GemSAFE™

For Netscape Communicator

User Guide

Version 1.1

GEMPLUS

21 December 1998

IMPORTANT NOTICE

Warranty

Gemplus warrants this GemSAFE™ product to be physically free of any defects in manufacturing and workmanship. No other warranties may be implied nor are enforceable according to international law and any authority.

Gemplus makes nor representation or warranties, either expressed or implied, by or with respect to this GemSAFE™ product, and shall not be liable for any implied warranties of merchantability, of non-infringement of third parties' rights, of error free and of fitness for a particular purpose or for any indirect, special or consequential damages.

Limitation of Liability

Except to the warranty provisions above, in no event shall Gemplus and its affiliates be liable for incidental, consequential or special damages, including loss of profit, loss of data, corrupted or misdirected data, whether based on contract, tort or any other legal theory. Gemplus bears no responsibility whatsoever with respect to the use, sale or other disposition of any product incorporating this GemSAFE™ product, in particular Gemplus makes no warranty as to the physical or logical security and protection of this GemSAFE™ product.

Copyrights

Any copyright, patent right, trademark, trade secret, mask work and any other intellectual and/or industrial property right in this GemSAFE™ product, its related documentation and this publication are and will remain the property of Gemplus. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by means electronic, mechanical, photocopying, recording or otherwise without the prior written consent of Gemplus. No patent liability is assumed with respect to the uses of any information contained herein.

Trademarks

"GemSAFE" is a trademark of Gemplus.

All written instructions or guidelines contained herein on the use of trademarks, copyrights or of other ownership rights of Gemplus shall be duly followed. All use of Gemplus trademarks and logos shall inure to the benefit of Gemplus.

CONTENTS

OVERVIEW

GemSAFE™ is a smart card-based solution which is primarily designed to secure electronic mail (e-mail) communication and web sessions on the Internet.

This solution combines the privacy, tamper-detection (integrity) and proof of origin (authentication) functionality provided by cryptographic algorithms with the simplicity, portability and convenience of smart cards.

What is GemSAFE Used For?

The GemSAFE™ smart card securely stores your personal secret information and thus prevents anyone from usurping your identity. Indeed, your password must be presented before your private keys can be used. The security improvement that the GemSAFE™ offers as compared to software-only solutions is that your keys are stored in your smart card and never leave it.

Internet services range from e-commerce, home-banking, to e-citizen, home-training and e-mail. Perspectives are huge, but the lack of proper security is impeding their implementation. Gemplus is now breaking down this major barrier with GemSAFE™, a reliable plug-and-play smartcard solution.

The latest standards such as SSL3 (for web access) or S/MIME (for e-mail) enable inter-operability (that is, compatibility) of security services between any browser interface and any web server. For example, although S/MIME is designed to exchange secure e-mails, you can also use the same mail application to send regular (unsecured) E-mails.

But the security hole in these protocols is the management of your personal keys and certificate. Indeed, these can easily be tampered with if you save them on your PC.

Travelling with your electronic identity in pocket, you can securely access on-line services with your personal smart card, protected by a PIN code, from any machine in the world. In addition, your card also performs cryptographic algorithms, so that your private keys never leave the card. Just plug your smart card into any reader connected to any internet terminal equipped with the GemSAFE™ software. With the GemSAFE™ solution, you no longer are dependent on your own computer.

Connecting to the GemSAFE Web Site

Additional information is available on the GemSAFE™ web site; its address is:
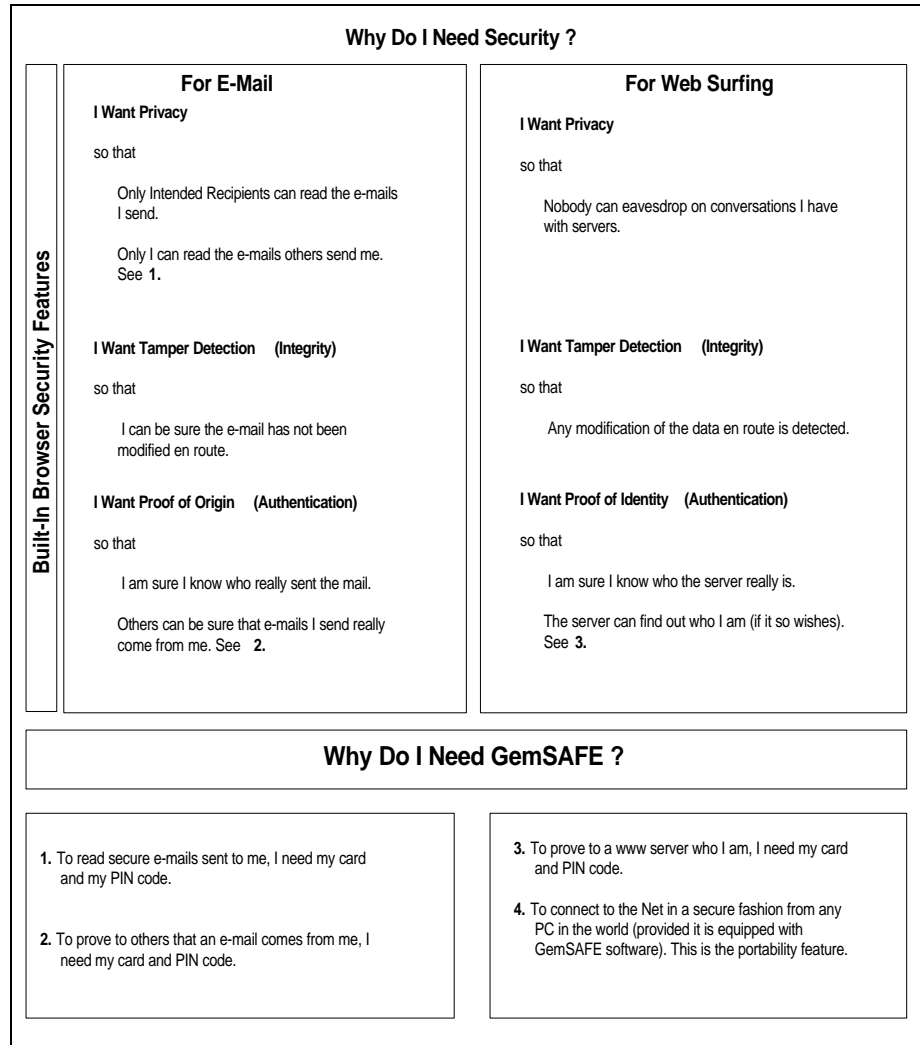
http://www.gemplus.com/GemSAFE

The Need for GemSAFE

Figure 1 illustrates in detail the way GemSAFE™ meets your electronic mail and internet surfing security needs, as it provides added privacy, tamper-detection and authentication.

