© Copyright 2008-2011 Gemalto N.V. All rights reserved. Gemalto and the Gemalto logo are trademarks and service marks of Gemalto N.V. and/or its subsidiaries and are registered in certain countries. All other trademarks and service marks, whether registered or not in specific countries, are the property of their respective owners.

GEMALTO, B.P. 100, 13881 GEMENOS CEDEX, FRANCE.

Tel: +33 (0)4.42.36.50.00 Fax: +33 (0)4.42.36.50.90

Printed in France.

Document Reference: D1228558A

June 10, 2011

Introduction		v
Classic Client	v
Who Should Read This Book	v
Documentation	v
Conventions	vi
Typographical Conventions	vi
Additional Resources	vi
For Further Help	vi
If You Find an Error	vi
Chapter 1	Installation	1
System Requirements	1
Computer	1
Operating Systems	1
Applications	2
Peripherals	2
Installing Classic Client 6.1 for Linux	2
Installing the Classic Client 6.1 for Linux Software	2
Connecting the Smart Card Reader	2
Configuring Gemalto Cryptographic Security Modules	3
Chapter 2	PIN Management	7
About PINs	7
PIN Types	7
The Administrator PIN	7
The User PIN	8
PIN Security Policies	8
Classic Client PIN Management Tool	9
PIN PAD Readers	9
PIN Management Tasks	9
Chapter 3	Fingerprint Authentication	13
About Fingerprints	13
Requirements	13
Authentication Process	13
Chapter 4	Tasks	15
How to Use E-mail Securely	15
About Secure E-mail	15
Working with Mozilla Thunderbird or Icedove.	16
How to View Secure Web Sites	23
Choosing a Certificate to Authenticate Yourself to Secure Web Sites	23
How to Sign PDF Documents	26

Appendix A	Security Basics	31
	Cryptography	31
	Secret Key Cryptography	32
	Public Key Cryptography	32
	What is Classic Client?	35
Terminology		37
	Abbreviations	37
	Glossary	38

List of Figures

Figure 1 - Encryption Tab in Advanced Dialog	3
Figure 2 - Device Manager	4
Figure 3 - The Load PKCS#11 Device Dialog Box	4
Figure 4 - Confirm Dialog	4
Figure 5 - Alert Dialog	5
Figure 6 - Cryptographic Modules Available	5
Figure 7 - Selecting a Smart Card Reader for the PIN Management Tool	9
Figure 8 - Classic Client PIN Management - Change PIN Function	9
Figure 9 - Classic Client PIN Management - Unblock PIN Function	11
Figure 10 - Fingerprint Capture Dialog Box	14
Figure 11 - Encrypt This Message	17
Figure 12 - Security Account Settings	18
Figure 13 - Enter Password	18
Figure 14 - Details of Selected Certificate	18
Figure 15 - "Use Same Certificate" Message	19
Figure 16 - Security Account Settings (2)	19
Figure 17 - New Msg Composition Window	20
Figure 18 - Message Security Info Window	21
Figure 19 - Mozilla Firefox Options Dialog	24
Figure 20 - Password Required	24
Figure 21 - Certificate Manager Window	25
Figure 22 - Security Settings Window	26
Figure 23 - Locate a PKCS#11 Module window	27
Figure 24 - Loaded Security Module	27
Figure 25 - Add ID in Security Settings Window	28
Figure 26 - Add Digital ID Dialog Box	28
Figure 27 - Digital ID Added in Security Settings Window	29
Figure 28 - Sign Document Dialog Box	29
Figure 29 - Save As Dialog Box	29

Welcome to Gemalto Classic Client for Linux.

You have made a wise investment by purchasing Classic Client as a safeguard for secure network services.

This chapter presents an overview of Classic Client, the documentation provided with it, and additional resources available for working with Classic Client.

Classic Client

Classic Client is for individual users, who want to use a smart card/token to protect information and transactions made via computers, including stand-alone workstations and Citrix client-server environments.

Note: A token is in fact a smart card embedded in a device that can be plugged into the USB port of a PC. In this document, “connecting a device” can mean inserting a card in a reader or plugging a token in the USB port of a PC.

With Classic Client you can use a digital certificate stored on a smart card/token to:

- Sign electronic documents.
- Open and verify signed documents.
- Send and receive secure e-mail using Mozilla e-mail software.
- Connect securely with a Web server.

Classic Client also includes features for managing certificates and smart card/token security.

This guide introduces you to Classic Client and provides easy-to-follow instructions. Read the entire guide for assistance in the installation, configuration, and use of Classic Client.

Who Should Read This Book

This guide is intended for Classic Client users who are familiar with smart cards/tokens and smart card reader technology, as well as PC hardware and software.

It is assumed that the user of Classic Client has:

- an understanding of the basic operations in a Linux OS.
- administrative privileges for the PC on which Classic Client will be installed.

Documentation

Classic Client is delivered with the following documentation:

- *Classic Client 6.1 for Linux* (this document). The file for this document is located on the Classic Client 6.1 CD and in the Classic Client installation folder.
- A *Classic Client 6.1 Release Notes* file. This contains any relevant information about the installation and the complete version history.

- *End User License Agreement (EULA)*

The EULA.rtf can be found after installation in the directory `usr/share/doc/libclassicclient`.

This document is best viewed with Adobe Acrobat Reader, version 7.0 or later. You can download Adobe Acrobat Reader from Adobe's Web site at: www.adobe.com.

Conventions

The following conventions are used in this document:

Typographical Conventions

Classic Client documentation uses the following typographical conventions to assist the reader of this document.

Convention	Example	Description
Courier	transaction	Code examples.
Bold	Enter libgclib.dylib	Actual user input or screen output.
>	Select File > Open	Indicates a menu selection. In this example you are instructed to select the “ Open ” option from the “ File ” menu.

Note: Example screen shots of the Classic Client for Linux software are provided throughout this document to illustrate the various procedures and descriptions. These screen shots were produced with Classic Client running on Debian.

Additional Resources

For further information or more detailed use of Classic Client, additional resources and documentation are available by contacting Gemalto technical support.

For Further Help

Further help is provided in the Gemalto Self Support portal at support.gemalto.com.

You can find information on how to contact your Gemalto representative by clicking **Contact Us** at the Gemalto web site, www.gemalto.com.

If You Find an Error

Gemalto makes every effort to prevent errors in its documentation. However, if you discover any errors or inaccuracies in this document, please inform your Gemalto representative. Please quote the document reference number found at the bottom of the legal notice on the inside front cover.

Installation

This chapter discusses information related to the installation of Classic Client 6.1 for Linux.

The installation requirements are outlined below.

This chapter describes:

- The hardware and software you need to use Classic Client 6.1 for Linux.
- How to install Classic Client 6.1 on your computer.

System Requirements

The following sections describe the hardware, operating systems, peripherals and software you need to use Classic Client 6.1. You must have administrator rights to the computer on which you are installing Classic Client.

Computer

The workstation must have at least 15 MB of available hard disk space and meet the normal system requirements to run the version of Linux installed.

Operating Systems

Classic Client for Linux supports the following operating systems:

- Ubuntu 10.04 LTS – 32-bit and 64-bit
- Other Linux operating systems upon request.
For details, contact your Gemalto technical consultant.

Gemalto recommends that your machine has a RAM at least equal to that normally recommended for the OS. If this RAM requirement is met, Classic Client for Linux should run normally.

Applications

For a detailed list of applications supported by Classic Client 6.1, please refer to the Release Notes. Here are some useful links where you can download the latest versions of some software applications free of charge:

- Mozilla Firefox and Thunderbird from www.mozilla.org.
- Adobe Acrobat and Adobe Acrobat Reader from www.adobe.com.
- Iceweasel 3.0 can be downloaded free of charge from various sites on the internet. Further information is available at <http://wiki.debian.org/Iceweasel>.

Peripherals

Classic Client 6.1 for Linux requires the following peripherals:

- A CD ROM drive (if the installation files are on a CD-ROM).
- An available USB port.

For a detailed list of the smart cards and smart card readers supported by Classic Client 6.1, refer to the Release Notes.

Installing Classic Client 6.1 for Linux

Installing the Classic Client 6.1 for Linux Software

Caution: Before installing the software make sure that your system has the latest version of the PC/SC Lite and CCID drivers.

To install Classic Client 6.1:

- 1 Begin by doing one of the following:
 - If your administrator has provided an installation CD-ROM, insert the CD-ROM into the CD-ROM reader of your PC.
 - If your administrator has made the installation program available from a network device, navigate to the network location and download the installation program to your computer.
- 2 Open a terminal and go to the directory where your installation program is. Start the installation program by typing the following line at the command prompt:
 - `DPKG -i libclassicclient_x.x.x-xx_i386_ubuntu.deb`
or
`apt-get install libclassicclient_x.x.x-xx_i386_ubuntu.deb`

Note: You can do this without going to the installation directory first, but if you do, you must type the full path for the file name.

Classic Client installs on your PC with no need for further intervention.

Connecting the Smart Card Reader

To use Classic Client on your workstation, you must connect a smart card reader to your computer.

If the card reader is not recognized on your workstation, you may need to install the latest card reader drivers. You can download these from <http://support.gemalto.com>.

Configuring Gemalto Cryptographic Security Modules

Security Modules are software add-ons that provide a variety of cryptographic services, such as secure browsing, and support the use of smart cards/tokens.

In Classic Client 6.1 for Linux, the PKCS#11 security module is installed automatically as it is included with the Classic Client software.

