



ProtectToolkit M

User Manual

© 2006-2008 SafeNet, Inc. All rights reserved.
Part Number: 002863-004 (Rev A)

All intellectual property is protected by copyright. All trademarks and product names used or referred to are the copyright of their respective owners. No part of this document may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, chemical, photocopy, recording or otherwise without the prior written permission of SafeNet.

1

SafeNet makes no representations or warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, SafeNet reserves the right to revise this publication and to make changes from time to time in the content hereof without the obligation upon SafeNet to notify any person or organization of any such revisions or changes.

SafeNet invites constructive comments on the contents of this document. These comments, together with your personal and/or company details, should be sent to the address below.

SafeNet, Inc.
4690 Millennium Drive
Belcamp, Maryland 21017
USA

Technical Support

If you encounter a problem while installing, registering or operating this product, please make sure that you have read the documentation. If you cannot resolve the issue, please contact your supplier or SafeNet support.

SafeNet support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between SafeNet and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Technical Support Contact Information:
Phone: 800-545-6608
Email: support@safenet-inc.com

Table of Contents

Chapter Overview	7
Chapter 1 Introduction	9
ProtectToolkit M Applications	9
The MSCAPI Model and ProtectToolkit M	9
<i>Cryptographic Service Providers</i>	9
<i>MSCAPI Implementation Using ProtectToolkit M</i>	9
<i>MSCAPI Keyset Model</i>	10
Further Documentation.....	11
Chapter 2 Installation/Uninstallation	13
Installation Requirements	13
Installation	13
Uninstallation.....	14
Chapter 3 Setup and Configuration	15
User Roles.....	15
Setup and Configuration Overview	15
Initial Configuration: Mandatory Steps	16
<i>Initializing the HSM</i>	16
<i>Setting Security Modes and Security Flags</i>	16
<i>Security Mode Descriptions</i>	17
<i>Security Mode Flag Descriptions</i>	19
<i>Allocating Keyset Space</i>	21
<i>Creating User Keysets</i>	21
<i>Commencing Normal Operation</i>	22
Configuration Options	22
<i>Registry Configuration</i>	22
<i>Error Log File Creation</i>	22
<i>Silent User Keyset Login</i>	23
<i>Work Load Distribution (WLD)</i>	23
Chapter 4 Administrative Tasks	25
Changing the Device Administrator Password.....	25
Allocating Keyset Space.....	25
De-allocating Keyset Space.....	26
Creating User Keysets	27
Deleting a Keyset.....	27
Setting the Adapter Transport Mode	28
Correcting Clock Drift.....	29
Viewing and Purging the HSM Event Log.....	29
Checking and Upgrading HSM Firmware.....	30
Tampering the HSM	31
Backing up a Keyset.....	32

Restoring a Keypset	34
Enabling Private Key Clear Export.....	35
Chapter 5 User Tasks.....	38
Creating Keypsets	38
Changing a Keypset Password.....	38
Adding a Key Container	39
Removing a Key Container.....	39
Generating a Key Pair.....	40
<i>Key Usage</i>	40
<i>Key Size</i>	40
Deleting a Key Pair	41
Displaying Key Pair Properties.....	41
Backing up and Restoring Keypsets	42
Chapter 6 Administration and User Utilities.....	43
Administration Utility	43
<i>Starting and Exiting the Administration Utility</i>	44
<i>User Interface</i>	44
<i>Menu Options</i>	47
<i>All Adapters Menu</i>	47
<i>Adapter Menu</i>	49
<i>Keypset Menu</i>	51
Keypset Management Utility	52
<i>Starting and Exiting the Keypset Management Utility</i>	52
<i>User Interface</i>	52
<i>Context Menus</i>	54
<i>Menu Options</i>	54
<i>Keypset Menu</i>	54
<i>Container Menu</i>	55
<i>Key Pair Menu</i>	55
CTKMU	57
CREATECERT Utility	60
Managing the Host Interface Master Key (HIMK).....	60
CTHIMK Utility	61
Chapter 7 Troubleshooting.....	63
Known Issues	63
<i>Self Signed Certificates and Netscape</i>	63
<i>Keypset Restore Error</i>	63
<i>Microsoft Internet Explorer (IE5)</i>	63
<i>Session Exists Error</i>	64
<i>Duplicate Container or Key Instances</i>	65
<i>Application Error</i>	66
Appendix A Integration with Microsoft CA	67
Setting Up a CA with ProtectToolkit M	67

Certificate Template Support for Safenet CSPs (Windows 2003)	70
CA Replication (Key Backup and Recovery).....	72
<i>Backing Up Keys for a CA Installation to Smart Cards</i>	72
<i>Replicating a CA Using Keys Restored from Backup Smart Cards</i>	72
Private Key Archiving and Recovery	74
<i>Private Key Archiving Example</i>	74
<i>Private Key Recovery Example</i>	84
Appendix B Integration With IIS.....	89
Creating a Certificate Overview	89
Creating a Certificate Using IIS	89
Creating a Certificate Using the Microsoft CA server	90
Creating a Certificate Using the createcert utility	95
Installing a Certificate for use with IIS	95
Appendix C PKCS#11 Attributes.....	101
Appendix D KAA Professional Smart Card Reader	103
Hexadecimal to Decimal Conversion Table.....	104
Appendix E Work Load Distribution (WLD)	105
Introduction	105
<i>Benefits from Using WLD</i>	105
<i>WLD Architecture</i>	105
The ProtectToolkit C Model.....	106
<i>Administration Objects</i>	107
<i>User Roles</i>	107
WLD Limitations.....	108
Setting Up a WLD System	108
WLD System Management	111
<i>Turning off WLD</i>	111
<i>Changing Configuration Under WLD</i>	112
WLD Technical Details	112
<i>Trust Management</i>	112
<i>Token Replication</i>	113
Appendix F Registry Configuration.....	115
Disclaimer.....	115
ptkcRuntime.....	115
CryptokiPath.....	115
Debug Level	116
Safenet RSA Full Cryptographic Provider	116
Safenet RSA SChannel Cryptographic Provider.....	117
Default RSA SChannel Cryptographic Provider Type.....	117
Default RSA Full Cryptographic Provider Type.....	117
Silent User Keyset Login Password	118

Appendix G Event Log Error Types119
Glossary121
Index123

Chapter Overview

Chapter 1 gives an overview of the ProtectToolkit M system and discusses some of the terms and concepts found throughout this manual.

Chapter 2 details the ProtectToolkit M software installation and uninstall procedure. This section is intended for the system administrator responsible for installation and removing software from the host system.

Chapter 3 describes how the ProtectToolkit M system is initially configured. The System Administrator or person responsible for the installation process should read this chapter.

Chapter 4 discusses the various operational tasks a ProtectToolkit M administrator may be required to perform. This chapter is applicable to the administrator.

Chapter 5 discusses the various operational tasks a ProtectToolkit M user may be required to perform. This chapter is applicable to the administrator and user.

Chapter 6 contains the reference sections for the tools provided with ProtectToolkit M. This chapter is applicable to both the administrator and normal user.

Chapter 7 contains a troubleshooting section, which details some of the common problems that could occur during setup or normal operation. This chapter is applicable to both the administrator and normal user.

Appendix A contains details on integrating ProtectToolkit M with Microsoft CA.

Appendix B contains details on integrating ProtectToolkit M with Microsoft IIS.

Appendix C contains PKCS #11 attribute information.

Appendix D covers the KAAAN professional smart card reader.

Appendix E contains work load distribution (WLD) details.

Appendix F contains Windows Registry configuration entries.

Appendix G details the type of entries which could occur in the HSM event log.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

This chapter gives an overview of the ProtectToolkit M and shows how the SafeNet components and terminology apply in the Microsoft Cryptographic API environment.

ProtectToolkit M Applications

With ProtectToolkit M installed, applications that call the Microsoft Cryptographic API (MSCAPI) can make use of the secure key storage and high speed cryptographic processing that SafeNet hardware security modules (HSMs) have to offer.

The Microsoft Cryptographic API (MSCAPI) provides services that allow for the easy incorporation of security into certain applications such as web based SSL processes.

Microsoft Certification Authority (MSCA) and Internet Information Services (IIS) (a Microsoft web server) use the MSCAPI and they are therefore examples of applications that may be integrated with ProtectToolkit M to implement hardware key storage and high speed cryptographic processing. A MSCA may store CA keys on a HSM, while IIS may use HSM key storage when establishing secure socket layer (SSL) communication.

The MSCAPI Model and ProtectToolkit M

Cryptographic Service Providers

ProtectToolkit M is implemented as a Microsoft Cryptographic Service Provider (CSP).

A CSP is a plug-in cryptographic module that integrates with Microsoft Windows and provides the underlying key storage and security operations for the Microsoft Cryptographic API (MSCAPI). The architecture of the MSCAPI supports the development of non-Microsoft CSPs such as ProtectToolkit M.

ProtectToolkit M includes both “RSA Full” and “RSA SChannel” cryptographic service providers. These can be used instead of the corresponding Microsoft CSPs to provide hardware based key storage and RSA encryption.

MSCAPI Implementation Using ProtectToolkit M

Figure 1 shows how SafeNet HSMs can be utilized as part of a MSCAPI system using ProtectToolkit M as a CSP.

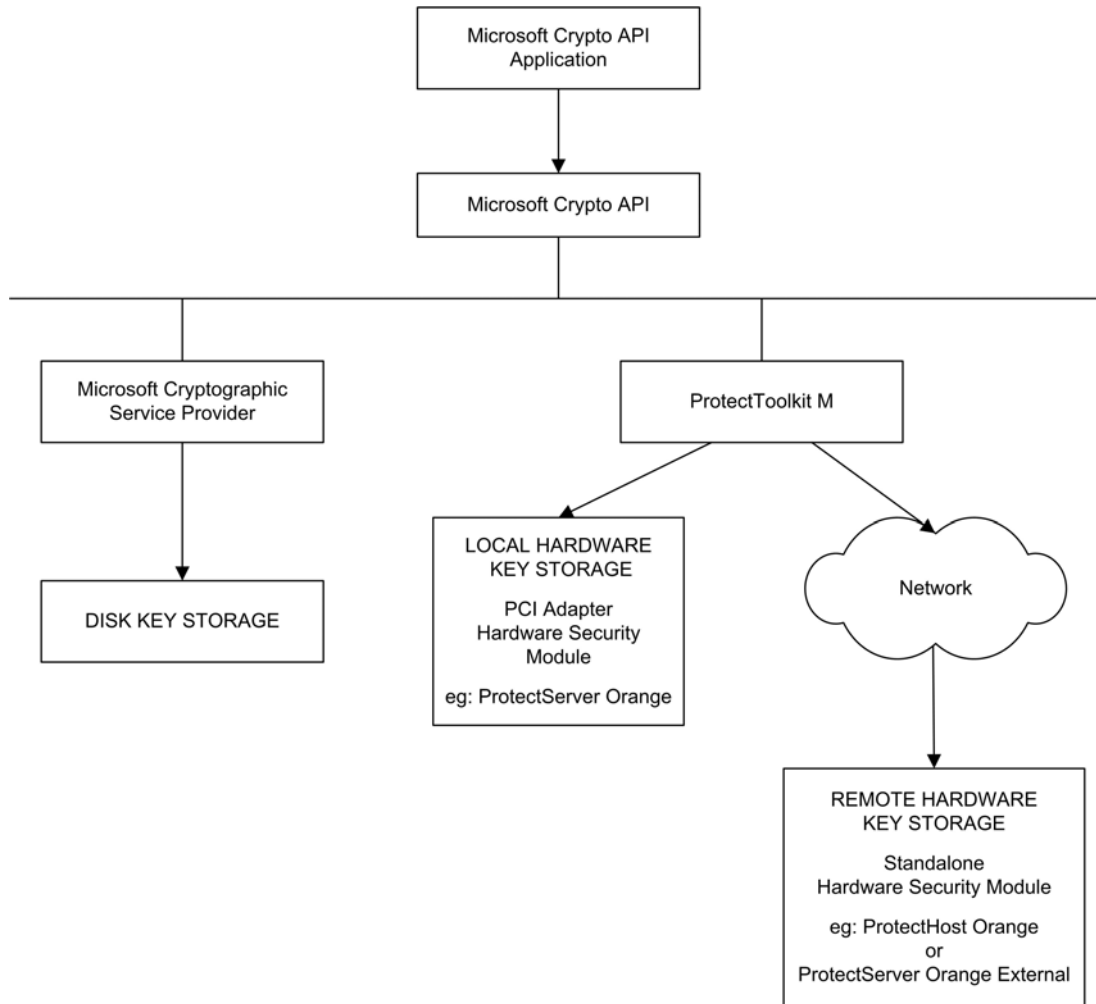
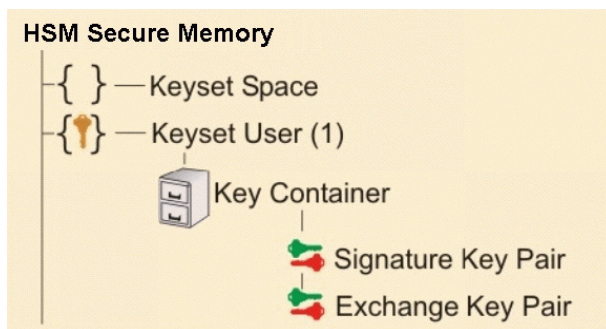


Figure 1 – ProtectToolkit M model

MSCAPI Keyset Model

Within MSCAPI (and hence ProtectToolkit M), *key pairs* are held within a *key container*, which in turn is stored within a *keyset*.



Each user requiring processing support from the ProtectToolkit M system will need a user keyset containing a key container. Key containers may contain up to 2 key pairs, a signature key pair and an exchange key pair.

Apart from this, there are two keysets which the ProtectToolkit M system requires for its internal processes. These are the SYSTEM keyset and the MACHINE keyset, which are visible to all system users. ProtectToolkit M cannot operate without either of these and will automatically create either set if they are not present or deleted. Shared keys (accessible by more than one user), such as those generated automatically when Microsoft CA is installed, will also be stored in one of these keysets when using a ProtectToolkit M CSP. Generally these shared keys are stored in the MACHINE keyset.

The physical storage location for each keyset is CSP dependant. By default, Microsoft CSPs store keys to disk, in user profiles. When using the “Safenet RSA Full” or “Safenet RSA SChannel” CSPs, all keys are secured by ProtectToolkit M within SafeNet hardware security modules (HSMs).

Further Documentation

The following reference material should be considered in addition to this user manual:

- ProtectServer Orange External Installation Guide
- ProtectServer Orange Installation Manual
- ProtectHost Orange Installation Guide
- Microsoft documentation on cryptographic service providers. See their web site.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Installation/Uninstallation

Installation Requirements

Before beginning with the ProtectToolkit M installation, please confirm that your system meets the following minimum requirements:

- IBM Compatible computer with a minimum of 128 MB RAM and 5 MB free hard disk space.
- Windows 2000 or Windows 2003 Server operating system.
- Microsoft IIS (Internet Information Services) should be installed, configured and working if integration with IIS is desired.
- A SafeNet hardware security module (HSM) must be available. Either an adapter can be installed in the local machine or a device may be made available via a network connection.

Installation

Preparation

If a previous version of ProtectToolkit M is installed, uninstall it prior to installing the newer software.

To install the ProtectToolkit M software:

- Open Windows Explorer and execute the file **ETmpprt.exe** found in the \Win32\ptkm directory of the installation CD.
- Follow the on-screen instructions to complete the installation. During installation you will be required to:
 - choose the directory where the software will be installed and to
 - nominate either a locally connected or network connected HSM as the cryptographic service provider.

Please take the time to read the **Readme.txt** file located in the root directory of the installation CD for last minute changes or update information concerning ProtectToolkit M.

Following the installation, please continue to *Chapter 3* for details on how to configure and setup the ProtectToolkit M product.

Note Regarding SSL Implementation

When using ProtectToolkit M as an SSL provider, the Dynamically Linked Library for the HSM provider (ethsm.dll) must be placed in the %WINDIR%\system32 directory. This is because the SSL provider runs in the context of the lsass.exe process, which has only the %WINDIR% and %WINDIR%\system32 directories in its search path, so the DLL will not be found if it is not copied into either of these directories.

You will find the file required in either the *network_hsm* or *pci_hsm* directory for network and PCI HSMs respectively, under:

\Program files\Safenet\ProtectToolkit M

Uninstallation

To uninstall the ProtectToolkit M software:

- If the key information stored on the HSM is no longer required, tamper the HSM in order to destroy it. See the *Tampering the Adapter* section for further instructions if needed.
- Select *Uninstall* from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -UnInstall. The uninstallation will proceed automatically.

Chapter 3

Setup and Configuration

User Roles

Prior to performing any configuration, it is important to understand the different ProtectToolkit M roles available and to determine which type of role an individual will assume.

There are two defined roles available. These are:

- i) A ProtectToolkit M device administrator; and
- ii) A ProtectToolkit M user

These roles are described below.

Device Administrator

The device administrator is responsible for tasks that involve management of the associated HSM and those applicable to ProtectToolkit M administration. Those assigned to this role are also responsible for performing backup and restore operations for MACHINE and SYSTEM keysets and allocation of space for user keysets.

User

A ProtectToolkit M user is responsible for the creation and management of their own keyset stored within a HSM. This includes responsibility for backup and restore of their own keyset, the key container and associated key pairs.

Setup and Configuration Overview

This chapter assumes that you have performed the instructions detailed in *Chapter 2*.

After installing ProtectToolkit M it is necessary for the device administrator to:

- initialize the HSM
- set the security mode
- allocate keyset space
- create user keysets (This is optional as users may also create their own keysets.)
- setup work load distribution (WLD) if required

After the device administrator has performed the above steps then users will typically need to undertake the following tasks:

- create keysets
- add containers to keysets

- generate key pairs in containers

To perform these tasks follow the procedures described in this chapter.

Initial Configuration: Mandatory Steps

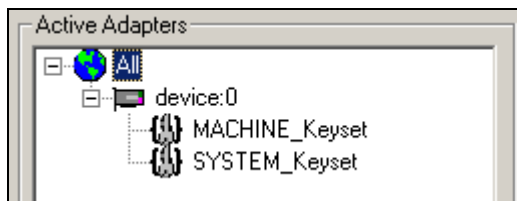
Initializing the HSM

ProtectToolkit M uses a hardware encryption HSM to store sensitive key information. The HSM will therefore need to be initialized prior to use, or following a tamper event. A tamper event occurs, for example, if the HSM detects that someone is trying to get inside the cover. It will then automatically erase it's secure memory.

HSM management tasks can only be performed by a device administrator. During HSM initialization, the device administrator password is set, and the HSM clock is synchronized with the host.

To initialize the HSM:

- Open the administration utility by selecting Start-Programs-Safenet-ProtectToolkit M -Administration. A prompt displays for the device administrator password.
- Enter the password in both the *Admin Password* and *Confirmation* fields and, if required, check *Keep Password For Session*. For added security, when the *Keep Password For Session* check box is clear, password entry is required to complete every task. For convenience, when the check box is checked, password entry will not be required again until after the utility is closed and reopened.
- Press *OK*. The MACHINE and SYSTEM keysets are created. The *Administration Utility* dialog box displays showing MACHINE and SYSTEM keyset icons under *Active Adapters* as shown below.



Setting Security Modes and Security Flags

The security mode for the HSM is chosen by specifying the value of a number of security flags. These flags affect both the services available to the various users of the system as well as specific security features of the HSM. These flags may be specified individually to set a custom security mode, however it is recommended that a standard security mode be used. When a standard security mode is selected the flags are assigned values automatically to meet the requirements for that mode.

Note: The security mode should be set prior to commencing normal operation. It is recommended that the impact of any proposed security mode change be carefully

assessed prior to implementation. Implementation should only occur after the proposed change has been assessed to be suitable\applicable.

To set a security mode:

1. If not already open, execute the *Administration Utility* from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M-Administration.
2. From the *All Adapters* menu select *Set Security Flags*. The *Set Security Flags – All Devices* dialog box displays.



3. Either:
 - change flag values to those required (See below for security mode flag descriptions) in the *Security Mode Flags* group box or
 - click a security mode button in the *Security Modes* group box to set a standard security mode. See below for security mode descriptions.
4. Click *OK* and enter the administration password if prompted. A *success* confirmation message displays.
5. Click *OK* to return to the *Administration Utility* dialog box.

Security Mode Descriptions

This section describes the security modes that can be selected from the *Security Modes* group box in the *Set Security Flags – All Devices* dialog box.

Set All and Clear All Modes

- Upon clicking the *Set All* button all security flags are set.
- Upon clicking the *Clear All* button all security flags are cleared.

FIPS 140 Mode

FIPS 140 mode refers to the security flag settings required to comply with the Federal Information Processing Standards (FIPS) 140 standard. A FIPS certification assures that a product is capable of operating under the highest levels of security and credibility.

It is important to note that the product can function outside the scope of this accreditation. Therefore, care should be exercised to ensure that the correct configuration is set if this level of FIPS secure operation is required.

The security mode flags set in FIPS 140 mode are shown in the table below.

Restricted Mode

The *Restricted Mode* security setting is a compromise between performance and security. If *Restricted Mode* is selected, then the HSM will require all users to identify themselves before cryptographic services are available. This mode also inhibits any clear PINs or sensitive key material from passing through the HSM's PCI bus interface but each individual request to the HSM does not need to be signed.

The security mode flags set in Restricted Mode are shown in the table below.

Security Mode Preconfigured Flag Settings

When the *FIPS* or *Restricted* security mode buttons are clicked in the *Set Security – All Devices* dialog box the status of the flags is changed as shown in the table below (default values). Those settings marked with an asterisk (*) are mandatory in order to implement the requirements for the mode concerned. Additional flags, marked with a plus (+), can be changed if required. See below for security mode flag descriptions.

	FIPS 140 Mode	Restricted Mode
Tamper Before Upgrade.	Set*	Cleared ⁺
No Public Cryptography	Set*	Set*
Entrust Compliant	Cleared*	Cleared*
No Clear PINs	Set*	Set*
Authentication Protection	Set*	Cleared*
Lock Security Mode	Set*	Set*
Increased Security Mode	Cleared ⁺	Cleared ⁺
Only Allow FIPS Approved Algorithms.	Set*	Cleared*
Full Secure Messaging Encryption	Cleared ⁺	Cleared ⁺

	FIPS 140 Mode	Restricted Mode
Full Secure Messaging Signing	Cleared ⁺	Cleared [*]

Security Mode Flag Descriptions

Tamper Before Upgrade

When this flag is set the HSM will automatically perform a soft tamper (erase all internal secure memory) as part of a firmware upgrade, FM download or FM disable operation.