With this solution, nobody can impersonate you, or read encrypted e-mails meant for you without:

- Having your GemSAFE™ card, and,
- Knowing your PIN code.

Why Do I Need Security ?

Built-In Browser Security Features

For E-Mail

I Want Privacy

so that

Only Intended Recipients can read the e-mails I send.

Only I can read the e-mails others send me. See **1.**

I Want Tamper Detection   (Integrity)

so that

I can be sure the e-mail has not been modified en route.

I Want Proof of Origin   (Authentication)

so that

I am sure I know who really sent the mail.

Others can be sure that e-mails I send really come from me. See **2.**

For Web Surfing

I Want Privacy

so that

Nobody can eavesdrop on conversations I have with servers.

I Want Tamper Detection   (Integrity)

so that

Any modification of the data en route is detected.

I Want Proof of Identity   (Authentication)

so that

I am sure I know who the server really is.

The server can find out who I am (if it so wishes). See **3.**

Why Do I Need GemSAFE ?

1. To read secure e-mails sent to me, I need my card and my PIN code.

2. To prove to others that an e-mail comes from me, I need my card and PIN code.

3. To prove to a www server who I am, I need my card and PIN code.

4. To connect to the Net in a secure fashion from any PC in the world (provided it is equipped with GemSAFE software). This is the portability feature.

Figure 1 - The Need for Security

The improvement provided by the use of the GemSAFE™ card (as opposed to software-only solutions) consists of:

- The fact the keys never leave your smart card
- PIN code protection
- GemSAFE™ portability: you are no longer restricted to a single PC. You can use your GemSAFE™ card from any PC connected to the Net anywhere in the world, provided the GemSAFE™ software is installed on it.

CRYPTOGRAPHY BASICS AND PUBLIC KEY ALGORITHMS

Note: *If you are familiar with cryptographic concepts and with public-key cryptographic systems in particular, you may wish to skip this chapter.*

Cryptography

In the GemSAFE™ secure e-mail and web context, cryptography consists of applying mathematical transformations to transmitted data in order to satisfy one or more of the three following objectives:

1. Privacy: the transmitted data can not be read by a third party

2. Tamper detection (also known as integrity): the recipient can be sure that the data was not modified en route

3. Proof of origin (also known as authentication): the recipient can be sure who really sent the data

In public-key systems, privacy is achieved by encryption; tamper detection and proof of origin are obtained by means of digital signatures.

Encryption

Encryption implies data is scrambled (ciphered) by the sender using an encryption key, before being sent to the recipient. When it arrives, the recipient uses a decryption key to unscramble (decipher) the data. Scrambling and unscrambling are performed using well known mathematical algorithms (with GemSAFE™, the RSA algorithm is used) and the encryption key is a number that feeds into these algorithms.

Public-Key Cryptography

In public-key (or asymmetric) cryptographic systems, the decryption and encryption keys are not the same. The sensitive element is the decryption key, since the person who has a copy of this key can decrypt the data. Normally, only the recipient has a copy of this decryption key which must be safely kept (for example stored in the GemSAFE™ smart card). However, the recipient freely distributes a copy the encryption key to anybody who is likely to need to send this person something securely.

The mathematical properties of these keys are such that anything encrypted with the encryption key (known as the public key because everybody has a copy) can only be decrypted with the corresponding decryption key (known as the private key because only the recipient has a copy). This means that anybody can send the recipient scrambled data, but only the recipient can unscramble it. The public and private keys together constitute what is known as a key-pair.

Who does the Key-Pair Belong To?

How does a sender know that a particular key-pair corresponds to you? You might have sent the public key by e-mail, or the sender might have retrieved it from a server, but this would not prevent a spy from changing it en route and substituting his/her own public key instead. The solution to this problem is digital certification (see the section entitled *Digital Certificates*).

Digital Envelopes

The mathematical algorithms involved in public key systems are slow. Thus, if there are multiple recipients (for example, if you send all your colleagues an encrypted e-mail) or if the data is large (several megabytes, for instance), the encryption of several megabytes will probably require several minutes and will have to be repeated for each recipient (since each has a different public-key). A solution known as digital envelopes is used to resolve this problem. Because symmetric or secret-key algorithms (where the decryption and encryption keys are the same, examples include RC2, RC4, DES and DES3) are much quicker, the following steps are carried out:

The sender:

1. Generates a random symmetric key (which is only a few bytes long)

2. Encrypts the bulk data with the random symmetric key (fast)

3. For each recipient, encrypts the random symmetric key with the recipient's public key

4. Transmits the output of steps 2 and 3 to all the recipients

The recipients:

1. Decrypt the random symmetric key using their private key

2. Use their random symmetric key to decrypt the bulk data (fast)

As a result, the bulk of the data is encrypted only once with a very fast algorithm.

Digital Signatures

A digital signature provides proof of origin and tamper detection. It consists of sending additional information (known as a signature) along with the original data which proves to the recipient that the received data is word for word identical to the data the sender intended to send.

Digital signing of data is completely independent from data encryption. Data can be both signed and encrypted, signed only, encrypted only and, of course, neither signed nor encrypted.

Since each person involved already has a private key (which is kept secret by its owner) and a corresponding public key (which everybody knows) for encryption purposes, a good system might consist of re-using these keys. The signature is calculated by the sender and sent along with the data to the recipient. Its value is a mathematical function of the sender's private key and the data to be sent. The construction of the algorithm is such that it is not possible to calculate this value without knowing the private key.

The recipient can verify that the data received corresponds to the data that was signed by the sender using another mathematical algorithm which relies upon the sender's public key, the signature, and the data received.

Digital Certificates

When the same key-pair is used for encryption and signature, this key-pair corresponds to a sort of on-line identity. You can use it to sign data (e-mails, expense claims, random challenges sent by web servers, etc.) and decrypt data that is meant only for you (incoming e-mails, etc.). The GemSAFE™ solution means that this identity (the private key) is securely stored in a smart card and it never comes out. Any calculations that are performed using this key are done by the card itself.

