

SSA

User Manager

User Guide

Version 1.2.5

Cybernetica

Akadeemia tee 21, EE0026 Tallinn, ESTONIA

<http://www.cyber.ee/>

© 1998 Cybernetics Ltd., all rights reserved.

Cybernetica and the Cybernetica logo are registered trademarks and Privador, Barricade, and SSA are trademarks of Cybernetics Ltd. All other brand, product or company names are trademarks or registered trademarks of their respective owners.

Contents

Introduction.....	3
Operation	5
Technology	6
User authentication. Certificates.....	7
Session reuse	8
Distinguished names.....	9
Strength of the cryptoalgorithms used	11
Model of the SSA system.....	13
Session.....	16
Licensing	17
SSA User Manager operations	19
User Manager key file.....	19
Creating the key file.....	20
Opening the key file	21
Saving the key file	22
Saving the key file under a new name	22
Closing the key file.....	22
Changing the key file passphrase.....	23
Users.....	23
Creating the user configuration file	23
User certification	25
Managing the public keys of other certification authorities.....	27
License management.....	29
Exporting the User Manager certificate	31
Changing the defaults	31
View issued certificates	33
SSA User Manager: A step-by-step guide.....	35

Step 1: Establish your certification policy and acquire the rights to use the SSA.....	35
Step 2: Create a new key file.....	36
Step 3: load the public key of the certification authority to the Users Manager	36
Step 4: load to the Users Manager the license allowing connecting to your server	36
Step 5: Enter the defaults conforming the certification policy	36
Step 6: Save the key file.....	37
Step 7: Close the key file	37
Creating the user configuration file	37
User certification.....	38
Security of the SSA User Manager	39
Theft of the private key	39
Necessary security measures	40
Installation	41
Problems	43
I lost my key file passphrase. What can I do?	43
Cybernetica Software End User License Agreement.....	45
SSLeay copyright notice	49

Introduction

The fast development of the computer and telecommunication technologies has brought about an explosive growth of the Internet and a massive use of client/server applications. However, this advance has its dark side — the information security concerns. As the Internet protocol stacks have no embedded safeguards, the data traffic can be easily eavesdropped or modified. The users are often not aware of the threats and transmit their sensitive information with no protection through the Internet.

Dedicated software with necessary safeguards for networked environments is often expensive, unavailable, unfriendly, inadequate or not existing at all.

Therefore, the problem needs to be solved how to protect with minimum costs the existing applications that are quite effective for their intended purposes. One has to consider as well the direct costs (the price of the security solution) as the indirect ones (user training, maintenance etc.).

The solution for the problem is the *Secure Sockets Agent (SSA)*.

The SSA is a system for securing the communication between unprotected or inadequately protected applications. The SSA consists of a pair of proxies, one of which works on the client computer and the other on the server computer. The proxies create an encrypted channel between them with all the traffic between the client and the server being tunneled into this channel.

This Guide explains the functioning, installation and configuring of the SSA software package, providing some examples of using the SSA. It would be useful to study the underlying principles of the SSA, as a profound understanding of these helps in learning to use the SSA more extensively and to assure the secure data communication.

Operation

SSA is designed to secure the client/server applications based on the TCP/IP protocol. Typically these applications have no protection from unauthorized eavesdropping or data modification.

In case of an ordinary client/server application, the client program in the client computer establishes the connection with the server program in the server computer. The traffic through the public network is public, so the intruder can monitor the data exchange between the client and server and/or change the transferred data.

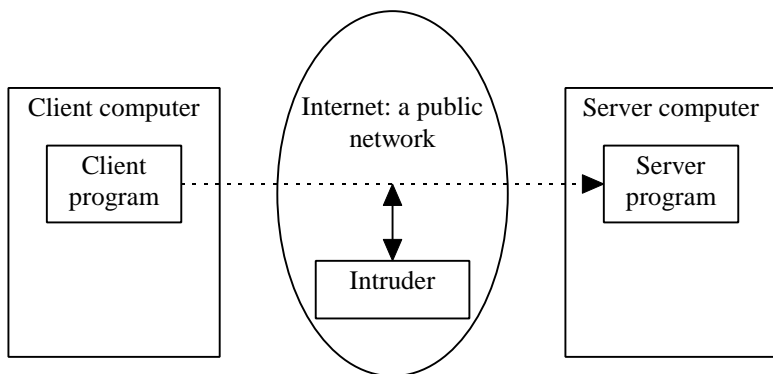


Figure 1. An unprotected client/server application

When using the same client/server application with the SSA an extra program is added to the client and server computers. The client starts the SSA Client and the server starts the SSA Server. Now the application to be protected does not directly access the server, but establishes the connection with the SSA Client running in the same client computer. This unprotected connection is made internally in one computer only, so the transferred data never reach the network. When the connection is made, the SSA Client connects to the SSA Server. To secure this connection, the SSL protocol is used providing for data confidentiality and integrity, and for the authentication of the communicating parties. The SSA Server connects to the server program after the secure channel is established and now a secure data exchange can begin between the client and the server.

The method is known as tunneling: in our case SSA tunnels the insecure TCP connection secure SSL channel (functioning via TCP). As a result, the intruder cannot monitor or change the data flows between the client and the server.

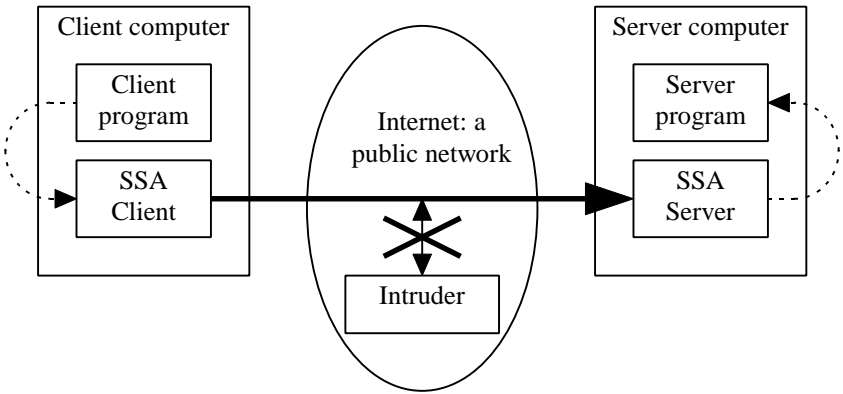


Figure 2. A client/server application protected with SSA

As the SSA proxies are self-contained programs, their use does not require any changes in the code of the application: the SSA allows protecting any TCP-based application that uses static ports.

The SSA can also mediate a couple of protocols which do not use any fixed TCP ports; these are the FTP, the Progress database system protocol and SQL*Net protocol from Oracle. With these protocols, the SSA will listen to the control connection and interpret the commands exchanged between the client and the server, to open a new TCP connection.

Technology

The SSA is based on the SSL (Secure Sockets Layer) protocol from the Netscape Communications Corporation, worldwide established as de facto standard and massively used for the WWW applications protection. The SSL allows creating a secure communication channel between two applications, assuring confidentiality (by using strong cryptographic algorithms, such as IDEA and 3DES) and integrity (by using message digest algorithms like MD5 and SHA) of the data transferred. The SSL protocol also assures the authenticity of the communicating parties (the client and the server), preventing the unauthorized use of the server resources and preventing the creation of fake servers to get the information from the client. For authentication of the communicating parties the SSL protocol uses digital signatures based on the asymmetric cryptographic algorithms (RSA) and

certificates issued by a trusted third party. The SSA sets no limitations to the lengths of the cryptographic keys.

The SSA supports the SSL versions 2 and 3. The older version contains several significant errors that have been corrected in the new version. The SSLv3 protocol has repeatedly been analyzed and found to be essentially correct. The SSL newer version also allows using the ephemeral Diffie-Hellman algorithm for key exchange, ensuring the perfect forward security.

By default, the client and server are authenticated when using the SSA. The client authentication can be switched off; in this case, a separate configuration file has not to be sent to each client: a single common file is sufficient, containing the session data and the Certification Authority public key for authenticating the SSA Server certificate.

User authentication. Certificates

The SSL uses asymmetric cryptographic algorithms for communication partner authentication.

In asymmetric cryptography, different keys are used for encryption and decryption. Only its owner, i.e. to the person who generated the keypair knows the private key of a keypair. The other key is public, i.e. everybody may know it. The keys are matched and complement each other: a ciphertext created with a public key can only be decrypted with the matching private key and vice versa – a ciphertext created with a private key can only be decrypted with the corresponding public key. It is not possible to derive the private key from a known public key.

These unique properties allow using asymmetric cryptoalgorithms for the authentication of the communication parties. Prior to this the parties must exchange their public keys in such a way that both parties are sure that only the other party has the private key corresponding to the valid public key.

With the SSL protocol, the client generates a random number when establishing the communication session, encodes it with the server public key and transmits the ciphertext to the remote party. The right server only, being a sole owner of the necessary private key, can decrypt the ciphertext. The number generated by the client will later have an important role in the SSL protocol, so the session cannot be continued without the previous action being successful. In case of any fake server not having the right private key the session will be immediately closed. The client authentication is carried out in a similar way.