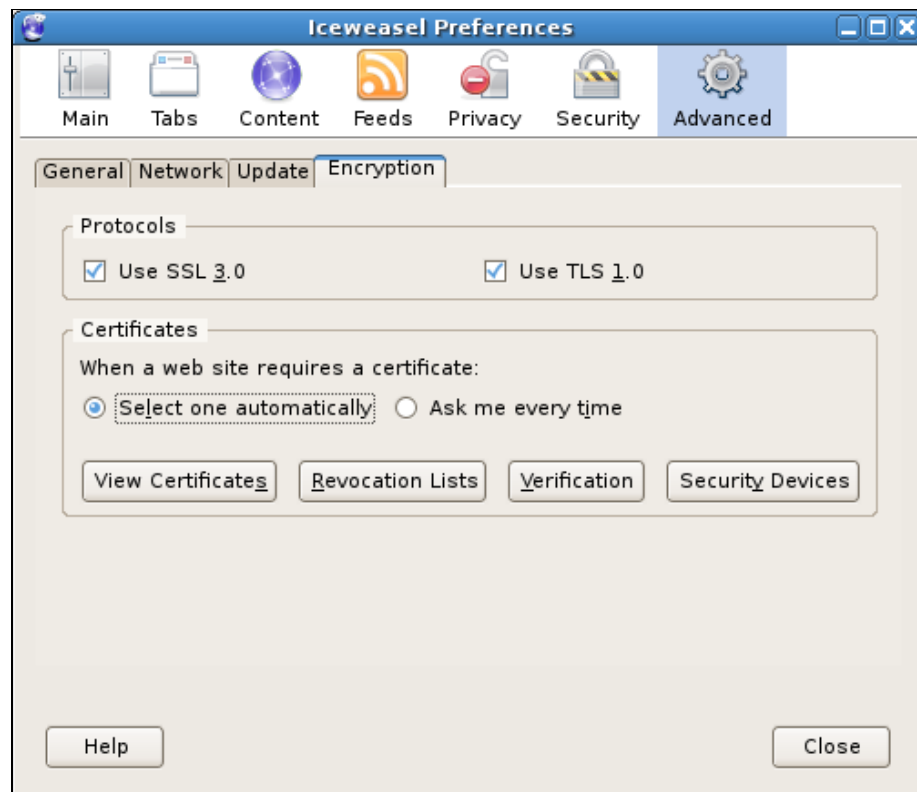
In order to enable the Mozilla applications Firefox and Thunderbird to communicate with Classic Client, the PKCS#11 security module must be registered in the Mozilla application.

Note: The screen shots in this section were taken on a PC running the Debian OS. In Debian, the Firefox browser is called Iceweasel, and its appearance is slightly different although its functionality is the same.

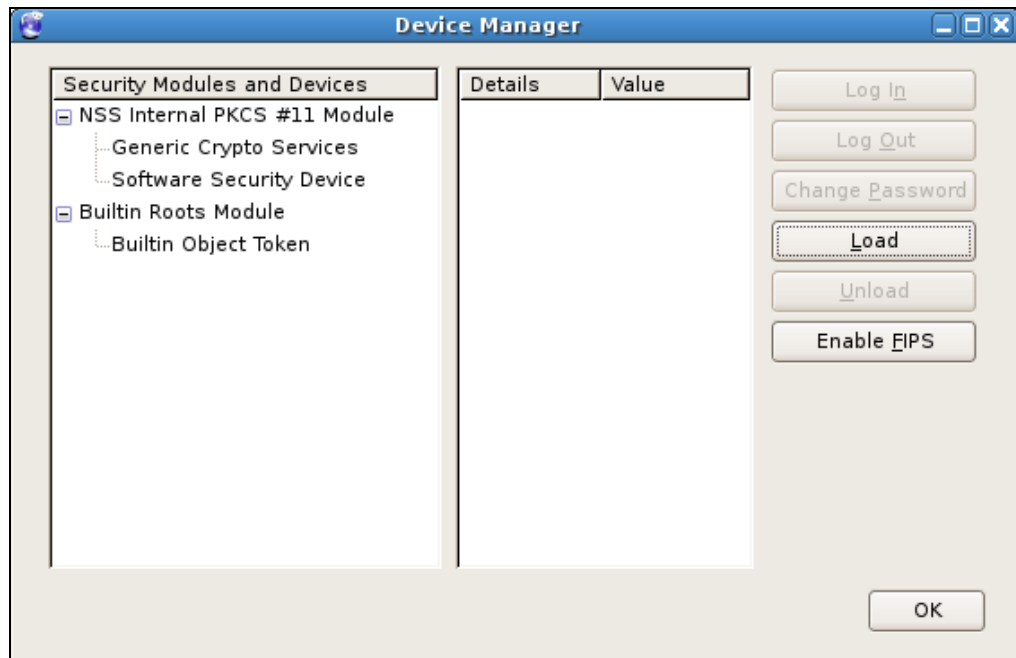
To configure Firefox (or Iceweasel) to recognize the security module:

- 1 Open **Firefox** and from the **Edit** menu choose **Preferences**.
- 2 In the dialog box that opens, click the **Advanced** icon, then the **Encryption** tab to display the settings as shown in “Figure 1”.

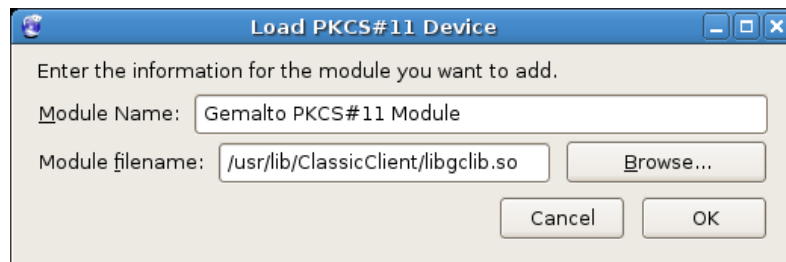
Figure 1 - Encryption Tab in Advanced Dialog



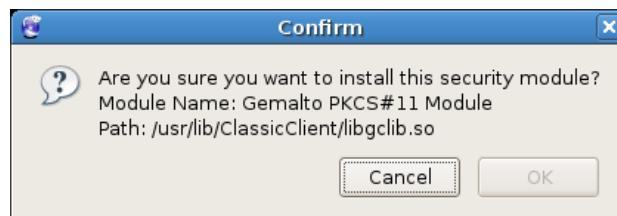
- 3 Click **Security Devices** to display the **Device Manager** window. This displays the modules currently available as shown in “Figure 2” on page 4.

Figure 2 - Device Manager

- 4 Click the **Load** button to the right in the dialog. This displays the **Load PKCS#11 Device** window, as shown in “Figure 3”.

Figure 3 - The Load PKCS#11 Device Dialog Box

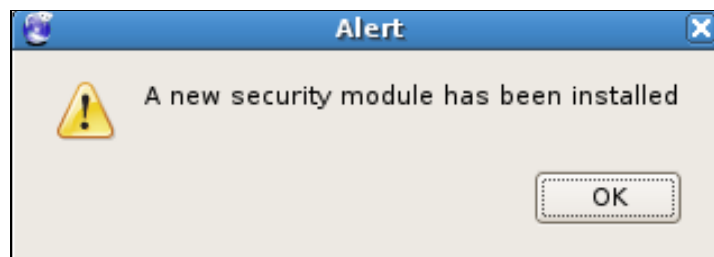
- 5 Enter a **Module Name**.
- 6 In **Module filename**, enter the full path and filename for the libgclib.so file:
By default, this is
`/usr/lib/ClassicClient/libgclib.so`
- 7 Click **OK**. The confirmation dialog appears as shown in the following figure:

Figure 4 - Confirm Dialog

- 8 Click **OK**.
A brief progress dialog appears indicating that the module is being loaded.

When this is completed the following **Alert** indicates that the module has been installed.

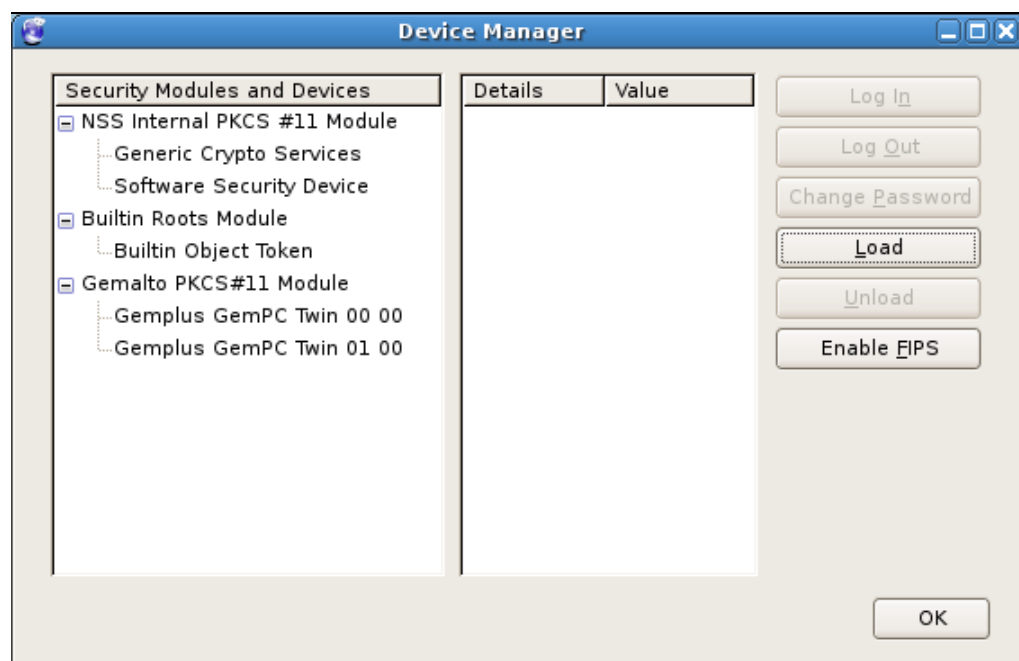
Figure 5 - Alert Dialog



- 9 Click **OK** to close this **Alert**.

The **Device Manager** indicates the presence of the new module as shown in "Figure 6".

Figure 6 - Cryptographic Modules Available



PIN Management

This chapter discusses the Classic Client PIN Management tool, the dedicated tool for managing PINs and the tasks it can be used to perform.

About PINs

PIN Types

Classic Client recognizes two types of PIN that may be in a smart card/token:

- Admin PIN – the PIN that is necessary to unblock the card/token (for example after too many consecutive incorrect presentations of the User PIN).
- User PIN – the standard PIN used by a user to access the card/token.

The Administrator PIN

This is the PIN used to unblock a User PIN. Normally only administrators know the value of this PIN.

The administrator PIN is an extremely important part of the security of the smart card/token. Knowledge of this PIN means you can change the value of all the user PINs on the card/token and unblock the card/token if the user PIN is blocked.

It is extremely important for smart card/token administrators to keep the value of the admin PIN secure and secret. The administrator must know the admin PIN value for all smart cards/tokens he or she has deployed. The admin PIN value of a card/token should never be shared with anyone else, and it is strongly recommended not to give this value to the card/token user, unless your security policy requests it.

Caution: Once an administration PIN has been entered incorrectly the requisite number of times, it becomes blocked and the card/token can never be used again.

The original Admin PIN value of a smart card/token is included in the packaging of the card/token. If you are an administrator you may want to change the Admin PIN value of the cards/tokens you deploy so that only you, the administrator, knows it.

The User PIN

A PIN (*Personal Identification Number*) is a private code. It can be a sequence of numeric or alphanumeric characters or a mix of the two and is used as a type of password. Your User PIN must be verified before you can perform security tasks with the card/token, such as logging on to a workstation, or creating a digital signature.

