Program

TokenUtility

Version

1.2

Description

TokenUtility is designed to configure a PKCS11 Cryptography token (such as the Chrysalis-ITS LUNA2 security token) for data signing and verification. Through a simple menu-based interface, it allows the user to:

- get information about the token
- initialize the token
- set and change PINs (passwords)
- generate key pairs
- export a public key as a certificate request
- save certificates on the token
- view all objects stored on the token

Both keys and certificates may be stored on the token. Keys are used primarily for data signing; certificates are used for email verification (e.g. by the AutoDRM program). Many objects may be stored on the token. Each stored object is assigned an ID number. The object type and ID number are used to specify a particular object on the token (e.g. to specify which private key should be used for signing data). Related objects will have the same object ID. For example, if you generate a public/private key pair, then obtain and store a certificate containing the public key, then all three objects (both keys and the certificate) will have the same object ID.

Running TokenUtility

Starting TokenUtility

TokenUtility runs within a command window under Solaris or Microsoft Windows NT. It is started manually from the command line, by typing:

tokenutility

It will open the first (main) menu and guide the user through the menu options.

Stopping TokenUtility

TokenUtility should always be shut down by typing "0 <CR>" in the main menu. This ensures that all sessions on the token are closed. Stopping the program any other way can cause a resource leak from the token.

Definition of User Interface Options

After starting TokenUtility program the following menu will appear:

PKCS#11 TOKEN UTILITIES v1.2

< 1 > Info
< 2 > Initialize Token
< 3 > Login as SO
< 4 > Login as User
< 0 > Quit

Type a number to select the corresponding option. Details for each option are given below:

Info

Displays a menu providing information about the available token slots and any security tokens installed in those slots. The Info menu has the following options:

Version Info	Gives information about the token manufacturer and library version.
Mechanism List	Prompts the user to select a slot, then gives the list of all supported algorithms for the token in the specified slot.
Slot Info	Prompts the user to select a slot, then gives the information about the selected slot.
Token Info	Prompts the user to select a slot, then gives the information about the token in that slot.
Slot List	Prompts the user to chose "All slots" or "Slots with Token", then lists the slots requested.
Return	Returns to the main menu.

Initialize Token

Allows initialization of the token. It is necessary to initialize a new token before use, to specify the Security Officer PIN, the token label, and the User PIN.

Warning: initializing a token will delete all keys, certificates and other data from the token.

Login as SO

Allows the user to login to the token as "Security Officer". The program will prompt the user for the Security Officer (SO) PIN. If the correct PIN is entered, the program will display the "Security Officer" menu with the following options:

Change PIN	Changes the security officer login PIN. Prompts the user to input the current SO PIN and the new SO PIN. If the current PIN is given correctly, the SO PIN will be set to the new value.
Initialize User PIN	Prompts the user to input the new User PIN and initializes the user area of the token. ***Warning: initializing the user PIN will delete all user data (including keys and certificates) from the token.***
Logout	Closes the SO session and returns to the main menu.

Login as User

Allows the user to login to the token as "User". The program will prompt the user for the User PIN. If the correct PIN is entered, the program will display the "User" menu, giving the following options:

Session Info	Gives the information about the current session.
Change PIN	Changes the user login PIN. Prompts the user to input the current User PIN and new User PIN. If the current PIN is given correctly, the User PIN will be set to the new value.
Delete Object	Deletes a user object from the token. Prompts the user for the Object ID (and type, if multiple objects have the same ID). The Object properties will be shown, and the user will be prompted to confirm the deletion.
List Objects	Lists summary information for all objects stored on the token. In a submenu, the user may choose to display keys, certificates, or all objects.
Display Object	Displays detailed information for a single object on the token. Prompts the user for the Object ID (and type, if multiple objects have the same ID).

Generate Key Pair Generates a DSA private / public key pair for data signing and verification. The keys will both be assigned the next available ID number, and labels provided by the user (or default label values).
Note: Only the DSA mechanism is supported by this program version.

Load Certificate Loads an X.509 certificate from file and stores it on the token. The user must specify the relative or absolute filename for the certificate; the certificate may be binary or base64 encoded.

If the certificate is self-signed, the user will be warned and asked whether or not to trust (and store) the certificate. If the certificate is not self-signed, the program will search the token for the issuer's certificate, and verify the new certificate using the issuer's public key. If the new certificate can be verified, it will be stored automatically on the token; otherwise it will not be stored. Unverified certificates will not be stored on the token.

Each certificate stored to the token will be given an object ID. If the public key contained in a certificate matches a public key on the token, it will get the same ID number as the key.

Generate Certificate Request Exports a public key in a standard format as a PKCS10 Certificate Request. The user is asked to specify the public key ID for which the certificate is requested, and to input country, organization, organization unit, and subject for the certificate. The program will save the certificate request in binary format (as **filename.crq**) and base64 format (as **filename.txt**) to a **filename** specified by the user.

Export Certificate Exports an X.509 certificate from the token and stores it as a file. The user must specify the object ID for the certificate to export, and a filename in which to store the certificate (the default file extension is "cer"). The certificate will be written to file in binary X.509 format.

Logout Closes the User session and returns to the main menu.

Quit

Selecting Quit in the main menu will exit TokenUtility program.

Typical Operation

Setting up a token for use with the AlphaStation or AutoDRM programs typically requires the following steps:

1. Initialize the token and set SO PIN, token label, and User PIN.
2. Login to the token as user, and generate a DSA key pair. You will use the private key to sign data (e.g. in AlphaStation) or email (e.g. in AutoDRM). Data receivers will use the public key to verify your signature.
3. Export the public key as a certificate request.
4. (optional) Obtain a certificate containing this public key by sending the certificate request to an appropriate certificate authority (CA). Store this on the token using the Load Certificate option (you must store the self-signed certificate of the CA first).
5. (optional) Store other trusted certificates on the token using the Load Certificate option. These certificates could be used, for example, to identify users authorized to request data via AutoDRM.
6. Send the public key (or certificate) to all proposed recipients of signed data or signed email.

7. Configure other programs which use the token (AlphaStation, AutoDRM), specifying the object ID of the private key to use for signing data.

Required Files

Windows: tokenutil.jar, cryst201.dll, pkcs11nmx.dll, and TokenUtility.bat.

Solaris: tokenutil.jar, libcrystoki2.so, libpkcs11nmx.so, and tokenutility.

Installation

1. Ensure that Java 1.2.2 or higher is installed on the machine.
2. Install the appropriate Chrysalis-ITS luna drivers for your operating system and PC-card reader.
3. Copy the required files (see above) to c:\nmx\user (Windows) or /nmx/bin (Solaris), and add that directory to your default path.

Trouble Shooting

TokenUtility is simple program, so that serious trouble shooting will not be necessary. In case of an error, the console window message will provide the tip for solving the problem.

Environment

TokenUtility is a Java program and should be run with Java 1.2.2 (or higher). Security token access libraries are available for Windows NT 4.0 and Solaris.

See Also

PKCS11 Cryptography Token Version 2.01

This document information

D:\Manuals\SoftwareReferenceManuals\TokenUtility\1.20\TokenUtility.lwp
Date created: 3/23/1998
Date last revised: 5/22/2001