The public key distribution problem has to be solved in this authentication scheme. The task may seem a simple one, but it is not. A public key is simply a large integer telling nothing about how it has been generated or who owns the matching private key. One public key is no better or worse of the other; the public key becomes a special and important one only with knowledge about the identity of the owner of the matching private key. The link between a private key and its owner is not obvious, one needs some additional information showing that the key K belongs to the person P . Moreover, this information must be trustworthy, it must be obtained and kept in a way preventing the use of any false keys.

For a system with two or three users the solution is simple: to get the public key, one can contact the owner personally. Only then can he or she be sure about the ownership of the key. Unfortunately, this scheme is not scalable; it will not work with more users and/or larger geographical area. We need a system allowing without personal contacts to verify a key belonging to a totally strange user or to a person in the other end of the world.

To solve the problem, a third party will be involved with the responsibility to issue certificates proving the ownership of the public keys. The certificate containing user's name and his public key is signed with the digital signature of the third party. This third party is described as the Certification Authority, with the procedure being called certification.

As the certificates are carrying the digital signature of the Certification Authority, they cannot be forged. One can always check the correctness of data in a certificate, using the public key of the Certification Authority. With this scheme, no personal contacts are needed any more; you have only to get the public key of the Certification Authority and now you can securely communicate with all users certified by this Authority.

Using the certificates provides for the scalability of the systems based on the SSL protocol and saves the end users from the necessity of making any critical decisions.

More about certification and cryptography can be found in the Internet, e.g.: <http://www.cyber.ee/>.

Session reuse

Some protocols make a lot of short-time connections (HTTP in the first place) and may slow down when secured with the SSA. The problem can be solved by reuse of the sessions.

With session reuse, the SSA Client and Server carry out the full authentication procedure only by establishing the first connection. For next connections the private keys will be used which were exchanged by establishing the first connection. Making a new connection with session reuse will be a magnitude faster compared to the option with full authentication.

To secure the reuse, each session must have a certain limited lifetime. When this time expires, the Client and the Server must carry out a new full authentication procedure. At present, the lifetime limit is set to 5 minutes.

Session reuse can be enabled and disabled from Client side as well as from Server side. Sessions will only be reused when enabled from both sides.

Distinguished names

The SSL protocol uses certificates complying to the X.509 standard, where the owner of the certificate and the issuer are identified with so-called distinguished names. A distinguished name is a hierarchical globally unique name defined as a string of relative distinguished names consisting of attribute/value pairs.

The uniqueness of distinguished names is assured by using a hierarchical naming scheme. For example, the four-level name

`/C=UK/O=IOC/OU=IT/CN=John Johnson`

tells us, that its owner works in the United Kingdom (C=UK), in the organization IOC (O=IOC), in the IT department (OU=IT) and his real name is John Johnson (CN=John Johnson). A lot of organizations worldwide can use the acronym IOC (e.g. the International Olympic Committee), but this one is located in the United Kingdom. Thousands of John Johnsons could be found in the world, but only one of them is working in this department of this organization.

The following attributes can be used to create the distinguished names:

Attributes related to the geographic location		
C	Country code	ISO 3166 standard country code. Examples: EE for Estonia, DE for Germany etc.
S	State, province, county, etc.	Examples: Ohio, Sussex, Dalarna
L	Location (city, village etc.)	Examples: Vienna, Bordeaux, Brighton
Attributes related to the organization		
O	Organization name	Examples: Fishfood Ltd., UCLA, Smith & Sons
OU	Department name	Examples: IT, Finances, R&D
Other attributes		
CN	Common name	Name under which the person is known. Can contain titles, abbreviations etc. Examples: Ann McClaren, Ph. D., Mr. Stanley Higgins
Email	E-mail address	Example: arne@mail.ee

Figure 3. Attributes in the distinguished names

For the Certification Authority of an organization the following name format can be used:

/C=XX/O=Company name/OU=CA

For server names the following format can be used:

/C=XX/O=Company name/CN=Name of the service provided by the server

For user names the following format can be used:

/C=XX/O=Company name/CN=First name Last name

Strength of the cryptoalgorithms used

DES (with 56-bit key) is the oldest public cryptoalgorithm, which has successfully survived all kinds of attack attempts. There is no effective method known to break DES faster than with brute-force attack (i.e. exhaustive search of the keyspace). DES is considered a little insecure nowadays only due to the fast development of the computer technology in the last decade, making brute-force attacks effective. Therefore, the **3DES** has to be preferred for more rigid security requirements; it is threefold slower, but gives a significantly higher security level. Its formal key length is 3×56 bits, making the effective length to be at least 80-90 bits.

IDEA (with 128-bit key) can be considered to be secure, as no effective breaking methods are known. Brute-force attacks of IDEA or 3DES will not be successful in the near future.

RSA is a thoroughly explored public-key cryptosystem. A great number of top researchers is studying the security of the RSA; for the present, the tools for breaking it are still rather theoretic and can only be effective for short (400-500 bits) keys. RSA with 768-bit or 1024-bit keys can be considered as a completely secure and practically unbreakable system.

The hash functions **MD5** (with 128-bit output) and **SHA** (with 160-bit output) have been explored rather well. By now, some success has only been achieved in breaking of MD5, but existing attack methods are still merely theoretic.

RC2 and **RC4** are variable key-length ciphers developed by the RSA Data Security. RC2 is a block cipher and RC4 is a stream cipher. These algorithms are proprietary and not very thoroughly analyzed. Therefore, some public and well-explored algorithm like DES or IDEA should be preferred.

The **Ephemeral Diffie-Hellman (EDH) algorithm** is used for the session key exchange. With EDH, the Perfect Forward Security (PFS) property will be ensured. RSA is only used for authentication of the Diffie-Hellman algorithm. It means that even if an attacker succeeds to get the RSA private key, he can decrypt no old sessions. With EDH, compromising the private key will not have any retroactive impact.

Security of the ephemeral Diffie-Hellman key exchange algorithm depends on parameters used. The parameters are public and have not to be concealed. Generating the parameters and checking their suitability is a very complicated and important procedure. The SSA uses a set of parameters defined by the RFC 2412 standard (see RFC 2412 standard "The OAKLEY Key Determination Protocol", appendix E.2 "The Well-Known Groups 2: A 1024 bit prime"). The

long integer in this set is derived from the decimals of pi and its primality has been carefully verified. This number has some favourable properties increasing the effectiveness and security of its use.

The following table lists all the combinations of cryptographic algorithms supported by the SSA. Numbers in parentheses show the length of the key (bits). It is recommended to use the algorithms from the beginning of the list. Weak algorithms with no export restriction can definitely not be recommended.

Cipher description	SSLv3	Key exchange	Encryption	MAC	Export
EDH-RSA-DES-CBC3-SHA	✓	EDH	3DES(168)	SHA1	
IDEA-CBC-SHA	✓	RSA	IDEA(128)	SHA1	
DES-CBC3-SHA	✓	RSA	3DES(168)	SHA1	
RC4-SHA	✓	RSA	RC4(128)	SHA1	
RC4-MD5	✓	RSA	RC4(128)	MD5	
EDH-RSA-DES-CBC-SHA	✓	EDH	DES(56)	SHA1	
DES-CBC-SHA	✓	RSA	DES(56)	SHA1	
EXP-EDH-RSA-DES-CBC	✓	EDH(512)	DES(40)	SHA1	✓
EXP-DES-CBC-SHA	✓	RSA(512)	DES(40)	SHA1	✓
EXP-RC4-MD5	✓	RSA(512)	RC4(40)	MD5	✓
EXP-RC2-CBC-MD5	✓	RSA(512)	RC2(40)	MD5	✓
IDEA-CBC-MD5		RSA	IDEA(128)	MD5	
DES-CBC3-MD5		RSA	3DES(168)	MD5	
RC4-MD5		RSA	RC4(128)	MD5	
RC2-CBC-MD5		RSA	RC2(128)	MD5	
RC4-64-MD5		RSA	RC4(64)	MD5	
DES-CBC-MD5		RSA	DES(56)	MD5	
EXP-RC4-MD5		RSA(512)	RC4(40)	MD5	✓
EXP-RC2-CBC-MD5		RSA(512)	RC2(40)	MD5	✓

Figure 4. Supported ciphers

Model of the SSA system

Let's take a look at a non-secured client/server application first, where the client accesses the server directly. Only one program in this system needs to be configured; it is the client program and to configure it, one must know the IP address of the server computer and the TCP port number of the server.