No Public Cryptography

When this flag is set no user can perform a cryptographic operation without having authenticated themselves.

When this flag is set, each token in the system will have the PKCS #11 CKF_LOGIN_REQUIRED flag set to indicate that applications must authenticate before operations. However, this security mode flag does not affect the Admin token which always requires authentication for use.

Note: This setting does not impede the ability to perform RSA or other public key processing. The setting implies that crypto services cannot be performed by unauthenticated users.

Entrust Compliant

When this flag is set Entrust Compliant Mode is operational. This is used to ensure compatibility with the Entrust range of applications. These applications require a specific security profile in order to operate correctly.

No Clear PINs

When this flag is set no user PINs or other sensitive information may be passed across the host interface in an unencrypted form. This method will enable secure messaging encryption between applications and the HSM. It will also disable certain functions that would otherwise result in the clear transmission of sensitive data. This flag will also not allow any keys to be created with the attribute CKA_SENSITIVE=FALSE.

Authentication Protection

This flag, when set, enforces secure messaging authentication between applications and the HSM. Each request to the HSM must be digitally signed and will be verified by the HSM. The key used for this signing process is derived from a key shared by the HSM and host application as well as the user PIN.

By setting this flag the applications will operate in a more secure manner, however this will also have the effect of decreasing HSM performance due to the increased operations required in signing each request and response message.

Lock Security Mode

The *Lock Security Mode* flag, when set, disables further modification of the security mode flag settings. Once set, this flag (or any other security mode flag) cannot be modified. Once in this mode a new security mode can only be implemented after a tamper operation is performed.

Increased Security Level

The *Increased Security Level* flag, when set, disables the mechanism CKM_EXTRACT_KEY_FROM_KEY and also does not allow the CKA_MODIFIABLE attribute to be changed from False to True.

Only Allow FIPS Approved Algorithms

The *Only allow FIPS Approved Algorithms* flag, when set, only enables the mechanisms that use FIPS Approved Algorithms. The algorithms that are **not FIPS approved** are MD2, MD5, RIPE, CAST, IDEA, RC2, RC4 and RC5.

Full Secure Messaging Encryption

The *Full Secure Messaging Encryption* flag, when set, is similar to the *No Clear PINs* flag except that every message is encrypted in both directions between the application and the HSM. The key used for the message encryption is generated using the PKCS#3 Diffie-Hellman Key Agreement Standard.

This flag only performs the two-way encryption when using the ProtectToolkit M client library in the client/server mode over TCP/IP.

By enabling this setting the applications will operate in a more secure manner, however this will also have the effect of decreasing HSM performance due to the increased operations required in encrypting and decrypting each request and response message.

Full Secure Messaging Signing

The *Full Secure Messaging Encryption* flag, when set, is similar to the *Authentication Protection* flag except that every request in both directions between the application and the HSM is digitally signed and verified. The key used for the message encryption is generated using the PKCS#3 Diffie-Hellman Key Agreement Standard.

This flag only performs the two-way encryption when using the ProtectToolkit M client library in the client/server mode over TCP/IP. By enabling this setting the applications will operate in a more secure manner, however this will also have the effect of decreasing HSM performance due to the increased operations required in signing and verifying each request and response message.

Allocating Keyset Space

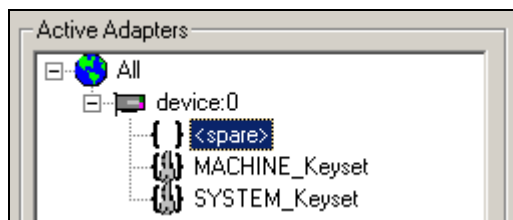
In order for applications to use the key storage facilities offered by ProtectToolkit M, it is necessary to allocate keyset space on the HSM. Enough space should be allocated to cover the number of users who will require key storage.

Allocation of keyset space is the responsibility of the device administrator and is performed using the ProtectToolkit M administration utility.

NOTE: It is important to determine how many key sets may be needed over time and to allocate sufficient space so that additional key sets can be created without the need for a server shutdown once the system is operational.

To allocate keyset space:

- If not already open, execute the administration utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M-Administration.
- Select the device on which to create keyset space from the *Active Adapters* list.
- Open the *Adapter* menu and choose *Allocate Space*.
- The administration utility will now prompt for the device administrator password.
- Following correct password entry, the new keyset space will be displayed under the device as shown below.



Creating User Keysets

User keysets are required for each individual that will use the ProtectToolkit M system. The keysets are stored on the HSM in available keyset spaces. This means that in order to create a user keyset, a free keyset space must be available (see above).

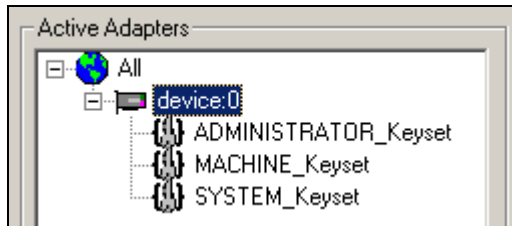
Creating user keysets is the responsibility of the device administrator and is performed using the ProtectToolkit M administration utility.

NOTE: Ideally all keysets required should be created prior to the system becoming operational. If this is not feasible, then it is important to estimate how many key sets may be needed over time and to allocate sufficient space so that additional key sets can be created without the need for a server shutdown once the system is operational.

To create a user keyset:

- If not already open, execute the administration utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M-Administration.

- Select the spare keyset space on which to create the keyset from the *Active Adapters* list.
- Open the *Keyset* menu and choose *Create Keyset*.
- The administration utility will now prompt for the *Keyset Name* and the *Keyset Password*. Enter the required information into the fields provided and press *OK* to create the new keyset. Note that the name of the keyset should match with the user login name. The new keyset displays under the device as shown below.



Commencing Normal Operation

Following the above steps, ProtectToolkit M is ready for use. Additional configuration may be required in certain circumstances, as covered in the [Configuration Options](#) section below.

There are also various operational tasks that may need to be performed by the device administrator or user during runtime usage. These tasks are covered in the following chapters.

Configuration Options

Registry Configuration

Entries made in the Windows registry during the installation of ProtectToolkit M are documented in *Appendix E*. These may be amended by expert users if required. Generally, the default values will not need to be changed. The exceptions are the *Debug Level* and *User Keyset Password* entries used to control error log file creation and silent user keyset login respectively. See the sections below for further information.

Error Log File Creation

The *Debug Level* registry key is used to control error log file creation. By default the value of this key is set so that no error log file is produced. Should it be necessary to create an error log file see the *Debug Level* entry in *Appendix F* for further information on the options available.

Silent User Keyset Login

Whilst access to the Machine and System keysets is open, access to a User keyset requires authentication.

Typically, User keyset access authentication is achieved by prompting the user for a password via a dialog box at the time access is requested. In some situations this is not convenient/permissible. For these situations, silent user keyset login is available.

To activate silent User keyset login:

Add the following value to the Windows registry:

```
HKEY_CURRENT_USER/Software/Safenet/ProtectToolkit M/  
UserKeysetPassword=<password>
```

where:

<password> is the clear text password for the User keyset.

Since this value is located in the *Current Users* registry hive (which is only accessible/visible when a user authenticates themselves to the Windows operating system) there is no security risk even though the password is stored in the clear.

Work Load Distribution (WLD)

If required, more than one hardware security module (HSM) can be used to implement Work Load Distribution (WLD).

WLD allows work to be balanced across a system by transferring units of work among HSM processing modules during execution. The demand placed on any particular processing module is thereby reduced. This results in an increase in the overall throughput of processing tasks for the system as a whole. Utilization of multiple HSMs in this way also provides redundancy in that if a HSM goes down the work will be shared amongst the remaining operational HSMs automatically.

For further information, including implementation and maintenance instructions, refer to *Appendix E Work Load Distribution (WLD)*.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Administrative Tasks

This chapter describes the operational procedures a device administrator may perform during normal ProtectToolkit M operation.

Changing the Device Administrator Password

At certain stages it may be necessary to change the device administrator password. The device administrator may perform a password change at any stage and on any token.

Changing the device administrator password may only be performed by the device administrator, using the ProtectToolkit M administration utility.

To change the device administrator password:

- If not already open, execute the administration utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Administration.
- Select the device on which to change the device administrator password from the **Active Adapters** list.
- Open the **Adapter** menu and choose **Change Device Admin Password**.
- The administration utility will now prompt for the current and new device administrator password. Enter the required information into the fields provided and press OK to change the password.

Note: Any existing backups of the MACHINE and SYSTEM keysets will no longer be useful following a device administrator password change, because the backup key is generated from the password. It is therefore recommended that new backups are taken after changing the password.

Allocating Keypset Space

When there is a requirement for additional user keysets, the system will need to be configured for additional keyset space. The number of allocated keyset spaces determines how many separate ProtectToolkit M users and hence keysets, the system can have.

Allocation of keyset space is the responsibility of the device administrator and is performed using the ProtectToolkit M administration utility.

To allocate keyset space:

- If not already open, execute the administration utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Administration.

- Select the device on which to create keyset space from the **Active Adapters** list.
- Open the **Adapter** menu and choose **Allocate Space**.
- The administration utility will now prompt for the device administrator password.
- Following correct password entry, the new keyset space will be displayed under the device.

Note: Additional keyset space cannot be allocated if an application is currently using ProtectToolkit M. See *Session Exists Error* in *Chapter 7 Troubleshooting* for further information.

To check if an application has any open sessions to ProtectToolkit M, check the value shown next to **Application Count** in the **System** section of the administration utility. This will need to be “1”, and the **Total Session Count** must be “0” in order for keyset de-allocation to succeed.

Note: If the value of **Application Count** is shown as “UNAVAILABLE”, your HSM firmware doesn’t support live application counting. In such a case, it is advisable to upgrade the HSM firmware to the latest version. Please refer to the section entitled *Checking and Upgrading HSM Firmware*, found later in this chapter for full details.

De-allocating Keyset Space

If there are keyset spaces which are not likely to be used, it is good practice to de-allocate spare spaces from the HSM in order to prevent memory exhaustion or invalid use.

De-allocation of keyset space is the responsibility of the device administrator and is performed using the ProtectToolkit M administration utility.

To de-allocate keyset space:

- If not already open, execute the administration utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Administration.
- Select any spare space from the **Active Adapters** list.
- Open the **Keyset** menu and choose **Deallocate Space**.
- The administration utility will now prompt for the device administrator password.
- Following correct password entry, the spare keyset space will be removed from the list of displayed keysets.

Note: Keyset space cannot be de-allocated if an application is currently using ProtectToolkit M. See *Session Exists Error* in *Chapter 7 Troubleshooting* for further information.

To check if an application has any open sessions to ProtectToolkit M, check the value shown next to **Application Count** in the **System** section of the administration utility. This will need to be “1”, and the **Total Session Count** must be “0” in order for keyset de-allocation to succeed.

Note: If the value of **Application Count** is shown as “UNAVAILABLE”, your HSM firmware doesn’t support live application counting. In such a case, it is advisable to upgrade the HSM firmware to the latest version. Please refer to the section entitled *Checking and Upgrading HSM Firmware*, found later in this chapter for full details.

Creating User Keysets

In order to create a new keyset for a specific user, you will firstly have to make sure that there is spare keyset space available on the HSM. This can be done by opening the ProtectToolkit M administration utility.

Should no spare space be available, you will have to allocate additional keyset space on the HSM. For details please refer to the appropriate section above.

Note: Users can create keysets for themselves once space exists.

To create a user keyset:

- If not already open, execute the administration utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Administration.
- Select the spare keyset space on which to create the keyset from the **Active Adapters** list.
- Open the **Keyset** menu and choose **Create Keyset**.
- The administration utility will now prompt for the Keyset Name and the Keyset Password. Enter the required information into the fields provided and press OK to create the new keyset.
Note: The name of the keyset should match with the user login name.
- The new keyset will be displayed under the device.

Deleting a Keyset

Deleting user keysets is the responsibility of the device administrator and is performed using the ProtectToolkit M administration utility.

To delete a user keyset:

- If not already open, execute the administration utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Administration.
- Select the desired keyset from the **Active Adapters** list.
- Open the **Keyset** menu and choose **Delete Keyset**.
- The administration utility will now prompt for the device administrator Password. Enter the required information into the fields provided and press OK to delete the selected keyset. Prior to deletion, the administration utility will prompt for confirmation that deletion is the requested operation.
- The keyset will be removed from the displayed keysets under the selected device.

Note: A keyset cannot be deleted if an application is currently using ProtectToolkit M. See *Session Exists Error* in *Chapter 7 Troubleshooting* for further information.

To check if an application has any open sessions to ProtectToolkit M, check the value shown next to **Application Count** in the **System** section of the administration utility. This will need to be “1”, and the **Total Session Count** must be “0” in order for the keyset deletion to succeed.

Note: If the value of **Application Count** is shown as “UNAVAILABLE”, your HSM firmware doesn’t support live application counting. In such a case, it is advisable to upgrade the HSM firmware to the latest version. Please refer to the section entitled *Checking and Upgrading HSM Firmware*, found later in this chapter for full details.

Setting the Adapter Transport Mode

The adapter transport mode is a facility that allows an adapter HSM to be removed from the host system PCI bus without causing a tamper condition. A tamper will remove all sensitive material from the adapter including the adapter configuration, all keys and certificates.

Setting the adapter transport mode is the responsibility of the device administrator and is performed using the ProtectToolkit M administration utility.

To set the adapter transport mode:

- If not already open, execute the administration utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Administration.
- Select the desired adapter from the **Active Adapters** list.
- Open the **Adapter** menu and choose **Set Transport Mode**.
- The device administrator is now prompted to choose one of three possible transport modes:
 - None - To be applied when adapter is installed and configured. This mode will tamper the adapter if removed from the PCI Bus.
 - Single - Adapter will not be tampered after removal from the PCI bus. Adapter will automatically change to “None” Transport Mode the next time the adapter is reset or power is removed and restored.
 - Continuous - Adapter will not be tampered by being removed from the PCI bus.
- The administration utility will now prompt for the device administrator password.

Note: The transport mode does not disable the tamper response mechanism entirely. Any attempt to physically attack the adapter will still result in a tamper response.

Correcting Clock Drift

Due to host system and HSM timing differences, such as clock drifts, it may become necessary, at certain stages, to adjust the internal time on the HSM.

Note that the HSM clock value cannot be specified directly. It is only possible to synchronize the HSM clock with the host system clock.

Synchronizing the HSM clock is the responsibility of the device administrator and is performed using the ProtectToolkit M administration utility.

To adjust the HSM clock:

- If not already open, execute the administration utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Administration.
- Select the desired HSM from the **Active Adapters** list.
- Open the **Adapter** menu and choose **Sync Clock**.
- The administration utility will now prompt for the device administrator password. Correct entry of the password will result in clock synchronization.

Viewing and Purging the HSM Event Log

SafeNet HSMs maintain event logs in order to provide a means of tracking serious hardware or consistent operational faults. It is the device administrator's task to view and purge HSM event log data. For full details on what the event log stores and how to interpret its data, please refer to *Appendix G*.

When the HSM event log is full, the HSM will no longer store new event records and will need to be purged.

Note: The HSM event log cannot be purged until it is full.

To view the HSM event log:

- If not already open, execute the administration utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Administration.
- Select the desired HSM from the **Active Adapters** list.
- Open the **Adapter** menu and choose **View Event Log**.
- The administration utility will now prompt for the device administrator password. Correct entry of the password will result in the event log being displayed.
- The event log is shown as a series of pages. If there are more than one page of event log entries, the operator can navigate through the pages via the “first”, “prev”, “next”, “last” buttons.

To purge the event log:

- If not already open, execute the administration utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Administration.
- Select the desired HSM from the **Active Adapters** list.
- Open the **Adapter** menu and choose **Purge Event Log**.
- The administration utility will now prompt for the device administrator password. Correct entry of the password will result in the event log being purged.
- Note that the event log can also be purged via the **View Event Log** dialog by pressing the "Purge" button.

Checking and Upgrading HSM Firmware

The ProtectToolkit M firmware that operates on the HSM can be upgraded to newer versions. This facility will only allow the HSM to be upgraded to firmware versions that have been digitally signed by SafeNet.

The firmware update package, formerly supplied identically on both the Ptk-C and Ptk-M CDs, is now available only on the Ptk-C software CD, along with a specific Update instruction document. The instructions in this section are generic in nature, intended only to show the scope of the operation. The authoritative, detailed instructions are always in the Update document that accompanies the update package. The Ptk-C CD is always available from SafeNet Customer Support.

Prior to performing a firmware upgrade, the firmware upgrade file should be checked to confirm that it is a valid SafeNet upgrade file.

Note: Depending on the security policy in place, the HSM may perform a soft-tamper before the upgrade process is executed. This tamper will erase all key and configuration data on the HSM. Prior to performing a firmware upgrade, ensure that you have performed the following:

- All important user data and keys have been backed up
- The current HSM configuration has been noted
- All applications using the HSM have been closed – this may require some services to be stopped (e.g. Certificate Services, IIS)

Upgrading the HSM firmware is the responsibility of the device administrator and is performed using the ProtectToolkit M administration utility.

To check the firmware upgrade file:

- If not already open, execute the administration utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Administration.
- Select the desired HSM from the **Active Adapters** list.
- Open the **Adapter** menu and choose **Check Firmware File**.

- The administration utility will now prompt for the location of the firmware upgrade file.
- The utility will show if the file is validated, or corrupt.

To upgrade the HSM firmware:

- If not already open, execute the administration utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Administration.
- Select the desired HSM from the **Active Adapters** list.
- Open the **Adapter** menu and choose **Upgrade Firmware**.
- The administration utility will now prompt for the location of the firmware upgrade file.
- If the file is validated, the utility will prompt for the device administrator password. Correct password entry will proceed with the firmware upgrade.

Note: During the firmware upgrade the administration utility will appear to stop functioning. This is normal since firmware upgrades can take up to 40 seconds to complete. If the utility does not respond after a number of minutes, shut down your system and re-boot the computer. If problems persist, please contact SafeNet technical support.

Tampering the HSM

The tampering of the HSM may be necessary at the end of its lifecycle or any other security sensitive event that requires all stored data to be immediately destroyed.

A tamper formats the secure memory of the HSM and thereby erases all configuration and key data.

Due to the highly destructive nature of this action, tampering the HSM is the responsibility of the device administrator and is performed using the ProtectToolkit M administration utility. Note that this action also requires that all sessions have been closed and that no user is accessing the HSM.

To tamper the HSM:

- If not already open, execute the administration utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Administration.
- Select the desired HSM from the **Active Adapters** list.
- Open the **Adapter** menu and choose **Tamper**.
- The administration utility will now prompt for the device administrator password. Correct entry of the password will show a final confirmation dialog to ensure that this is the desired course of action.
- Press “Yes” to tamper the HSM, or “No” to Cancel.

Note that the above action cannot tamper the HSM whilst other applications are active. Due to this, the administration utility will indicate if the tamper operation was successful. A white cross on a red background shown next to the selected HSM indicates that the device is tampered.

Backing up a Keyset

Individual, HSM stored keysets can be backed up to a secure disk file or one or more smart cards. Backed up keysets can then be restored in the event of a tamper to the HSM or if the keysets are otherwise lost.

Note that users are responsible for backing up their own keysets and the ProtectToolkit M device administrator is responsible for backing up the MACHINE and SYSTEM keysets.

A triple-DES *BackupKey* is used to encrypt each keyset prior to storage on a smart card. A different *BackupKey* is automatically created for each keyset when the keysets are created but these keys are not visible under normal ProtectToolkit M operation. A *BackupKey* for a keyset is derived from a combination of the password used to secure that particular keyset and the keyset name. In the case of the MACHINE and SYSTEM keysets, the device administrator's password and the keyset name are used to derive the key. Thus to restore a keyset that was previously backed up, the same password and keyset name must be used.

Keyset backup and restore is accomplished with the command line utility *ctkmu*. Please refer to *Chapter 6* for the complete *ctkmu* reference.

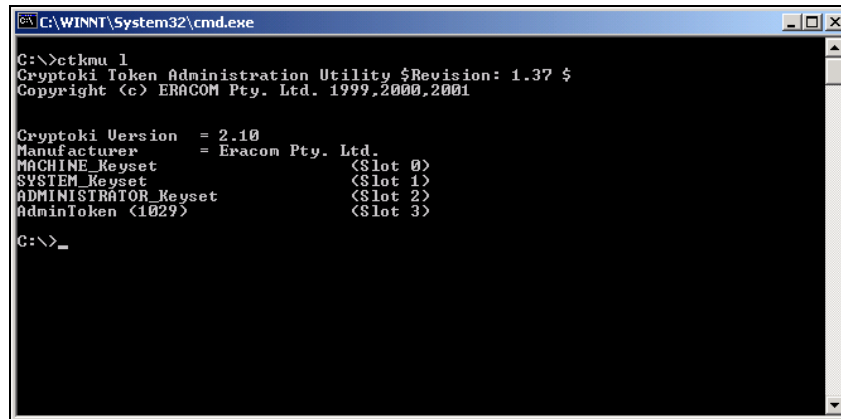
Preparation

Prior to attempting a keyset backup, please ensure that you have:

- a valid keyset that can be backed up
- if backing up to smart cards, a smart card reader connected to the HSM, and
- sufficient smart cards or disk space to back up the required data.

Procedure

1. Obtain a listing of all keysets by executing *ctkmu* with the **l** option from a command prompt.



```

C:\WINNT\System32\cmd.exe
C:\>ctkmu 1
Cryptoki Token Administration Utility $Revision: 1.37 $
Copyright (c) ERACOM Pty. Ltd. 1999,2000,2001

Cryptoki Version = 2.10
Manufacturer = Eracom Pty. Ltd.
MACHINE_Keyset (Slot 0)
SYSTEM_Keyset (Slot 1)
ADMINISTRATOR_Keyset (Slot 2)
AdminToken (1029) (Slot 3)

C:\>_

```

Figure 2 – Getting Keyset Information For Backup

- Record the slot number for the keyset you wish to backup.
- To backup a keyset to a file, from a command prompt, type the following, substituting the *n* with the slot number of the keyset to backup and *fileName* with name of the file to backup to:

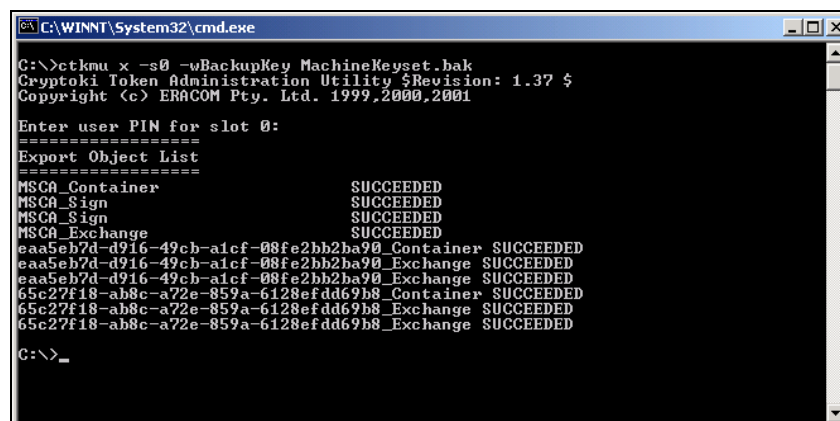
```
ctkmu x -sn -wBackupKey fileName
```

Note: When backing up the *MACHINE_Keyset* or the *SYSTEM_Keyset*, enter the default value, *password* as the user password. The device administrator's password and the keyset name will be used to derive the *BackupKey* in these instances.