The user PIN of a smart card/token may be the original PIN value set at the time of manufacture or it may be a PIN value assigned by the administrator.

The user PIN should be unique to your card/token and known only to you. It is standard practice, upon reception of a smart card/token, to change the user PIN value so that only you, the user, knows it. Your administrator can even force you to change the PIN value upon first use in the software.

To perform a security operation, you must prove that you know the User PIN. Software that performs a security operation usually displays a window requesting you to enter the PIN before performing the security operation.

- When creating a digital signature, successful PIN validation proves that you are the real card/token holder and enables you to sign with the selected key.
- By using the PIN to log on a network, you prove both that your card/token is valid in the system and that you card/token holder, is physically there.

Caution: Do not allow the User PIN for your card/token to be blocked. If, for example, you forget the user PIN and enter a predetermined number of failed validation attempts (the PIN is entered incorrectly), the card/token becomes blocked and you cannot perform any further security operations with it. If you know the Admin PIN you can unblock your card/token as described in “How to Unblock a User PIN” on page 10. However most companies’ security policy does not allow this, in which case you must ask your Classic Client system administrator to unblock the card/token using the Administrator PIN. Sometimes card/token technology or software on-board the card/token limits the absolute number of these unblocking operations. For more information, see your card/token technology documentation.

PIN Security Policies

PIN policies are established according to a company’s security policy, but they are also established in relation to the particular type of smart card/token you use and the on-board software the card/token features. For example, some cards/tokens allow a user PIN to be a minimum of 4 characters, and other cards/tokens allow a minimum of 6 characters. Please see your card/token documentation for more information.

Classic Client PIN Management Tool

The Classic Client PIN Management tool allows you to make changes to the PINs associated with a particular smart card/token.

PIN PAD Readers

You can use the Gemalto PIN PAD, "PC Pinpad" with the PIN Tool. PC Pinpad behaves like a normal reader in transparent mode.

PIN Management Tasks

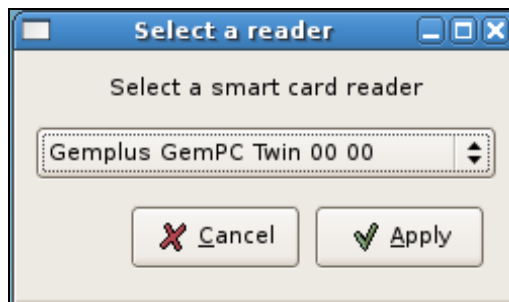
This section describes the tasks that you can perform with the PIN Management Tool.

How to Access the Classic Client PIN Management Tool

To access the PIN Tool:

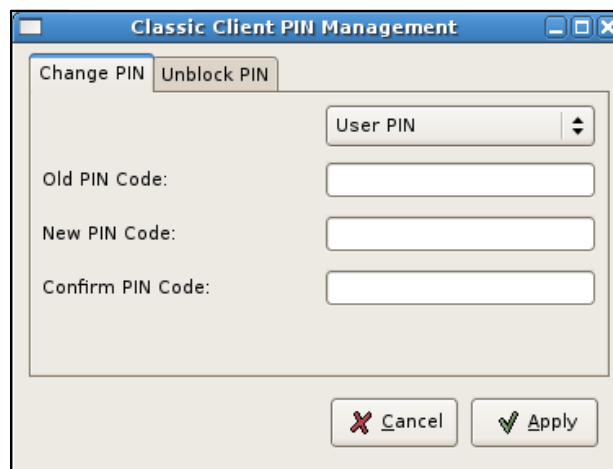
- 1 Make sure that your smart card/token is connected to your computer.
- 2 Either browse to /usr/bin/ and double-click **CCChangePinTool** or open a terminal, go to /usr/bin/ and type **./CCChangePinTool**.
- 3 When the window shown in "Figure 7" appears, select a smart card reader from the list and click **Apply**.

Figure 7 - Selecting a Smart Card Reader for the PIN Management Tool



This opens the **Classic Client PIN Management** Window as shown in "Figure 8".

Figure 8 - Classic Client PIN Management - Change PIN Function



How to Change an Administration PIN or User PIN

To change the Admin PIN, you will need to know its current value. This means that normally you will not be able to change an Admin PIN unless you are an administrator.

To change a PIN

- 1 Connect the smart card/token whose Admin PIN or User PIN you want to change to the PC.
- 2 Open the PIN Management window as described in “How to Access the Classic Client PIN Management Tool” on page 9.
- 3 If it is not already selected, click **Change PIN** at the top of the window (see “Figure 8” on page 9).
- 4 Select the PIN whose value you want to change from the list, **Admin PIN** or **User PIN**.
- 5 Enter the current value of the PIN in **Old PIN Code**, and the new value in **New PIN Code** and again in **Confirm PIN Code**.
- 6 Click the **Apply** button at the bottom of the window. A pop-up window appears to confirm a successful PIN change or to display an error message if unsuccessful.

How to Unblock a User PIN

Note: It is not possible to unblock an Admin PIN. If the Admin PIN becomes blocked, the smart card/token can no longer be used.

If you know the Admin PIN for your card/token, you can unblock your User PIN by using the Classic Client PIN Management tool.

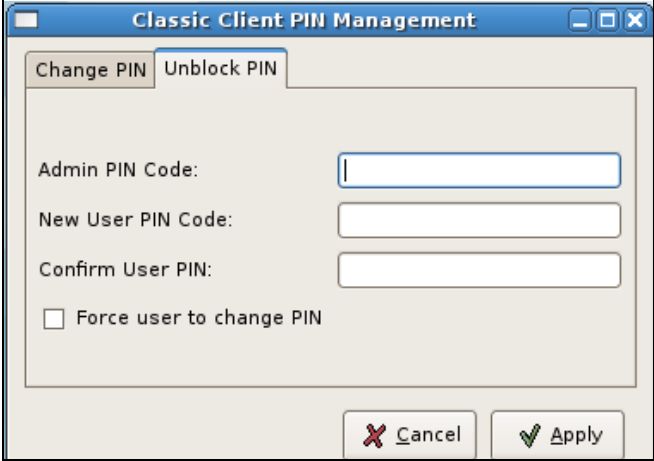
In most cases, if you are not an administrator you will not know the Admin PIN – it depends on your company’s security policy. In such cases, there are two possibilities;

- The administrator must unblock the smart card/token for you. You must return the smart card/token to the administrator so he or she can unblock it on his or her PC.

To unblock a PIN as an administrator:

- 1 Connect the blocked smart card/token to your administrator PC.
- 2 Open the Classic Client PIN Management window as described in “How to Access the Classic Client PIN Management Tool” on page 9.
- 3 If it is not already selected, click **Unblock PIN** at the top of the window as shown in “Figure 9” on page 11.

Figure 9 - Classic Client PIN Management - Unblock PIN Function



The screenshot shows a dialog box titled "Classic Client PIN Management" with two tabs: "Change PIN" and "Unblock PIN". The "Unblock PIN" tab is active. It contains three text input fields: "Admin PIN Code:", "New User PIN Code:", and "Confirm User PIN:". Below these fields is a checkbox labeled "Force user to change PIN". At the bottom right, there are two buttons: "Cancel" (with a red X icon) and "Apply" (with a green checkmark icon).

- 4 Enter the Admin PIN in **Admin PIN Code**, and the new value for the User PIN in **New User PIN Code** and again in **Confirm User PIN**.
- 5 For security reasons, Gemalto recommends that you check the box **Force user to change PIN**. This is particularly useful if the user whose PIN is being unblocked is not the administrator (as in most cases).
- 6 Click the **Apply** button at the bottom of the window. A pop-up window appears to confirm a successful **Unblock PIN** operation or to display an error message if unsuccessful.

Fingerprint Authentication

This chapter provides information on fingerprint authentication in the Classic Client. Fingerprint authentication can be used as an alternative to PIN authentication. Fingerprint authentication is supported in the tasks mentioned in "Chapter 4 - Tasks".

About Fingerprints

For cards that contain the IAS Classic Applet V3 and the Match On Card (MoC) applet, fingerprints can be used as an alternative to presenting a PIN. For fingerprint authentication, you must have a fingerprint scanner connected to the computer. (Please refer to the Classic Client Release Note to know which fingerprint scanners are supported.) To authenticate, place a finger on the sensor of the reader. Classic Client compares the digital fingerprint of the finger with the corresponding fingerprint stored in the MoC applet.

Caution: As with PINs, the number of attempts to perform a fingerprint authentication is limited. After a pre-defined number of failed attempts, you can no longer perform operations that require fingerprint authentication. **YOU CANNOT UNBLOCK FINGERPRINT AUTHENTICATION USING CLASSIC CLIENT.**

Requirements

For fingerprint authentication to work, the fingerprints must already be present on the smart card. The smart card must have the MoC (Match on Card) algorithm loaded on it.

Authentication Process

If the requirements are fulfilled and the applications are configured properly, the applications should prompt the user to authenticate with a fingerprint as shown below.

Figure 10 - Fingerprint Capture Dialog Box



When the above dialog box appears:

- 1 Choose the finger that you want to use for the authentication by clicking the option button next to the corresponding finger.
- 2 Place the finger on the scanner.
If successful, the fingerprint window disappears.

Tasks

This chapter discusses information related to specific tasks that you will most often be required to carry out when using the Classic Client 6.1 for Linux software and where to find the information about them.

These tasks are:

- “How to Use E-mail Securely” on this page.
- “How to View Secure Web Sites” on page 23
- “How to Sign PDF Documents” on page 26

Tasks concerning PINs are described in “Chapter 2 - PIN Management”.

How to Use E-mail Securely

The following sections explain how to send secure e-mail using Classic Client 6.1 for Linux.

About Secure E-mail

With Classic Client 6.1 for Linux, you can improve e-mail security by using the digital certificate on your smart card/token to:

- Sign your e-mail so that the recipient can verify that the message is really from you and has not been altered.
- Encrypt, or “scramble” a message so that only the intended recipient can read it. This eliminates concerns about intercepted messages and e-mail monitoring.
- Sign or encrypt your message using one e-mail program, while your intended recipient can read it with any other S/MIME-enabled e-mail program.
- Receive signed and encrypted e-mail messages.

Setting up Secure E-mail

You must do the following before you can send secure e-mail:

- **Configure the application to recognize the PKCS#11 security module**
- **Configure security settings**
Set the security settings for digitally signing and/or encrypting the contents and attachments of outgoing messages.
- **Specify certificates to be used for signing and encryption**

Choose the digital certificate(s) that you will use to encrypt and digitally sign your e-mails. You can use the same certificate for both operations or two different ones. These certificates are associated with your e-mail account.