The system relies on the fact everybody knows that a particular key-pair is linked to you. This is the purpose of digital certificates. A key pair without a corresponding digital certificate is effectively useless.

The Role of the Certificate Authority

By issuing a certificate, the Certification Authority (CA) basically states that "Public key 1234… corresponds to a private key that only Mr. Smith or Company XYZ has access to ". Anybody who trusts the CA can, for example, encrypt an e-mail for Mr. Smith's eyes only, or verify a digital signature created by Company XYZ.

This binding of a real-world identity (Mr. Smith in this example) to a digital identity (Mr. Smith's key pair) is performed using a digital signature. The CA has its own key-pair which is used to sign the concatenation of Mr Smith's public key and the name "Mr. Smith" (along with a host of other useful things such as the certificate's validity date, etc.).

Certification Authorities usually charge a fee for this binding task. Indeed, depending on the company's policy, the CA may pay a visit to the person it is vouching for to verify it actually is who it claims to be, or it may need a letter from this person's employer certifying he/she works there, etc. The CA may also offer other value-added services, such a public directory of the certificates that it has issued.

To determine whether you should trust a particular CA, you first need to look at the CA's policy statement to check that it performs a type of check that you find appropriate before issuing its certificate (what guarantees does the CA provide? what is the legal position?).

Secondly, you need to recover a copy of the CA's public key so you can verify the CA's digital signatures. It is convenient to recover this public key within a certificate (since the certificate also provides validity dates and other relevant elements).

This certificate may be signed by the CA itself or by yet another CA whom you already trust. In the former case, the certificate cannot be independently verified and its integrity must be validated by other means (for example, using the certificate's independently transmitted 'fingerprint').

GEMSAFE SECURITY FEATURES

SSL and S/MIME

The GemSAFE™ solution complements two key security standards:

- The SSL/TLS (Secure Socket Layer/Transport Layer Security) is a protocol between the server and the browser, which operates over the Internet.

- The S/MIME (Secure Multipart Internet Mail Encoding) is a message format designed to secure e-mail messages.

Note: *TLS is the latest standardized version by the Internet Engineering Task Force (IETF) of SSL.*

The SSL Protocol

SSL is an on-line protocol which may provide privacy over the internet as it allows client/server applications to communicate in a way that cannot be eavesdropped.

SSL offers the following basic features: message privacy, message integrity and mutual authentication.

Message Privacy

Message privacy is achieved through encryption. All traffic between an SSL server and an SSL client can be encrypted using a key and an encryption algorithm negotiated during the SSL handshake (see *The SSL Handshake*).

Message Integrity

The message integrity service ensures that SSL session traffic is not modified on the way to its final destination. SSL uses the combination of a shared secret and special mathematical functions (called hash functions) to provide the message integrity service.

Mutual Authentication

Mutual authentication is the process whereby the server convinces the client of its identity and the user convinces the server of its identity. These identities are coded in the form of public-key certificates (X509), and these certificates are exchanged during the SSL handshake.

To demonstrate that the entity presenting the certificate is the legitimate certificate owner (i.e., that this entity has access to the private key which corresponds to the public key in the certificate) rather than a mere impostor, SSL requires that the certificate presenter must digitally sign data exchanged during the handshake (see *Digital Signatures*).

The entities sign protocol data (which includes their certificates) to prove they are the legitimate owner of the certificate. This prevents someone from masquerading as you by presenting your certificate. The certificate itself does not authenticate; the combination of both the certificate and the proof that you have access to the private key does, however (see *Digital Certificates*).

Note: *Server authentication is systematically performed by the browser with the SSL protocol, whereas client authentication may or may not be systematically required by the server.*

S/MIME

S/MIME is an off-line message format standard implemented for use with the Netscape Messenger mail application, which is designed to encrypt and digitally sign electronic mail.

S/MIME offers users the following basic features:

- Encryption for message privacy
- Sender authentication with digital signatures
- Tamper detection
- Compatibility with any other S/MIME-compliant software

Private Messaging

S/MIME's encryption helps ensure that your messages remain private. Netscape Messenger software supports domestic and export-level public key and symmetric key encryption.

Sender Authentication and Tamper Detection

S/MIME authenticates the message sender by reading the sender's digital signature (the recipient can see who signed the message and view the certificate for additional detail).

Compatibility

Because S/MIME is an open standard, the mail software client can operate with other S/MIME-compliant clients (for example, if you are operating with Netscape Messenger, you can correspond with someone equipped with Microsoft Outlook Express (which is S/MIME-compliant)).

CERTIFICATE MANAGEMENT

If you do not already have a preferred Certificate Authority (CA) to issue your certificate, you can find a list of CAs on the GemSAFE™ site (**http://www.gemplus.com/GemSAFE**). Some of these CAs offer free certificates for testing and demonstration purposes.

Obtaining Your Own Certificate

1. Contact a CA.

2. Fill out the required information. This data varies with each CA, but it usually includes your name, e-mail address and your key length.
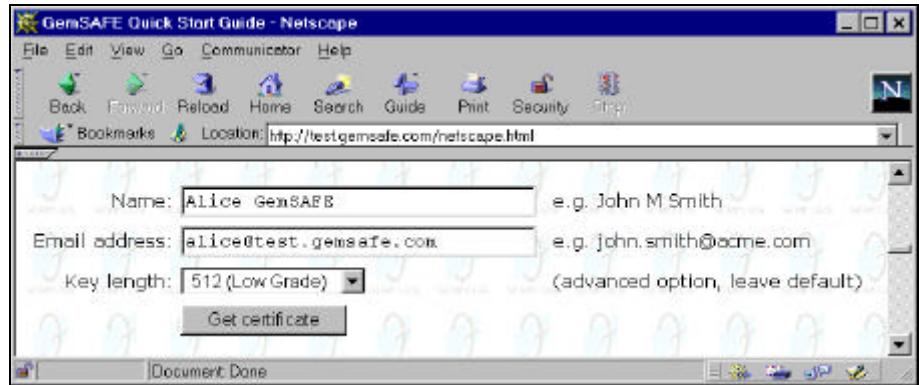
Figure 4 - Getting a Certificate