Example

In the example below, the keyset on slot 0 will be securely encrypted using the key *BackupKey* (created from the user password for the keyset and the keyset name) and backed up to the disk file named *MachineKeyset.bak*. This operation will prompt for the user password for the keyset.

```
ctkmu x -s0 -wBackupKey MachineKeyset.bak
```



```

C:\WINNT\System32\cmd.exe
C:\>ctkmu x -s0 -wBackupKey MachineKeyset.bak
Cryptoki Token Administration Utility $Revision: 1.37 $
Copyright (c) ERACOM Pty. Ltd. 1999,2000,2001

Enter user PIN for slot 0:
=====
Export Object List
=====
MSCA_Container          SUCCEEDED
MSCA_Sign               SUCCEEDED
MSCA_Sign               SUCCEEDED
MSCA_Exchange          SUCCEEDED
eaa5eb7d-d916-49cb-alc-f-08fe2bb2ba90_Container SUCCEEDED
eaa5eb7d-d916-49cb-alc-f-08fe2bb2ba90_Exchange SUCCEEDED
eaa5eb7d-d916-49cb-alc-f-08fe2bb2ba90_Exchange SUCCEEDED
65c27f18-ab8c-a72e-859a-6128efdd69b8_Container SUCCEEDED
65c27f18-ab8c-a72e-859a-6128efdd69b8_Exchange SUCCEEDED
65c27f18-ab8c-a72e-859a-6128efdd69b8_Exchange SUCCEEDED

C:\>_

```

Figure 3 - Backup Keyset

Restoring a Keypset

Precautions

- Extreme care should be taken to ensure that keys which are being restored DO NOT already exist on the ProtectToolkit M system. A restore operation DOES NOT replace existing keys, but will restore a second instance of the same key pair. If you have accidentally created multiple instances of the same key pair, ProtectToolkit M will mark the affected keyset as being invalid. Please refer to the troubleshooting section in *Chapter 7* for details on how to address this type of problem.
- To restore a key that was previously backed up, the same password and keyset name must be used.

Procedure

1. Create a new keyset with the same name and password as the original. See the section [Creating User Keysets](#) in [Chapter 3, Setup and Configuration](#) for the procedure.
2. To restore a keyset from file, from a command prompt type the following, substituting the slot number of the keyset to restore for *n* and the name of the file containing the keyset for *fileName*.

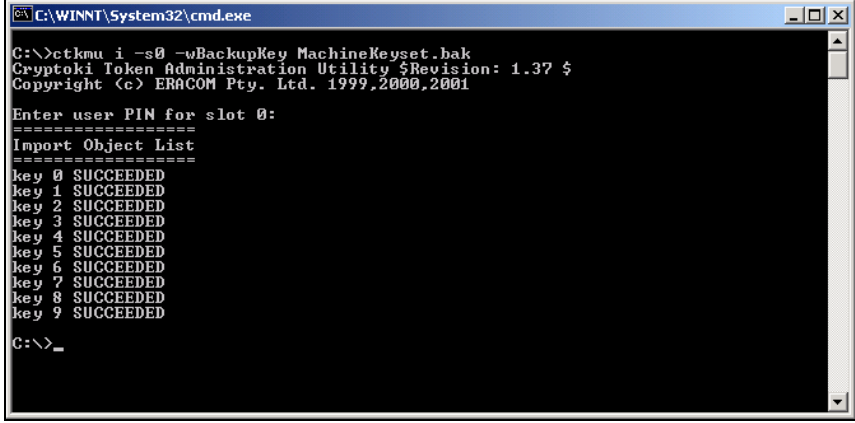
```
ctkmu i -sn -wBackupKey fileName
```

Note: When restoring the *MACHINE_Keypset* or the *SYSTEM_Keypset*, enter the default value, *password* as the user password. The same device administrator's password as was used to create the backup must also be used.

Example

The example below will import a keyset to the token in slot 0 from a disk file named *MachineKeypset.bak* and unwrap, or decrypt, the data with the key which has a label of *BackupKey*. This operation will prompt for the user password for the keyset.

```
ctkmu i -s0 -wBackupKey MachineKeypset.bak
```



```
C:\WINNT\System32\cmd.exe
C:\>ctkmu i -s0 -wBackupKey MachineKeyset.bak
Cryptoki Token Administration Utility $Revision: 1.37 $
Copyright (c) ERACOM Pty. Ltd. 1999,2000,2001

Enter user PIN for slot 0:
=====
Import Object List
=====
key 0 SUCCEEDED
key 1 SUCCEEDED
key 2 SUCCEEDED
key 3 SUCCEEDED
key 4 SUCCEEDED
key 5 SUCCEEDED
key 6 SUCCEEDED
key 7 SUCCEEDED
key 8 SUCCEEDED
key 9 SUCCEEDED

C:\>_
```

Figure 4 - Restore Keyset

Enabling Private Key Clear Export

In order to support the Windows 2003 key archival process, it must be possible to obtain the value of the private key in the clear in the host machine. See ***Error! Reference source not found. Error! Reference source not found.*** for further information on how this functionality is used.

Since having the value of the private key in the clear in the host machine is a security risk, the ability to do this is controlled by the *Allow Clear Export of Private Keys* flag. This is a “secure configuration item”.

A *secure configuration item* is a configuration item which is open for reading but requires authentication for writing. Such configuration items are stored on the HSM protected by the password of the device administrator.

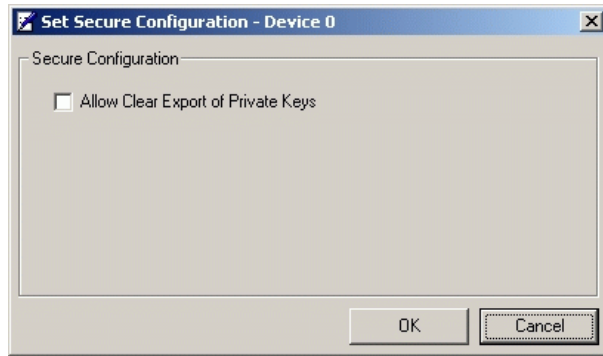
If *Allow Clear Export of Private Keys* is set to True, then using the Microsoft Crypto API (MSCAPI) it is possible to obtain the value of a private key in the clear (this will allow the key archival process to succeed).

If *Allow Clear Export of Private Keys* is set to False, then any requests to obtain the value of a private key in the clear are denied (this will cause the key archival process to fail).

The ProtectToolkit M *Administration Utility* is used to manipulate the value of the *Allow Clear Export of Private Keys* flag.

To set or clear the *Allow Clear Export of Private Keys* flag:

- If not already open, execute the administration utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Administration.
- Select the desired HSM from the *Active Adapters* list.
- Open the *Adapter* menu and choose *Set Secure Configuration*. The *Set Secure Configuration* dialog box displays.



- Set or clear the *Allow Clear Export of Private Keys* flag as required, then click *OK* to action the change.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

User Tasks

This chapter describes the operational procedures a User may perform during normal ProtectToolkit M operation.

Creating Keysets

In order to create a new keyset you will firstly have to make sure that there is spare keyset space available on the HSM. This can be done by opening the ProtectToolkit M Administration Utility.

Should no spare space be available, an administrator will have to allocate additional keyset space on the HSM. For details please refer to the previous chapter.

To create a keyset:

- If not already open, execute the administration utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Administration.
- Select the spare keyset space on which to create the keyset from the **Active Adapters** list.
- Open the **Keyset** menu and choose **Create Keyset**.
- The administration utility will now prompt for the Keyset Name and the Keyset Password. Enter the required information into the fields provided and press OK to create the new keyset.
Note: The name of the keyset should match with the user login name.
- The new keyset will be displayed under the device.

Changing a Keyset Password

A keyset password may need periodic change, which the user may perform at any stage.

Changing a user keyset password is performed by the keyset owner, using the ProtectToolkit M keyset management utility.

To change the keyset password:

- If not already open, execute the keyset management utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Keyset Management.
- Select the keyset for which to change the password from the displayed list.
- Open the **Keyset** menu and choose **Change Password**.

- The user is prompted for the current and new keyset password. Enter the required information into the fields provided and press OK to change the password.

Note: Any existing keyset backups will no longer be useful following a keyset password change, because the backup key is generated from the password. It is therefore recommended that new backups are taken after changing the password.

Adding a Key Container

Key containers are created within a user's keyset, so that the keyset can hold key pairs.

Adding a key container is performed by the keyset owner, using the ProtectToolkit M keyset management utility.

To add a key container:

- If not already open, execute the keyset management utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Keyset Management.
- Select the keyset to which you want to add a container, from the displayed list.
- Open the **Keyset** menu and choose **Add Container**.
- The user is prompted for the keyset password and key container name. Enter the required information into the fields provided and press OK to create the key container.

Removing a Key Container

Key containers which are no longer required or hold obsolete key pairs may be removed from a keyset.

Removing a key container is performed by the keyset owner, using the ProtectToolkit M keyset management utility.

To remove a key container:

- If not already open, execute the keyset management utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Keyset Management.
- Select the keyset container which you wish to remove.
- Open the **Keyset** menu and choose **Remove**.
- The user is prompted for the keyset password and confirmation that the container removal is the required action. Press OK to remove the key container.

Generating a Key Pair

Key pairs are used by Crypto API to encrypt or sign data. There are two types of key pairs, and either of these must be created inside a key container. For details on how to create key containers please refer to the appropriate section above.

Generating a key pair is performed by the keyset owner, using the ProtectToolkit M keyset management utility.

To generate a key pair:

- If not already open, execute the keyset management utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Keyset Management.
- Select the keyset container in which to generate a key pair.
- Open the **Container** menu and choose **Generate Key Pair**.
- The user is prompted to enter the keyset password. Correct password entry will display the generate key pair dialog.
- The generate key pair dialog will prompt for the key usage and key size.
- Choose **Exchange** or **Sign** depending on the required key pair usage.
- Select a **Key Size** from the drop-down list.
- Check the **Exportable** checkbox if you want to be able to backup this key pair.
- Press “OK” to generate the key pair.

Key Usage

Key pairs that are generated using the keyset management utility have one of two usage attributes. These are :

- Exchange - This type of key pair is used to encrypt session keys for the user during normal ProtectToolkit M operation.
- Sign - This type of key pair is used to create digital signatures for the user during normal ProtectToolkit M operation.

Each user will generally require both types of keys within their particular keyset.

Key Size

Key size is an important consideration when using encryption as a security measure. When discussing key size, the value is given as a “bit” length, referring to how many digits are represented in the key value.

As a general guideline, the higher the “bit” length, the longer the key length and hence the greater the security offered by the encryption process. On the other hand, choosing a larger key size value is detrimental to the speed of the actual encryption process because of the increased mathematics that are involved.

Key size is a compromise of speed versus security. It is up to the individual to choose which of those attributes are more important for their particular application. Generally, a key size of 1024 bits offers good speed and security for most applications.

Deleting a Key Pair

Deleting a key pair is performed by the keyset owner, using the ProtectToolkit M keyset management utility.

To delete a key pair:

- If not already open, execute the keyset management utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Keyset Management.
- Select the key pair which you wish to delete.
- Open the **KeyPair** menu and choose **Delete**.
- The user is prompted to enter the keyset password. Correct password entry will delete the selected key pair.

Displaying Key Pair Properties

Key pair properties can be displayed by any user of the ProtectToolkit M keyset management utility.

To display the properties of a key pair:

- If not already open, execute the keyset management utility from the Start menu by selecting Start-Programs-Safenet-ProtectToolkit M -Keyset Management.
- Select the key pair for which to display its properties.
- Open the **KeyPair** menu and choose **Properties**.

Information shown includes the following:

- **Keyset** Shows the name of the keyset on which the selected key pair resides.
- **Container** Displays the key container name, in which the selected key pair resides.
- **Usage** Shows the key usage attribute of the selected key pair. This value will either be “EXCHANGE” or “SIGN”.
- **Size** Shows the key size for the selected key pair.
- **Private Key Held** This indicates if the private key for the selected key pair is present as part of the key pair. Since it is possible to

import a public key only, this value will either be “TRUE” or “FALSE”.

- Exportable Indicates if the selected key pair can be backed up.

Backing up and Restoring Keysets

A user is responsible for backing up their own keysets. The procedures involved in backing up and restoring key pairs or keysets are detailed in *Chapter 4*.

Keyset backup or restore operations should not be attempted without thorough knowledge of the procedure and the possible consequences of incorrect actions. It is strongly advised that the device administrator is consulted prior to performing a keyset backup or restore operation.

Chapter 6

Administration and User Utilities

Administration Utility

The administration utility is designed exclusively for use by the ProtectToolkit M device administrator, and allows for the following operations:

- Initialization of HSMs
- Synchronization of HSM clock with the system clock
- Setting of the adapter transport mode
- Setting security flags
- Changing of the ProtectToolkit M device administrator password
- Upgrade of the HSM firmware
- Allocation of keyset space
- De-allocation of keyset space
- Creation of keysets
- Deletion of keysets
- Viewing the HSM event log
- Purging the HSM event log
- Tampering the HSM

Please note that this section is only intended as a reference for the administration utility. When performing administrative tasks, the reader is strongly advised to refer to *Chapter 4* for full details regarding each task.

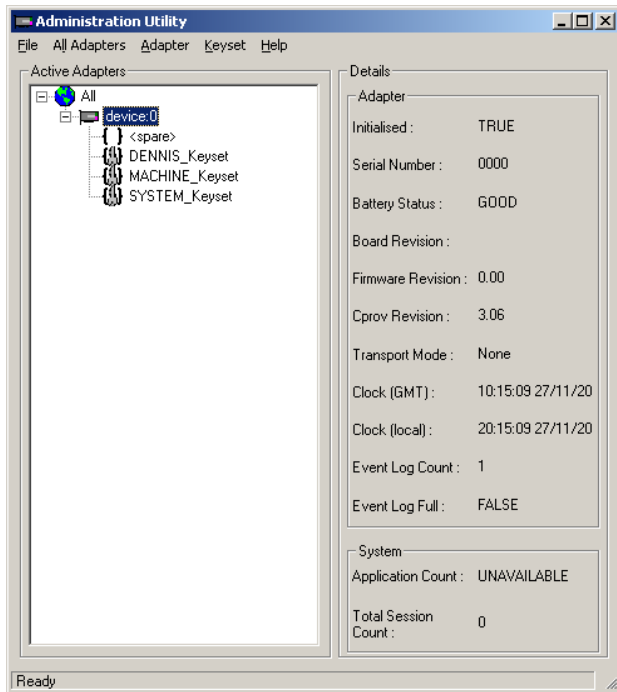


Figure 5 – Administration Utility User Interface.

Starting and Exiting the Administration Utility

To start the administration utility select *Start-Programs-Safenet-ProtectToolkit M - Administration*. After an initial splash screen, the main user interface is shown.

To exit from the utility, select **Exit** from the **File** menu.

User Interface

The administration utility is presented as a Graphic User Interface (GUI), which is divided into three main areas. These are:

1. The menu bar – which is shown along the top of the utility. All available utility commands can be activated via these menus
2. The **Active Adapters** display pane – shows all hardware HSMs found on the host system and their associated keysets. These are represented as a hierarchical tree view, with HSMs being the highest member and keysets or keyset spaces shown beneath each HSM.
3. The **Details** pane – is broken up into two sub-groups and displays the following information.

Adapter Details	
Initialized	Shows if the currently selected HSM has been initialized. Values are either TRUE or FALSE.

Adapter Details	
Serial Number	The serial number of the selected HSM.
Battery Status	Indicates the charge of the onboard battery of the selected HSM. This may either be GOOD or LOW. If this indicates LOW, the selected HSM may not be able to retain stored key information in the event of a system power failure. The HSM should be returned to your nearest SafeNet service centre for battery replacement.
Board Revision	Shows the hardware revision of the currently selected HSM.
Firmware Revision	Shows the firmware revision of the currently selected HSM.
Cprov Revision	Shows the revision of ProtectToolkit C found on the currently selected HSM. This is a software component which forms part of the HSM firmware. This value may need to be quoted when contacting SafeNet support.
Transport Mode	Shows the transport mode which is set for the currently selected HSM. This value will be either “None”, “Single”, or “Continuous”. Refer below for details.
Clock (GMT)	Shows date and time (GMT) on the currently selected HSM.
Clock (local)	Shows the local date and time on the currently selected HSM.
Event Log Count	Gives a total for the number of event log entries on the currently selected HSM.
Event Log Full	Shows if the event log is full and needs purging. This value may be either “FALSE”, indicating that the log is not full or “TRUE” which indicates the opposite.

System Details	
Application Count	Shows the number of applications which are currently accessing the ProtectToolkit M system. This value may show as “UNAVAILABLE” which denotes that the firmware on the selected HSM does not support application counting.
Total Session Count	Shows the number of open sessions to the ProtectToolkit M system.

Password Entry Dialog Boxes

Most actions performed within the administration utility will require entry of the device administrator password (see *Figure 6*). The device administrator password is case sensitive and may consist of any alphanumeric characters, of between 4 and 32 characters in length.



Figure 6 – Device administrator password entry dialog box.

Keep Password Feature

The password dialog has a facility to remember the device administrator password whilst the administration utility remains open. This eliminates having to repeatedly enter the password when performing multiple operations.

To enable this facility, check the box next to “Keep Password For Session” within the device administrator password dialog. Correct password entry followed by the “OK” button will enable the “Keep Password” feature.

Note: When this feature is enabled, care should be taken not to leave the administration utility unattended. To ensure that unauthorized people do not obtain management access to HSMs, close the administration utility once you have finished with your assigned task.

Keyboard Shortcuts

All available menu choices may be selected via a series of keyboard shortcuts. The menu bar can be selected by pressing the “Alt” key. Commands may then be selected by pressing the first unique letter of the required command. For example: “Alt” followed by “A” will open the Adapter menu.

There are also a number of key combination shortcuts which will immediately activate a command. To activate these shortcuts you must hold down the first key, whilst pressing the second key. These are as follows:

- “CTRL”+“I” = Initialize HSM
- “CTRL”+“A” = Allocate Keyset Space
- “CTRL”+“V” = View Event Log
- “CTRL”+“P” = Change Admin Password
- “CTRL”+“U” = Upgrade Firmware
- “CTRL”+“T” = Tamper HSM
- “CTRL”+“D” = De-allocate Space
- “CTRL”+“K” = Create Keyset

Context Menus

Right-clicking on an item, which has been selected from the **Active Adapters** display pane, will bring up a context menu showing available commands specific to that item.

For details about these commands, please refer to the appropriate section below.

Menu Options

The following details each available menu option which can be accessed from the administration utility.

All Adapters Menu

The **All Adapters** menu is only of value if there is more than one HSM installed on the system. This menu allows the device administrator to affect all installed HSMs with a single command. The following actions can be performed via this menu:

- Initialize HSMs
- Synchronize Clocks
- Set Transport Modes
- Set Security Flags
- Set Secure configuration
- Change Admin Passwords
- Upgrade Firmware
- Tamper All HSMs

Initialize Adapters

This menu option performs the initialization for all uninitialized HSMs found on the system. Since only initialized HSMs can store key information, this command is used to initialize a new system after installation.

Synchronize Clocks

This menu option:

Performs the synchronization of all HSMs found on the system with the value of the host system clock.

Set Transport Modes

This menu option is used to set the adapter transport mode for all adapters found on the system. The adapter transport mode is a facility that allows an to be removed from the host system PCI bus without causing a tamper condition. A tamper will remove all sensitive material from the adapter including the adapter configuration, all keys and certificates.

The device administrator is prompted to choose one of three possible transport modes:

- None - To be applied when adapter is installed and configured. This mode will tamper the adapter if removed from the PCI Bus.
- Single - Adapter will not be tampered after removal from the PCI bus. Adapter will automatically change to “None” Transport Mode the next time the adapter is reset or power is removed and restored.
- Continuous - Adapter will not be tampered by being removed from the PCI bus.

Set Security flags

This menu option allows the setting of a security mode using security flags. These flags affect both the services available to the various users of the system as well as specific security features of the HSM. The flags may be specified individually to set a custom security mode, however it is recommended that a standard security mode be used. When a standard security mode is selected the flags are assigned values automatically to meet the requirements for that mode. For further information see the section ***Error! Reference source not found.*** in *Chapter 4, Administrative Tasks* for further information.

Set Secure configuration

This menu option allows the setting of security configuration items.

A *secure configuration item* is a configuration item which is open for reading but requires authentication for writing. Such configuration items are stored on the HSM protected by the password of the device administrator.

A single item is currently supported *Allow Clear Export of Private Keys*. See *Enabling Private Key Clear Export* in *Chapter 4, Administrative Tasks* for further information.

Set Admin Passwords

This menu option changes the current device administrator password to a new password value for all HSMs on the host system. The device administrator is prompted to enter the current password as well as enter and confirm the new password.

Upgrade Firmware

This menu option performs an upgrade of the firmware for all HSMs on the host system.

The device administrator is prompted to enter the path to the firmware update file.

Tamper All Adapters

This menu option causes a tamper of all HSMs found on the system. A tamper formats the secure memory of the HSM and thereby erases all configuration and key data.

Adapter Menu

The **Adapters** menu is used to perform the following administrative actions on a selected HSM:

- Allocate Space
- Change Admin Password
- Check Firmware File
- Initialize
- Purge Event Log
- Set Secure Configuration
- Set Security Flags
- Set Transport Mode
- Synchronize Clock
- Tamper
- Upgrade Firmware
- View Event Log

Initialize

This menu option initializes the selected HSM. Since only initialized HSMs can store key information, this command is used to initialize a new or tampered HSM.

Allocate Space

This menu option allocates one keyset space on the selected HSM. Keyset space is required in order to create user keysets.