- **Send yourself a digitally-signed e-mail**

When you send a signed e-mail, you sign it with the private key. The recipient receives the corresponding public key with the mail which he or she uses to decipher your mail.

Before you can send e-mails to anybody else, you need to send a signed message to yourself in order for Thunderbird to store your public key.

Then you can send your public key to other people, for example by sending them a signed message. Once they have your public key, they can use it to encrypt mails they send to you (which you decipher using your private key).

The following sections describe how to perform the above operations using the Mozilla Thunderbird e-mail program. The dialog boxes shown may differ slightly from your own software, depending on what version you are using.

Working with Mozilla Thunderbird or Icedove.

The following sections explain how to set up and send secure e-mail with Mozilla's Thunderbird e-mail program. The screenshots in this section were made using Icedove, which is the Debian version of Thunderbird. The two applications are identical except for their appearance.

There are three stages:

- 1 Configure Thunderbird to recognize the Security Module, described in the following section.
- 2 Configure the security settings and specify the certificates to use for signing and encryption, described on page 16.
- 3 Send a digitally signed e-mail to yourself in order to store your public key in Thunderbird, described on page 20.

Configure Thunderbird to Recognize the Security Module

You only need to do this once.

To configure Mozilla Thunderbird

- 1 Make sure your smart card/token is connected.
- 2 Start **Mozilla Thunderbird**.
- 3 Enter your password if you are prompted for it and click on **OK**.
- 4 For the rest of the procedure, follow the instructions in "To configure Firefox (or Icedove) to recognize the security module:" on page 3, except that in step 2 of those instructions, choose the **Certificates** tab instead of the **Encryption** tab.

This new module will be used with all e-mail you send with Thunderbird.

Configuring Settings and Specifying Certificates

You only need to do this the first time you use your card/token to sign or encrypt an e-mail.

Note: Although selecting the certificates is mandatory, this does not mean that you must sign and encrypt e-mails.

- 1 Make sure your smart card/token is connected.


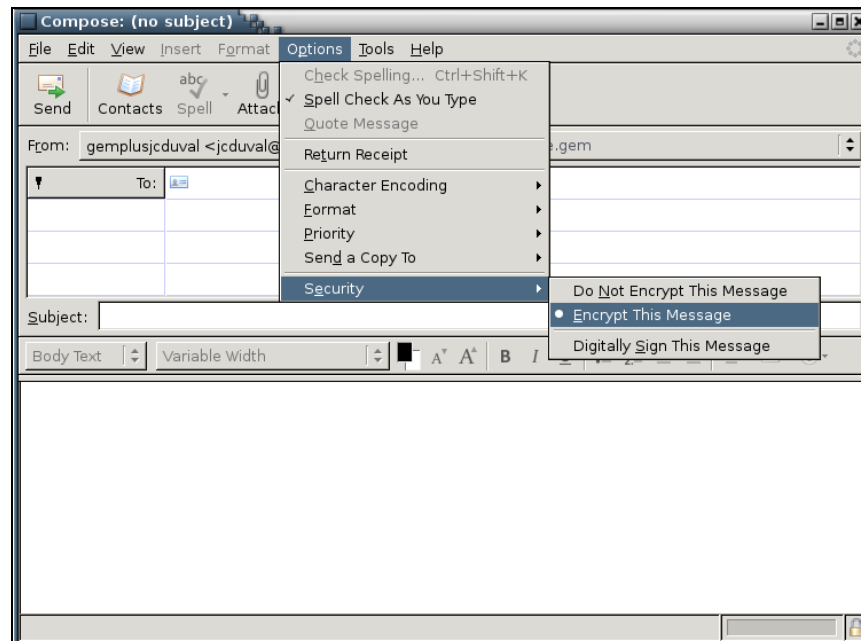
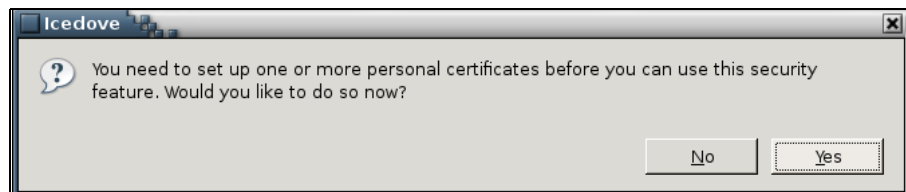
- 2 Start **Mozilla Thunderbird**.
- 3 Enter your password if you are prompted for it.
- 4 In **Thunderbird**, click the **Write** icon  .
This opens the **Compose** window.
- 5 In the **Compose** window's **Options** menu, choose **Security > Encrypt This Message** as shown in "Figure 11".

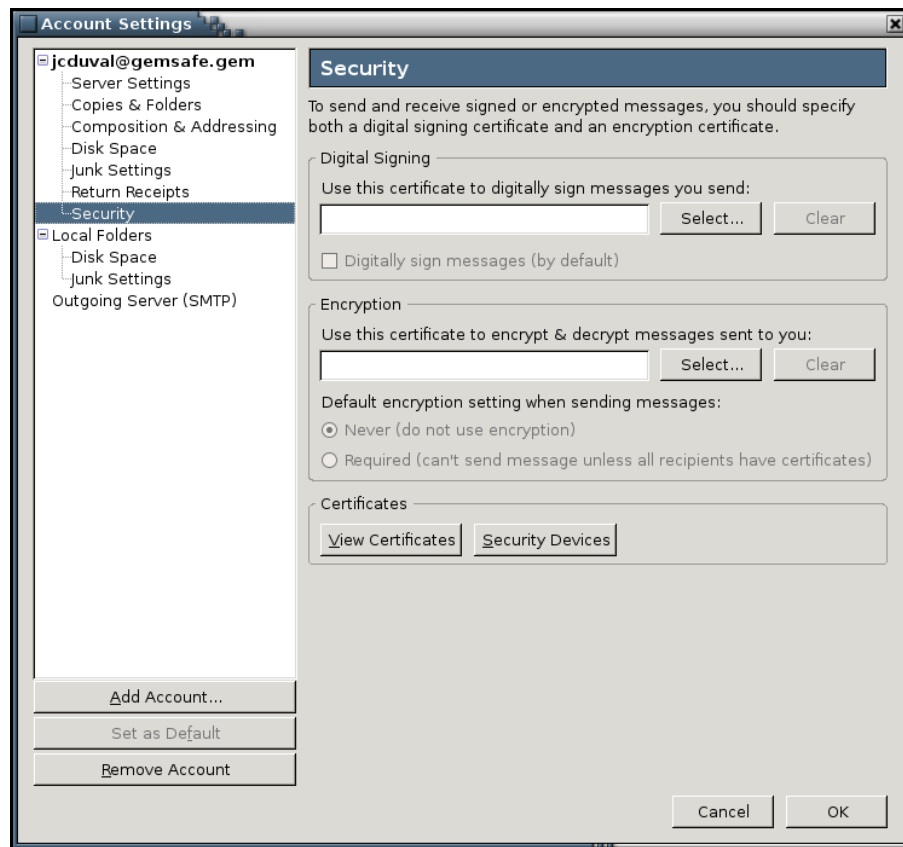
Figure 11 - Encrypt This Message



As the certificates in the card/token are not yet set up, the following message appears:

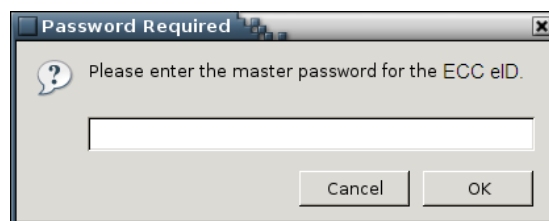


- 6 Click **Yes**. This opens the security account settings window for your e-mail account as shown in "Figure 12" on page 18.

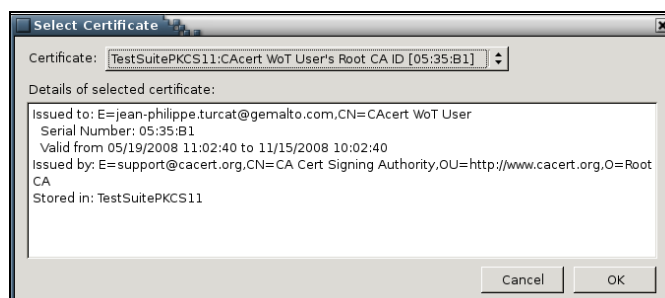
Figure 12 - Security Account Settings

- 7 In **Digital Signing**, click **Select** and choose the certificate you want to use from the list that appears.

Note: You may be prompted to enter a “master password” as shown in “Figure 13”. If so, enter the PIN for the card and click **OK**.

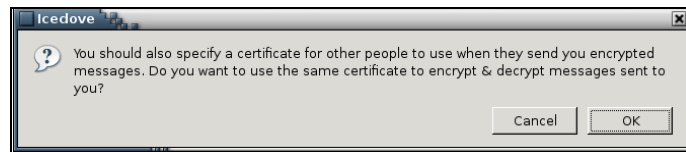
Figure 13 - Enter Password

The details of the selected certificate appear, as shown in “Figure 14”.