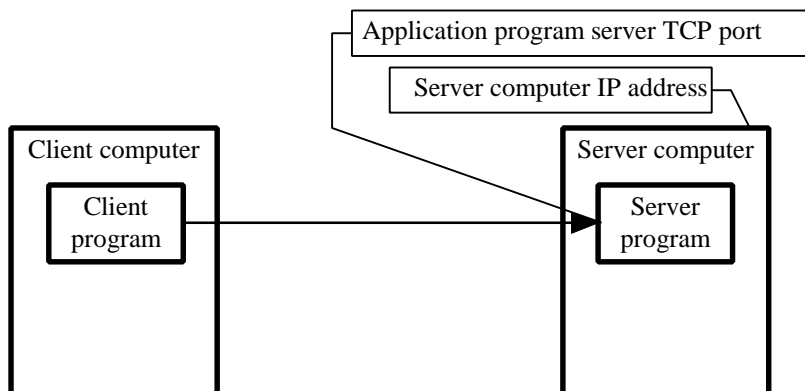


Figure 5. The model of a non-secured client/server system

With the SSA, the system will be more complex. Now, three programs need to be configured: the client program, the SSA Client, and the SSA Server. The user manages the first two of them; the server administrator manages the third one.

To configure the client program, one still needs two parameters: the client computer loopback IP address, which is almost always 127.0.0.1 and the TCP port number of the SSA Client, which can be chosen by the user himself.

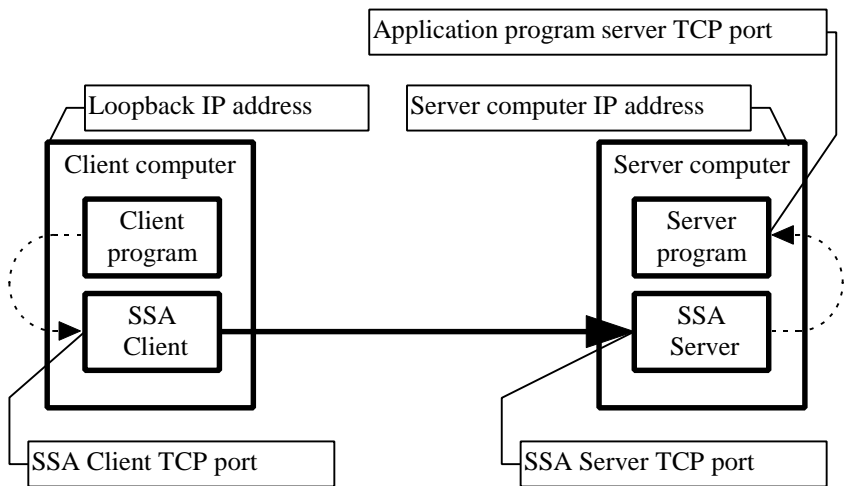


Figure 6. The model of the client/server system secured with the SSA

To configure the SSA Client, more data are needed. One must know the SSA Client TCP port, the server computer IP address and the SSA Server TCP port. The client's port number can be chosen by the user, the server computer IP address is, as a rule, set and known from the times when the non-secured application was used; the SSA Server port number is assigned by the server administrator.

To configure the SSA Server, the server administrator must know the SSA server TCP port number, the server computer IP address, and the application program server's TCP port number. The SSA Server port number can be chosen by the administrator; the server computer IP address is, as a rule, fixed and known, and the same holds for the application program server TCP port as well.

In addition to the network addresses and port numbers, the security attributes have to be defined. First of all, the owner of the server has to choose the (-ies) whose services he will use. He has various options: for example, he can use a public certification authority whose security policy is acceptable for the server owner and for the users. In case of so-called "closed applications" having a limited user community, it would be better for the server owner to create his own certification authorities for his clients and servers. In any case, the SSA Client has to be provided with the public key from the certification authority which certified the SSA Server and vice versa – the SSA Server has to be provided with the public key from the certification authority which certifies the SSA Clients.

The system will be more secure when the Clients and the Server are certified by different certification authorities.

The public key of the certification authority has a form of a self-signed X.509 certificate. Note that this key will be used for the final authentication of the client and the server, therefore the authenticity of this key must be verified through independent channels (e.g. by taking a phone call to the partner etc.) before it can be put in use.

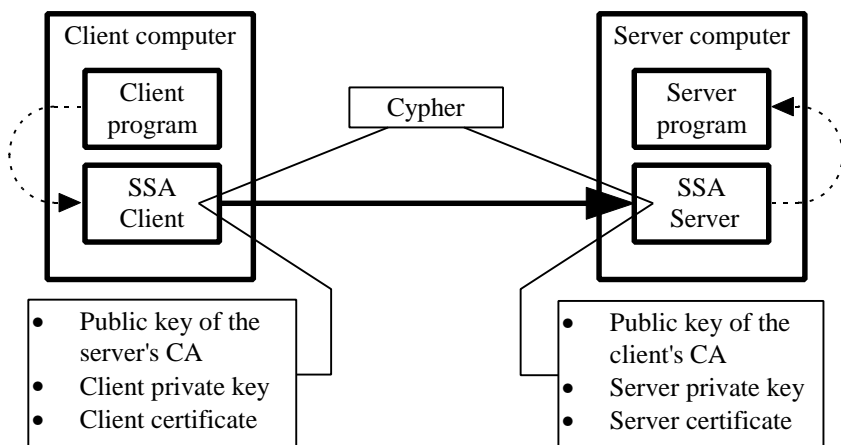


Figure 7. Security attributes in the SSA system

The owner of the server also determines the cipher to be used for the creation of the secure channel.

Finally, the client and the server shall generate a RSA keypair for them and certificate them in the chosen Certification Authority.

The table below makes a summary of the parameters needed for the SSA system configuration, grouped by the points of application:

Parameter	Source	Point of use
Loopback IP address	Fixed. 127.0.0.1	Application program client
SSA Client TCP port	User-defined	Application program client, SSA Client
Client private key	User-generated	SSA Client
Client certificate	Certification Authority	SSA Client
Server computer IP address	Server administrator	SSA Client, server
SSA Server TCP port	Server administrator	SSA Client, server
Public key of the client's CA	Client's Certification Authority	SSA Server
Public key of the server's CA	Server's Certification Authority	SSA Client
Cryptoalgorithm	Server administrator	SSA Client, Server
Application program server TCP port	Fixed or variable	SSA Server
Server private key	Server administrator	SSA Server
Server certificate	Certification Authority	SSA Server

Figure 8. The SSA system parameters

Session

The notion of session is central in the whole SSA system. The session is a set of parameters describing the secure communication channel created with the SSA. All the parameters together explained in the previous subdivision describe one session.

The session parameters are twofold: SSA Client parameters and SSA Server parameters. We talk about Client sessions and Server sessions correspondingly.

The Client session consists of SSA Client TCP port, server computer IP address, SSA server TCP port, client private key, client certificate, public key of the server's CA, and the cipher.

The Server session consists of SSA Server TCP port, server computer IP address, application program server TCP port, server private key, server certificate, public key of the client's CA, and the cipher.

Session parameters are stored in the SSA configuration file. To prevent modification and copying, the file is encrypted using symmetric cryptoalgorithm with the key derived from the passphrase entered by the user.

Licensing

The SSA software is being freely distributed, but to use it legally, the customer has to sign an agreement with the Cybernetica giving him the right to use the SSA software under conditions stipulated in the agreement. By signing the agreement, the customer will also get from the Cybernetica the licensing information file(s) containing data about the licensee, the licensed product, and the conditions of its use.

```
License owner:      AS Wonderbar
License number:     1234567
Product:           SSA Server
License type:       Commercial
Number of licenses: 1
Number of clients:  unlimited
```

Figure 9. The data stored in the licensing information file

This file is secured with Cybernetica's digital signature, so the data in the file cannot be modified. To use the licensing information file, it has to be loaded into the configuration file of the corresponding SSA program. During the loading, the digital signature of the file and the pertinence of the license for the given program will be checked. Each SSA program requires a pertinent license: the SSA Server refuses to use a SSA Client license and vice versa.

The license will also be checked by the SSA Client and Server before the session activation. If the license is missing, invalid, or impertinent, a corresponding warning will be recorded in the log file, but the procedure will proceed. If the license is OK, the data from the license will be stored in the log file.

Two types of licenses exist for the SSA Server:

- the license for using the SSA Server to secure a public server (*SSA Public Server License*). This license does not allow the SSA Server to authenticate the clients.
- the license for using the SSA Server to secure applications having a limited users community. This license allows the SSA Server to authenticate the clients. The license gives rights for servicing a preset number of clients. If the preset limit is violated, the SSA Server will log a corresponding warning, but will proceed.

SSA User Manager operations

The User Manager runs in the Windows environment and is equipped with the graphical user interface.

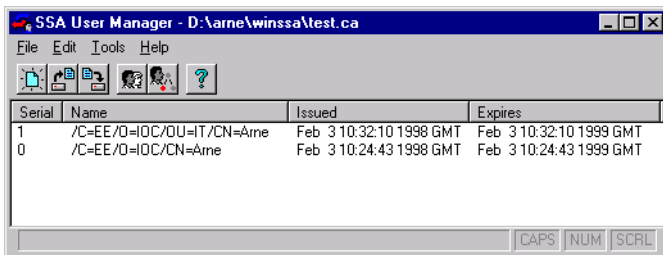


Figure 10. The graphical user interface of the SSA User Manager

The main window of the User Manager shows the names of the certified users and the details of certificates issued.