View Event Log

This menu option opens the event log viewer. The dialog shows event log entries in chronological time order, with the most current event showing last. The “first”, “prev”, “next” and “last” buttons can be used to navigate through the event details, should there be more than one page of entries.

If the event log is full, it can be purged by pressing the “Purge” button.

Note: The “Purge” button is disabled until the event log is full.

Purge Event Log

This menu option purges the event log.

Note: This menu option is disabled until the event log is full.

Synchronize Clock

This menu option synchronizes the clock of the selected HSM with the host system clock.

Set Transport Mode

This menu option is used to set the adapter transport mode for the selected adapter. The adapter transport mode is a facility that allows the adapter to be removed from the host system PCI bus without causing a tamper condition. A tamper will remove all sensitive material from the adapter including the adapter configuration, all keys and certificates.

The device administrator is prompted to choose one of three possible transport modes:

None - To be applied when adapter is installed and configured. This mode will tamper the adapter if removed from the PCI Bus.

Single - Adapter will not be tampered after removal from the PCI bus. Adapter will automatically change Transport Mode to “None”, the next time the adapter is reset or power is removed and restored.

Continuous - Adapter will not be tampered by being removed from the PCI bus.

Change Admin Password

This menu option changes the current device administrator password to a new password value for the currently selected HSM. The device administrator is prompted to enter the current password as well as enter and confirm the new password.

Upgrade Firmware

This menu option performs an upgrade of the firmware for the selected HSM. The device administrator is prompted to enter the path to the firmware update file.

Check Firmware Upgrade File

This menu option is used to check the validity of a firmware upgrade file. The device administrator is prompted to enter the path to the firmware update file.

Tamper

This menu option causes a tamper of the selected HSM. A tamper formats the secure memory of the HSM and thereby erases all configuration and key data.

Set Security flags

This menu option allows the setting of a security mode using security flags. These flags affect both the services available to the various users of the system as well as specific security features of the HSM. The flags may be specified individually to set a custom security mode, however it is recommended that a standard security mode be used. When a standard security mode is selected the flags are assigned values automatically to meet the requirements for that mode. For further information see the section ***Error! Reference source not found.*** in *Chapter 4, Administrative Tasks* for further information.

Set Secure configuration

This menu option allows the setting of security configuration items.

A *secure configuration item* is a configuration item which is open for reading but requires authentication for writing. Such configuration items are stored on the HSM protected by the password of the device administrator.

A single item is currently supported *Allow Clear Export of Private Keys*. See *Enabling Private Key Clear Export* in *Chapter 4, Administrative Tasks* for further information.

Keyset Menu

The **Keyset** menu is used to perform the following administrative actions on a selected keyset:

- Delete
- Create Keyset
- Deallocate

Delete

This menu option will delete the currently selected keyset.

Create Keyset

This menu option creates a keyset within the currently selected keyset space.

Deallocate

This menu option removes the selected spare keyset space.

Keyset Management Utility

The keyset management utility is designed for the ProtectToolkit M user, and allows for the following operations:

- Create keysets
- Generate key pairs
- Delete key pairs
- Show key pair properties
- Add key containers
- Remove key containers
- Change passwords

Please note that this section is only intended as a reference for the keyset management utility. When performing administrative tasks, the reader is strongly advised to refer to *Chapter 5* for full details regarding each task.

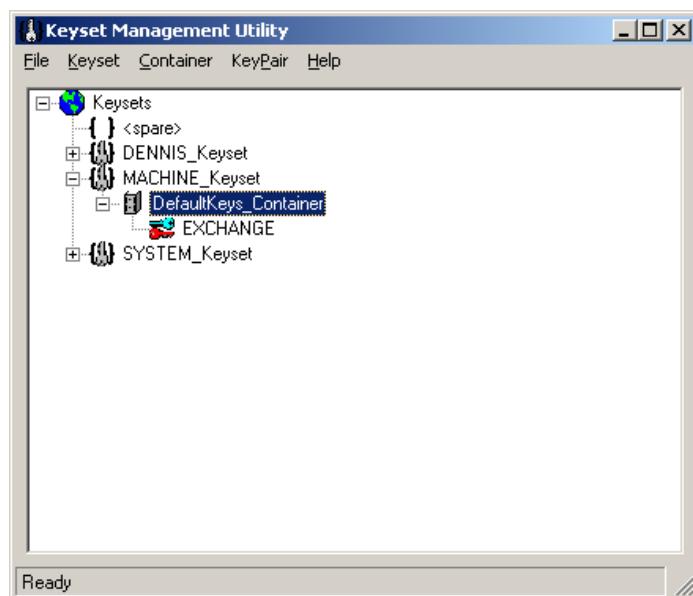


Figure 7 – Keyset Management Utility User Interface.

Starting and Exiting the Keyset Management Utility

To start the keyset management utility select *Start-Programs-Safenet-ProtectToolkit M* and click on *Keyset Management*. After an initial splash screen, the main user interface is shown (see *Figure 7*).

To exit from the utility, select **Exit** from the **File** menu.

User Interface

The administration utility is presented as a Graphic User Interface (GUI), which is divided into two main areas. These are:

1. The menu bar – which is shown along the top of the utility. All available utility commands can be activated via these menus
2. The main display pane – shows all keysets, spare keyset spaces, key containers and key pairs. These are represented as a hierarchical tree view, with keysets being the highest member, key containers and key pairs shown beneath each keyset.

Password Entry Dialogs

Most actions performed within the keyset management utility will require entry of the keyset user password (see *Figure 8*). The keyset password is case sensitive and may consist of any alphanumeric characters, of between 4 and 32 characters in length.



Figure 8 – Keyset password entry dialog.

Keep Password Feature

The password dialog has a facility to remember the keyset password whilst the keyset management utility remains open. This eliminates having to repeatedly enter the password when performing multiple operations.

To enable this facility, check the box next to “Keep Password For Session” within the keyset password dialog. Correct password entry followed by the “OK” button will enable the “Keep Password” feature.

Note: When this feature is enabled, care should be taken not to leave the utility unattended. To ensure that unauthorized people do not obtain access to a user keyset, close the keyset management utility once you have finished with your assigned task.

Keyboard Shortcuts

All available menu choices may be selected via a series of keyboard shortcuts. The menu bar can be selected by pressing the “Alt” key. Commands may then be selected by pressing the first unique letter of the required command. For example: “Alt” followed by “A” will open the Adapter menu.

There are also a number of key combination shortcuts which will immediately activate a command. To activate these shortcuts you must hold down the first key, whilst pressing the second key. These are as follows:

- “CTRL”+”A” = Add Container
- “CTRL”+”R” = Remove Container
- “CTRL”+”K” = Create Keyset
- “CTRL”+”P” = Change Password
- “CTRL”+”G” = Generate Key Pair
- “CTRL”+”D” = Delete Key Pair

Context Menus

Right-clicking on an item, which has been selected from the display pane, will bring up a context menu showing available commands specific to that item.

For details about these commands, please refer to the appropriate section below.

Menu Options

The following details each available menu option which can be accessed from the keyset management utility.

Keyset Menu

The **Keyset** menu is used to perform the following user actions on a selected keyset:

- Create Keyset
- Add Container
- Change Password

Create Keyset

This menu option creates a keyset within the currently selected keyset space.

Add Container

This menu option adds a key container to the selected keyset.

Change Password

This menu option changes the keyset password from the current to a new value for the selected keyset. The user is prompted to enter the current password as well as enter and confirm the new password.

Container Menu

The **Container** menu is used to perform the following user actions to a keyset container:

- Remove
- Generate Key Pair

Remove

This menu option:

Removes the selected key container. Note that this action destroys any key pairs contained within the selected container.

Generate Key Pair

This menu option generates a key pair within the selected container. The user is prompted for the key usage and key size:

- Valid key usage is either “Sign” or “Exchange”
- Valid key lengths are either 512, 768, 1024, 2048 or 4096 bits

Key Pair Menu

The **Key Pair** menu is used to perform the following user actions to a selected key pair:

- Delete
- Properties

Delete

This menu option deletes the selected key pair

Properties

This menu option displays properties for the selected key pair. The following attributes are shown:

Keyset	Shows the name of the keyset on which the selected key pair resides.
Container	Displays the key container name, in which the selected key pair resides.
Usage	Shows the key usage attribute of the selected key pair. This value will either be “EXCHANGE” or “SIGN”.
Size	Shows the key size for the selected key pair.

Private Key Held	This indicates if the private key for the selected key pair is present as part of the key pair. Since it is possible to import a public key only, this value will either be “TRUE” or “FALSE”.
Exportable	Indicates if the selected key pair can be backed up.

CTKMU

Key Management Utility for the ProtectToolkit M environment.

SYNOPSIS

```
ctkmu c [-s<slot>] [-z<size>] [-g] [-k<numb>] [-p] -a<attribute> -n<name> -t<type>
ctkmu d [-s<slot>] -n<name>
ctkmu e -c<slot>
ctkmu i [-s<slot>] [-2] [-w<name>] (-y [-m] -a<attribute> -n<name> -t<type> | -c<slot> | <filename>)
ctkmu j -n<name> [-s<slot>] -a<attribute> <filename>
ctkmu l [-s<slot>] [-n<name>]
ctkmu m [-s<slot>] -a<attr> -n<name>
ctkmu p [-s<slot>] [-O]
ctkmu s -c<slot>
ctkmu t [-s<slot>] [-l<label>]
ctkmu x [-s<slot>] [-n<name>] [-w<name>] (-y [-m] | -c<slot> | <filename>)
```

DESCRIPTION

The *ctkmu* utility is used for ProtectToolkit M token management. This includes the operations required by a token's SO such as setting user PINs and re-initializing tokens as well as those operations required by the normal user such as object management.

A number of commands can be used with the *ctkmu* utility to help with key creation, deletion, import, export, as well as token initialization and PIN change.

COMMANDS

- c** **The Create Key command 'c'** is used to generate new keys on the specified token. The **-a** parameter is used to specify the attributes, the **-n** parameter specifies the key's label and the **-t** parameter the new key type. *Appendix C, PKCS#11 Attributes* contains further information on key attributes. Common uses for this command are generation of a random key, import of a split custodian key (using the **-k** flag), or creation of a split custodian key (using the **-g** and **-k** flags). When importing a split custodian key, optionally, a supported pin pad device can be used (using the **-p** flag) to ensure that the key components are entered directly to the device. See *Appendix D, KAAAN Professional Smart Card Reader* for further information.
- d** **The Destroy Key command 'd'** is used to delete a key on the specified token. This command will permanently destroy the key with the label specified with the **-n** parameter.
- e** **The Erase Smart Card command 'e'** is used to erase a smart card in the specified slot and will leave the smart card in an uninitialized state.
- i** **The Import Key command 'i'** is used to import keys previously exported with the export command (see below).

- j** **The Import a private Key command ‘j’** imports a private key and a Certificate from a PKCS#12 file.
- l** **The List Information command ‘l’** used to display information on the objects stored on the token in the specified slot. This command will list the actual keys, certificates and other objects, or, if the token is a smart card token previously used with the key export function information on that key backup set.
- m** **The Modify Attributes command ‘m’** is used to toggle the specified attributes, i.e. change from TRUE to FALSE and vice versa or add the attribute if it does not exist.
- p** **The Pin command ‘p’** is used to initialize the User PIN or to change an existing PIN (either the User or SO PIN). If the specified slot contains a token without an initialized User PIN this command will prompt for the current SO PIN and then for the new User PIN. If the PIN is initialized the current PIN will be prompted for before the new PIN may be specified. To change the SO PIN specify the **-O** option.
- s** **The Smart Card status command ‘s’** is used to display information on the smart card token currently inserted in the specified slot. Details of the keys exported to the token will be displayed.
- t** **The Initialize/Reset Token command ‘t’** allows for existing tokens to be initialized or re-initialized. If the specified token contains an initialized token the current SO PIN will be prompted for before a new Token label may be specified and the token re-initialized. If the token is un-initialized this command will only operate if the *No clear PINs flag* is not specified for the HSM. In this case the new SO PIN and label may be specified. Once the token has been reset or initialized a new user PIN may also be set.
- x** **The Export Key command ‘x’** allows for keys to be exported to one or more smart cards or to a file. This command supports two key methods; keys may be exported as split custodian in which case they will be encrypted using a randomly generated key and then distributed to a number of smart card tokens. Alternatively a key wrapping key may be specified which will then be used to encrypt the key specified for backup. This encrypted data can then be written to a smart card token or to a file.

PARAMETERS

- a<attributes> --attributes=<attributes>** Specifies attributes for an object / key
- Valid attributes are:
- P CKA_PRIVATE=1
 - M CKA_MODIFIABLE=1
 - T CKA_SENSITIVE=1
 - W CKA_WRAP=1
 - w CKA_EXPORT=1
 - U CKA_UNWRAP=1
 - X CKA_EXTRACTABLE=1

		x	CKA_EXPORTABLE=1
		R	CKA_DERIVE=1
		E	CKA_ENCRYPT=1
		D	CKA_DECRYPT=1
		S	CKA_SIGN=1
		V	CKA_VERIFY=1
		L	CKA_SIGN_LOCAL_CERT=1
		C	CKA_USAGE_COUNT=1
		I	CKA_IMPORT=1
-c <slot>	--sc-slot-num =<slot>		Specifies the Smart Card slot to export to or import from
-g	--gen-comp		Generate key components
-h, -?	--help		Display usage information
-k <numb>	--num-comp =<numb>		Number of key components required to be entered or number to be generated (when -g parameter is specified)
-l <label>	--label =<label>		Specify label.
-m	--multi-part		Do a multi part key entry for console import/export
-n <name>	--name =<name>		Name of the object to operate on
-O	--SO-PIN		Change the Security Officer PIN. Used in conjunction with the change PIN command.
-p	--pinpad		Use a supported pin pad device for entering key components.
-s <slot>	--slot-num =<slot>		Specifies the slot to operate on. Default is 0 (zero), however must be specified when using the l command and -v option for Slot 0.
-t <type>	--type =<type>		The type of key to create. Options are: (aes des des2 des3 rc2 rc4 cast idea rsa dsa)
-v	--verbose		This will display the attributes that ctkmu may change
-w <name>	--wrap-key =<name>		Name of the key used to wrap or unwrap
-y	--console		Import/Export using the console

-z <size>	--size= <size>	Size of the key to create/import (for aes, rc2, rc4, cast, rsa, dsa and generic secret)
-2	--Cprov2	Import keys from a Cprov 2 formatted file. This is used when migrating keys from an older Cprov 2 key format to the current format.

CREATECERT Utility

Utility for creating a self signed certificate.

SYNOPSIS

createcert <X509 Name>

DESCRIPTION

The **createcert** utility is used as a quick and simple method of creating a self signed certificate for the ProtectToolkit M machine.

Note: The logged on user must be logged on as administrator to use this utility.

PARAMETERS

X509 Name The X.509 Certificate name. For example, CN=<machinename>.

Managing the Host Interface Master Key (HIMK)

HSMs may implement an optional Trusted Channel called the Secure Message System (SMS). This system enables applications to securely communicate with the HSM over the PCI bus interface. The SMS is enabled when either the *No Clear PINs* or *Authentication Protection* security flags are enabled. Refer to the section *Security Mode Flag Descriptions* for more details regarding these flags.

A Trusted Channel (TC) is created on demand by the operator but may be terminated by either the HSM or the operator. Either the HSM or application may be configured to require a TC to be created before cryptographically sensitive services can be provided. For the HSM to be compliant to FIPS 140 operation the HSM must be configured in this way. However it is also possible for the application to request and use a TC even though the HSM is not configured to require them.

The HSM can manage multiple simultaneous Trusted Channels, each of which will have its own set of randomly generated session keys for message encryption / decryption and message signing / verification. The negotiation of these session keys is based on a shared secret known by both the application and the HSM. This secret is a triple length DES keys and is known as the Host Interface Master Key (HIMK).

After initial installation, the HSM will possess a single default HIMK. Administrators may create new HIMKs and destroy existing HIMKs (including the default). The HIMK can be updated on the host or on the HSM using either direct key input or a specified key file. This is achieved using the *cthink* utility.

To setup the HIMK on the HSM:

- From a command prompt type the following:

```
cthink
```

The utility will now proceed through the following steps:

- The HIMK key value is prompted for. This will need to be entered and confirmed using 48 hexadecimal digits.
- Following confirmation of the key input, the Key Check Value (KCV) will be shown.
- A prompt will ask if the KCV is correct. Enter Y if the value is as expected.
- The administrator password will have to be entered next.
- Following will be the prompt to enter a label for the new HIMK.
- Should an existing HIMK be found on the chosen HSM, the utility will prompt you to delete the old HIMK. Press Y if you want to replace the existing key or N to keep the existing key and store the new HIMK in addition to the old.
- The utility will now prompt the user to update the *cryptoki.ini* file with the new HIMK settings if required. Press Y if you want the host side updated with the new HIMK.
- The next action is the prompt to generate a key file for HIMK distribution or backup. Press Y if you want to make a key file for the new HIMK. Press N if you do not want to generate a file and exit from the utility.
If the user answered Y to the key file generation prompt, a file name must be entered.
- The utility will also prompt for entry and confirmation of a password which will protect the HIMK key file.
- The above file output procedure can be repeated as many times as required by pressing Y at the *Export to another file* prompt. Pressing N will exit from the utility.

For further information about the *cthink* utility see below.

CTHIMK Utility

Host Interface Master Key (HIMK) management tool.

SYNOPSIS

```
cthink [-?h] [-d<device>] [-i<file>] [-r] [-u]
```

DESCRIPTION

The HIMK is used for secure messaging, therefore at intervals it may be necessary to change the HIMK. By default, the *cthink* utility will set the new HIMK on both the user (host) side and the HSM. It is also possible to set the new HIMK on the user (host) side only. See the *Managing the Host Interface Master Key (HIMK)* section above for further information about using this utility.

OPTIONS

The following options are supported:

-d <device>	--device =<device>	Device Number. This will specify which device to update with the new HIMK. By default device 0 will be updated.
-h	--help	Display usage information
-i <file>	--import =<file>	Import HIMK from file. The default mode requests the key to be entered from the console.
-r	--reset	Reset the HIMK in <i>cryptoki.ini</i> to default (none).
-u	--user	User Mode. Only the HIMK on the user or host side will be modified. The default mode will update both the user side and the HSM. In addition, in the user mode the option to back up the HIMK will not be presented.

Chapter 7

Troubleshooting

This section details some of the known issues that can occur due to incorrect configuration or usage of the ProtectToolkit M product.

Should you encounter any difficulties not detailed within this section, please contact SafeNet technical support for advice.

Known Issues

Self Signed Certificates and Netscape

Note that self signed certificates will not work with Netscape. Only Microsoft Internet Explorer will accept the use of such a certificate, but will prompt the user with a security alert.

Problem: Netscape browser does not load page using https.

Cause: The Netscape browser is unable to validate the root of the self-signed certificate. IE displays a warning and allows the user to accept the certificate and continue, whereas Netscape does not, it just does not load the page.

Solution: Use the IE browser or use a certificate issued by the Microsoft CA.

Keyset Restore Error

When attempting a keyset restore and not using the same key and password during backup, an error message may be displayed.

Problem: Error message during keyset restore. e.g. “The key used to import was not the same as key to export”

Cause: There is a mismatch between the backup and restore key.

Solution: After a tamper, when the keyset is being created, if you want to restore from a previous backup you must set the keyset password to be the same as when the backup was taken.

Microsoft Internet Explorer (IE5)

In some instances, Microsoft IE5 will not be configured to accept 128-bit encryption. You can find out what your browser cipher strength is, by selecting Help/About from the

menu bar of IE5. If the dialog tells you that the cipher strength is 56-bit, you will need to obtain an upgrade file from Microsoft.

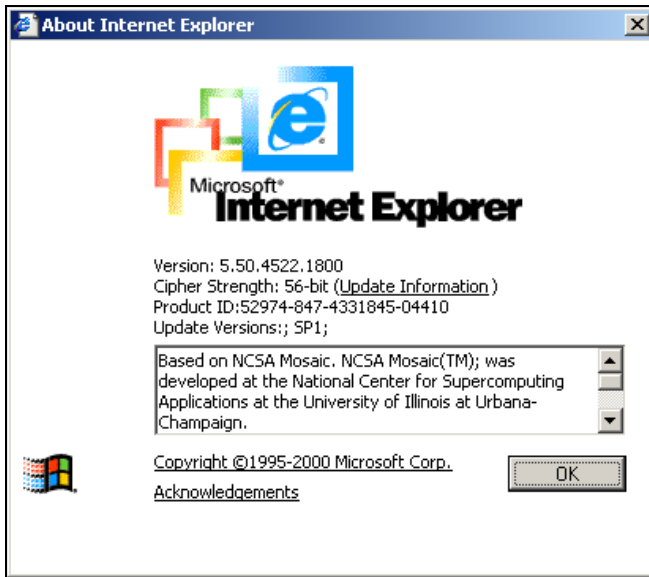


Figure 9 - IE5 help/about dialog

Session Exists Error

This error may occur during an attempt to allocate additional keyset space or during a delete keyset operation.

- Problem:** Error message during keyset delete or during space allocation / de-allocation.
- Cause:** There are applications that have open sessions to ProtectToolkit M. Certain administrative operations require exclusive use of the system as a security measure; these include keyset sensitive tasks such as space allocation and keyset deletion.
- Solution:** Close or temporarily stop any applications or services that may be using ProtectToolkit M such as Certificate Services, IIS etc.

To check if an application has any open sessions to ProtectToolkit M, check the value shown next to *Application Count* in the *System* section of the administration utility. This will need to be “1”, and the *Total Session Count* must be “0” in order for the chosen action to succeed.

If this error persists, try re-booting your machine and check for any self- or auto-starting applications which may open sessions to ProtectToolkit M.

The *Certification Authority* service: *CertSvc* is one application that may be using ProtectToolkit M. If after reboot, the application count is still > 1, try disabling the service, performing the Admin operation and then re-enabling the service.

Also try the following if applicable:

- Stop the CA
- Deactivate *Directory Security* (IIS)
- Reboot machine
- Run the E8KRESET utility (PCI HSM only).

Note: If the value of *Application Count* is shown as “UNAVAILABLE”, your HSM firmware doesn’t support live application counting. In such a case, it is advisable to upgrade the HSM firmware to the latest version. Please refer to the section entitled *Checking and Upgrading HSM Firmware* in *Chapter 4 Administrative Tasks* for full details.

Duplicate Container or Key Instances

It is possible that following a key restore operation, there may be more than one instance of the same container or key within a particular keyset.

Problem: Duplicate key or container instance showing in keyset management utility (see *Figure 10*).