Note: *As a rule, the key length should be left at the default setting since this normally is the maximum authorized length for your country.*
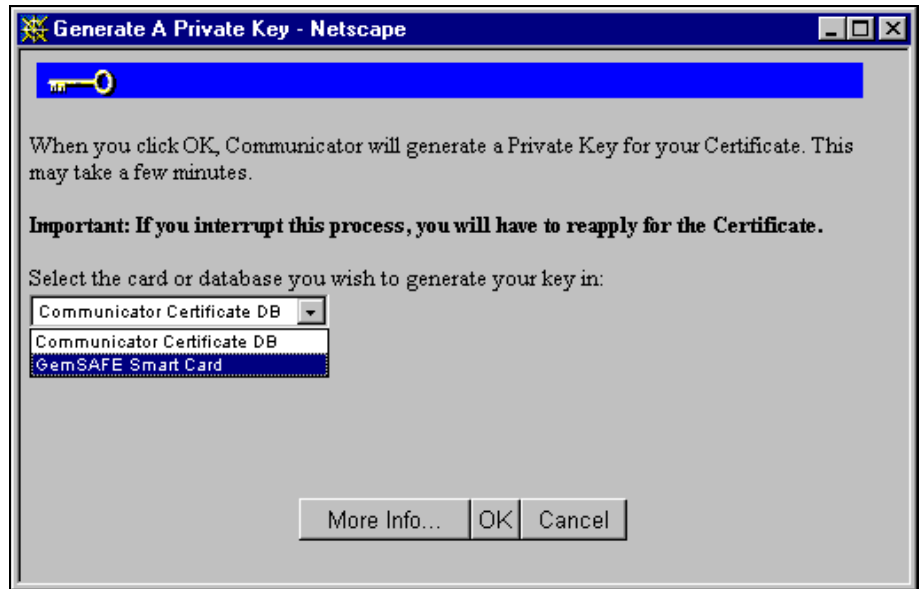
Figure 5 - Generating a Private Key

Note: *If you are using your GemSAFE™, a list box is displayed. Make certain this indicates that you have selected **GemSAFE Smart Card** (do not select **Communicator Certificate DB**).
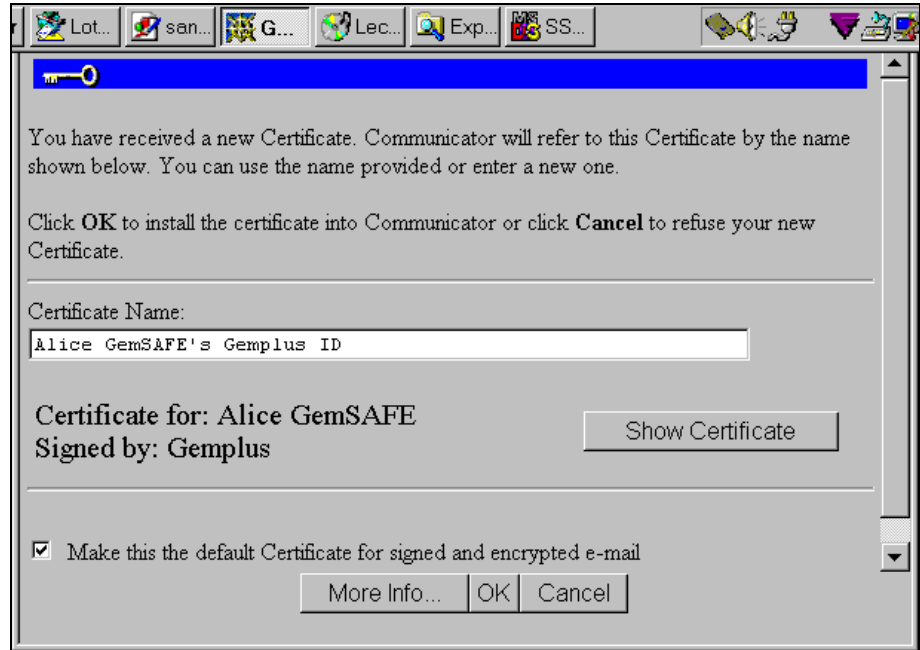Once the certificate is downloaded, it can no longer be exported.*

Figure 6 - The New Certificate

Figure 7 - Saving the Certificate

Warning: *Do not save the downloaded certificate on your disk as suggested by Netscape.*

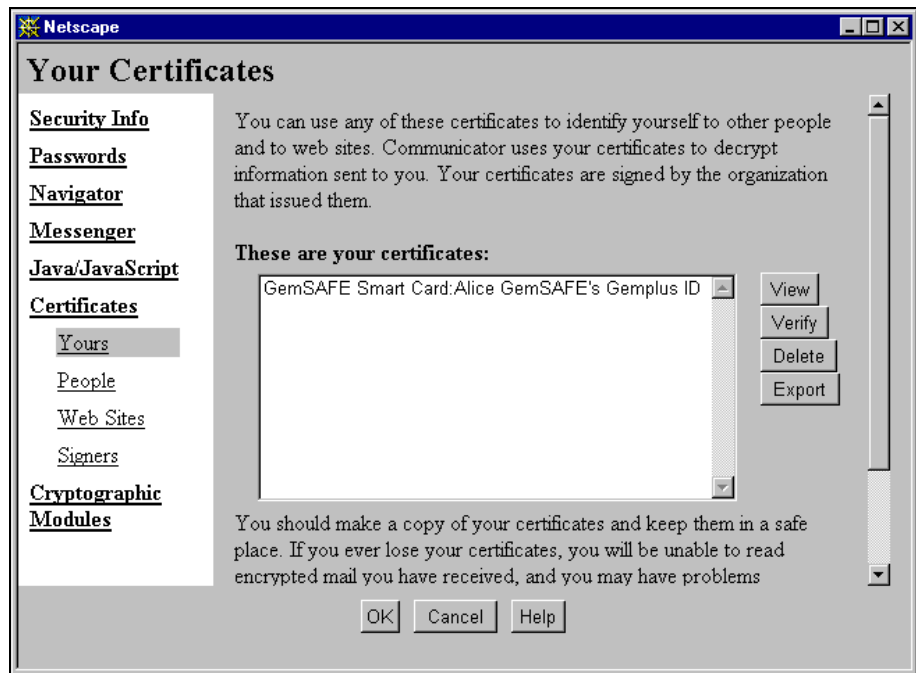If you need to change the certificate in your card, see *Deleting a Certificate*.

Figure 8 - List of Your Certificates

Once you have received your certificate, click **Security**, **Certificate**, **Yours** to view it. All your certificates are then displayed.

Note: *You can only store one certificate in your card at a time. All other certificates are located in the software.*

Who Do You Trust?