Figure 14 - Details of Selected Certificate

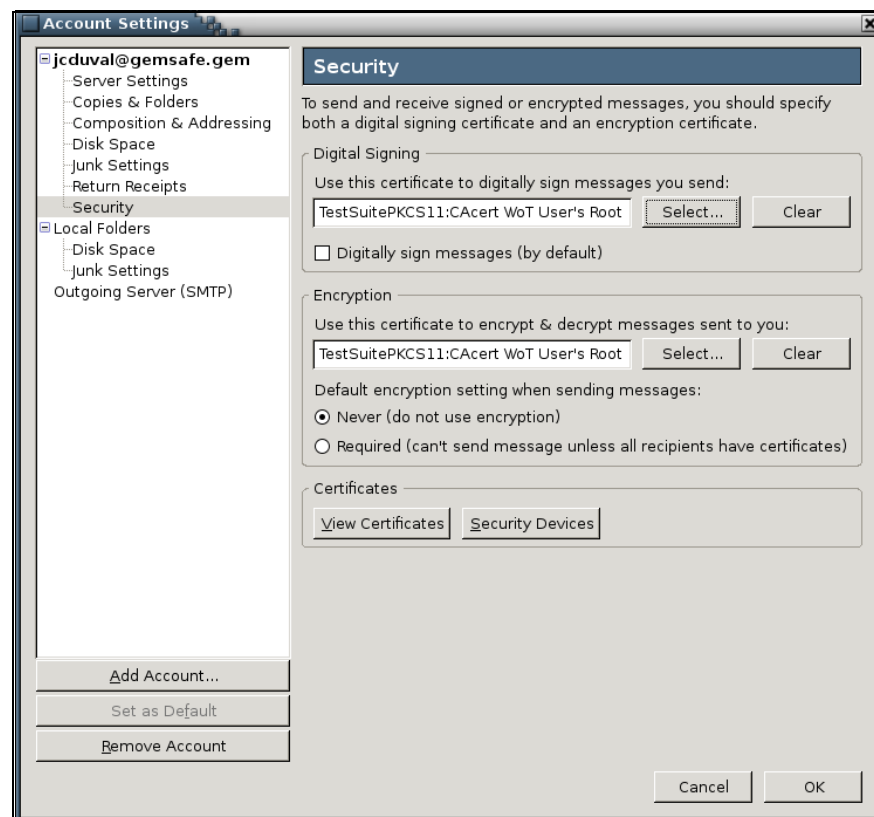
- 8 Click **OK**. The following message appears:

Figure 15 - "Use Same Certificate" Message



- 9 If you want to use the same certificate to encrypt and decrypt messages, click **OK**. This selects the certificate for you in the **Encryption** panel as shown in "Figure 16". Otherwise click **Cancel**.

Figure 16 - Security Account Settings (2)



- 10 If you want all of your e-mails to be digitally signed by default, check the box **Digitally sign messages (by default)**.
- 11 In **Encryption**, if you chose not to use the same certificate as the one used for digital signing, click **Select** and choose the certificate from the list that appears. A message similar to the one in "Figure 15" on page 19 appears, but this time asking if you want to use the Encryption certificate for digital signing. This is just in case you select your encryption certificate before you select your digital signature certificate.
- 12 In **Default encryption setting when sending messages**, choose one of the option buttons **Never** or **Required**.
- 13 Click **OK** to close the **Security Account Settings** window.

Note: If you want to modify the account settings at any point, open the **Account Settings** window from the **Tools** menu by choosing **Account Settings**. This can be done either from the **Compose** window or directly in Thunderbird.

Sending Digitally Signed E-mail with Mozilla Thunderbird

To send a signed e-mail to yourself with Mozilla Thunderbird


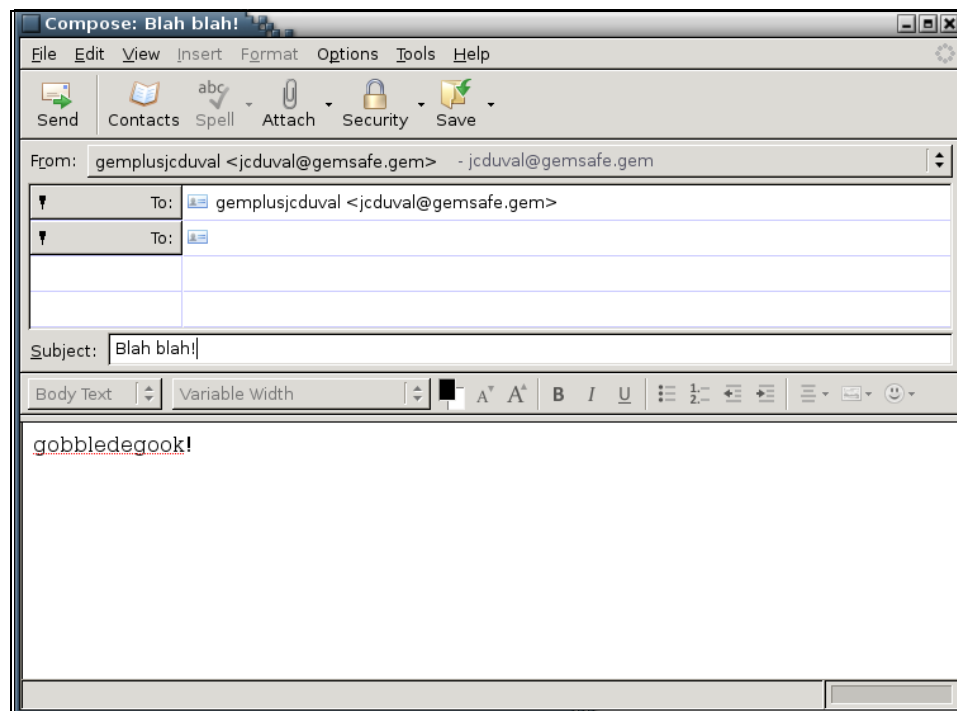
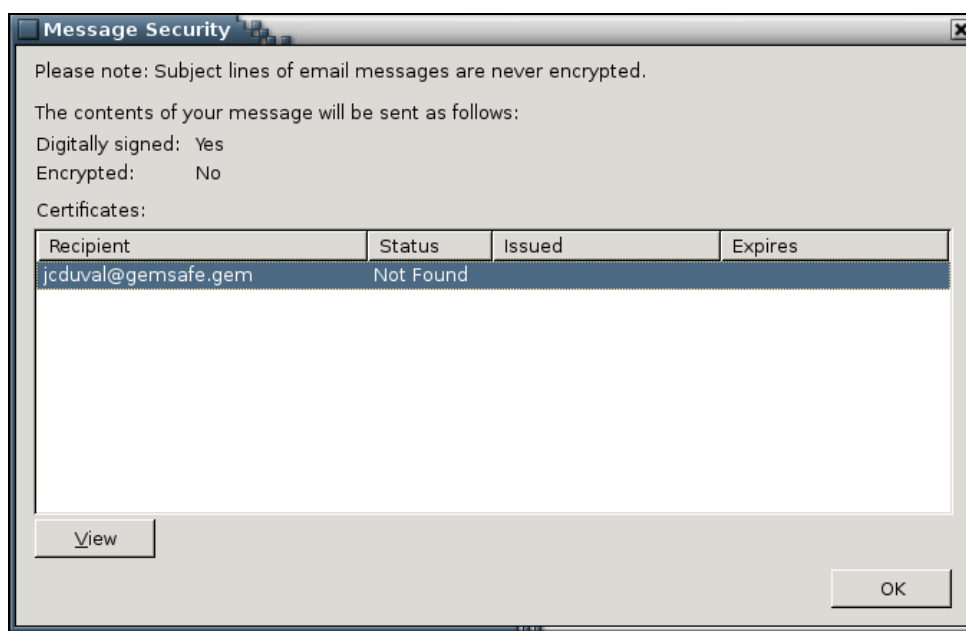
- 1 Make sure your smart card/token is connected.
- 2 Start **Mozilla Thunderbird**.
- 3 Enter your password if you are prompted for it.
- 4 In **Thunderbird**, click the **Write** icon  .
This opens the **Compose** window.
- 5 In the **Compose** window, write a short message *addressed to yourself*.
Be sure to include a subject heading.

Figure 17 - New Msg Composition Window



- 6 From the **Options** menu in the **Compose** window, choose **Security > Digitally Sign this Message** in order to sign the message.

Note: You can check the security settings for your message in the **Compose** window by choosing **View > Message Security Info**. This displays the **Message Security Info** window as shown in “Figure 18” on page 21.


Figure 18 - Message Security Info Window

You can display details about the certificate by clicking **View**.

- 7 Click **OK** to close the **Message Security** window.
- 8 Back in the **Compose** window, click **Send**.

If you are prompted for a master password for your security module, as shown in “Figure 13” on page 18, then enter the User PIN for your smart card/token.

Note: This is the most usual case, but you could equally be asked to authenticate yourself using the PIN Pad reader or by fingerprint authentication (see “Authentication Process” on page 13).

- 9 Open the message you sent yourself from in your inbox.
Notice the  icon showing you that the message has been signed.
You have successfully sent yourself a digitally signed e-mail.
Now that Thunderbird recognizes your public key, you can send signed messages to other people, thus sending them your public key.

Sending Encrypted E-mail with Mozilla Thunderbird

Once you have configured your e-mail account in **Mozilla Thunderbird**, you can retrieve a person’s public key when he or she sends a signed message to you. When you send e-mail to that person, you use his or her public key to encrypt the e-mail. This is done automatically by Thunderbird or Icedove; you just need to specify the recipient(s) of the mail. Since no one except the person who has the private key can decrypt it, the e-mail is secure.

To send an encrypted e-mail:

Follow the same steps as “To send a signed e-mail to yourself with Mozilla Thunderbird” on page 20, except in the **Compose** window, choose **Encrypt this message** from the **Options** menu.

Reading Encrypted E-mail with Mozilla Thunderbird

When you open an encrypted e-mail, the application prompts you for a password.

Enter the User PIN of your smart card/token to decrypt and read the e-mail.

Note: This is the most usual case, but you could equally be asked to authenticate yourself using the PIN Pad reader or by fingerprint authentication (see “Authentication Process” on page 13).

How to View Secure Web Sites

Communicating and conducting business on the Web is quickly becoming the most convenient, effective means of transaction. Therefore, Web sites must be secure to protect the corporation, the individual and the information exchanged.

With your Classic Client smart card/token, you can browse secure Web sites knowing that your private key and digital certificate are safely stored on your smart card/token instead of your hard drive, where they might be susceptible to unauthorized access.

Note: All secure Web site addresses must begin with https://. Browsers display a lock icon at the bottom of the browser window indicating that the site is secure. A closed lock indicates that you are operating in secure mode. You may need to configure your organization's network to allow secure browsing.

When you connect to a secure Web site, your certificate must be specified in your browser so that you can authenticate yourself to the Web server. For example, when you bank online, your bank must be sure that you are the correct person to get account information. Your certificate confirms your identity to the online bank.

The following sections explain how to check that your certificates are correctly registered in your browsers when authenticating with secure web sites using Mozilla Firefox.

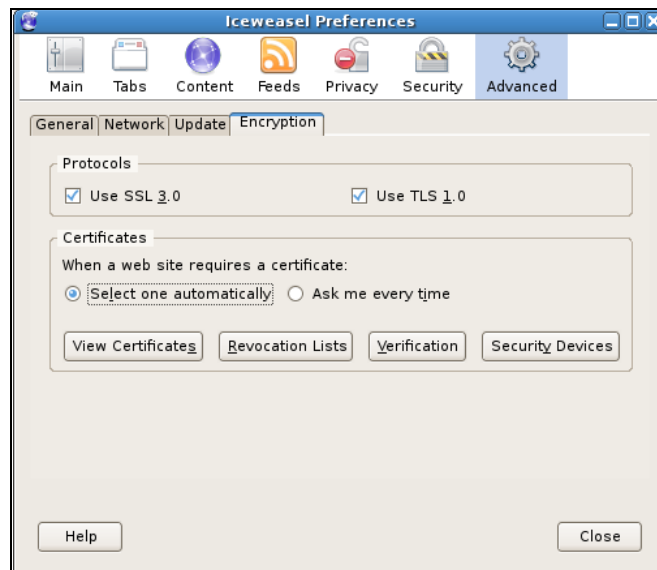
Choosing a Certificate to Authenticate Yourself to Secure Web Sites

To authenticate using the Mozilla Firefox browser, your certificate must be registered in the browser. This section describes how to check that a certificate is registered and also how to tell the browser whether it should select the certificate itself, or ask you.