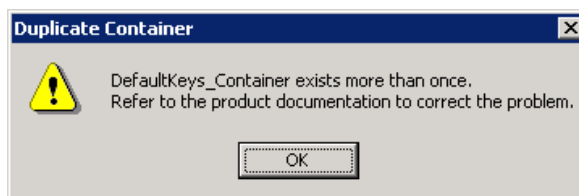


Figure 10 – Duplicate Container Error.

Cause: This is caused by performing a key restore whilst the same keys are already in existence on the selected keyset. ProtectToolkit M does not replace existing keys during a key restore. Multiple instances of the same key will cause the keyset management utility to show the keyset as being invalid.

Solution: Close any applications that are using ProtectToolkit M.

There are two methods which can be employed to address this problem:

1. It is possible to use the CTKMU utility to manually delete one of the duplicate keys or containers.

To delete a duplicate key object:

- Ascertain the slot on which the duplicate object resides by performing the following command:

ctkmu l

- List the contents of the slot. For example:

ctkmu l -s<slot>

Answer Yes to view private <user> objects.

- Note the name of the object which appears twice

- Delete one of the duplicate objects. For example:

ctkmu d -s<slot> -n<object name>

The above command will show a list of objects. The only method of determining which to delete will be to look at the date of creation.

2. An alternative to the above is to delete the affected keyset using the administration utility.

Note: This can only be performed by the device administrator and destroys all containers and key pairs on the selected keyset.

Following deletion of the keyset, it will have to be recreated, and key containers may then be restored from a backup.

Application Error

Problem: An application which was functioning correctly prior to ProtectToolkit M installation is now not working.

Cause: This may be caused by the replacement of the default “RSA SChannel” provider. During installation, ProtectToolkit M changes the default provider to be the “Safenet RSA SChannel” provider. In some cases this provider is incompatible with certain applications.

Solution: Restore the default previous provider. To ascertain which provider was used prior to ProtectToolkit M installation, open the file “uninst.ini”, found in your ProtectToolkit M installation directory. The last line of the file will detail the name of the provider prior to the installation.

You must edit your registry and change the required value. Do not perform this if you are uncertain on how to alter the Windows registry. Obtain advice from your system administrator, or alternatively uninstall the ProtectToolkit M product to see if this fixes the problem.

Appendix A

Integration with Microsoft CA

Setting Up a CA with ProtectToolkit M

This section explains how to configure ProtectToolkit M to be used with the Microsoft CA.

ProtectToolkit M, in conjunction with Microsoft CA, provides the ability to securely store keys related to signing certificates.

IMPORTANT NOTE: Before starting, ensure the following:

- Make certain that you have read and understood Chapters 1 through to 3.
- Make sure that Microsoft CA has NOT been installed prior to the ProtectToolkit M installation.
- Ensure that the current logged on user has Windows administrator privileges.
- A keyset must exist for the logged on user. For Administrative Tasks please refer to *Chapter 4*.

An example of how to setup the CA with ProtectToolkit M follows. Microsoft Windows 2000 screen shots are used. Where differences exist between Windows 2000 and Windows 2003 Server these are noted. See also the next subsection with respect to Windows 2003.

NOTE: The field entries shown in the screen shots are exemplary only. Actual values should be chosen as required, to suit each particular installation.

To set up the CA with ProtectToolkit M:

- From the Windows Control Panel, select *Add/Remove Programs* and then the *Add/Remove Windows Components* option.
- Select *Certificate Services* and press *Next* to continue.
- Select the required CA type. Make sure that either the *Advanced options* (Windows 2000) or the *Use custom settings to generate the key pair and CA certificate* checkbox is checked.

To use the **key backup and recovery feature** (Windows 2003 server only) the CA type must be either *Enterprise root CA* or *Enterprise subordinate CA*.

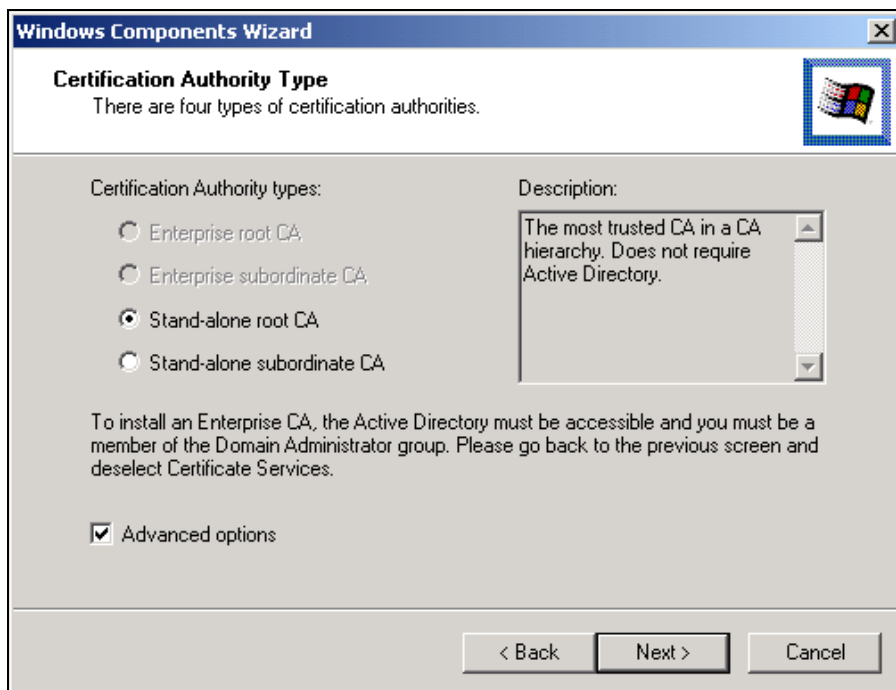


Figure 11 - Certification Authority Type selection dialog

- Press *Next* to continue.
- On the *Public and Private Key Pair* page of the wizard, under *CSP*, select *Safenet RSA Full Cryptographic Provider*. Also choose the type of hash algorithm to use and the required key length.

Note: If the *Safenet RSA Full Cryptographic Provider* is not shown in the list, it means that there is no keyset for the currently logged on user. If this is the case, cancel the current setup and create a keyset for the current logged on user before attempting to setup Microsoft CA again.

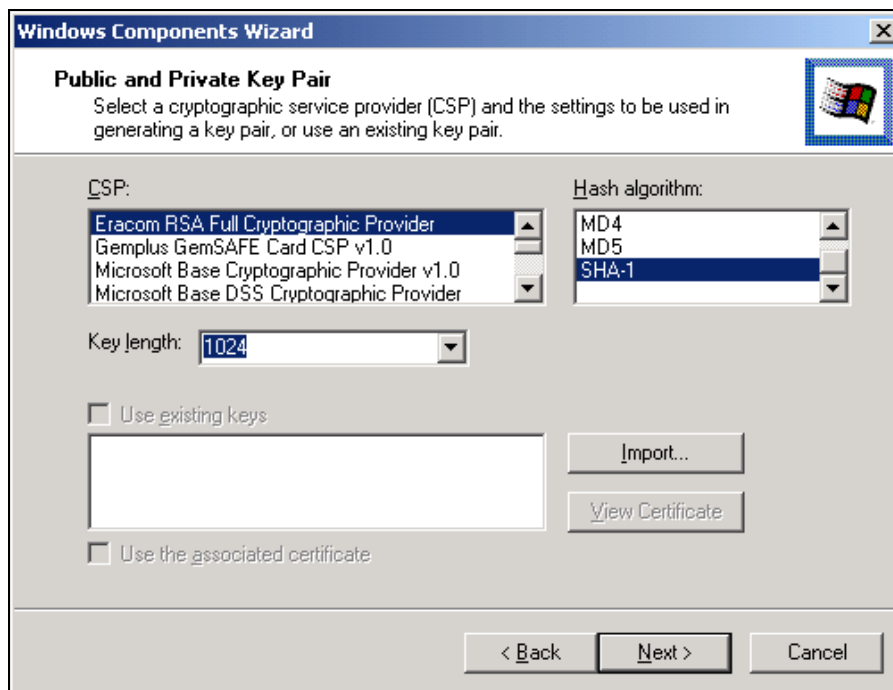


Figure 12 - CSP Selection dialog. (Setting of SHA-1 and 1024 are examples only)

- Press *Next* to continue.
- You will now be prompted to input CA Identifying Information. Enter all details as required and press *Next* when ready to continue.

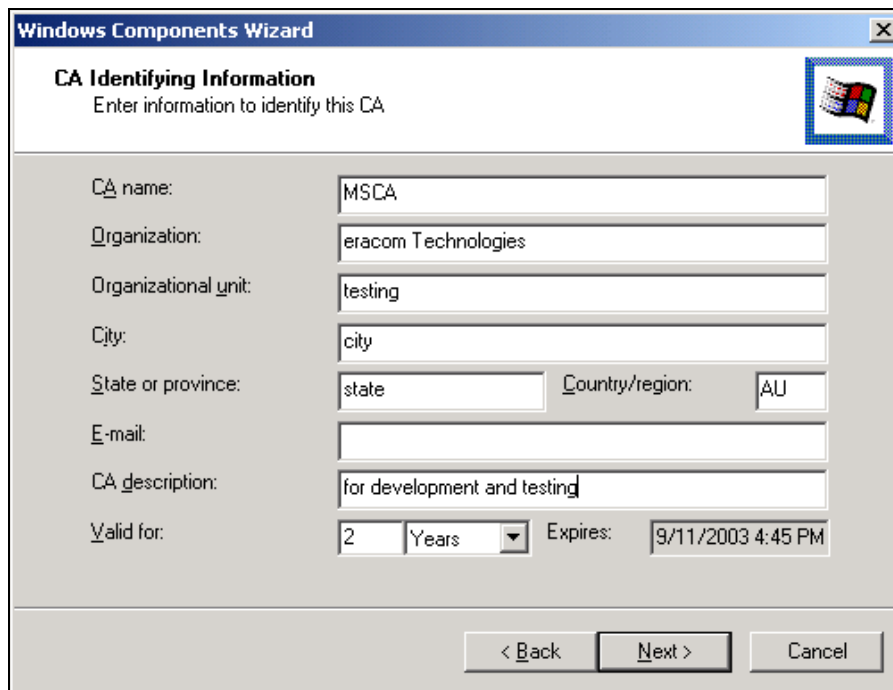


Figure 13 - CA Identifying Information dialog. (Entries are examples only)

- The final dialog prompts you to enter the storage location for the configured data. Type location paths or use the *Browse* buttons to locate them as required and press *Next* to complete the Microsoft CA setup.

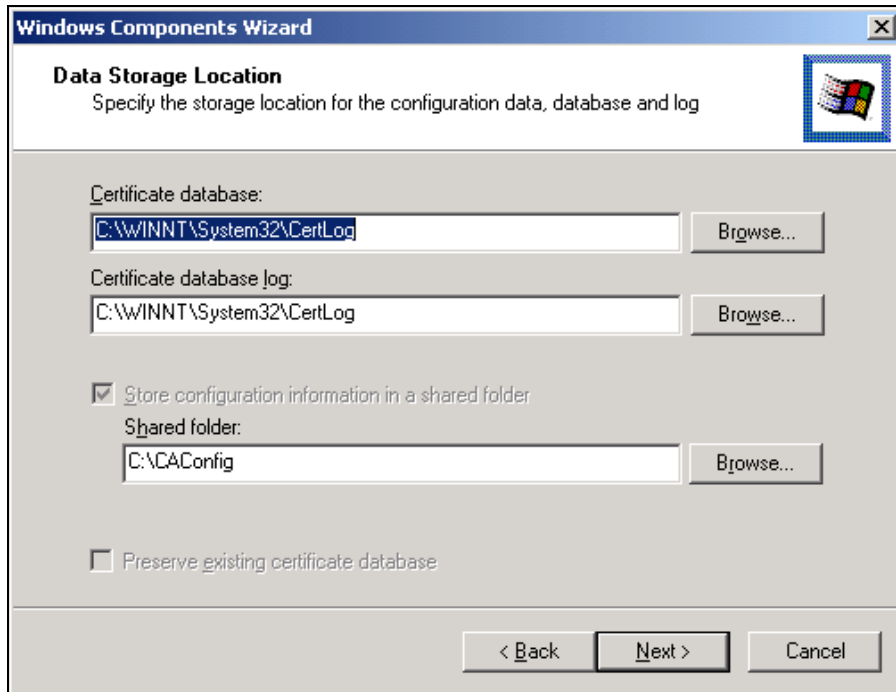


Figure 14 - CA Data Storage Location dialog

Following the successful completion of the above steps, ProtectToolkit M is now selected as the CSP for Microsoft CA operations. For further details regarding the Microsoft CA, please refer to your Microsoft documentation.

Certificate Template Support for Safenet CSPs (Windows 2003)

For Windows 2003, the current list of certificate templates in the CA **do not** make use of the Safenet CSP. New templates must be created in the Certificate Templates store and then issued from the CA templates store. For example, a *web server* certificate template only supports the Microsoft DH and RSA providers.

In order to create new templates that support the Safenet CSP, perform the following procedure. The procedure is basically the same for any certificate that you need to issue using the Safenet CSPs.

To create a new template that supports the Safenet CSP:

Note that the use of the *User* template in this procedure is exemplary only. Substitute this for any other template to meet your particular requirements.

1. Start a new *MMC* session and add both the *Certification Authority* and *Certificate Template* snap-ins.
2. Expand the *Certificate Templates* object and locate the *User* template.
3. Right click on the *User* template and select *Duplicate template*. This will display the new template properties.
4. Enter a *Template display name*. Choose an appropriate name e.g. *Users*. Note that you cannot give it the same name as the template that already exists.
5. Go to the *Request Handling* tab and click on the *CSP* button. Either click the radio button *Requests can use any CSP available on the subject's computer* or make sure that the Safenet RSA providers are checked.
6. Check the *Issuance Requirements* and *Security* tabs to ensure that the appropriate permissions are correct.
7. Click *OK* to complete.
8. Now go to the CA object and select *Certificate Templates*.
9. Right click, *New* and select *Certificate Template to Issue*.
10. Locate the *Users* template that was created in steps 1-7 and click *OK*. Close the MMC console session.
11. To test that the Safenet provider is now available, open a new MMC console and choose the *Certificates* snap-in. Select *My User Account* when prompted. The Administrator's personal certificate store is now available.
 - a) Right click on the personal object and select: *All Tasks, Request new certificate*. The *Certificate Request Wizard* displays.
 - b) Click next to reveal the certificate types available and click the *Users* certificate and the *Advanced* check box. Click *Next*.
 - c) On the *CSP* page that now displays, note that the Safenet providers are now listed. Choose the *RSA full provider* and any other appropriate settings such as *Key is Exportable* etc.
 - d) Complete the process by clicking *OK*. The certificate is generated and visible in the personal store.

CA Replication (Key Backup and Recovery)

Typically, to replicate a CA installation, keys may be backed up to smart cards and then restored from the smart cards to establish the new CA installation. One smart card per keyset is required.

A triple-DES *BackupKey* is used to encrypt each keyset prior to storage on a smart card. A different *BackupKey* is automatically created for each keyset when the keysets are created but these keys are not visible under normal ProtectToolkit M operation. A *BackupKey* for a keyset is derived from a combination of the password used to secure that particular keyset and the keyset name. In the case of the MACHINE and SYSTEM keysets, the device administrator's password and the keyset name are used to derive the key. Thus to restore a keyset that was previously backed up, the same password and keyset name must be used.

Backing Up Keys for a CA Installation to Smart Cards

1. Obtain a listing of all keysets by executing *ctkmu* with the **l** option from a command prompt. A list of all keysets and associated slots displays.

2. Decide which keysets to backup.

Important: At a minimum the MACHINE_Keyset must be backed up as this is where the CA keys are stored.

3. Record the slot number for each keyset that you wish to backup.

4. To backup the MACHINE_Keyset to smart card, from a command prompt, type the following, substituting the slot number of the MACHINE_keyset for *n* and the slot number representing the smart card reader for *b*. Both *n* and *b* can be found in the listing obtained at step 1.

```
ctkmu x -sn -wBackupKey -cb
```

5. Upon receiving a prompt for a user password enter "password".

6. Insert a new smart card and repeat steps 4 and 5 for the SYSTEM_Keyset if required.

7. Insert a new smart card and repeat step 4 for each of the other keysets required.

Replicating a CA Using Keys Restored from Backup Smart Cards

Key Points

The procedure that follows takes account of the following key points.

- On the machine where the replica is to be created ProtectToolkit M must be installed before the Microsoft CA.

- To allow installation of a CA that utilizes the Safenet CSP for the HSM storage of keysets, both the MACHINE_Keyset (where the CA stores keys) and a user keyset for the current user must be available. At CA installation time if either or both of these keysets are missing, the Safenet CSP will not display in the list of CSPs available for selection.
- All keyset names and associated passwords created when establishing the replica must match the originals that are to be restored from the backup smart cards.

Procedure

1. Install ProtectToolkit M.
2. Start the ProtectToolkit M administration utility. This can be done via the Windows Start menu. Select Start, Programs, Safenet, ProtectToolkit M, Administration.

The lack of a MACHINE_Keyset and a SYSTEM_Keyset will be detected and these will be created. Later on, the MACHINE_Keyset created here will be replaced with the version containing the CA keys that was backed up to smart card.

The device administrator password will be requested, or must be set if this is the first time the HSM has been accessed.

The Administration Utility default view displays.
3. Under *Active Adapters*, expand *All* to reveal the device and the Machine and System key sets just created on that device.
4. Highlight the device entry and select *Adapter* on the menu bar. Now select *Allocate Space* to create a keyset space.
5. Under *Active Adapters*, select the spare keyset space.
6. Select *Keyset* on the menu bar and then choose *Create Keyset*. The Administration Utility will now prompt for a keyset name to use and the password for the currently logged on user as authorisation. The default name should be accepted.
7. If additional user keysets containing keys are to be restored from smart card then for each keyset to be restored create an empty replica keyset on the HSM that has the same name and that is protected by the same user password as the original. To do this, repeat steps 4 to 6 for each keyset using the correct keyset name and user password each time.
8. Obtain a listing by name of all the keysets that now exist on the HSM and their corresponding slot numbers. To do this execute the following command from a command prompt:

```
ctkmu l
```
9. Import a keyset from smart card to the HSM. To do this insert the smart card containing the keyset and execute the following command from a command prompt:

```
ctkmu i -sN -wBackupKey -cM
```

where: **N** is the slot number of the keyset on the HSM discovered in step 8

M is the smart card reader slot number. This will also be shown in the listing obtained at step 8.

10. Upon receiving a prompt for a user password enter the value for the keyset being restored. In the case of the machine and system keysets the value to enter is “password”.
11. Insert a new smart card and repeat step 4 for each of the other keysets until all have been restored.
12. Install the Microsoft CA
13. Select the Safenet CSP from the drop down box during installation. If the Safenet option is not present, this means that the keyset for the currently logged in user does not exist. Ensure the user is the same as the user who did the backing up of the CA initially.
14. After selection of the Safenet CSP, click on the *Use existing keys* box and select the key that corresponds to the CA key pair.

Private Key Archiving and Recovery

When requesting a certificate using the Windows 2003 CA, users now have the option of selecting to have their private key archived by the CA. In situations such as a catastrophic system failure that results in the user losing their entire system, this feature allows the user’s private key to be recovered.

Support for this archival and recovery process is included in ProtectToolkit M. The following sections, *Private Key Archiving Example* and *Private Key Recovery Example* demonstrate the use of this capability.

Private Key Archiving Example

Here are the tasks required to archive a private key using a Microsoft certification authority (CA).

1. Create a key recovery agent account
2. Acquire the key recovery agent certificate
3. Configure the certification authority to allow key recovery
4. Create a new certificate template that allows key archiving
5. Acquire a user certificate that has an archived key

Prerequisites

Before doing these tasks:

- You must have a Windows Server 2003 domain controller.
- The Windows Server 2003 domain controller must also be configured as an enterprise root or subordinate CA.
- A user keyset for the user must exist. Refer to *Creating User Keysets* on page 21 for further information.
- The *Allow Clear Export of Private Keys* flag must be set. See the section *Enabling Private Key Clear Export* in *Chapter 4, Administrative Tasks* above for the procedure.

Task 1—Creating a Key Recovery Agent Account

Configure and add the *Key Recovery Agent certificate* template as a template that can be issued by the enterprise CA.

1. Verify who can enroll the Key Recovery Agent template
 - a. Log on as administrator.
 - b. Click *Start, Run* and type *certtmpl.msc* then press *Enter*. This opens the *Certificate Templates* snap-in in the *Microsoft Management Console*.
 - c. In the console tree, click *Certificate Templates*.
 - d. In the details pane, right-click *Key Recovery Agent* and click *Properties*.
 - e. In *Key Recovery Agent Properties*, click the *Security* tab.

By default, the security groups that can enroll the Key Recovery Agent certificate template are Domain Administrators and Enterprise Administrators.
 - f. To allow other users or groups to enroll the Key Recovery Agent certificate template, click *Add* to add the user or group and grant them *Read* and *Enroll* permissions.
2. Change the default issuance behavior of the Key Recovery Agent template
 - a. In *Key Recovery Agent Properties*, click the *Issuance Requirements* tab.
 - b. Clear the *CA certificate manager approval* check box and click OK.
 - c. Close the *Microsoft Management Console*.
3. Change the request handling to allow the Safenet CSP's

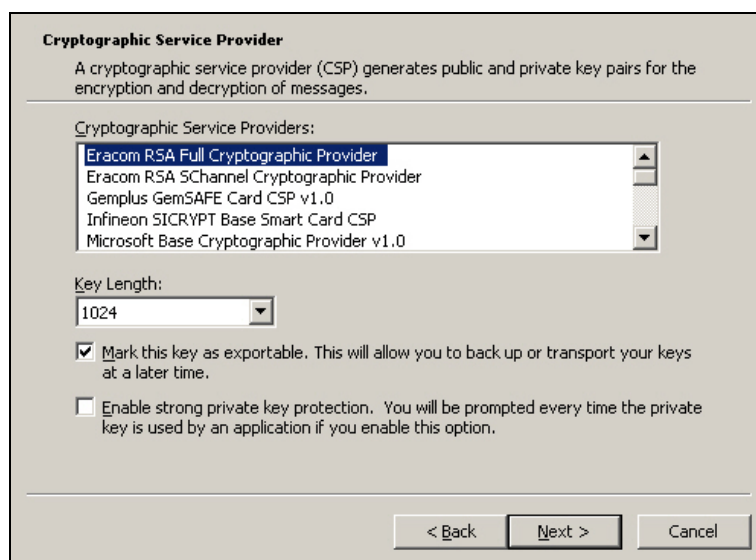
- a. In the *Key Recovery Agent Properties*, click the *Request Handling* tab.
 - b. Check that *Allow Private Key to be Exported* is ticked.
 - c. Click on the *CSP* button and click on the radio button to allow requests to use any CSP available on the subject's computer.
4. Configure the Certification Authority (CA) to issue Key Recovery Agent certificates.
- a. On the *Administrative Tools* menu, click *Certification Authority*. This opens the *Certification Authority* snap-in in the *Microsoft Management Console*.
 - b. In the console tree, double-click the CA, and then click *Certificate Templates*.
 - c. Right-click *Certificate Templates*, then click *New, CertificateTemplate to Issue*.
 - d. In *Enable Certificate Template*, click *Key Recovery Agent*, and then click *OK*.