User Manager key file

The private key for signing the users' certificates is stored in the User Manager key file. For confidentiality and integrity protection of the key, the file is equipped with check codes and encrypted. In addition to the key this file also contains the self-signed certificate of the User Manager and the default values used for creating user certificates and configurations.

Closely related to the key file is the user file, containing data about all certificates issued and about configuration files. When you open, close, save etc. the key file, the same operation will be automatically carried out with the users file.

If you try to close without prior saving an opened key file containing any unsaved changes, the User Manager will give a corresponding warning:

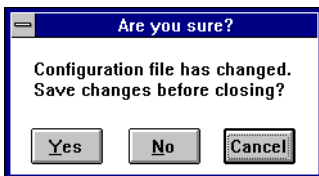


Figure 11. Warning about unsaved changes

Now you can choose saving the file, not saving it or canceling the operation.

Creating the key file

In the *File* menu select *New...* The Key Wizard appears, helping the administrator to generate a new RSA private key and a self-signed certificate.

Enter the length of the RSA key to be generated. To get adequate security, the User Manager private key must be at least 1024 bits long.

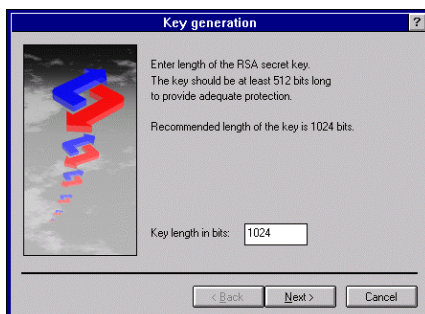
After entering the key length click *Next*.

Type the key file passphrase. To get adequate security, the passphrase must consist of at least 11 characters. The User Manager does not accept passphrases shorter than 8 characters. The passphrases should be regularly changed.

To avoid errors, the passphrase must be entered twice. When this is done, click *Next*.

Fill in the User Manager distinguished name. The name should contain the country code and the name of the organization. The *Organization unit* field should refer to the SSA User Manager.

After entering all the components of the distinguished name click *Next*.

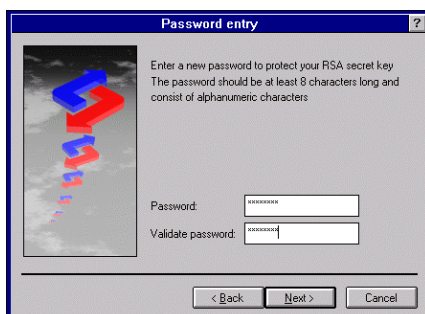


Key generation

Enter length of the RSA secret key.
The key should be at least 512 bits long to provide adequate protection.
Recommended length of the key is 1024 bits.

Key length in bits:

< Back Next > Cancel



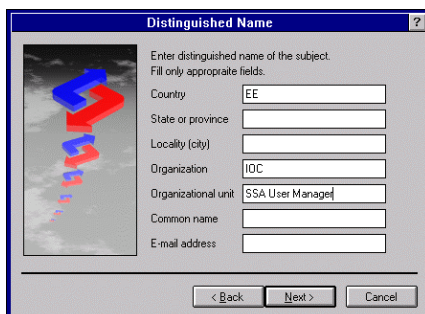
Password entry

Enter a new password to protect your RSA secret key.
The password should be at least 8 characters long and consist of alphanumeric characters.

Password:

Validate password:

< Back Next > Cancel



Distinguished Name

Enter distinguished name of the subject.
Fill only appropriate fields.

Country:

State or province:

Locality (city):

Organization:

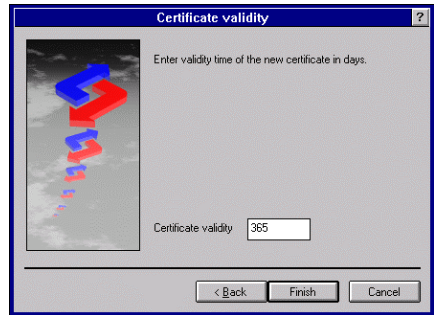
Organizational unit:

Common name:

E-mail address:

< Back Next > Cancel

Enter the validity period of the Certification Authority certificate. After expiration of this period a new key must be generated and all the users must be re-certified. As the procedure is rather troublesome, the validity period should be a couple of years. On the other hand, an excessive validity period would facilitate attacks.



After entering the validity period click *Finish*.

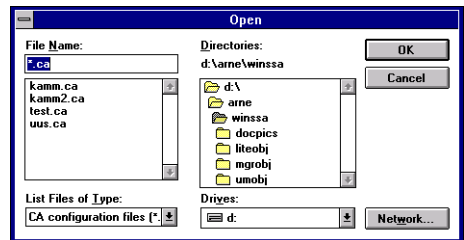
The User Manager will generate a private key of the specified length and create the self-signed certificate with the specified options.



Opening the key file

In the *File* menu select *Open...*. The key file opening dialog box appears.

NOTE. The dialog box may have different appearance, depending on the operating system.



Select the key file you want and click *OK*. The passphrase entering dialog box appears.



Type the passphrase and click *OK*. The list of the certified users appears in the User Manager window.



Saving the key file

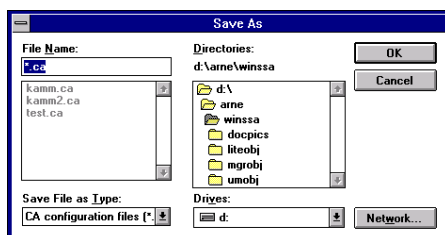
In the *File* menu select *Save*. The User Manager will save the data in the current key file which name is shown on the title bar of the window.

Saving the key file under a new name

In the *File* menu select *Save as...*. The key file saving dialog box appears.

NOTE 1. The dialog box may have different appearance, depending on the operating system.

NOTE 2. When you save the key file under a new name, the associated users file will be also copied.



Enter the new name for the key file and click *OK*.

Closing the key file

In the *File* menu select *Close*. The key file will be closed and all sensitive data will be removed from the memory.

Changing the key file passphrase

In the *File* menu select *Change password...* The passphrase dialog box appears.



Choose a new passphrase and enter it in the dialog box. Click *OK*.

Users

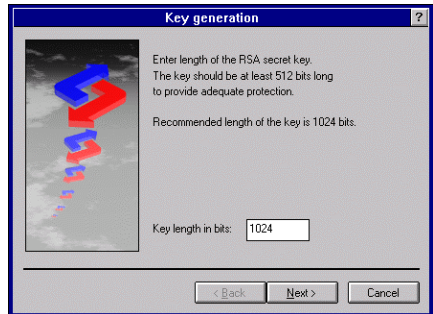
The main function of the User Manager is to create the SSA configuration files for the SSA Lite Client. The User Manager also allows certifying the users of the full-featured version of the SSA Client. To facilitate the work of the administrator, the User Manager saves default values for the most of the fields in the key file; so the administrator has not to enter them again and again.

Creating the user configuration file

In the *Tools* menu select *Create user...* The User Wizard appears, helping the administrator to generate the new SSA configuration file.

Enter the length of the RSA key to be generated. To get adequate security, the user private key must be at least 1024 bits long.

After entering the key length click *Next*.



Type the key file passphrase. To get adequate security, the passphrase must consist of at least 11 characters. The User Manager does not accept passphrases shorter than 8 characters.

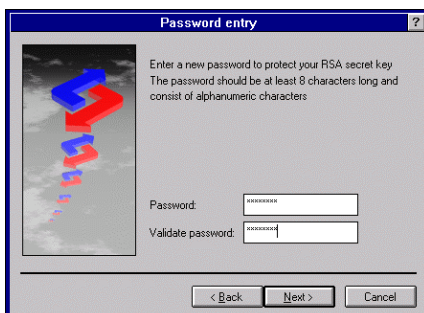
To avoid errors, the passphrase must be entered twice. When this is done, click *Next*.

Fill in the user distinguished name. The name should contain the country code, the name of the organization, the name of the organization unit and the first and last name of the user.

After entering all the components of the distinguished name click *Next*.

Enter the validity period of the user certificate. After expiration of this period a new key must be generated for the user and he must be re-certified. A reasonable validity period for the user certificate would be one year.

After entering the validity period click *Next*.



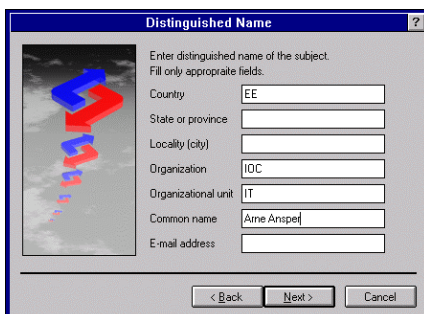
Password entry ?

Enter a new password to protect your RSA secret key
The password should be at least 8 characters long and consist of alphanumeric characters

Password:

Validate password:

< Back Next > Cancel



Distinguished Name ?

Enter distinguished name of the subject.
Fill only appropriate fields.