When you first try to access a secure web site or to receive a secure e-mail, you may receive a message stating that the CA which signed the server's certificate (or another user's certificate) is unknown. This simply means that your browser does not have a certificate for this CA and that you need to add it to your list if you want to proceed with a secure web session.

Note: *Your browser already has a number of certificate authorities it accepts by default. To see this list of CAs click **Security**, then click **Signers**.*
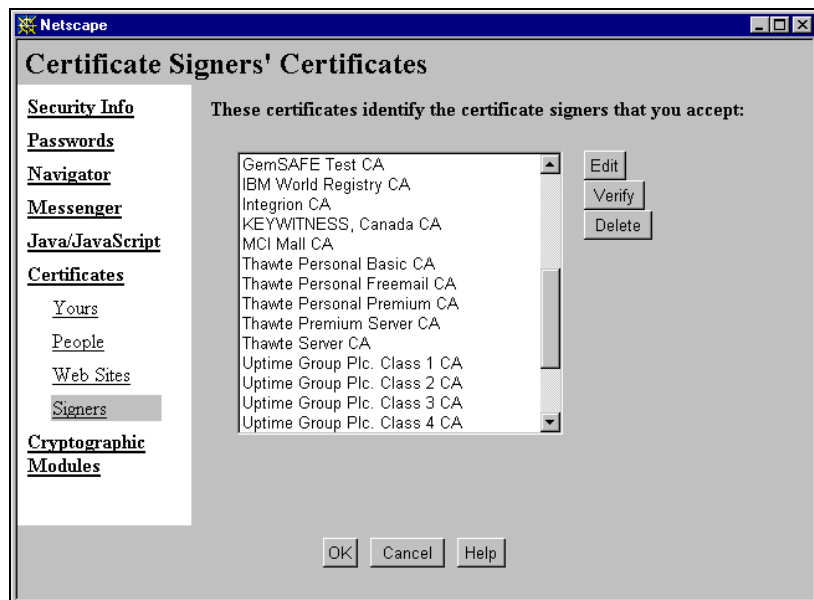

Figure 9 - Certificate Authorities Accepted by Your Browser

Figure 10 - Viewing the Certificate

Identification   To verify that the CA's certificate is valid, you need the CA's public key beforehand (it is located in the CA's certificate). It may be obtained in various ways (including an e-mail from the CA, the CA's web site, a directory...).
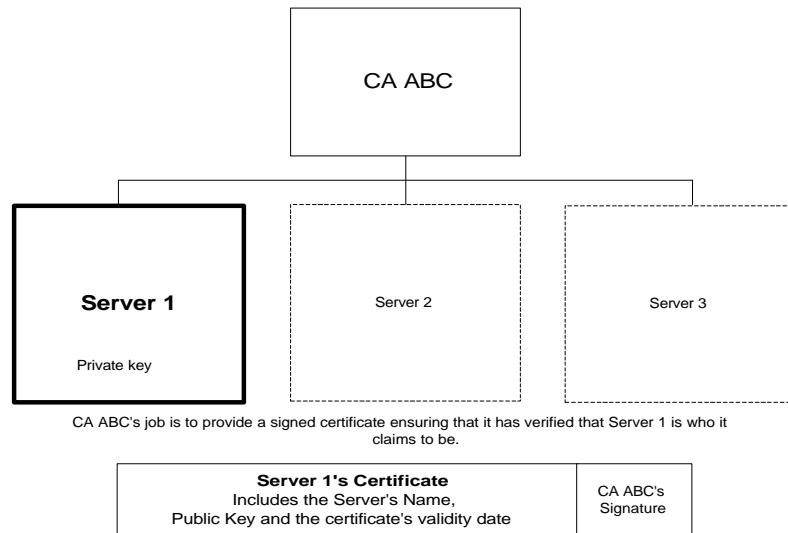

Figure 11 - The Server, the CA and the Signed Certificate

When surfing, you may choose to access a web site without effectively verifying its certificate. In this case, you may select access for this session only, or permanent access. Warning messages are displayed to remind you of the risks of connecting to a site without verifying its certificate first.

Adding a CA

The process of adding a CA to your browser's list consists of getting the CA's certificate over the internet and verifying its integrity by checking that its fingerprint (a digest of the certificate) matches the fingerprint sent to you by independent means.
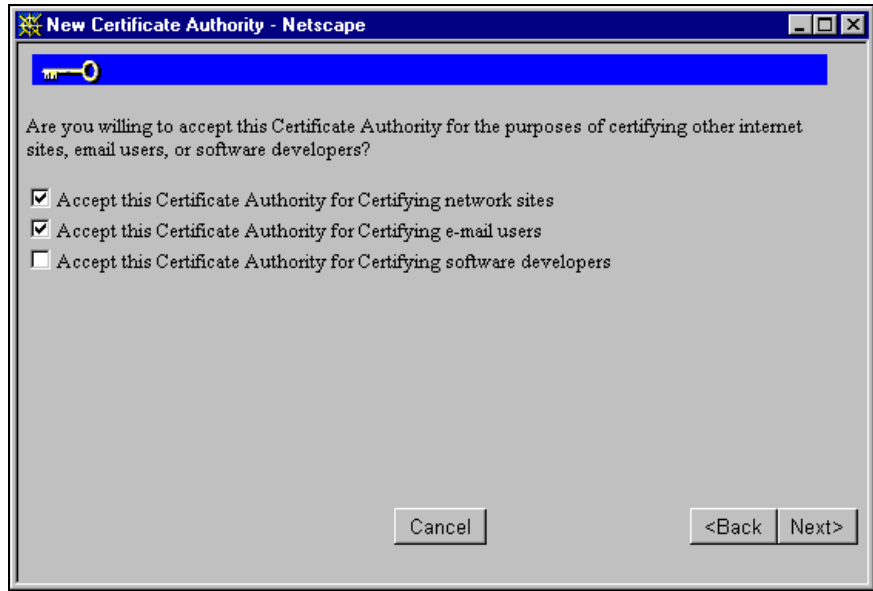