Task 2—Acquiring the Key Recovery Agent Certificate

In this series of steps, you will acquire a Key Recovery Agent Certificate for the purpose of recovering private keys.

1. Create an MMC console with the *Certificates* snap-in loaded.
 - a. Ensure that you are logged on as the administrator.
 - b. On the taskbar, click the *Start* button, and then click *Run*.
 - c. In *Run*, type *mmc*, and then click *OK*.
 - d. On the *File* menu, click *Add/Remove Snap-in*.
 - e. In *Add/Remove Snap-in*, click *Add*.
 - f. In *Add Stand-alone Snap-in*, click *Certificates*, and then click *Add*.
 - g. In *Certificates*, click *My User account* and then click *Finish*.
 - h. Click *Close*, and then click *OK*.
2. Acquire a Key Recovery Agent certificate.
 - a. In the newly-created MMC console, in the console tree, double-click *Certificates - Current User*.
 - b. In the console tree, right-click *Personal*, click *All Tasks, Request New Certificate*.

- c. In the *Certificate Request Wizard*, click *Next*.
- d. In *Certificate Types*, select *Key Recovery Agent* and the *Advanced* checkbox, and then click *Next*.
- e. On the *CSP* page that now displays (see example below) choose the Safenet provider for HSM key storage and any other appropriate settings such as *Key is Exportable* etc. Then click *Next* and *Next* again.



- f. On the *Certificate Friendly Name and Description* page, in the *Friendly name* field, type *Key Recovery*, and then click *Next*.
- g. In *Completing the Certificate Request Wizard*, click *Finish*.
- h. In the console tree, double-click *Personal* and then click the *Certificates* folder.
- i. Ensure that a certificate with the friendly name of *Key Recovery* exists.
- j. Close the console without saving changes.

Task 3—Configuring the CA to allow key recovery

In this series of steps, are used to configure the enterprise CA to use the Recovery Agent certificate acquired in Task 2. The CA must load the public key for the Key Recovery Agent to be used for encrypting the recovery data.

1. Configure the Recovery Agent to be the Administrator's Key Recovery Agent certificate.
 - a. Ensure that you are logged on as the administrator.

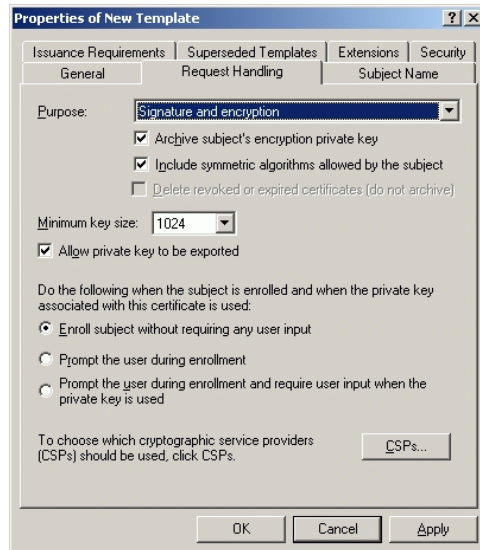
- b. In *Administrative Tools*, open *Certification Authority*. This opens the *Certification Authority* snap-in in the *Microsoft Management Console (MMC)*.
 - c. In the console tree, click *the CA*.
 - d. Right-click *the CA*, and then click *Properties*.
 - e. On the *CA Properties*, on the *Recovery Agents* tab, click *Archive the key* and then click *Add*.
 - f. In *Key Recovery Agent Selection*, click the certificate that is displayed, and then click *OK*. The *key recovery agent certificate* is shown with a status of *Not loaded*.
 - g. Click *OK* and when prompted to restart the CA, click *Yes*.
 2. Open the Certificates console, focused on the local computer.
 - a. On the taskbar, click the *Start* button, and then click *Run*.
 - b. In *Run*, type *mmc*, and then click *OK*.
 - c. On the *File* menu, click *Add/Remove Snap-in*.
 - d. In *Add/Remove Snap-in*, click *Add*.
 - e. In *Add Standalone Snap-in*, click *Certificates*, and then click *Add*.
 - f. In *Certificates snap-in*, click *Computer account* and then click *Next*.
 - g. In *Select Computer*, click *Local Computer*, and then click *Finish*.
 - h. Click *Close*, and then click *OK*.
 3. Verify the installation of the Key Recovery Agent (KRA) certificate.
 - a. In the console tree, double-click *Certificates (Local Computer)*, double-click *KRA*, and then click *Certificates*.
 - b. In the details pane, double-click the certificate.
 - c. Verify that the intended use of the certificate is *Key Recovery Agent* and the certificate is issued to *Administrator*. This procedure ensures that the Key Recovery Agent has been successfully configured.
 - d. Click *OK* and then close the console without saving changes.

Task 4 — Creating a new certificate template that allows key archiving

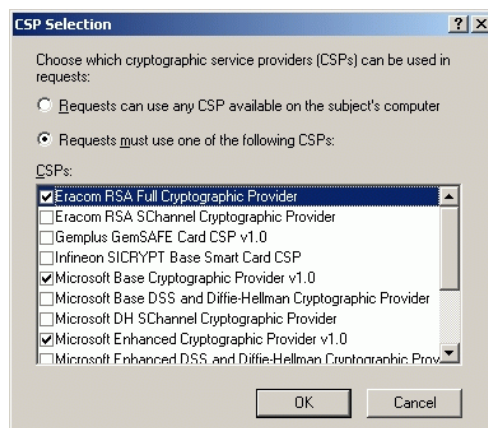
In this series of steps, you will define a new template that allows Key Archival and HSM key storage by using the *Certificate Templates* console. This will allow hardware key storage within a HSM at the client computer and key recovery in the domain in the event that the private key is lost or corrupted at the client computer.

1. Open the *Certificate Templates* console.
 - a. Log on as the administrator.
 - b. On the taskbar, click the *Start* button, and then click *Run*.
 - c. In *Run*, type *mmc*, and then click *OK*.
 - d. On the *File* menu, click *Add/Remove Snap-in*.
 - e. In *Add/Remove Snap-in*, click *Add*.
 - f. In *Add Standalone Snap-in*, click *Certificate Templates*, and then click *Add*.
 - g. Click *Close*, and then click *OK*.
2. A duplicate of the *Users certificate* template is now created and named *Archive User*. This is a shortcut to creating a template with permissions that will allow both *Domain Administrator* and *Domain User* certificate enrollments. The template is then modified so that certificate enrollments made using this template will enable both key archival and the ability to use Safenet as a CSP.
 - a. In the console tree, click *Certificate Templates*.
 - b. In the details pane, right-click the *User* template, and click *Duplicate Template*.
 - c. In the *Properties of New Template* dialog box, in the *General* tab, in the *Template display name* box, type *Archive User*.
 - d. In the *Request Handling* tab, enable the *Archive subject's encryption private key* option (see the screen shot below).

The *archive key* option makes it possible for a Key Recovery Agent to recover the private key from the certificate store.



- e. Click the *CSPs* button to enable HSM key storage using one or more Safenet CSPs.



The dialog box allows selection of particular CSPs or all CSPs may be enabled by selecting the appropriate radio button.

Typically, only the *Safenet RSA Full Cryptographic Provider* is required. The *SChannel Provider* is only needed where SSL processing will be carried out.

- f. After finalizing selections click *OK* and *OK again* to apply changes and close the dialog boxes.
- g. Close the console without saving changes.

Task 5—Acquiring a User certificate that has an archived key

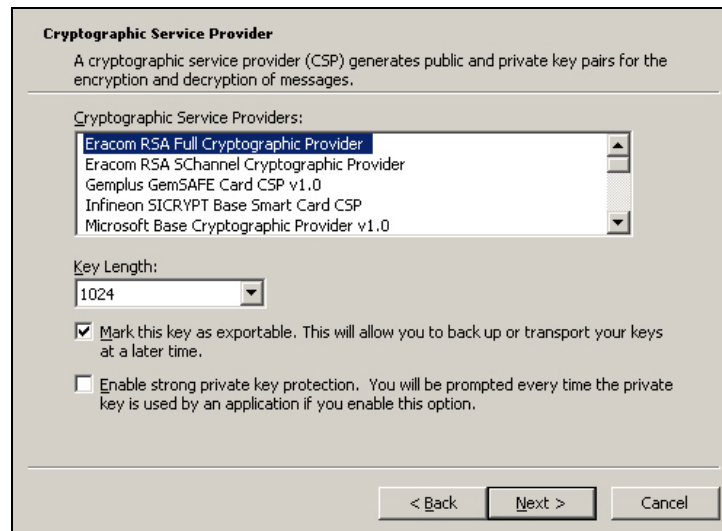
In this series of tasks, you will configure the certification authority (CA) to issue *Archive User* certificates. Using a newly created account, you will act as a user to acquire an

Archive User certificate from the CA and record the certificate's serial number for later use.

1. Configure CA to issue the new *Archive User* certificate template.
 - a. Ensure that you are logged on as the administrator.
 - b. From *Administrative Tools*, open *Certification Authority*.
 - c. In the console tree, double-click the CA name, and then click *Certificates Templates*.
 - d. Right-click *Certificate Templates*, click *New*, and then click *Certificate Template to Issue*.
 - e. In *Enable Certificate Templates*, click *Archive User* and then click *OK*.
 - f. The *Archive User* certificate template now appears in the details pane.
 - g. Close *Certification Authority*.
2. Create a new user account
 - a. In *Administrative Tools*, open *Active Directory Users and Computers*.
 - b. Double-click the domain.
 - c. Select *Users* then click the *Create a new user in the current container* button.
 - d. Complete the following fields to create a user account.
 - First name
 - Last name
 - User logon name e.g. JSmith@xxxx.com
 - User logon name (pre-Windows 2000) e.g. JSmith
 - Password
 - e. Click *Next*, and then click *Finish*.
 - f. Double-click the new user account, select the *General* tab and enter the email address. e.g. JSmith@xxxx.com. This is required if the option to include the email name is set in the template used to create the user (*Subject Name* tab).
 - g. For the purpose of demonstration here, add the user to the *Server Operators* group so they are able to log on locally to the domain controller. This would not normally be required.
 - Select the *Member of* tab.
 - Click *Add*, in *Select Groups*, type *Server Operators*, click *Check Names*, and then click *OK*.

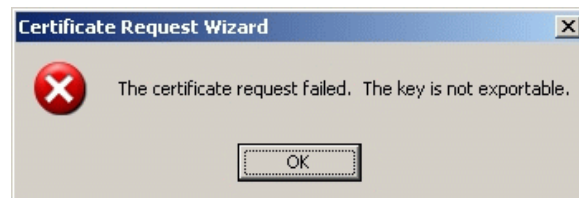
- Click *OK* to close *Properties*.
 - h. Close *Active Directory Users and Computers*.
 - i. Close all open windows and log off the computer.
3. Log in as the user and open the *Certificates* console.
 - a. Log on as the user.
 - b. On the taskbar, click the *Start* button, and then click *Run*.
 - c. In *Run*, type *mmc*, and then click *OK*.
 - d. From the *File* menu, click *Add/Remove Snap-in*.
 - e. In *Add/Remove Snap-in*, click *Add*.
 - f. In *Add Stand-alone Snap-in*, click *Certificates*, click *Add*, and then click *Close* to close the *Add Stand-alone Snap-in* dialog box.
 - g. Click *OK* to close the *Add/Remove Snap-in* dialog box.
 4. Use the *Certificates MMC* to acquire an *Archive User* certificate
 - a. In the newly-created MMC console, in the console tree, double-click *Certificates (Current User)*.
 - b. In the console tree, right-click *Personal*, click *All Tasks*, and then click *Request New Certificate*.
 - c. In the *Certificate Request Wizard*, click *Next*.
 - d. Under *Certificate types*, select *Archive User* and check the *Advanced* checkbox. Then click *Next*.

- e. On the *CSP* page that now displays (see example below) choose the Safenet provider for HSM key storage and any other appropriate settings such as *Key is Exportable* etc. Then click *Next* and *Next* again.



- f. In *Friendly name*, type *Archive User*, and then click *Next*.
- g. On *Completing the Certificate Request Wizard*, click *Finish*.

Note: If the dialog box shown below displays, the most likely cause of the problem is that the *Allow Clear Export of Private Keys* flag has not been set. See the section *Enabling Private Key Clear Export* in *Chapter 4, Administrative Tasks* for further details.



- h. Double-click *Personal*, and then click *Certificates*.
- i. In the details pane, double-click the certificate with the friendly name of *Archive User*.
- j. In *Certificate*, click the *Details* tab.
Note that the certificate template used to generate this certificate was *Archive User*, then click *OK*.
- k. Close the new console without saving changes
- l. Close all windows and log off of the computer.

Private Key Recovery Example

Here are the tasks required to recover a lost private key previously archived using a Microsoft certification authority (CA).

1. Perform key recovery
2. Import the recovered private key

Task 1—Performing a Key Recovery

In this series of tasks, you will perform a key recovery by using Certutil.exe. For more information on *Certutil*, see your Microsoft documentation.

1. Ensure that the private key is recoverable by viewing the *Archived Key* column in the Certification Authority console and obtain the certificate serial number required for recovery.
 - a. Log on as the administrator.
 - b. From *Administrative Tools*, open *Certification Authority*.
 - c. In the console tree, double-click the CA, and then click *Issued Certificates*.
 - d. From the *View* menu, click *Add/Remove Columns*.
 - e. In *Add/Remove Columns*, in *Available Column*, select *Archived Key*, and then click *Add*. *Archived Key* should now appear in *Displayed Columns*.
 - f. Click *OK* and then, in the details pane, scroll to the right and confirm that the last issued certificate to The user has a *Yes* value in the *Archived Key* column.

Note: A certificate template must have been modified so that the *Archive bit* and *Mark Private Key as Exportable* attributes were enabled. The private key is only recoverable if there is data in the *Archived Key* column.
 - g. Double-click the *Archive User* certificate.
 - h. Click the *Details* tab

Write down the serial number of the certificate. (Do not include spacing between digit pairs.) This is required for recovery.

The serial number is a 20 character, hexadecimal string. The serial number of the private key is the same as the serial number of the certificate. For the purposes of this walkthrough, the serial number is referred to as *serialnumber*.

 - i. Click *OK*.
 - j. Close *Certification Authority*.

2. Recover the private key into a BLOB output file by using certutil.exe.

- a. On the taskbar, click the *Start* button, click *Run*, type *cmd*, then click *OK*.
A command prompt window opens.
 - b. Type *cd * and then press *ENTER*.
 - c. Ensure that you are in the *c:* directory.
 - d. At the command prompt, type:
Certutil -getkey serialnumber outputblob
 - e. At the command prompt, type *dir outputblob*
Note: If the file *outputblob* does not exist, you probably typed the serial number incorrectly for the certificate.

The *outputblob* file is a PKCS#7 file containing the KRA certificates and the user certificate and chain. The inner content is an encrypted PKCS#7 containing the private key (encrypted to the KRA certificates).
3. Recover the original private/public key pair using Certutil.exe
 - a. On the taskbar, click the *Start* button, click *Run*, type *cmd*, then click *OK*.
A command prompt window opens.
 - b. At a command prompt, type:
Certutil -recoverkey outputblob <username>.pfx
 - c. When prompted, enter the following information:
 - Enter new password: *password*
 - Confirm new password: *password*
 - d. Type *exit*, and then press *ENTER*.
 - e. Close all windows and log off as the current user.

Task 2—Importing the recovered private key

Restoration of the recovered private key to the users certificate store by importing the *<username>.pfx* file.

1. Log on as the user and start the Certificates mmc.
 - a. Log on as the user.
 - b. On the taskbar, click the *Start* button, and then click *Run*.

- c. In *Run*, type *mmc*, and then click *OK*.
 - d. On the *File* menu, click *Add/Remove Snap-in*.
 - e. In *Add/Remove Snap-in*, click *Add*.
 - f. In *Add Standalone Snap-in*, click *Certificates*, click *Add*, and then click *Close* and *OK*.
2. Delete all certificates issued by the CA to simulate a re-installed computer.
 - a. Right-click *Certificates - Current User*, and then click *Find Certificates*.
 - b. In *Find Certificates*, in *Contains*, type the CA and then click *Find Now*.
 - c. On the *Edit* menu, click *Select All*.
 - d. On the *File* menu, click *Delete*.
 - e. In *Certificates*, click *Yes*.
 - f. In *Root Certificate Store*, click *Yes*.
 - g. Close *Find Certificates*.
 3. Import the certificate at *c:\ <username>.pfx* and let the certificates be placed automatically.
 - a. In the console tree, right-click *Personal* and then click *All Tasks* and then click *Import*.
 - b. In the *Certificate Import Wizard*, click *Next*.
 - c. On *Files to Import*, in the *File name* box, type *c:\ <username>.pfx*, and then click *Next*.
 - d. In *Password*, type *password* and then click *Next*.
 - e. On *Certificate Store*, click *Automatically select the certificate store based on the type of certificate* and then click *Next*.
 - f. On *Completing the Certificate Import Wizard*, click *Finish*.
 - g. If the *Root Certificate Store* dialog box appears, click *Yes*.
 - h. In *Certificate Wizard Import*, click *OK*.

Two certificates were imported. The *Archive User* certificate for the user is located in the *Personal* certificates store and the CA certificate is located in the *Trusted Root Certification Authorities* store.

4. Verify the serial number of the imported certificate.
 - a. In the console tree, double-click *Personal* and then click *Certificates*.
 - b. Double-click the certificate
 - c. In *Certificate*, click the *Details* tab. Verify that the serial number matches the original.
 - d. Close all open windows and log off.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix B

Integration With IIS

One of the uses for the Microsoft Cryptographic API is for Secure Socket Layer (SSL) processing. This section explains the steps necessary to configure and use ProtectToolkit M in conjunction with IIS.

Prior to performing any of the following, please ensure that ProtectToolkit M is correctly installed and configured. For details, please refer to Chapters 1 to 3.

Creating a Certificate Overview

In order for ProtectToolkit M and the HSM to be used for SSL processing, a certificate needs to be setup that specifies the details of the ProtectToolkit M machine.

There are multiple methods in which a certificate for the machine can be created. For example:

1. Using IIS
2. Using Microsoft's CA to create the certificate.
3. Using the supplied command line utility **createcert**, which creates a self-signed certificate. Note that self signed certificates are only of use for testing purposes.

Creating a Certificate Using IIS

When using IIS to install a certificate on the host machine, the following has to be performed:

1. Creating a certificate request
2. Sending the certificate request to be signed by a CA
3. Installing the signed certificate into IIS

To create a certificate request using IIS:

- Start the Internet Services Manager from the Administrative Tools menu.
- Highlight the "Default Web Site" entry, and right-click to open a context menu. Select Properties. The default web site properties dialog opens. Select the Directory Security Tab.
- Click on the Server Certificate button. This will start the IIS Certificate Wizard.
- Choose "Create a new certificate" from the available options and press Next to continue.

- Choose “Prepare the request now, but send it later” from the available options and press Next to continue.
- In Windows 2003 it will be necessary to select Safenet as the security provider. (In Windows 2000 this is the default setting). For Windows 2003 users only, on the *Name and Security Settings* page that now displays, check the *Select cryptographic service provider (CSP) for this certificate* checkbox. Click *Next* to continue.
- For Windows 2003 users only, select an Safenet CSP from the list displayed on the *Available Providers* page.
- Continue to follow the on-screen prompts until the certificate request is completed.

The IIS Certificate Wizard will create the certificate request as a file. You should now forward this file onto your CA in order to have it signed. The CA will return a new file which will be the signed certificate.

Refer to the section entitled *Installing a Certificate for use with IIS* for details on how to proceed to install the signed certificate.

Creating a Certificate Using the Microsoft CA server

The Microsoft CA server provides a standard internet browser interface for the creation of certificates.

To create a certificate using MS CA server:

Note: Before starting the following procedure, ensure that the current logged on user has Windows administrator privileges and has a valid keyset.

- Start the MS CA services interface by opening your web browser and specifying the Microsoft CA server URL. For example: `http://hostname/certsrv`

You will be presented with the opening screen for CA services.

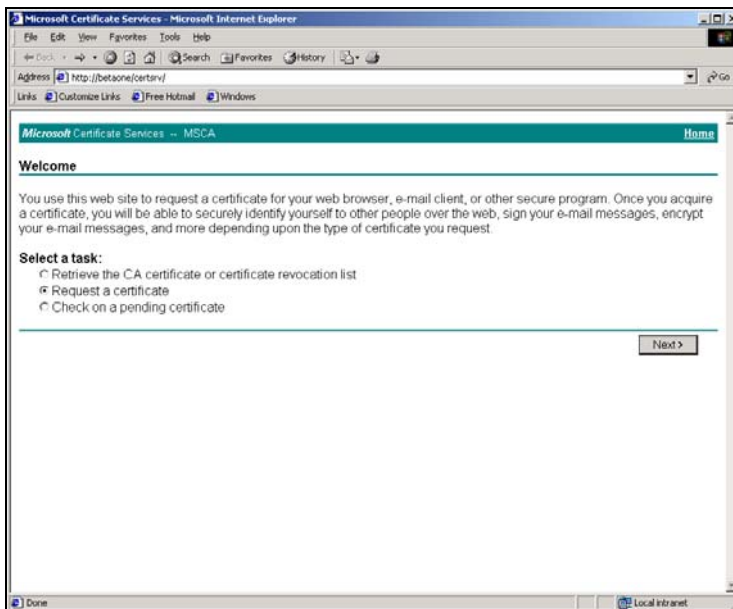


Figure 15 - CA Services Welcome screen

- Select the “Request a certificate” option and press Next to continue. You will then be prompted to select the request type.