Country:

State or province:

Locality (city):

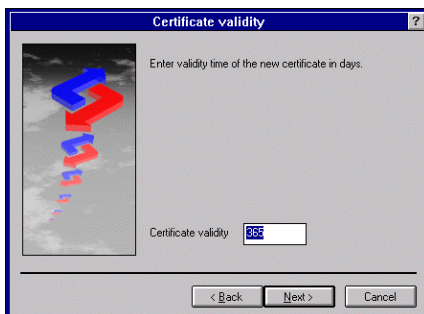
Organization:

Organizational unit:

Common name:

E-mail address:

< Back Next > Cancel



Certificate validity ?

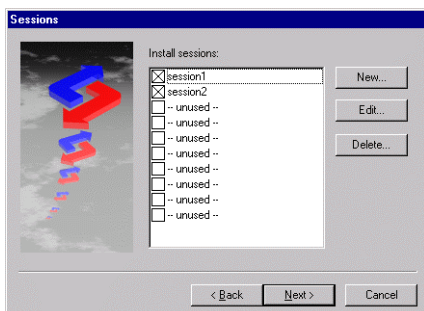
Enter validity time of the new certificate in days.

Certificate validity:

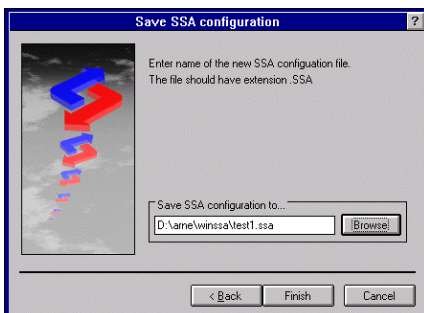
< Back Next > Cancel

Mark the sessions to be stored in the user configuration file. You can add or modify sessions as appropriate. More information about session parameters can be found in the “Session” section, p. 16.

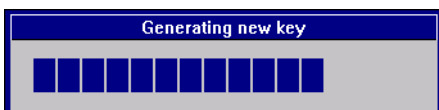
After marking the sessions click *Next*.



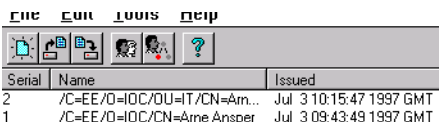
Enter a name for the user configuration file and click *Finish*.



The User Manager will create and save the SSA configuration file.



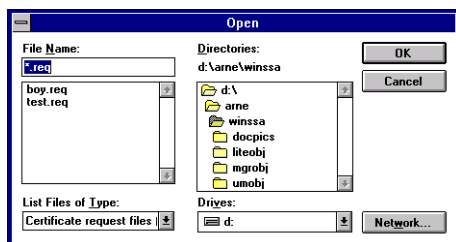
The name of the new user appears in the User Manager window.



User certification

In the *Tools* menu select *Certify user...*. The Certificate Wizard appears, helping the administrator to certify the user.

Select the file containing the user certification request and click *OK*.



The User Manager shows the distinguished name fetched from the certification request, and the public key. Prior to issuing the certificate the administrator has to check that the user has the right to use this name, the name satisfies the established rules, and the user's public key has not been modified.

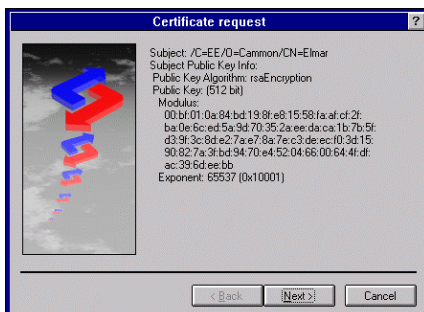
After the checks click *Next*.

Edit the distinguished name chosen by the user. The name should contain the country code, the name of the organization, the name of the organization unit, and the first and last name of the user.

After entering all the components of the distinguished name click *Next*.

Enter the validity period of the user certificate. After expiration of this period a new key must be generated for the user and he must be re-certified. A reasonable validity period for the user certificate would be one year.

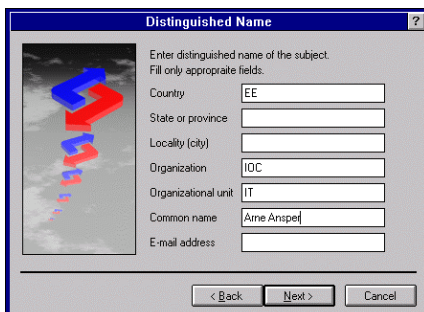
After entering the validity period click *Next*.



Certificate request

Subject: /C=EE/O=Cammon/CN=Elmar
 Subject Public Key Info:
 Public Key Algorithm: rsaEncryption
 Public Key: [512 bit]
 Module:
 00:bf:01:0a:84:bd:19:9f:e8:15:58:1a:af:cf:2f:
 ba:0e:6c:ed:5a:3d:70:35:2a:ee:da:ca:1b:7b:5f:
 d3:9f:3c:8d:e2:7a:e7:8a:7e:c3:de:ec:f0:3d:15:
 90:82:7a:3f:bd:94:70:e4:52:04:66:00:64:4f:df:
 ac:33:fd:ee:bb
 Exponent: 65537 (0x10001)

< Back Next > Cancel

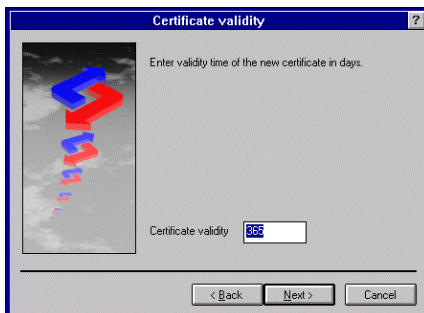


Distinguished Name

Enter distinguished name of the subject.
 Fill only appropriate fields.

Country:
 State or province:
 Locality (city):
 Organization:
 Organizational unit:
 Common name:
 E-mail address:

< Back Next > Cancel



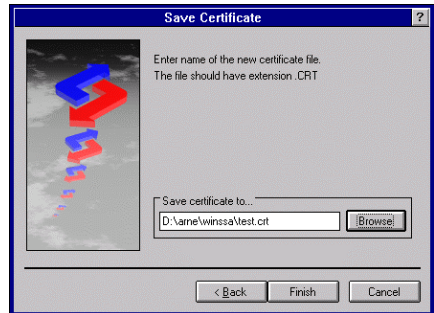
Certificate validity

Enter validity time of the new certificate in days.

Certificate validity:

< Back Next > Cancel

Enter the user's certificate file name, then click *Finish*.



The User Manager saves the user certificate in the selected file. The name of the new user appears in the User Manager window.

File Edit Tools Help		
Serial	Name	Issued
3	/C=EE/O=Cammon/CN=Elmar	Jul 3 11:50:30 1997 GMT
2	/C=EE/O=IOC/OU=IT/CN=Am...	Jul 3 10:15:47 1997 GMT
1	/C=EE/O=IOC/CN=Arne Anspen	Jul 3 09:43:49 1997 GMT

Managing the public keys of other certification authorities

When creating configuration files for the SSA Clients, the public key of the Server's certification authority has to be stored in these files. If the Client and the Server have been certified by different authorities, the key has prior to this to be loaded to the configuration file of the Users Manager certifying the Clients. The dialog for the management of the public keys of certification authorities allows the administrator to view, add, and delete the public keys. When a new SSA configuration file is being created, all the necessary keys will automatically be stored with session descriptions in this file.

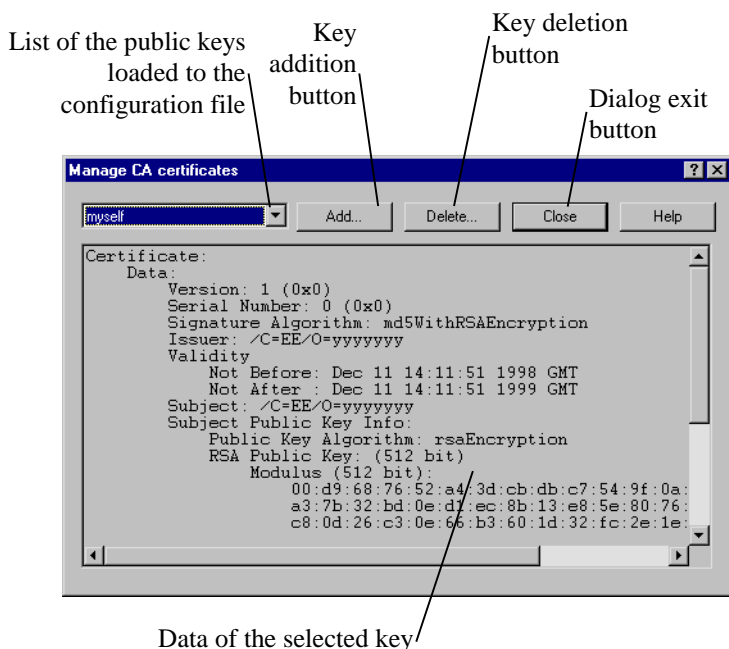
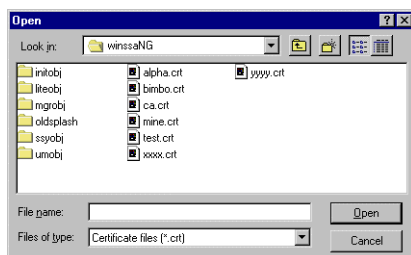


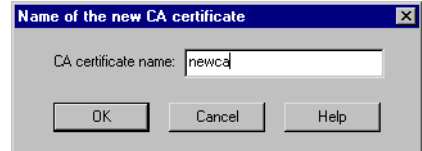
Figure 12. Certification authorities' public keys management dialog

When the configuration file of the Users Manager is being created, the public key of the Manager itself will be included to the public key list, under the name *myself*. Where the Server and Clients are certified by the same certification authority, this key has to be used.