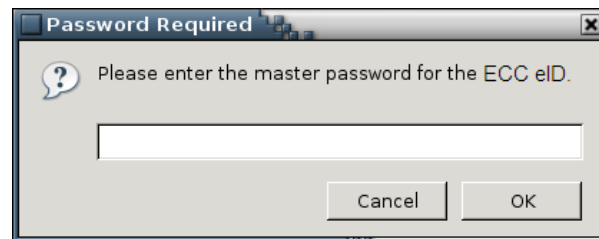
The screenshots in this section were made using Iceweasel, which is the Debian version of Mozilla Firefox. The two applications are identical except for their appearance.

To check certificates registered in Mozilla Firefox:

- 1 Make sure your card/token is connected.
- 2 Open Mozilla Firefox.
- 3 From the **Edit** menu choose **Preferences**.
- 4 Click the **Advanced** icon, then the **Encryption** tab as shown in "Figure 19".

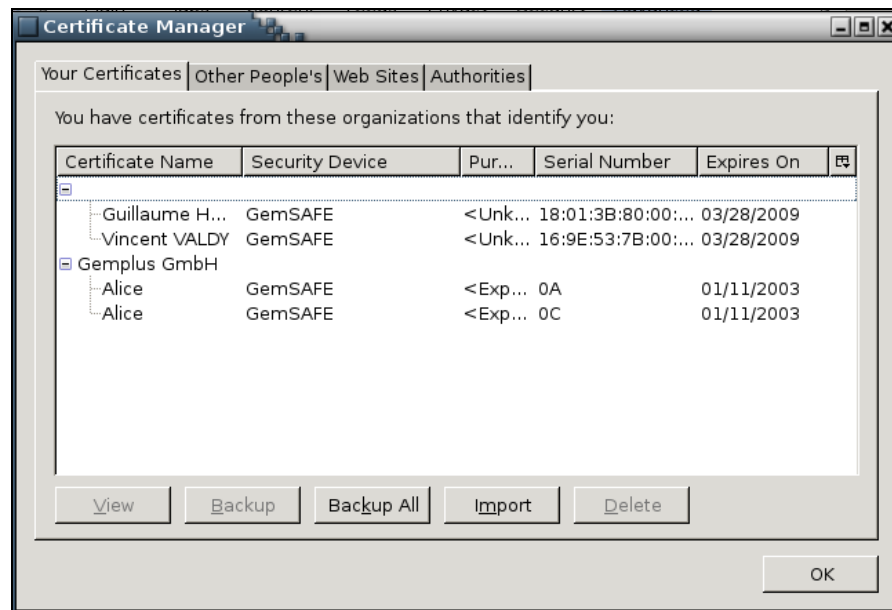
Figure 19 - Mozilla Firefox Options Dialog

- 5 In **Certificates**, choose one of the options for the action to take when a web site requires a certificate:
 - Select one automatically
 - Ask me every time
- 6 To display the certificates that are on your card/token, click **View Certificates**. You will be prompted for a password as shown in “Figure 20”.

Figure 20 - Password Required

Note: This is the most usual case, but you could equally be asked to authenticate yourself using the PIN Pad reader or by fingerprint authentication (see “Authentication Process” on page 13).

- 7 Enter the User PIN for your card/token.
The **Certificate Manager** window appears.

Figure 21 - Certificate Manager Window

- 8 Under **Your Certificates** appears the certificates that are stored on the card/token. To display the properties of a particular certificate, select it and click **View**.

How to Sign PDF Documents

This section explains how to digitally sign PDF documents using Adobe Reader.

There are three stages:

- 1 Configure the Adobe Reader to recognize the Security Module.
- 2 Configure the security settings and specify the certificate to use for signing.
- 3 Sign the PDF document.

Configure Adobe Reader to Recognize the Security Module

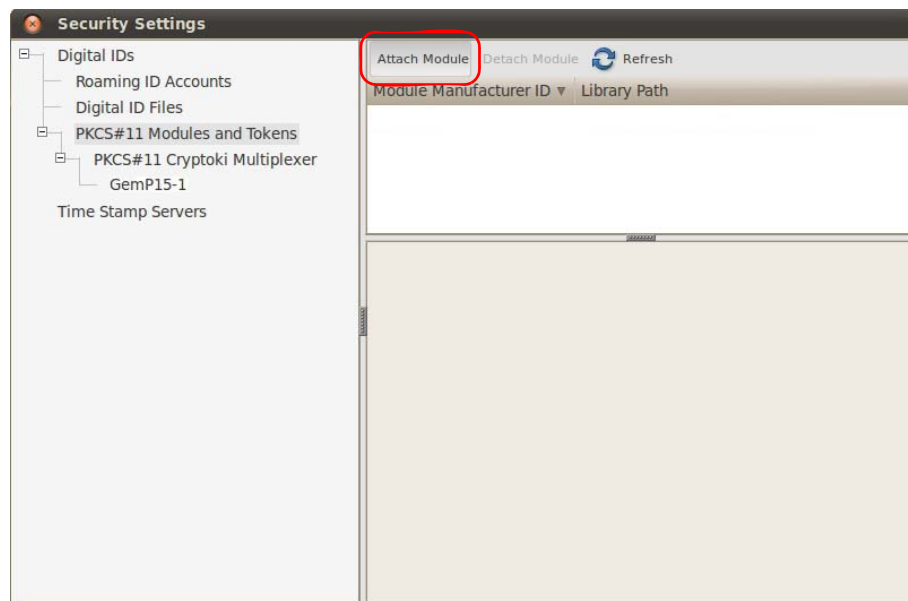
You only need to do this once.

To configure Adobe Reader

- 1 Make sure your smart card/token is connected.
- 2 Start a Terminal window and type this command:

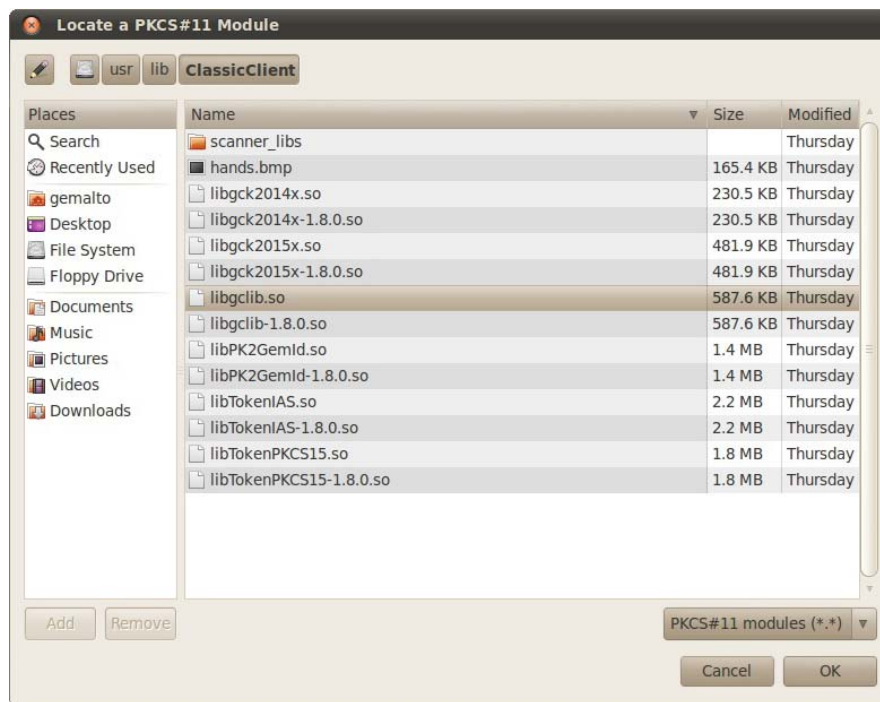
```
mv /usr/lib/xulrunner-1.9.2.16 /usr/lib/xulrunner-bak
```
- 3 Start **Adobe Reader**.
- 4 From the **Document** menu, choose **Security Settings**.
- 5 In the **Security Settings** window, select **PKCS#11 Modules and Tokens**, and then click the **Attach Module** button.

Figure 22 - Security Settings Window



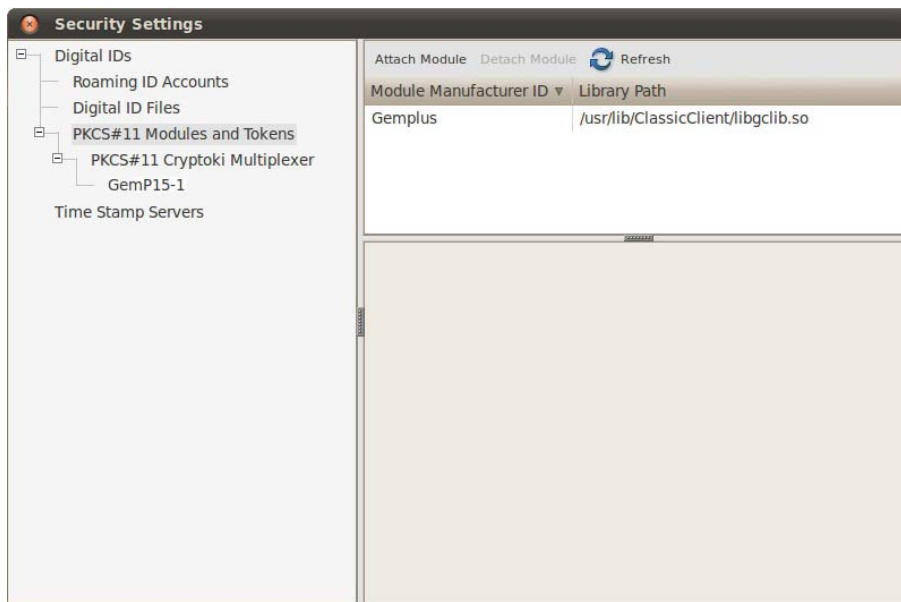
- 6 In the window that appears, select **File System**, browse to **/usr/lib/ClassicClient/libgclib.so**, and then click the **OK** button

Figure 23 - Locate a PKCS#11 Module window



You should see the loaded security module.