Figure 12 - Adding a Certificate Authority

Other People's Certificate List

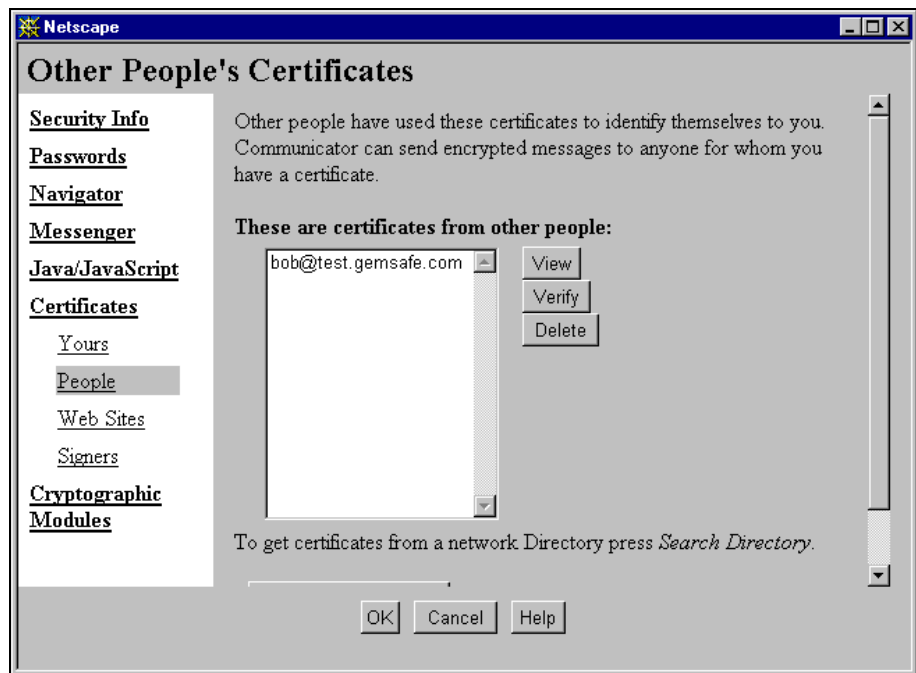You may decide not to take such a risk and opt to verify the server's certificate.

Figure 13- Other People's Certificates Your Browser Knows

To send someone a secure mail, you will need to add this person's certificate to your "**Other People**" certificates list. In order to do this, you must first receive a signed mail from the intended recipient, or look up this recipient's certificate in a public directory.

**Deleting a
Certificate**

A GemSAFE™ card is only designed to store one certificate at a time. If you have a certificate on your card and wish to obtain a new one, you have to delete the initial one first. To do so, click **Security**, **Certificates**, **Yours** and **Delete**.

ACCESSING SECURE SITES

The GemSAFE™ web site contains a list of secure servers that can be used for testing purposes.

To access the GemSAFE™ site, type the following location from your Netscape browser:

http://www.gemplus.com/GemSAFE

Identifying Secure Web Sites

Secure sessions rely on the SSL protocol. All secure web sites are accessed using the **https://** prefix (that is, the address must start with this prefix if you want to communicate with a secure site).

If your browser has a certificate on file for the CA corresponding to the secure site you wish to access (and if your browser has a certificate to present to confirm who you are, in the event client authentication is requested by the server), then you may proceed with your secure session.
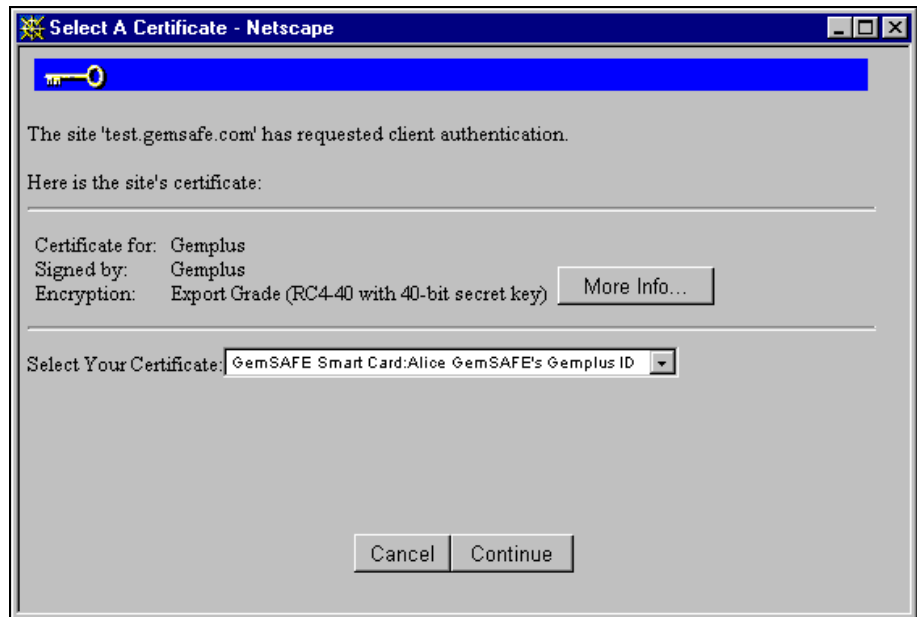

Figure 14 - Selecting a Certificate

You can get your browser to automatically select your certificate by clicking:

- **Security**
- **Navigator**
- **Certificate**

Figure 15 - Sample Secure Web Site

Notice the lock in the lower-hand corner of Figure 15. It is in closed position which means you are operating in secure mode. Figure 16 illustrates the lock in unsecure mode. When the lock is open, you are not using GemSAFE™ and are not operating in secure mode!

Figure 16 - Unsecure Operating Mode

Viewing the Server Page Information

Click:

- **View**

- **Page Info**

to display all the information regarding the web server's certificate and the algorithm used to cipher the secure page.
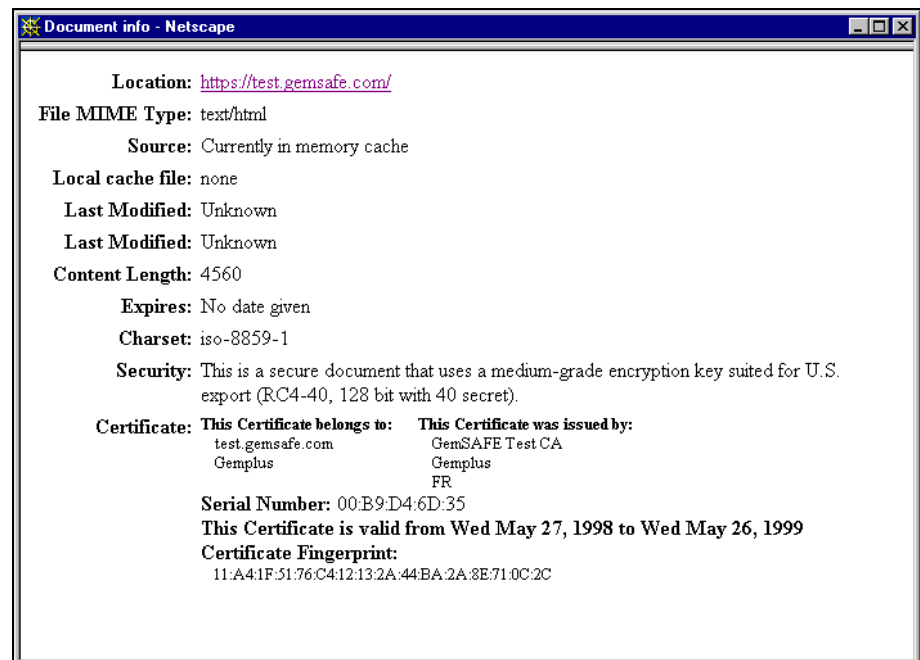