To add other keys click *Add...* Type the name of the file containing the public key of the certification authority and click *Open*.



The dialog for entering the name of the certification authority public key will appear. Type the name you chose and click *OK*.



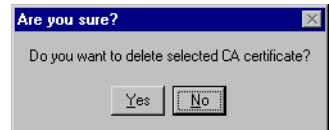
The data of new public key will appear on the screen. From now on this key can also be selected in the session descriptions modification dialog.

With the public key of the certification authority the Client verifies the authenticity of the Server's public key and vice versa – the Server verifies the Client's public key. Therefore, it is very important to have the right and valid public key of the certification authority. Otherwise the authentication of the partner and the whole resulting security provided by the SSA are not trustworthy.

Keys acquired over network can be never trusted without additional check through external channels. As the certification authority's public key is usually distributed in the form of a certificate, one can get a wrong impression of the data in the certificate being protected like they are protected in case of the Client's certificate. Actually, there is a great difference between a user certificate and a certification authority's certificate – the last is a self-signed certificate allowing no automatic verification of the authenticity of its data.

In other words – prior to adding a public key of a certification authority to the configuration file, the authenticity of the key must be assured.

To delete a public key click *Delete...* The dialog appears asking confirmation to the deletion. Click *Yes*.



If you unintentionally deleted your public key, you can still add it by exporting the public key of the Users Manager to a certificate file (see "Exporting the User Manager certificate") and adding it to the configuration file again.

License management

When configuration files are being created for the SSA Clients, the licens for connecting to your server can be stored to this file. Prior to this, the license has to be loaded to the configuration file of the Users Manager certifying the Clients. The dialog for the license management allows the administrator to view, add, and delete licenses. When a new SSA configuration file is being created, all the necessary licenses will automatically be stored with session descriptions in this file.

The license of the SSA Users Manager itself has also to be loaded to the configuration file.

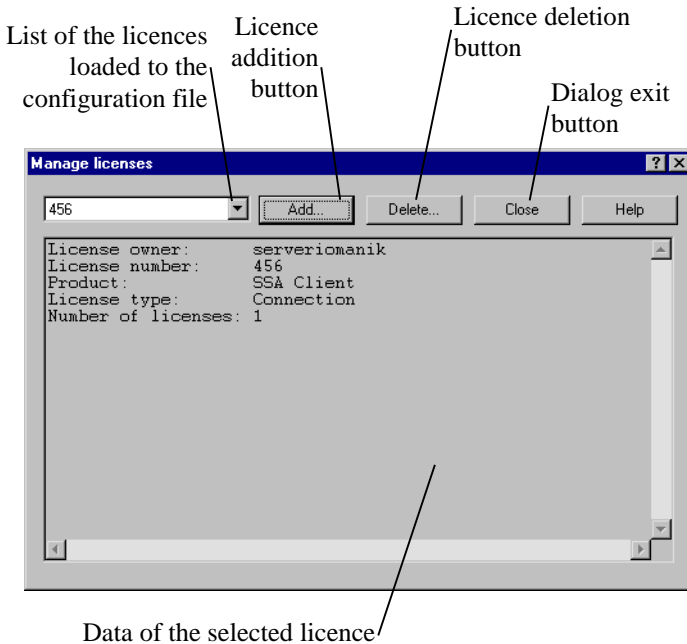
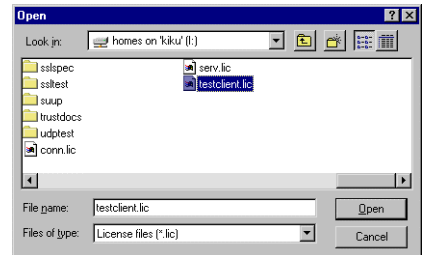


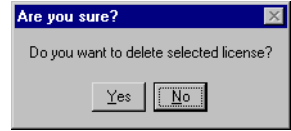
Figure 13. The license management dialog

To add a license click *Add...* Type the name of the licensing information file and click *Open*.



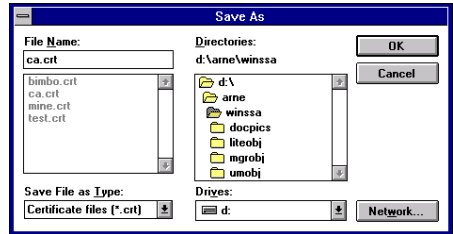
The data of the new license will appear. From now on, the license can also be selected in the session descriptions modification dialog.

To delete a license click *Delete...* The dialog appears asking confirmation to the deletion. Click *Yes*.



Exporting the User Manager certificate

In the *Tools* menu select *Export self-signed certificate...* Enter the name of the file for saving the User Manager self-signed certificate and click *OK*.

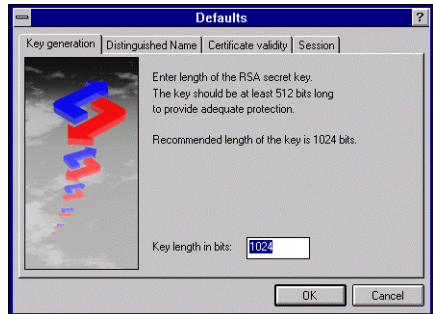


The User Manager will save the certificate in the selected file.

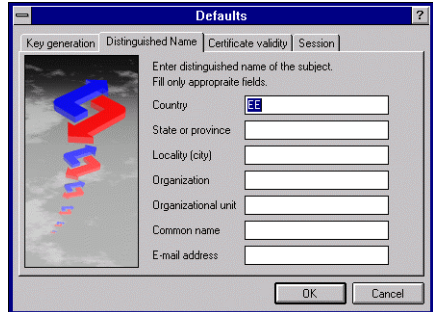
Changing the defaults

In the *Tools* menu select *Defaults...* The User Manager default options dialog box appears.

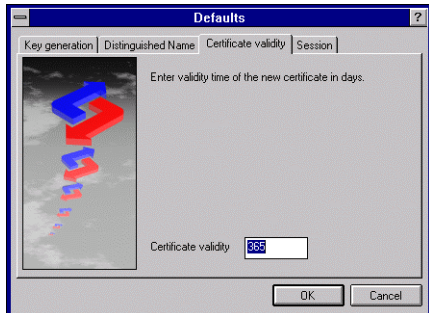
In the *Key generation* sub-dialog you can change the default length of the keys.



In the *Distinguished name* sub-dialog you can change the default distinguished name for users.

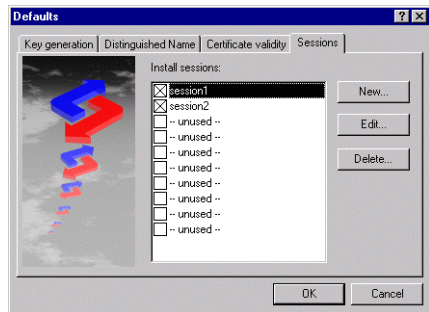


In the *Certificate validity* sub-dialog you can change the default validity period of the certificates.



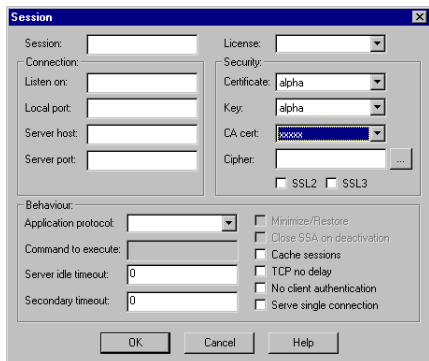
In the *Sessions* sub-dialog you can edit the sessions included to the user configuration file. By default, all the marked sessions will be proposed.

The User Manager allows saving the data of up to 10 sessions.



To add a new session, click *New...*. The new session addition dialog box appears. Enter the session parameters and click *OK*.

More information about session options can be found in the “Session” section, p. 16.



To modify the data of an existing session, click *Edit...*. The session editing dialog box appears. Modify the session options and click *OK*.