Figure 24 - Loaded Security Module

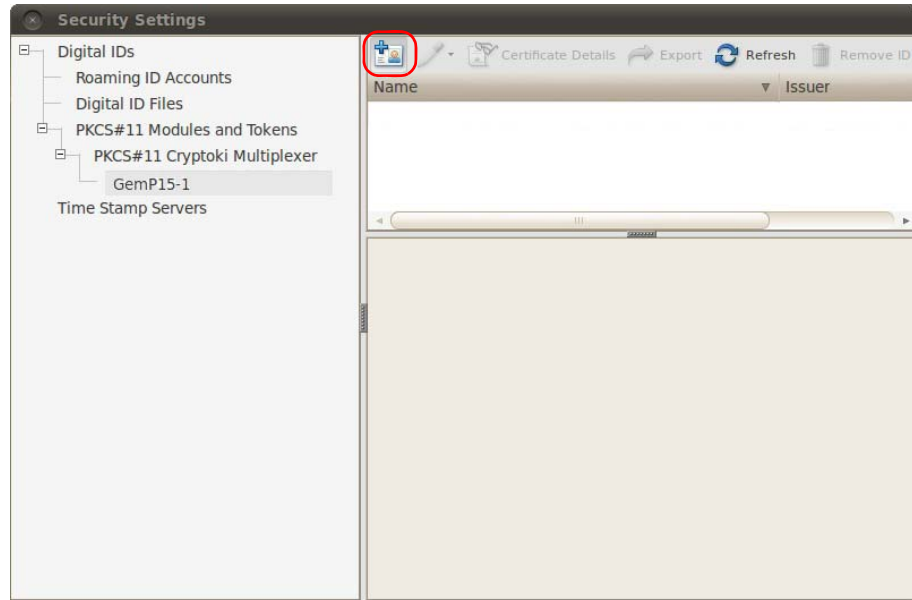


Configuring Settings and Specifying Certificates

Continuing from the previous section:

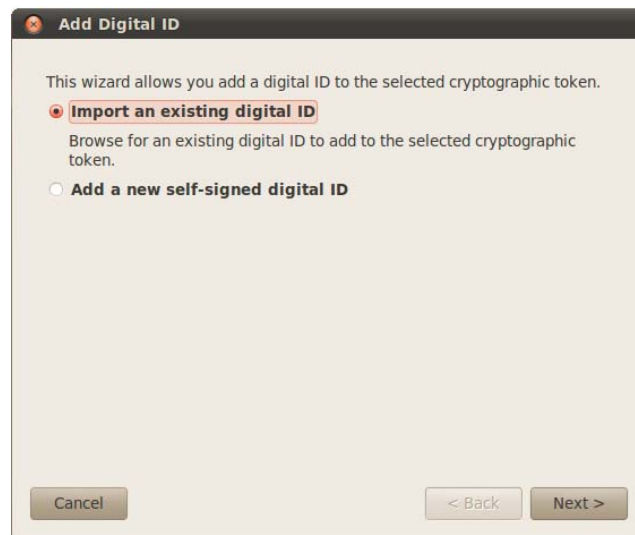
- 1 Select the card label (“GemP15-1” in this example), and then click the **Add ID** icon as shown in “Figure 25”.

Figure 25 - Add ID in Security Settings Window

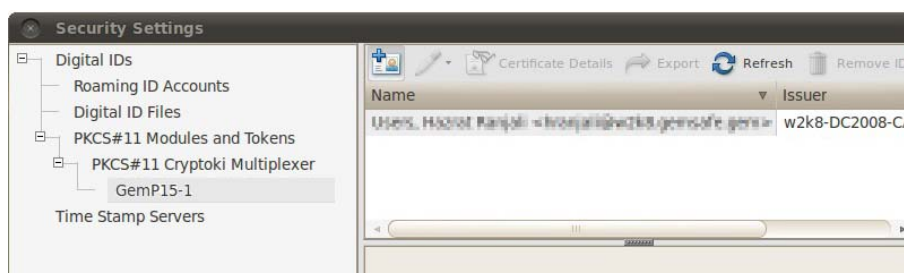


- 2 In the **Add Digital ID** dialog box, select **Import an existing digital ID**, and then click the **Next** button.

Figure 26 - Add Digital ID Dialog Box



- 3 In the dialog box that appears, select the **Cancel** button. You should see the added ID.

Figure 27 - Digital ID Added in Security Settings Window

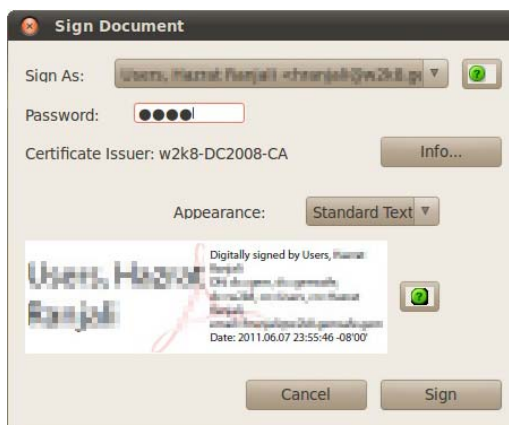
- 4 Close all settings windows.

Signing PDF Documents

After configuring Adobe Reader, you are ready to digitally sign PDF documents.

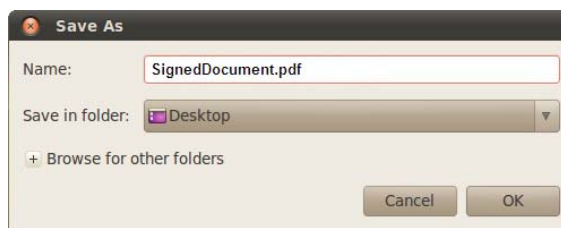
To sign PDF documents

- 1 In Adobe Reader, from the **Document** menu, select **Sign > Sign Document**.
- 2 When prompted for the password in the **Sign Document** dialog box, enter your User PIN and then click the **Sign** button.

Figure 28 - Sign Document Dialog Box

Note: This is the most usual case, but you could equally be asked to authenticate yourself using the PIN Pad reader or by fingerprint authentication (see “Authentication Process” on page 13).

- 3 In the dialog box that appears, enter the filename you want and then click the **OK** button.

Figure 29 - Save As Dialog Box

Security Basics

This chapter introduces you to the IT security standards integral to Classic Client.

Cryptography

Communicating and conducting business electronically is quickly becoming the most convenient, effective means of transaction. An essential condition for the continued growth toward an electronic market is security. The identities of both corporations and individuals must be authentic. The integrity and privacy of information must be guaranteed.

Encryption/decryption enables you to send and receive secure e-mail and documents to protect confidential or private information. You can use the signature function to sign your messages. By signing messages, you can prove to the recipient that you are who you claim to be.

The IT industry uses cryptography to render information secret and known only by authorized entities.

There are two types of cryptography:

- Secret Key Cryptography.
- Public Key Cryptography

Both cryptographic systems use *keys* to digitally sign or encrypt/decrypt data. A key is a value in electronic format used to perform cryptographic functions on electronic data.

The differences between secret key and public key cryptography include:

- Key management.
- Complexity of the key structure.

Key management is central to having a successful crypto system. If keys are not managed in a secure environment, the overall security of the crypto system is at risk. Keys must also be convenient to use.

The complexity of a key length is determined by the degree of mathematical properties applied to the random numbers that comprise the key.

Secret Key Cryptography

Secret key cryptography is the traditional crypto system, which remains in widespread use even today. Secret key cryptography uses a single secret key to digitally sign or encrypt/decrypt electronic data. The most widely used secret key crypto systems are DES and RC2 (also known as symmetric key cryptography).

The sender and receiver must use the same secret key for the session in which secure information is exchanged. The sender uses the secret key to encrypt the message; the receiver uses the same secret key to decrypt the message.

The primary advantage of secret key cryptography is the speed at which data can be encrypted/decrypted.

The primary weakness of secret key cryptography regards key management. Because sender and receiver must share knowledge of the secret key, there must be a transfer of the secret key at some point. Introducing a third party (such as a telephone line or courier) to deliver the secret key to the receiver presents a security risk.

Secret keys are included in the cryptographic functionality of Mozilla e-mail and browser products.

Public Key Cryptography

Public key cryptography was introduced in 1976 and is the most advanced, secure crypto system for digitally signing and encrypting/decrypting electronic data. Public key cryptography refers to a crypto system that uses key pairs. The most popular and widely-used public key crypto system uses the RSA key pair.

A key pair is a matched set of keys used to digitally sign or encrypt/decrypt electronic data. RSA key pairs, like secret keys, are strings of random numbers. However, RSA keys are not only significantly longer than secret keys, they also possess complex mathematical properties.

A single user *owns* an RSA key pair. One key is private, while the other key is public. The private key remains private and accessible only to the owner of the key pair. The public key is made available by the owner to public users. The public key is used to encrypt data. The private key is used to decrypt data.

The strengths of using an RSA key pair is that the need for sender and receiver to share knowledge of the single secret key used in secret key crypto systems is eliminated.

Classic Client takes advantage of the speed the secret key offers and the robust security and convenience of the RSA key pair. When you use Classic Client to send secure e-mail, the actual message data is encrypted using a secret key. The secret key is then encrypted using the public key of the intended recipient. Only the recipient's private key can decrypt the secret key. Only the secret key can decrypt the message data.

Classic Client offers the most advanced digital security at the greatest speed and convenience.

What is a digital certificate?

A digital certificate is an electronic document that serves as your digital passport. Your digital certificate stores your public key and other personal information about you and the certificate.

The most widely accepted standard for digital certificates is defined by *International Telecommunications Union standard ITU-T X.509*. Version three is the most current version of X.509.

The X.509v3 certificate includes the following data:

- Version.
- Serial number.
- Signature algorithm ID.
- Issuer name.
- Expiration Date.
- User name.
- User public key information.
- Issuer unique identifier.
- User unique identifier.
- Extensions.
- Signature on the above fields.

As a convenience to recipients, it is standard practice to attach your digital certificate to every secure e-mail that you send. The recipient uses your public key, included in your digital certificate, to encrypt e-mail addressed to you. If you do not attach your digital certificate to outgoing e-mails, recipients must retrieve your public key from a public directory if they want to reply to you with an encrypted e-mail.

What is a Certificate Authority?