**Figure 17 - Secure Page Information**

The SSL Handshake

The SSL handshake takes place each time you start a secure web session. This operation identifies the server and it is automatically performed by your browser.

Note: *The SSL handshake can only succeed if the server's certificate is still valid.*

The SSL Process

If you want to have a secure web session with Server 1, you may want to make certain that Server 1 is indeed what it claims to be (the real Server 1 and not another entity impersonating it). In this case, you will want a certificate ensuring this from Server 1's CA. Of course, you also will have to make sure that the certificate you receive comes from the genuine CA and not from some entity impersonating it!

1. You start the SSL operation by sending the server a random number. The server returns a certificate and a signature of the random number. The certificate provides the server's public key, and the signature proves that the server currently has the private key corresponding to the certificate it is sending.
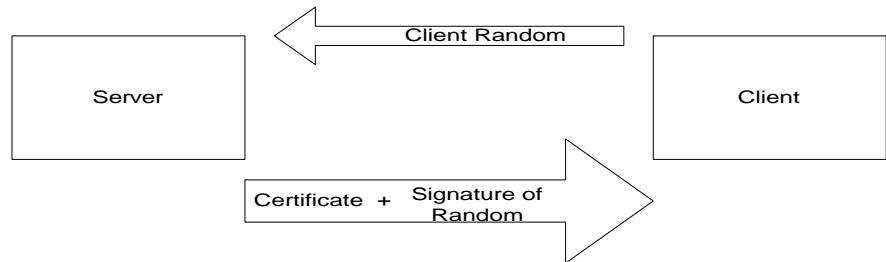
Figure 18 - The SSL Process

2. Now the CA's signature needs to be verified. This is done by comparing the CA's public key (obtained in the certificate received in the previous step) to the public key in a certificate obtained by other means (usually from a directory or a public list).

3. Your browser now checks that the name of the web site in the certificate matches the name you typed (for example, test.gemsafe.com).

SENDING SECURE MESSAGES

To access your secure e-mail, click **Communicator**, click **Security** and click **Messenger**. For some versions of Netscape, for your secure e-mail to operate properly, you need to make certain that the settings entered on this screen are the same as those you plan to enter in the mail you are about to send.

In other words, you can check any and every one of the following options, as long as you check the same options in the mail message:

- Encrypt mail messages, when it is possible

- Sign mail, when it is possible

- Sign discussion (News) messages, when it is possible

Figure 19 - The Messenger Screen

There may be several reasons why it is not possible to use one of these options. These will be detailed later on.

Before you are ready to send someone a encrypted message, you need a current certificate for this person. You thus need to receive a plain-text mail from this person with a signed certificate, or you need to look up this person's certificate (from a directory, for example).

Similarly, when you want to send an encrypted mail to a list of persons, you need to have a certificate for each addressee, or your mail will not go out to anyone.

Once you have received a signed message from someone, this person's certificate is automatically stored by your browser so that this operation does not have to be repeated systematically.

To view the list of certificates your browser already knows, click **Security**, then click **Certificates** and click **People**.

Figure 20 - Other People's Certificate Screen

Your browser also automatically checks that the name in the e-mail address you are writing to corresponds to the name in the certificate.

Similarly, when you receive a message, your browser checks that the e-mail address of the sender matches the sender's certificate.

Sending Conditions

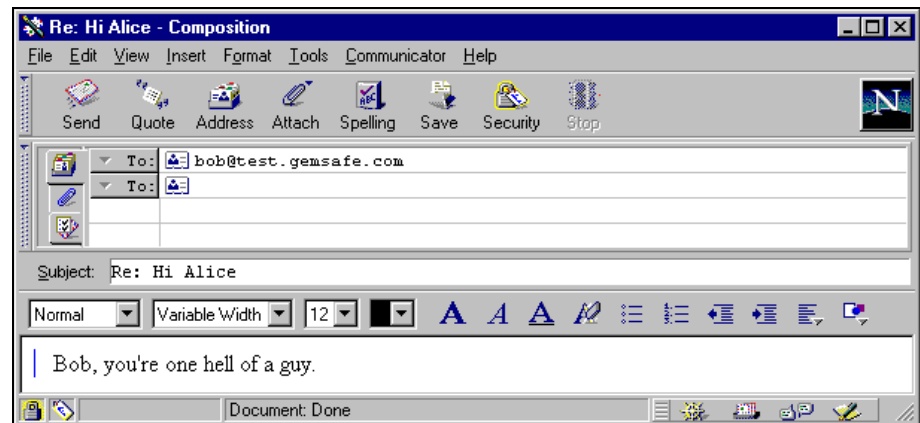If you click the sending conditions icon, the following screen is displayed:

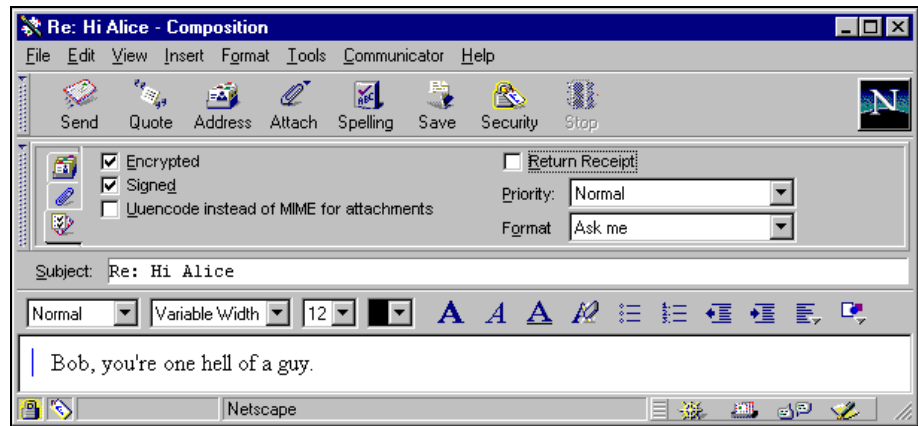**Figure 21 - Setting the Sending Conditions**