The **Session** dialog box contains the following fields and options:

- Session:** newsession
- License:** 123
- Connection:**
 - Listen on:** (empty)
 - Local port:** 3333
 - Server host:** server.cyber.ee
 - Server port:** 4444
- Security:**
 - Certificate:** alpha
 - Key:** alpha
 - CA cert:** xxxxx
 - Cipher:** EDH-RSA-DES-CBC3-S (dropdown)
 - ☐ SSL2 ☒ SSL3
- Behaviour:**
 - Application protocol:** (dropdown)
 - Command to execute:** (empty)
 - Server idle timeout:** 0
 - Secondary timeout:** 0
 - ☐ Minimize/Restore
 - ☐ Close SSA on deactivation
 - ☐ Cache sessions
 - ☐ TCP no delay
 - ☐ No client authentication
 - ☒ Serve single connection

Buttons: OK, Cancel, Help

To delete a session, click *Delete...*. The deletion dialog box appears, asking confirmation to deletion. Click *Yes*.

The **Are you sure?** dialog box contains the text: "Do you want to delete selected session?"

Buttons: Yes, No

When all the needed changes are done, click *OK*.

View issued certificates

Select from the list the certificate(s) you want to check.

Serial	Name
2	/C=EE/O=IOC/OU=IT/CN=Arne Ansper
1	/C=EE/O=IOC/OU=IT/CN=Arne
0	/C=EE/O=IOC/CN=Arne

From the *Tools* menu, select *View certificate...* The Certificate dialog box shows the contents of the certificate selected. To close the dialog box, click *OK*.

The **View** dialog box displays the following certificate details:

- Certificate:**
 - Data:**
 - Version: 1 (0x0)
 - Serial Number: 2 (0x2)
 - Signature Algorithm: md5WithRSAEncryption
 - Issuer: /C=EE/O=IOC/CN=XXX
 - Validity:**
 - Not Before: Feb 3 11:16:35 1998 GMT
 - Not After: Feb 3 11:16:35 1999 GMT
 - Subject:** /C=EE/O=IOC/OU=IT/CN=Arne Ansper
 - Subject Public Key Info:**
 - Public Key Algorithm: rsaEncryption
 - RSA Public Key: (512 bit)
 - Modulus (512 bit):

Buttons: OK

SSA User Manager: A step-by-step guide

The examples in this guide are based on a simple client/server database.

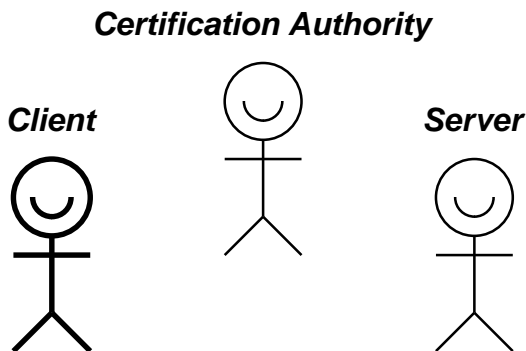


Figure 14. The subjects in the sample system

Step 1: Establish your certification policy and acquire the rights to use the SSA

The SSA Client and Server need for their functioning the certificates issued by a trusted third party. The certificate issued to the client gives it access to the service provided by the server. To ensure a reasonable use of the service, the rules of users' certification need to be established. These rules are known as certification policy. The operator of the Certification Authority must strictly follow the certification policy; an authorized auditor must check the conformity to the policy. The certification policy prescribes:

- requirements to a certifiee. For example, signing a special agreement for using the service may be required from the customer. The policy can limit the customer base according to their organizations etc.
- the certification procedure; the documents and actions needed to identify the certifiee candidate and his capabilities; the rules for generating customers' distinguished names; the validity period of the certificates

If the SSA User Manager will be used for certification of the users of a specific service, the certification policy can just constitute a part of the service provision agreement.

Before implementing the SSA, the corresponding rights to use it must be acquired. Information concerning the SSA licensing policy can be found from <http://www.cyber.ee/ssa/>.

Step 2: Create a new key file

Create a new key file (see “Creating the key file”, p. 20). The passphrase for protection of this file should be at least 11 characters long and contain digits, capitals, and lowercase letters: it makes brute-force braking of the passphrase practically impossible. To provide adequate security, the key itself should be at least 1024 bits long. The distinguished name of the Certification Authority should consist of country code, organization name, and organization unit name referring to the Certification Authority (for example: “*SSA User Manager*”). When setting the validity period of the Certification Authority certificate, you should consider that after certificate expiration you have to generate new private key and re-certificate all users; therefore, too short validity period is impractical. On the other hand, a very long period is insecure, as the attacker has more time for breaking the key contained in the certificate. Optimal validity period is two or three years.

Step 3: Load the public key of the Certification Authority to the User Manager

Open the certification authority public key management dialog (see “Managing the public keys of other certification authorities”, p. 27) and load the public key of the authority certifying the servers to the configuration file.

Step 4: Load to the User Manager the license allowing connecting to your server

Open the license management dialog (see “License management”, p. 29) and load to the configuration file the license allowing connecting to your server.

Step 5: Enter the defaults conforming the certification policy

Open the defaults dialog box (see “Changing the defaults”, p. 31) and fill in the defaults corresponding to the certification policy: length of the key to be generated, certificate validity period, user’s distinguished name, and the parameters of the session to be added to the configuration file.

Step 6: Save the key file

Save the key file you just created (see “Saving the key file under”, p. 22).

Step 7: Close the key file

Never leave your computer with the key file opened: an attacker having access to the computer can covertly get the Certification Authority private key. The key file has to be closed immediately after completing the work (see “Closing the key file”, p. 22).

Creating the user configuration file

During the creation of the configuration file needed for the SSA Lite Client, the SSA User Manager generates the client private key, issues the certificate to the client and creates the configuration file for storing these items; using this file gives the client access to the secure service. The configuration file will be created according to the agreement made with the client; this agreement gives to the user the right to use the service and obliges him not to transfer his configuration file to any third parties.

To create the configuration file, the administrator must know the following:

Parameter	Source
Length of the key to be generated to the user	Defined by certification policy. At least 1024 bits recommendable
User's distinguished name	Derived from the user data according to the certification policy
User's certificate validity period	Defined by the certification policy. For example, one year is suitable.
User session options	Specified by the SSA Server administrator. See also the SSA Server Guide.
Passphrase of the user configuration file	May be fixed; in this case the user must change it himself. Alternatively, a unique passphrase can be generated for each user.

The configuration file can be created and transferred in connection with making the service use agreement. When the agreement is made, the user is given a

floppy containing the SSA Lite Client and the configuration file created specially for this user; he gets also an envelop with the configuration file passphrase. If no formal agreement is made and the user prefers a remote delivery, the Lite Client and the configuration file can be sent him e.g. by email. Nevertheless, the configuration file passphrase must never be sent by mail! The passphrase must in any case be transferred through some channel external to the system.

User certification

Although the SSA User Manager allows to issue certificates conforming to the X.509 standard, it has not meant to be used as a universal Certification Authority. The User Manager still lacks of the certificate cancellation function required for a Certification Authority. However, the User Manager can be used for certification of the users having the full-featured SSA Client; those users can themselves generate the keys and certification requests. In this case, the certification policy has to be extended correspondingly.

The user can be certified only when he satisfies the requirements of the certification policy. On certification, the administrator must check compliance of the distinguished name in the user's certification request with the rules established by the certification policy. If necessary, the administrator can change the name or refuse to issue the certificate.

The administrator also has to check the public key of the user; the key must be of sufficient length and unmodified. If the user sends his certification request to the administrator e.g. by mail, the authenticity of the public key must always be checked using some channel external to the system.

After the certificate is created, it has to be sent to the user. It can be done e.g. by mail, as the correctness of the data contained in the certificate can be checked with the public key of the Certification Authority.

To use the certificate, the client needs the Certification Authority public key. Save the key in a file and send it to the user (see "Exporting the User Manager certificate", p. 31). The customer must always check the authenticity of the public key through the channels external to the system.

It should be noted that the certificate issued to the user gives him the same options for connecting to the SSA Server, as does the configuration file.

Security of the SSA User Manager

SSA successfully solves the data communication security problem between client and server, but to make the whole system secure, the SSA itself needs protection. In the following chapters some SSA User Manager security issues are discussed and solutions are given for security problems.

Theft of the private key

The private key is needed to create a secure communication channel for user authentication. Theft of this key gives the attacker the capability to covertly operate under the name of the user. Therefore it can be considered to be the most serious attack against the SSA. As a rule, to steal the private key one must have physical access to user's computer. The private key is stored in the SSA configuration file that is protected with user's passphrase from unauthorized reading. The ways to steal the key are following (with ascending complexity to do it):

1. The administrator has opened his SSA configuration file and left his computer unsecured. The attacker having physical access to the computer can change the password of the configuration file and save it to a floppy. To prevent this kind of attack, close your SSA configuration file before leaving your computer.
2. The attacker watches how the administrator enters his passphrase when he opens the configuration file. Later on, the attacker can copy the configuration file and open it with the stolen passphrase. There is a lot of ways to do such kind of spying: from peeping over the shoulder to dedicated covert cameras and various other spy equipment. As a rule, the attacker has to enter the user's room at least once. To copy the configuration file, he has typically to access user's computer physically, but it can be done through the network, if the computer is poorly configured. General security measures and frequent change of the passphrase (it sets narrower time limits for copying attempts) are the remedy for this attack.
3. The attacker copies the configuration file and tries to guess the passphrase with brute force. To copy the file, he has to have physical access to the administrator's computer. To guess the passphrase is more difficult when the passphrase is long. Each entered character will add about 6 bits of key information, if the passphrase contains digits, capitals, and small letters. As up to 56 bit keys can be broken nowadays (with rather great computing

effort), a passphrase must be at least 11 characters long. So, using sufficiently long passphrases will defeat this attack.