Certificate Authorities (CAs) are trusted third parties that issue digital certificates. CAs vouch for the identity of the individual or enterprise to whom they are issuing a certificate. CAs provide a transfer of trust from CA to the individual or enterprise. When you trust the CA certificate, you can transfer that trust to all certificates published by that CA.

When you obtain your digital certificate, you provide the CA with your public key and any personal information requested by the CA. The CA verifies your personal information and the integrity of your public key. After the verification process, the CA signs your public key, stores appropriate personal information and your public key on the digital certificate, and issues your digital certificate to you.

CAs issue certificates with varying levels of identification requirements. CA policies and the level of identification of the digital certificate determine the method and requirements for proving your identity to the CA. The most simple digital certificate only requires your e-mail address and name. However, some CAs require a driver's license, notarized certificate request form, or any other personal documentation attesting to your identity. Some CAs may even go as far as requiring biometric data such as fingerprints.

The CA public key must be widely available so that users can validate the authenticity of all certificates published by this CA.

What is a digital signature?

A digital signature is a piece of information created using message data and the owner's private key. Digital signatures provide message authentication, non-repudiation of origin, and data integrity.

Digital signatures are created by mathematical, or *hash*, and private signing functions. The one-way hash function produces a message digest, a condensed version of the original message text. The message digest is encrypted using the sender's private key, turning it into a digital signature.

The digital signature can only be decrypted using the public key of the same sender. The recipient of the data decrypts the digital signature and compares the result with a message digest, recalculated from the original message text. If the two are identical, the message was not manipulated, thus is authentic.

What is S/MIME?

Secure/Multipurpose Internet Mail Extensions (S/MIME) is an open protocol standard, that provides encryption and digital signature functionality to Internet e-mail. S/MIME uses public key cryptography standards to define e-mail security services.

S/MIME enables you to encrypt and digitally sign Internet e-mail using Web messaging applications such as Mozilla Thunderbird. S/MIME also enables you to authenticate incoming messages.

S/MIME provides the following security functions:

- **Sender Authentication** to verify the sender's identity. By reading the sender's digital signature, the recipient can see who signed the message and view the certificate for additional details.
- **Message Encryption** to ensure that your messages remain private. Mozilla Thunderbird supports domestic and export-level public key and secret key encryption.
- **Data Integrity** to guard against unauthorized manipulation of messages. S/MIME uses a secure hashing function to detect message tampering.
- **Inter-operability** to work with other S/MIME-compliant software.

What is SSL?

Secure Sockets Layer (SSL), developed by Netscape Communications, is a standard security protocol that provides security and privacy on the Web. The protocol allows client/server applications to communicate securely. SSL uses both public and secret key cryptography.

The SSL protocol is application independent, which enables higher-level protocols such as Hyper Text Transfer Protocol (HTTP) to be layered on top of it transparently. Therefore, the client can negotiate encryption and authentication with the server before data is exchanged by the higher-level application.

The SSL Handshake Protocol process includes two phases:

- **Server Authentication** in which the client requests the server's certificate. In response, the server returns its digital certificate and signature to the client. The server certificate provides the server's public key. The signature proves that the server currently has the private key corresponding to the certificate.
- **Client Authentication** (optional) in which the server requests the client's certificate. In response, the client sends the digital certificate and signature to the server. If the SSL Server requests it, the client is prompted to enter a PIN to visit a secure Web site.

The SSL process is repeated for every secure session you attempt to establish unless you specify a permanent session. The SSL process will not proceed if the Web server's certificate is expired.

Note: In some instances, the SSL Handshake takes place between the Web server and the browser and does not require the client's certificate.

SSL provides the following security functions:

- **Data Encryption** to ensure data security and privacy. Both public key and secret key encryption are used to achieve maximum security. All traffic between an SSL server and SSL client is encrypted using both public key and secret key algorithms. Encryption thwarts the capture and decryption of TCP/IP sessions.
- **Mutual Authentication** to verify the identities of the server and client. Identities are digital certificates. The entity presenting the certificate must digitally sign the data to prove ownership of the certificate. The combination of the certificate and signature authenticates the entity.
- **Data Integrity** to ensure that SSL session data is not manipulated en route. SSL uses hash functions to provide the integrity service.

What is Classic Client?

Classic Client is a smart card-based solution designed to secure e-mail communications and Internet transactions. Classic Client smart cards/tokens support encryption/decryption and signature functions.

Classic Client and a smart card/token provide the following advantages:

- Your private key is never removed from your smart card/token.
- The smart card/token is hardware-based security.
- The PIN code protects key use.
- Classic Client is portable and convenient.

The encryption/decryption function enables you to send and receive secure e-mail to protect confidential or private information. You can use the signature function to sign your messages. By signing messages, you can prove to the recipient that you are who you claim to be.

Classic Client combines the privacy, integrity, and authentication functionality provided by cryptographic algorithms with the simplicity, portability, and convenience of smart cards/tokens. Your private key, digital certificate, and other personal information are securely stored on your Classic Client smart card/token to prevent fraudulent use of your electronic identity.

The latest industry standards such as SSL3 (for Web access) and S/MIME (for e-mail) enable inter-operability of security services between any browser interface and any Web server. However, the security hole in SSL3 and S/MIME is the management of your private key and digital certificate. Without Classic Client, your private key and digital certificate are stored on your hard drive, which makes them susceptible to unauthorized access and fraudulent use. Without Classic Client, your electronic identity is at risk.

Classic Client provides double-barreled security! Classic Client, you get the hardware-based security inherent in smart cards/tokens and software-based encryption security, as well as the added advantage of individual PIN codes. Hardware-based security is a principal security advantage. It is significantly more secure than software-only solutions. Without the possession of your smart card/token and knowledge of your PIN code, no one can use your identity.

Classic Client is your electronic passport to the digital world.

What is a Smart Card/Token?

A smart card is the size of a conventional credit card. But unlike the credit card, which has a magnetic stripe, the smart card has a silicon microprocessor chip to store and process electronic data and applications. The advantage of the smart card is **security**.

Gemalto manufactures various types of smart cards. Contact smart cards use a microprocessor chip to store and process data. They must be inserted into a smart card reader. Contactless smart cards use a microprocessor chip and antenna to store and process data.

Smart cards can also be embedded in tokens such as USB devices, that you can plug directly into a PC.

Smart cards/tokens provide the most sophisticated security available on the market.

What is the Classic Client Smart Card/Token?

Your Classic Client smart card/token stores your private key and digital certificate. In the past, your only option was to store your private key on your local hard drive, rendering it susceptible to theft and fraudulent use. With Classic Client, your electronic identity is secure. You must have both the smart card/token and PIN code to use the smart card/token.

The Classic Client smart card/token is tamper resistant. The structure and operating system of the smart card/token make it practically impossible to penetrate, probe, or pilfer smart card/token data.

Perhaps the most convenient aspect of the Classic Client smart card/token is portability. With Classic Client, you can carry your electronic passport with you at all times and use it on any Classic Client–equipped computer in the world.

The Classic Client smart card/token has a robust and flexible design. These features offer greater freedom and enhanced security.

On-board Key Generation

The Classic Client smart card/token offers on-board key generation. With this feature, every time you enroll a new certificate on your smart card/token, a new key pair is generated on your smart card/token. In other words, you are not limited to using the same key pair for every certificate that you enroll.

One significant advantage of onboard key generation is the ability to monitor and control the life span of your RSA key pairs and that the generated key pair is unique.

Increased Certificate Storage

You can store up to six key pairs and multiple digital certificates on your Classic Client smart card/token, depending upon the size of your certificates and space available on your smart card/token. This feature provides the convenience of using up to eight digital certificates for whatever purposes you want; for example, you can use certificates with varying degrees of encryption (from 1024-bit to 2048-bit RSA key pairs) to communicate securely with contacts in various parts of the world.

Another reason for obtaining more than one digital certificate is the level of certification that the Certificate Authority (CA) requires. You may want to obtain and use a digital certificate from a CA that requires stringent identity certification if you are using the certificate for sensitive business communications or financial transactions. However, if you want to encrypt/sign data for personal communications, you may decide that a certificate from a CA that requires minimal identity certification meets your needs.

The costs of obtaining a digital certificate from a CA are somewhat based on the degree of identity certification the CA requires.

Abbreviations

CA	Certificate Authority
ID	Identification
IMAP	Internet Message Access Protocol
OS	Operating System
PIN	Personal Identification Number
PKCS	Public Key Cryptography Standard
PKCS#11	Public Key Cryptography Standard #11. For further information about this and other PKCS standards, refer to the RSA Laboratories web sit at http://www.rsa.com/rsalabs/
POP	Post Office Protocol
RHEL	Red Hat Enterprise Linux
RSA	Rivest, Shamir, Adleman (inventors of public key cryptography standards)
S/MIME	Secure/Multipurpose Internet Mail Extensions
SSL	Secure Sockets Layer A protocol, v.3.0.v, for securing TCP/IP sessions

Glossary

Algorithm	A mathematical formula used to perform computations that can be used for security purposes.
Certificate	A certificate provides identification for secure transactions. It consists of a public key and other data, all of which have been digitally signed by a CA. It is a condition of access to secure e-mail or to secure Web sites.
Certificate Authority	An entity with the authority and methods to certify the identity of one or more parties in an exchange (an essential function in public key crypto systems).
Cryptography	The science of transforming confidential information to make it unreadable to unauthorized parties.
Digital Signature	A data string produced using a Public Key Crypto system to prove the identity of the sender and the integrity of the message.
Encryption	A cryptographic procedure whereby a legible message is encrypted and made illegible to all but the holder of the appropriate cryptographic key.
Key	A value that is used with a cryptographic algorithm to encrypt, decrypt, or sign data. Secret key crypto systems use only one secret key. Public key crypto systems use a public key to encrypt data and a private key to decrypt data.
Key Length	The number of bits forming a key. The longer the key, the more secure the encryption. Government regulations limit the length of cryptographic keys.
Public Key Crypto system	A cryptographic system that uses two different keys (public and private) for encrypting data. The most well-known public key algorithm is RSA.
SSL	Secure Sockets Layer: A Security protocol used between servers and browsers for secure Web sessions.
SSL Handshake	The SSL handshake, which takes place each time you start a secure Web session, identifies the server. This is automatically performed by your browser.
S/MIME	A Standard offline message format for use in secure e-mail applications.
Token	In a security context, a token is a hardware object like a smart card, but it could also be a pluggable software module designed to interact with a specific hardware module, such as a smart card. Token-based authentication provides enhanced security because success depends on a physical identifier (the smart card) and a personal identification number (PIN).