Figure 22 - Checking that the Sending Conditions Match

Note: *Check that the sending conditions' configuration is exactly the same here as what is displayed when you click **Security** and **Messenger**. Indeed, if sending conditions in these two locations do not match, current versions of Netscape will claim that you do not have a certificate, although you may actually have one. If such a situation occurs, remember to set the sending conditions properly.*

Ways to Send Messages

1. My message is "Hello". I send it in plain-text.

2. My message is "Hello". I encrypt it using the recipient's certificate (which includes the recipient's public key). I have obtained this certificate either from a public source of information, such as a directory, or from a previous mail the recipient sent me in the past. The message becomes "gobbledigook". The recipient deciphers it using his/her private key.
    This provides confidentiality and privacy.

3. My message is "Hello". I send it in plain-text with my signature and my digital certificate (which includes my public key). The certificate may be used by the recipient to verify my signature.
    This provides authenticity and integrity.

4. My message is "Hello". I concatenate this message with my signature and my certificate and encipher the result.
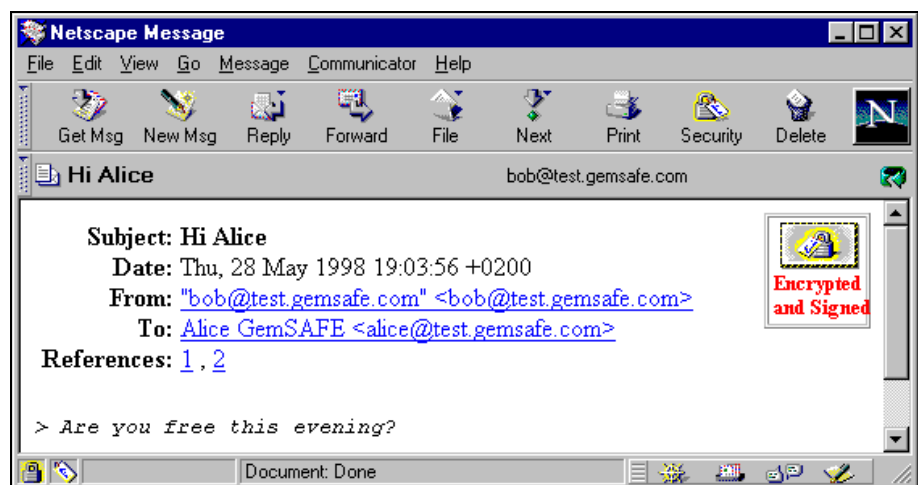   This is equal to 3. plus 2. and it therefore provides privacy, authenticity and integrity.

Figure 23 - Receiving a Secure e-mail

Key Length and Secure E-Mails

There are various levels of security depending on the length of the keys used. This length is primarily dictated by the laws of the country you are in. If you are located in the USA, for instance, you may be able to generate keys of a length of 168 bits. However, in the international version of Netscape the maximum authorized key length is 40 bits. Thus, a message sent from the USA using a 168-bit key cannot be deciphered in France, for example (where browsers do not have a 168-bit option for the key length).

Therefore, bear in mind what the addressee's cryptographic capacity is before sending the message, or it may be impossible for this person to read your message!

For additional information about key lengths, see *Levels of Security*.

LEVELS OF SECURITY

There are different versions of the Netscape Navigator browser. The length of the keys your browser uses depends on U.S. export rules and on applicable regulations in your country.

Public Key and Symmetric Key Length

Outside of the United States, the length of the RSA public/private key pair (which is used for signing, verifying, etc.) is often limited to 512 bits, while the symmetric key (used for bulk encryption) usually is 40 bits long.

Therefore, depending on the country you are located in, you may find you have many or few options to choose from to set the key length.

Note: *The symmetric key length is determined by both the browser and the card.*

THE CARD DETAILS TOOL

PIN Code Management

The Card Details Tool provides a status of the certificate and keys in the card, and PIN code management.

To use the Card Details Tool, click:

* The Windows **Start** Menu
* **Programs**
* **GemSAFE Card Details**

Follow the instructions as they appear on the screen.

Your GemSAFE™ card is protected by a PIN code. This PIN code must be four to eight digits long.

Three wrong PIN code presentations lock the card and therefore prevent its further use. To access the PIN code management section from the GemSAFE Card Details, click:

* **PIN**
* **Verify User PIN**

Unblocking the PIN Code

If your PIN code is blocked, click:

* **PIN**
* **Unblock**

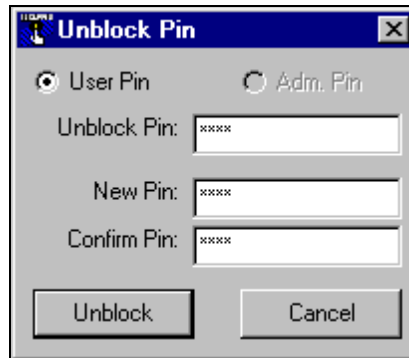Follow the instructions as they appear on the screen.

Figure 24- The Unblocking Code

Changing a Pin Code

To change either your user or your adminstration (unblocking) pin code, click:

* **PIN**
* **Change**

Select either your user PIN code or your administrative PIN code. Enter your old password and the new one as per instructions on the screen.

Figure 25- Changing the PIN Code

Your PIN code must not exceed a length of eight digits.

Note: *The default is 1 2 3 4 for both the user and the administrative PIN codes.*

GLOSSARY

Certificate   A certificate is identification for secure
                                     transactions. It consists of a public key and other
                                     data, all of which has been digitally signed by a
                                     Certificate Authority (CA). It is a condition for
                                     access to secure e-mail or to secure web sites.

Fingerprint   A digest of a certificate.

SSL   (Secure Socket Layer/Transport Layer Security)
                                     Communicating 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. It is automatically performed by your
                                     browser.

S/MIME   Off-line message format standard for use in
                                     secure mail applications.