4. The attacker plants a Trojan in the administrator's computer. This is a very dangerous and versatile attack. The rather simple method is to store all administrator's keystrokes to a file from which the attacker can later read out the passphrase. In a more sophisticated case the attacker himself can connect to the computer and send the collected data covertly. Here the the remedies are safeguarding of the whole working environment and regular sanitation.

When a doubt arises about the key file private key being compromised, one must immediately report it to the server owner. The key must be cancelled and security measures must be applied to prevent any recurrence of the incident. After that, the new key file has to be generated and all the users have to be re-certified.

Necessary security measures

The SSA is no ultimate solution for all security problems. It is a component helping to build a secure system. To beat the threats described in this chapter, additional IT safeguards, organizational measures and physical measures must be applied. To sum up: the effective use of the SSA User Manager presupposes the following:

- a separate offline computer for the User Manager
- regular sanitation of this computer to prevent the Trojans
- using long passphrases and changing them frequently

Installation

The SSA User Manager works in the Windows 95, Windows NT 3.51 and Windows NT 4.0 environments. You need 2 MB free space on disk to install it. To install the software, start the **setup.exe** program on the first installation diskette.

The Setup program allows choosing the directory for the SSA User Manager to be installed; the program copies necessary program files and auxiliary files to this directory and creates icons for the installed programs.

Problems

I lost my key file passphrase. What can I do?

By creating the SSA system a great care was taken to prevent the possibility of reading the key file without knowing the passphrase. It is necessary for the protection of the Certification Authority key against attacks. No “master passphrase” exists which would allow reading all the key files.

When you lose your passphrase, you have to create the new key file and to re-certify all the users.

Cybernetica Software End User License Agreement

IMPORTANT - READ CAREFULLY!

WHEREAS

- this product contains computer software (Software), of which Küberneetika AS (Cybernetica) with its principal place of business at Akadeemia tee 21, 12618 Tallinn, Estonia, is the sole proprietor and that is protected by the laws of the Republic of Estonia and international copyright conventions,
- Cybernetica only grants you the right to use, copy and distribute the Software according to the terms and conditions set forth in this Cybernetica Software End User License Agreement (CS-EULA),
- you have been informed by Cybernetica about the foregoing and
- by using, copying or distributing the Software, you are expressing your free will;

NOW, THEREFORE, by using, copying or distributing the Software you are consenting to be bound by this CS-EULA and the legal Agreement between Cybernetica and you as the Customer or Customer's legal representative be entered into from the moment of your performing of any such act.

IF YOU DO NOT AGREE WITH THE TERMS AND CONDITIONS SET FORTH IN THIS AGREEMENT, CYBERNETICA GRANTS YOU NO RIGHT WHATSOEVER TO USE, COPY OR DISTRIBUTE THE SOFTWARE.

1. LICENSE

Cybernetica hereby grants the Customer a personal, permanent, nonexclusive and nontransferable license to use, copy and distribute the Software as follows:

(a) SSA Client, SSA Server, SSA User Manager (SSA Evaluation License)

Customer may use the Software during a 30-day evaluation period for the purpose of evaluating the Software and its suitability to Customer's further purposes. Upon the expiration of the evaluation period, Customer shall terminate the use of the Software unless he obtains an appropriate license from Cybernetica.

Customer may produce an unlimited number of identical copies of the Software and distribute such copies to any third parties provided that (a) no fee is charged for such copies or (b) the recipient has been informed of his right to obtain a free copy.

The object code of the Software carries the license.

(b) SSA Client, SSA Server, SSA User Manager (SSA Non-commercial / Academic License)

Customer may produce an unlimited number of copies of the Software and take such copies into non-commercial or academic use.

The license information file carries the license.

(c) SSA Server (SSA Public Server License)

Customer may use one copy of the Software for providing services to unlimited number of users without possibility of authenticating the users. Customer may also produce one additional backup copy of the Software.

The license information file carries the license.

(d) SSA Server (SSA Commercial Server License, N users)

Customer may use one copy of the Software for providing services to N users. Customer may also produce one additional backup copy of the Software.

Customer may and shall provide all users with a license information file that carries an appropriate SSA Connection License.

The license information file carries the license.

(e) SSA Client (SSA Connection License)

Customer may use one copy of the Software for establishing connections with the SSA Server specified in the license information file. Customer may also produce one additional backup copy of the Software.

The license information file carries the license.

(f) SSA Client (SSA Client License, N users)

Customer may produce and use N copies of the Software and license carrier. Customer may also produce one additional backup copy of the Software.

The license information file carries the license.

(g) SSA User Manager (SSA User Manager License)

Customer may use one copy of the Software. Customer may also produce one additional backup copy of the Software.

The license information file carries the license.

2. USING THE SOFTWARE

The Software is considered to be “in use” or “used” when its object code is loaded into the random access memory of any computing device. When used with multi-user operating systems, such number of copies are considered to be in use as there are “users”, according to the specifications of the operating system, who may directly interact with the Software.

2.1. NON-COMMERCIAL USE means using the Software by a physical person for his/her own private purposes that are not related to work or business.

2.2. ACADEMIC USE means using the Software by the members or students of an academic institution within the context of education process, and using the Software for providing publicly available services by a public library.

2.3. COMMERCIAL USE means using the Software in any situation not described in p.2.1 or p.2.2 above.

3. RESPONSIBILITY OF PARTIES

Because by entering into this agreement, Customer has accepted no obligation to pay license fees, Cybernetica disclaims and waives all warranties and liabilities, including, but not limited to, any liabilities for any indirect damages. **IN NO CASE SHALL CYBERNETICA'S ENTIRE LIABILITY EXCEED THE TOTAL AMOUNT OF LICENSE FEES PAID BY THE CUSTOMER TO CYBERNETICA OR ITS AGENTS.**

Should the Customer produce, use or distribute any copies of the Software notwithstanding with the terms and conditions set forth in this CS-EULA, he shall be held responsible to the maximum extent allowed by applicable law. **IN NO CASE SHALL CUSTOMER'S ENTIRE LIABILITY BE LESS THAN THE TOTAL AMOUNT OF LICENSE FEES UNPAID TO THE CYBERNETICA BECAUSE OF THE CUSTOMER'S BREACH OF THIS AGREEMENT.**

4. NEW VERSIONS OF SOFTWARE

Cybernetica may license new versions of the Software as “patches” or “updates”.

Any software licensed as a patch becomes an integral part of the original software. The original CS-EULA shall also govern the use of the “patch”.

If Cybernetica believes that the Customer holds at least one valid “free” license of “updateable software”, Cybernetica may license the Software to the Customer as an “update”. For the purposes of this paragraph, a license is considered “free” if it has never been used for obtaining “update” licenses. The original software becomes an integral part of the “update” and the new license agreement shall

govern the use of the resulting software. For the purposes of this paragraph, such “update” licenses are considered “free”.

Cybernetica may, at its own option, modify the list of “updateable” software. Cybernetica may require the Customer to prove that he holds a valid free license of “updateable” software.

5. RESERVED RIGHTS

Except as expressly specified in this agreement or required by applicable law, Customer shall not (a) separate the Software from the carrier of license, (b) rent, lease, sell or otherwise transfer the Software or his rights under this agreement to any third parties, or (c) decompile, disassemble, decipher or reverse engineer the Software. This paragraph survives the termination of this agreement.

6. TERM AND TERMINATION

This agreement is effective until termination. Customer may terminate the agreement by destroying all copies of the Software and license carriers.

7. JURISDICTION

This agreement shall be construed and governed in accordance with the laws of the Republic of Estonia and is considered to have been entered into on the territory of the Republic of Estonia. Customer shall attorn to the jurisdiction of the courts in the Republic of Estonia. This paragraph survives the termination of this agreement.

SSLeay copyright notice

Copyright (C) 1997 Eric Young (eay@cryptsoft.com)

All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are aheared to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Please note that MD2, MD5 and IDEA are publically available standards that contain sample implementations, I have re-coded them in my own way but there is nothing special about those implementations. The DES library is another mater :-).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)" The word 'cryptographic' can be left out if the rouines from the library being used are not cryptographic related :-).

4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement:
"This product includes software written by Tim Hudson
(tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The license and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution license [including the GNU Public License.]

The reason behind this being stated in this direct manner is past experience in code simply being copied and the attribution removed from it and then being distributed as part of other packages. This implementation was a non-trivial and unpaid effort.