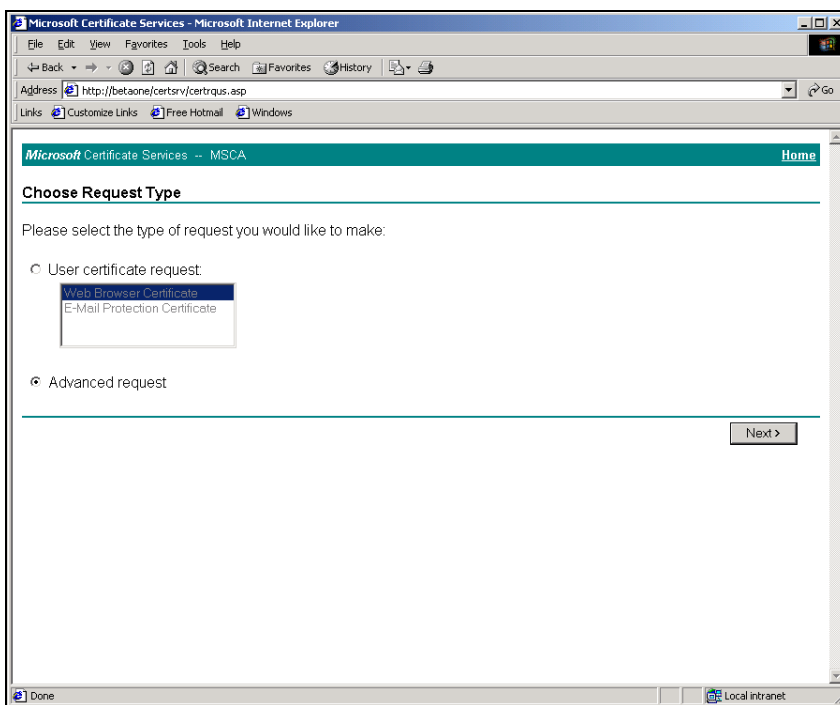


Figure 16 - Choose Request Type screen

- Choose, “Advanced request” and press Next to continue. You will be presented with the Advanced Certificate Requests screen.

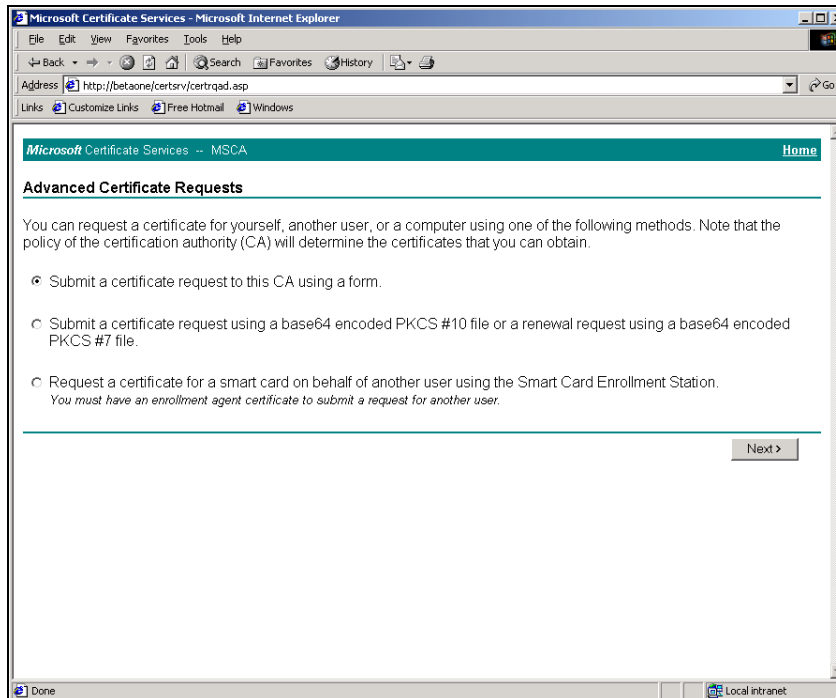


Figure 17 - Advanced Certificate Request screen

- Select “Submit a certificate request to this CA using a form”, and press Next to continue. You will be presented with a form to input the certificate details.

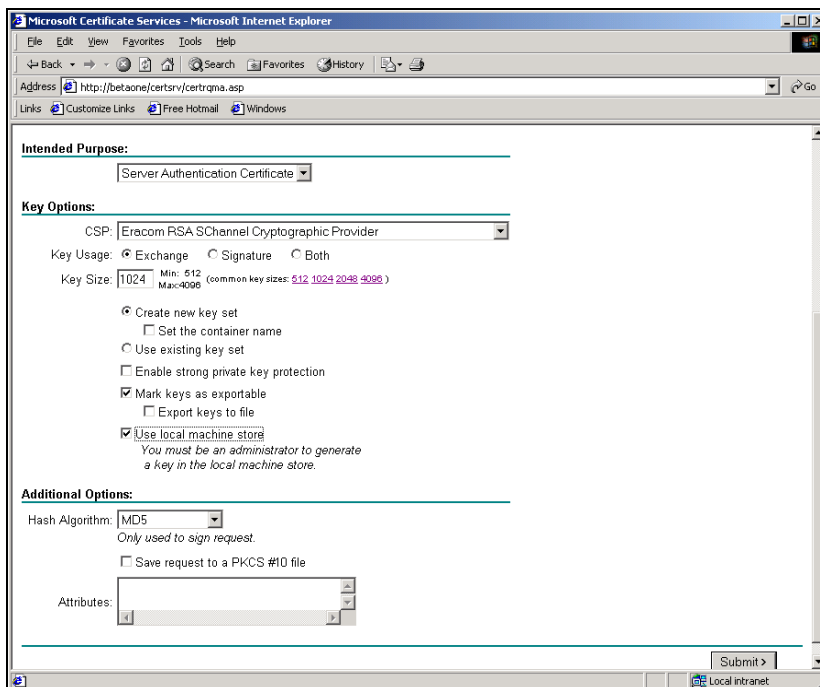


Figure 18 - Certificate Details input form

- Enter the details for the certificate into the fields provided.

For the Certificate Name, enter the host machines name. This can be ascertained by executing the standard Windows command **hostname** from a command prompt.

For the Intended Purpose, choose “Server Authentication Certificate”.

For the Key Options: - choose “Safenet RSA SChannel Cryptographic Provider” as the CSP,

- choose “Exchange” as the Key Usage,

- enter the Key Size as required, eg. “1024”

- select “Create new key set”

- if you want to be able to backup the keys associated with the certificate at a later date, choose “Mark keys as exportable”

- choose “Use local machine store”

For the Additional Options: - choose Hash Algorithm, e.g. “MD5”

Note: If the current logged on user’s keyset does not exist when the Safenet CSP is selected, the “Hash Algorithm” list box at the bottom of the screen will be empty. Should this be the case, abort this operation and create a keyset for the currently logged on user before attempting this task again.

- Press the Submit button when you have confirmed your inputs. If the Microsoft CA was configured to “Auto Issue” certificates, you will then be presented with the Certificate Issued dialog.

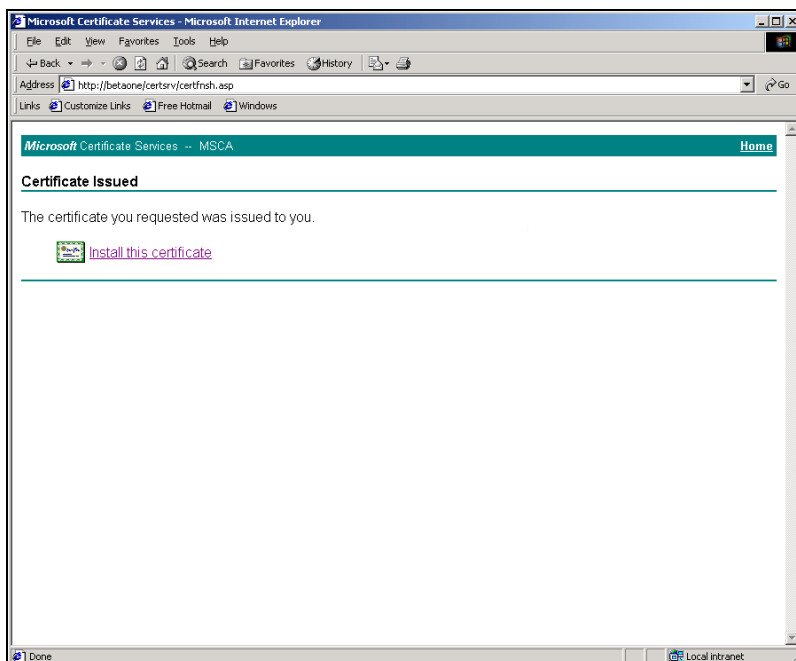


Figure 19 - Certificate Issued screen

If CA Services is not configured to auto-issue certificates, the dialog will state that your certificate request is pending. You will have to check on the status of the certificate using the CA services at a later time. When the certificate is ready, you will be presented with the Certificate Issued screen.

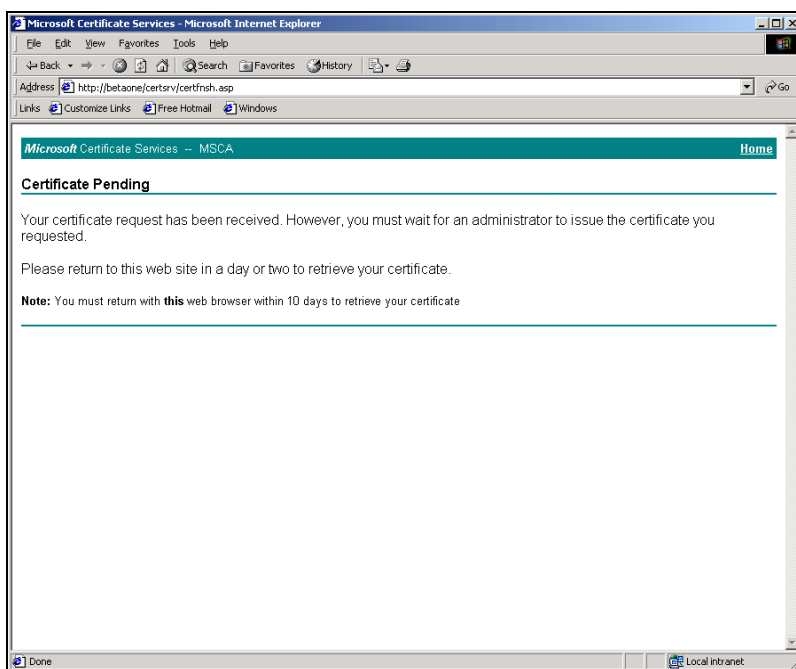


Figure 20 - Certificate Pending screen

- Click on “Install this certificate” to complete the certificate request and installation.

Creating a Certificate Using the createcert utility

The **createcert** utility is provided as a simple means to create a self signed certificate for the ProtectToolkit M host machine.

Note that these certificates are intended for development and testing purposes only. Also ensure that the current logged on user has Windows administrator privileges and has a valid keyset.

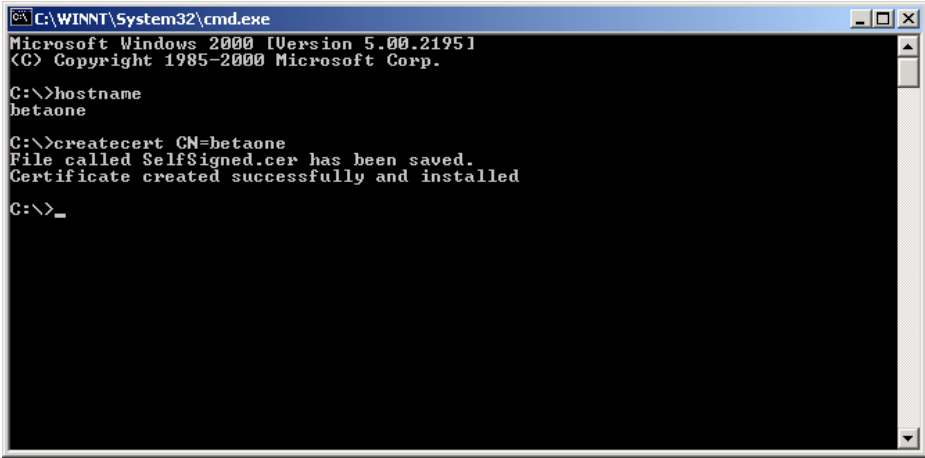
You will need to know the machine name for the ProtectToolkit M system. This can be ascertained by executing the standard Windows command “**hostname**” from a command prompt.

To create a self-signed certificate using createcert utility:

- From a command prompt, execute the utility **createcert**, specifying the machine name. For example, if the machine name is “betaone”, the command would be as follows:

```
createcert CN=betaone
```

Successful execution of the above will result in an RSA key pair being created, as well as a certificate which is saved in the file “selfsigned.cer”. This certificate is then automatically installed ready for IIS use.

A screenshot of a Windows command prompt window. The title bar reads "C:\WINNT\System32\cmd.exe". The window content shows the following text:

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>hostname
betaone

C:\>createcert CN=betaone
File called SelfSigned.cer has been saved.
Certificate created successfully and installed

C:\>_
```

Figure 21 - Creating a self signed certificate

Installing a Certificate for use with IIS

In order to make use of the certificate in IIS, it will need to be assigned to a website.

To install the certificate with IIS:

- Start the Internet Services Manager from the Windows Start/Programs/Administrative Tools/ menu.

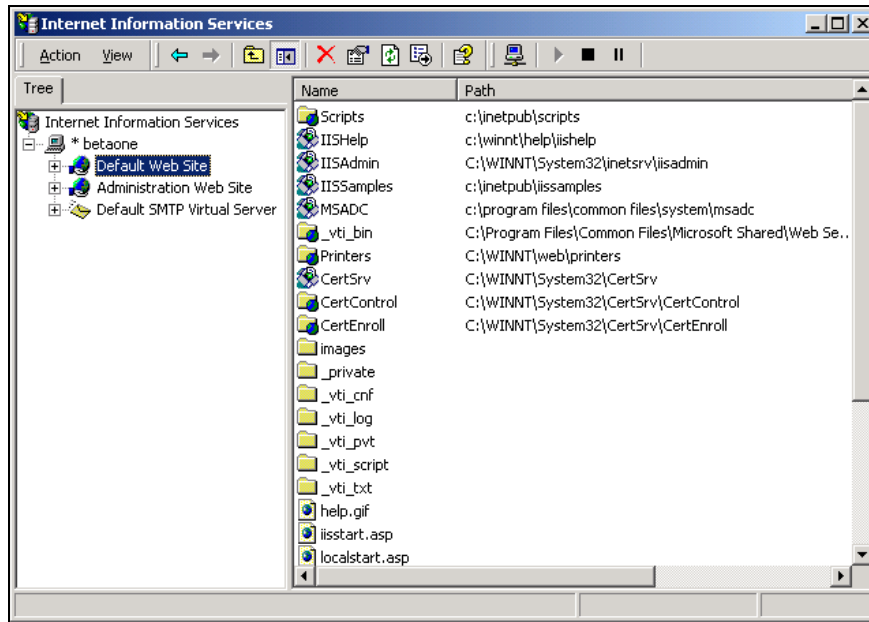


Figure 22 - IIS Manager dialog

- Highlight the “Default Web Site” entry, and right-click to open a context menu. Select Properties. The default web site properties dialog opens. Select the Directory Security Tab.

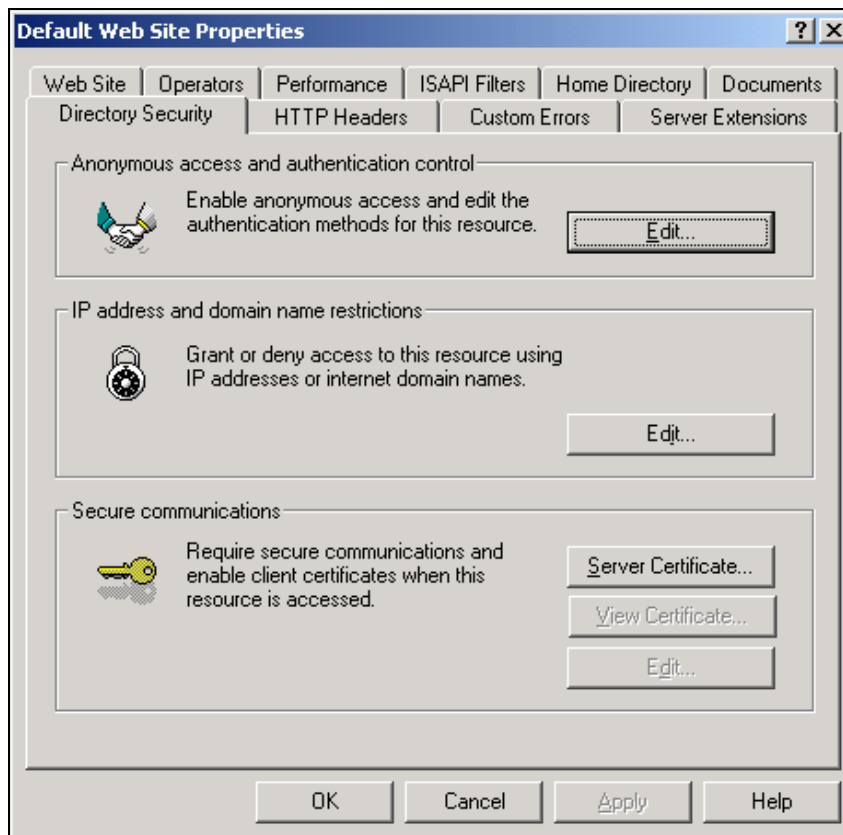


Figure 23 - Default Web Site Properties dialog

- Click on the Server Certificate button. This will start the IIS Certificate Wizard.
- Depending on how the certificate was generated, the IIS Certificate Wizard will prompt for the following:

If there is an outstanding certificate request:

- The IIS Certificate Wizard will inform the user that there is a pending certificate request.
- When prompted, choose “Process the pending request and install the certificate”.
- Continue to follow the on-screen prompts until the certificate is installed.

If there is no outstanding certificate request:

- the IIS Certificate Wizard will prompt the user to assign a certificate using one of three possible methods.
- Choose “Assign an existing certificate” from the available options and press Next to continue.

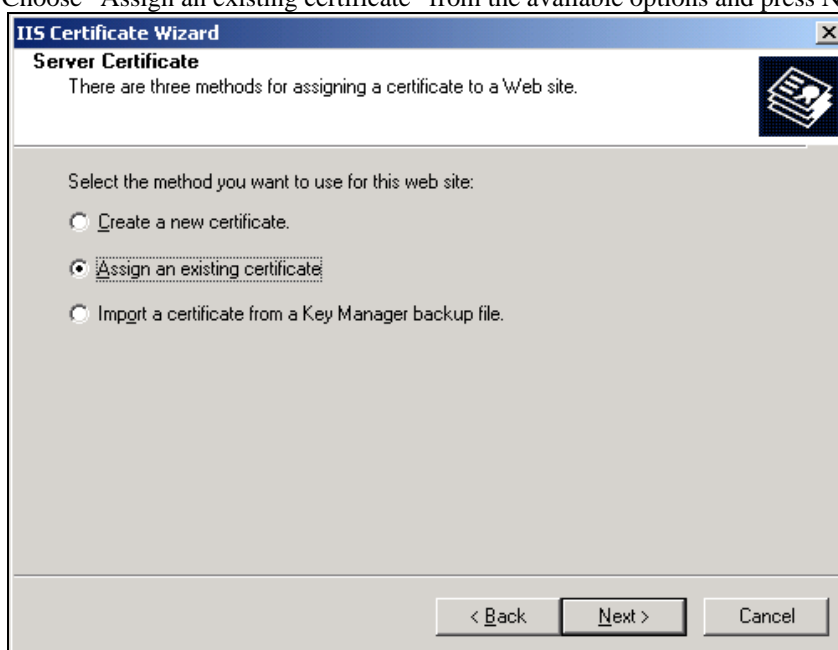


Figure 24 - Assign an existing certificate

The example below shows that there are three certificates currently installed on the machine.

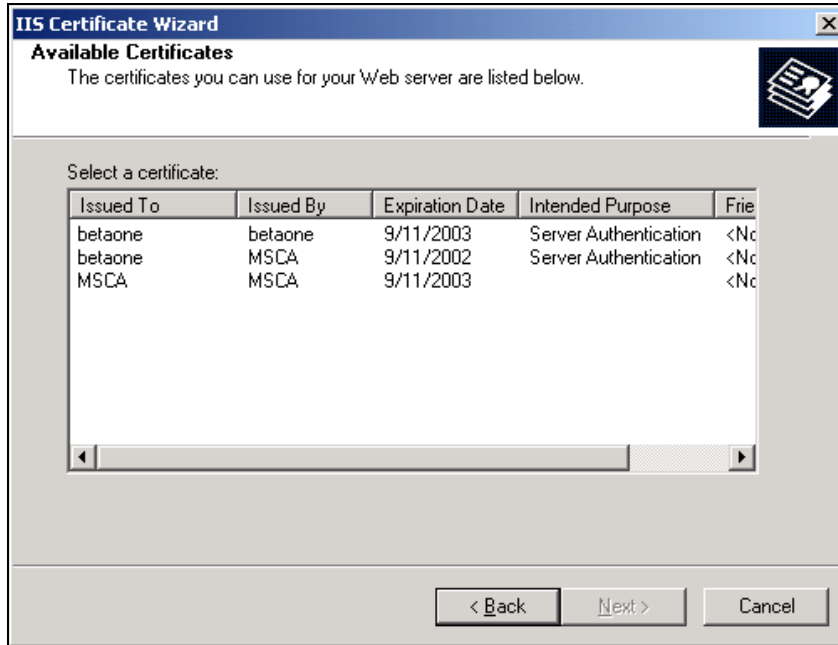


Figure 25 - Available Certificates dialog

The first listing is a self signed certificate, created using the **createcert** utility. Note that the “Issued To” and “Issued By” fields are the same. If you decide to install this type of certificate, you will receive a “Security Alert” when trying to access the web site using Microsoft Internet Explorer. When using self signed certificates, you will not be able to acquire a secure connection using Netscape 6.0. For more details please refer to *Chapter 7, Troubleshooting*.

The second listing in the example is a certificate generated using Microsoft CA. Note that in this case the “Issued To” and “Issued By” fields reflect that this is not a self signed certificate.

The last listing in the example dialog is the Microsoft CA certificate used to sign certificate requests. This certificate appears in the list because the CA was set up on the same machine as the one being configured.

- Select the certificate you wish to install and press Next to proceed with the certificate installation.

This concludes the certificate installation for IIS, and SSL connections can now be made to the default web site.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix C

PKCS#11 Attributes

Objects as described by PKCS#11 consist of a number of attributes that define both the object and its access policy. In general the ProtectToolkit M system will define the object's value attributes however the access policy should be provided by the user based on the particular requirements. The following attribute descriptions are intended to assist with these decisions.

CKA_LABEL

This attribute specifies a textual label for an object. This label is used to assist in differentiating the various objects stored on a token. Note that although ProtectToolkit M does not require this attribute to be unique, various tools may do so.

CKA_CLASS

This attribute is assigned by the system when an object is created. There are a number of classes in common use:

CKO_PUBLIC_KEY
CKO_PRIVATE_KEY
CKO_SECRET_KEY
CKO_CERTIFICATE
CKO_CERTIFICATE_REQUEST
CKO_DATA

CKA_KEY_TYPE

This attribute specifies the key type associated with the object. There are many key types supported by ProtectToolkit M. For example:

CKK_AES, CKK_DES, CKK_DES2, CKK_DES3, CKK_RSA, CKK_DSA

CKA_ENCRYPT

CKA_DECRYPT

CKA_SIGN

CKA_VERIFY

CKA_WRAP

CKA_UNWRAP

The previous attributes describe the cryptographic operations the key may be used for. Careful consideration should be given when assigning these attributes to avoid key misuse.

CKA_IMPORT

CKA_IMPORT is similar to the standard CKA_UNWRAP attribute. It is used to determine if a given key can be used to unwrap encrypted key material. The important difference between these attributes and their standard counterparts is that if CKA_IMPORT is set to True and CKA_UNWRAP attribute is set to False, then the only unwrap mechanism that can be used is CKM_WRAPKEY_DES3_CBC. With this combination, the error code CKR_MECHANISM_INVALID will be returned for all other mechanisms.

CKA_EXPORT

This attribute is similar to the CKA_WRAP attribute in that it specifies that the key may be used to encrypt a second key so that it may be extracted from the adapter in an encrypted form. Unlike the CKA_WRAP attribute however only the Security Officer may specify this attribute.

CKA_SENSITIVE

This attribute specifies that the key object may not be extracted from the token in the clear. Generally this attribute should be specified to ensure the key material is not exposed. When the 'No Clear PINs' flag is set only sensitive keys may be created on the adapter.

CKA_EXTRACTABLE

CKA_EXPORTABLE

These attributes are used to specify that the key may be extracted from the token in an encrypted (e.g. wrapped) form. These attributes determine how the key may be backed up. Please consult the key backup section in *Chapter 4* for more information.

Appendix D

KAAN Professional Smart Card Reader

The KAAAN Professional smart card terminal meets ZKA specifications. In addition to being another supported smart card reader, it is used by ProtectToolkit M as a key entry device.

To enter keys using this device use the `ctkm` console utility `create` command, in conjunction with the `-p` (or `--pinpad`) switch. The command syntax is:

```
ctkm c -p -t<keyType> -k<numComponents> -n<keyName> -a<keyAttributes>
```

Enter each hex byte of the key component into the Kobil KAAAN device as a three digit decimal value. The table over the page can be used to convert the hex byte values to decimal for key entry. Its use is best explained by way of an example.

Example key in hex form: *00A3442B*.

1. The hex digits must be taken in pairs, representing one byte (octet) of binary data, and converted to their decimal equivalents using the table.
2. For each pair look up the decimal equivalent value.

For the example key:

Value in Hex Form	Decimal Equivalent
00	000
A3	163
44	068
2B	043

Thus the sequence of digits to be entered using the keypad in this example is:
000163068043.

Hexadecimal to Decimal Conversion Table

Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec
00	000	20	032	40	064	60	096	80	128	A0	160	C0	192	E0	224
01	001	21	033	41	065	61	097	81	129	A1	161	C1	193	E1	225
02	002	22	034	42	066	62	098	82	130	A2	162	C2	194	E2	226
03	003	23	035	43	067	63	099	83	131	A3	163	C3	195	E3	227
04	004	24	036	44	068	64	100	84	132	A4	164	C4	196	E4	228
05	005	25	037	45	069	65	101	85	133	A5	165	C5	197	E5	229
06	006	26	038	46	070	66	102	86	134	A6	166	C6	198	E6	230
07	007	27	039	47	071	67	103	87	135	A7	167	C7	199	E7	231
08	008	28	040	48	072	68	104	88	136	A8	168	C8	200	E8	232
09	009	29	041	49	073	69	105	89	137	A9	169	C9	201	E9	233
0A	010	2A	042	4A	074	6A	106	8A	138	AA	170	CA	202	EA	234
0B	011	2B	043	4B	075	6B	107	8B	139	AB	171	CB	203	EB	235
0C	012	2C	044	4C	076	6C	108	8C	140	AC	172	CC	204	EC	236
0D	013	2D	045	4D	077	6D	109	8D	141	AD	173	CD	205	ED	237
0E	014	2E	046	4E	078	6E	110	8E	142	AE	174	CE	206	EE	238
0F	015	2F	047	4F	079	6F	111	8F	143	AF	175	CF	207	EF	239
10	016	30	048	50	080	70	112	90	144	B0	176	D0	208	F0	240
11	017	31	049	51	081	71	113	91	145	B1	177	D1	209	F1	241
12	018	32	050	52	082	72	114	92	146	B2	178	D2	210	F2	242
13	019	33	051	53	083	73	115	93	147	B3	179	D3	211	F3	243
14	020	34	052	54	084	74	116	94	148	B4	180	D4	212	F4	244
15	021	35	053	55	085	75	117	95	149	B5	181	D5	213	F5	245
16	022	36	054	56	086	76	118	96	150	B6	182	D6	214	F6	246
17	023	37	055	57	087	77	119	97	151	B7	183	D7	215	F7	247
18	024	38	056	58	088	78	120	98	152	B8	184	D8	216	F8	248
19	025	39	057	59	089	79	121	99	153	B9	185	D9	217	F9	249
1A	026	3A	058	5A	090	7A	122	9A	154	BA	186	DA	218	FA	250
1B	027	3B	059	5B	091	7B	123	9B	155	BB	187	DB	219	FB	251
1C	028	3C	060	5C	092	7C	124	9C	156	BC	188	DC	220	FC	252
1D	029	3D	061	5D	093	7D	125	9D	157	BD	189	DD	221	FD	253
1E	030	3E	062	5E	094	7E	126	9E	158	BE	190	DE	222	FE	254
1F	031	3F	063	5F	095	7F	127	9F	159	BF	191	DF	223	FF	255

Appendix E

Work Load Distribution (WLD)

Introduction

If required, more than one hardware security module (HSM) can be used to implement Work Load Distribution (WLD).

Benefits from Using WLD

WLD allows work to be balanced across a system by transferring units of work among HSM processing modules during execution. The demand placed on any particular processing module is thereby reduced. This results in an increase in the overall throughput of processing tasks for the system as a whole.

Utilization of multiple HSMs under WLD also provides redundancy in that if a HSM goes down, with the exception of the *master HSM*, the work will be shared amongst the remaining operational HSMs automatically. If the master HSM goes down this will most likely cause system failure. See the section [Admin Token Cannot Be Distributed - Single Point of Failure](#) below for further information.

WLD Architecture

The SafeNet ProtectToolkit C model is used as the basis for implementing WLD for ProtectToolkit M systems and this model must be understood if WLD is to be implemented and maintained successfully.

There are a number of integral components within a system that deploys WLD using the ProtectToolkit C model, as follows.

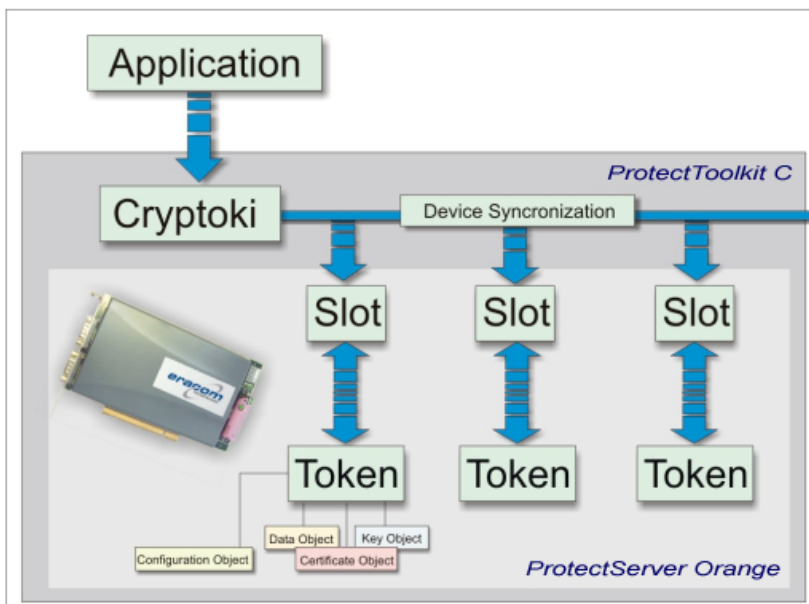
- A distribution engine portions the requests for work, which it then distributes to an appropriate HSM to perform.
- The distribution engine implements a distribution scheme to determine which HSM is selected.
- The tokens that are utilized within the scheme must be replicated across the HSMs, as appropriate to the system design.
- A good system design should address throughput requirements, resource portioning and fault tolerance/disaster recovery.
- The ctident utility provides the mechanism to establish trust between HSMs that share tokens. The ctkmu utility provides the mechanism to replicate a token once trust has been established.

The ProtectToolkit C Model

The ProtectToolkit C model is the SafeNet implementation of the PKCS #11 Cryptographic Token Interface Standard.

In this model, a *slot* represents a device interface and a *token* represents an actual cryptographic device such as a HSM processing module. In the first instance these are created automatically when keysets are created as part of ProtectToolkit M initial configuration.

The diagram below illustrates how the PKCS #11 interface is used by an application to communicate its requests to tokens.



A number of different slot types are supported. ProtectToolkit M users will encounter: *user slots* and *admin slots*.

- User slots are created by the Administrator for each user of a HSM and are designated for standard application usage. Each HSM may have a configurable number of user slots.
- The admin slot is designated for the administrator and is used for configuration and administration of the HSM. Each HSM will have a single admin slot.

As shown in the diagram above, each token may contain a number of *objects*. The PKCS #11 standard allows for different types of objects which are classified as follows:

- Data objects, which are defined by an application
- Certificate objects, which represent digital certificates such as X.509, for example

- Key objects, which can be public, private or secret cryptographic keys

Each object in the system is comprised of a number of *attributes*. These attributes describe the actual object as well as the *access policy* for that object. For example, each object may be classified as *public* or *private*; this classification determines who may access the object. A *public object* is visible to any user (or application), whereas a *private object* is only visible once the user is authenticated to the token where that object is stored.

Administration Objects

In addition to the object classes defined within PKCS #11, ProtectToolkit C introduces a new set of objects known as *administration objects*. The administration objects reside on a special token referred to as the *Admin token*. This token in turn resides only in the *Admin slot*.

The administration objects represent the hardware and contain HSM configuration settings. The administration objects can be queried by the application and some can be modified by an administrator. The default administration objects are automatically created on the Admin token when ProtectToolkit C initializes.

User Roles

As part of the ProtectToolkit C configuration process different *user roles* are assigned to those responsible for the application's administration and use. These roles are:

- *Security Officer (SO)*
- *Token Owner or User*
- *Administration Security Officer (ASO)* and
- *Administrator*

Standard PKCS #11 defines the first two of these, the *Security Officer (SO)* and the *Token Owner or User*. Each slot and its associated token will have an SO and a User, each with their own respective PINs.

- A Security Officer grants and revokes access to a token and assists with key backups
- A Token Owner uses the token for the application

Two additional roles are defined that are only available on the Admin token. The holders of these roles handle HSM level administration and management. These are the *Administration Security Officer (ASO)* and the *Administrator*. These roles effectively mirror their standard PKCS #11 counterparts.

WLD Limitations

Read-Only

Using ProtectToolkit M 2.0 as a CSP under the current version of WLD is severely limited. The current version of WLD does not support write/create operations. Because of this, the CSP cannot be used to create certificates when in WLD mode, as this involves creating a key pair. The CSP can, however, be used to sign certificate requests that have been generated by a client, provided the client also generated their own key pair.

Admin Token Cannot Be Distributed - Single Point of Failure

The admin token of HSMs contains information relevant to the HSMs configuration. WLD does not allow replication of admin tokens. This is not expected to change under future versions of WLD. ProtectToolkit M has the notion of a 'secure configuration', which is a collection of configuration items that are stored on a data object on the admin token and are therefore readable but not modifiable by anyone other than the administrator. Currently, the only relevant secure configuration item is 'clear export'. This specifies whether or not keys may be exported in the clear. Since admin tokens cannot be replicated, it is necessary to expose the admin token of one of the HSMs in the array, called the *master HSM*. If the master HSM fails, the admin token will no longer be available, which will most likely cause system failure.

Setting Up a WLD System

Background Information

This section covers the specifics required to complete implementation of WLD. An understanding of the technicalities covered in this section is not needed however further background information, which may be of interest to some users, can be found in the [WLD Technical Details](#) section.

Step 1: HSM Installation

Install the HSMs to be configured as part of the WLD system and ensure that they are available to the host.

The HSMs that may be used in a Work Load Distribution system include the Orange Series PCI and Network HSMs. A PCI HSM is defined as a HSM located in the same machine as the ProtectToolkit C installation and refers to the ProtectServer Orange HSM. A network HSM is defined as a HSM not located in the same the machine as the ProtectToolkit C installation but accessible via a communications link (typically a TCP/IP LAN/WAN) and refers to the ProtectServer Orange – External HSMs. It is anticipated that WLD trust management will be supported on the Protect Host Orange in the future.

Support for other SafeNet HSMs is not currently implemented and integration with non SafeNet HSMs is not possible due to the proprietary nature of the implementation.

A Work Load Distribution system may consist of any number of PCI HSMs or any number of network HSMs. The use of PCI and network HSMs concurrently is not yet supported.

Step 2: Slot Creation and Token Replication

To configure ProtectToolkit M to operate using WLD, it is necessary to first configure a single HSM, called the master HSM, then replicate its slots to all other HSMs.

1. To configure the master HSM, the other HSMs must either be halted (PCI adapter card HSMs installed in the host system) or removed from the *serverlist* (network connected standalone HSMs).
2. Once only a single HSM is recognized, set up all required key sets following standard procedure for ProtectToolkit M.
3. When all keysets have been established on the master HSM, enable all HSMs so they are once again visible.
4. Each HSM requires a *HSM ID key pair*. The *ctident* tool is used to generate these. The command is `ctident gen all`. Running this command should produce output similar to the following:

```
ProtectToolkit C HSM Identity Key Management Utility $Revision
Copyright (c) SafeNet, Inc.
Generating HSM ID key pair for dev:0 sn:2729 slot:3 ...
Please enter the Admin SO pin for device dev:0 sn:2729 slot:3
Success
```

Where the last three lines occur for each HSM in the WLD array.

5. Once each HSM has a HSM ID key, trust needs to be established between the master HSM and the slaves so the token replication may take place. For more information about trust management see the [Trust Management](#) section below. Again, the *ctident* utility is used. The command is `ctident trust all all`. This command establishes trust between all the HSMs and not just the master and the slaves. This is more than is required however it is the simplest to implement. Running this command should produce output similar to the following:

```
ProtectToolkit C HSM Identity Key Management Utility $Revision:
1.2.2.13 $
Copyright (c) SafeNet, Inc.
Please enter the Admin SO pin for device dev:0 sn:2692 slot:1
dev:0 sn:2692 slot:1 trusting dev:1 sn:1129 slot:3 ... success
dev:0 sn:2692 slot:1 trusting dev:2 sn:1490 slot:5 ... success
```

```
dev:1 sn:1129 slot:3 trusting dev:0 sn:2692 slot:1 ... success
dev:1 sn:1129 slot:3 trusting dev:2 sn:1490 slot:5 ... success
dev:2 sn:1490 slot:5 trusting dev:0 sn:2692 slot:1 ... success
dev:2 sn:1490 slot:5 trusting dev:1 sn:1129 slot:3 ... success
```

6. Once trust has been established across all HSMs the next step is to prepare the slave HSMs to receive replicated tokens from the master HSM. This is done by creating slots on the slave HSMs to match those on the master HSM.

Use the *ctconf* tool. The command is:

```
ctconf -c X -a Y,
```

where X is the number of additional slots required on each HSM and Y is the HSM number where the slots will be created.

The *ctconf -v -a X* command may be used to determine how many slots a HSM has, where X is the zero based HSM number.

Each slave HSM would generally be set up to have the same number of slots as the master HSM. It possible however to vary the number of slots created as a way of varying the proportion of work allocated to each HSM.

7. Determine which tokens should be replicated to which slots by running the *ctkmu 1* command. This will produce output similar to the following:

```
ProtectToolkit C Key Management Utility $Revision: 3.10.2.9 $
Copyright (c) SafeNet, Inc.
Cryptoki Version = 2.10
Manufacturer = SafeNet, Inc.
MACHINE Keyset (Slot 0)
SYSTEM Keyset (Slot 1)
AdminToken (2692) (Slot 2)
<uninitialised token> (Slot 3)
<uninitialised token> (Slot 4)
AdminToken (1129) (Slot 5)
<uninitialised token> (Slot 6)
<uninitialised token> (Slot 7)
AdminToken (1490) (Slot 8)
```

Each HSMs Admin Token is its last token listed and all of a HSMs token are listed contiguously. So, from the output it is evident that HSM 0 has 3 slots (slot 0,1 and 2), HSM 1 has 3 slots (slot 3,4 and 5) and HSM 2 has 4 slots (slot 6, 7 and 8). This is how the configuration should be to replicate HSM 0's tokens. Token 0 would be replicated to slot 3 and 6, and token 1 would be replicated to slot 4 and 7.

8. Once all HSMs have the correct number of slots replicate the tokens and the keys sets they contain from the master HSM to all other HSMs.

The command to replicate the tokens is

```
ctkmu rt -sx -dy
```

where x is the token to replicate and y is the slot that will accept the replicated token.

Step 3: WLD Token Configuration

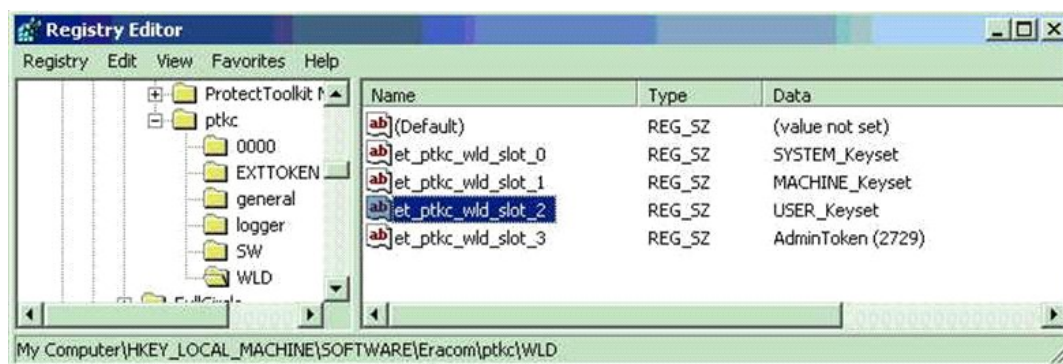
Now that the tokens have been replicated, the WLD tokens should be configured. This is done in the registry, the entries being made in the following key:

```
HKLM\Software\Safenet\ptkc\WLD\
```

An entry is made for each that is to be visible under WLD. Each entry must be named:

```
et_ptkc_wld_slot_X
```

where X is the **zero based** number of the slot. The value of the entry is the actual name of the slot. See the example below.



Step 4: Switch to WLD Mode

Now the ET_PTKC_GENERAL_LIBRARY_MODE configuration item may be set to WLD to switch the Cryptoki library to Work Load Distribution mode. This is done in the registry, the entries being made in the following key:

```
HKLM\Software\Safenet\ptkc\GENERAL\
```

WLD is the value assigned to the entry ET_PTKC_GENERAL_LIBRARY_MODE to switch the Cryptoki library to Work Load Distribution mode.

WLD System Management

Turning off WLD

To switch off Work Load Distribution mode, assign the value NORMAL to the ET_PTKC_GENERAL_LIBRARY_MODE configuration item. To do this:

1. Open *Regedit* and go to the *HKLM\Software\Safenet\ptkc\GENERAL* key.
2. The ET_PTKC_GENERAL_LIBRARY_MODE configuration item is an entry under this key. Assign the value NORMAL to it.

Changing Configuration Under WLD

Note that when the library is in WLD mode, no new objects should be created, as they will not be replicated across all HSMs.

To create any new objects, or add new keysets, turn the library mode back to normal, make the changes to the master HSM, and replicate any affected tokens to the other HSMs. The switch back to WLD mode.

WLD Technical Details

It is not necessary to understand the technical details covered in this section in order to implement WLD under ProtectToolkit M, however the information may be of interest to some users.

Trust Management

Trust management comes into play, where there is a need to transfer secure data or keys from one HSM to another HSM. Environments where Work Load Distribution (WLD) is used are an example of such a system.

When a WLD system is configured, it is necessary to replicate tokens across those HSM User slots that are associated with a common WLD virtual slot. For the HSM that imports the token, it is essential that the token is deemed trustworthy before it is utilized; in other words, that the token must not have been altered during transmission and that the token was imported from a trustworthy source. For the HSM that exports the token, it is essential that the HSM that imports the token is also deemed trustworthy.

Public key cryptography is used to establish trust between HSMs. Private keys are used for signing extracted information and unwrapping tokens. Public keys are used for wrapping tokens and verifying signed information. A RSA key-pair must be generated on the administrative token of each device. This key-pair is referred to as the **local** HSM Identity Key-Pair. The public half of the key-pair is termed the HSM Identity Public-Key, while the private portion is called the HSM Identity Private-Key. A HSM trusts another HSM (the peer HSM) when the HSM holds the HSM Identity Public-Key of the peer in its administrative token. This is referred to as the **peer** HSM Identity Public-Key.

Figure 26 shows an example of a system where simple trust relationships have been established between HSMs. The arrows in the figure indicate the trust relationship. In this system, HSM A trusts HSM B, i.e. HSM A holds the HSM Identity Public-Key of HSM B in its administrative token. However, HSM B does not trust HSM A. HSM B and HSM C share a relationship of mutual trust. In this system token replication could only be performed between HSM B and HSM C (with either device originating the tokens) as token replication requires a relationship of mutual trust between HSMs.



Figure 26 – Simple trust relationships

Typically, when token replication is performed in a WLD configuration, a HSM is selected to hold the master tokens and tokens are then replicated to the other HSMs. Figure 27 illustrates a system in a typical WLD configuration. In this system, HSM A has been selected to hold the master tokens. The arrows indicate the relationships of mutual trust between HSM A and the other HSMs that are necessary for token replication to be performed. The figure also illustrates that it is not necessary to establish trust amongst the HSMs that the tokens are replicated to, in other words, no trust need be established amongst HSM B, HSM C, HSM D and HSM E.

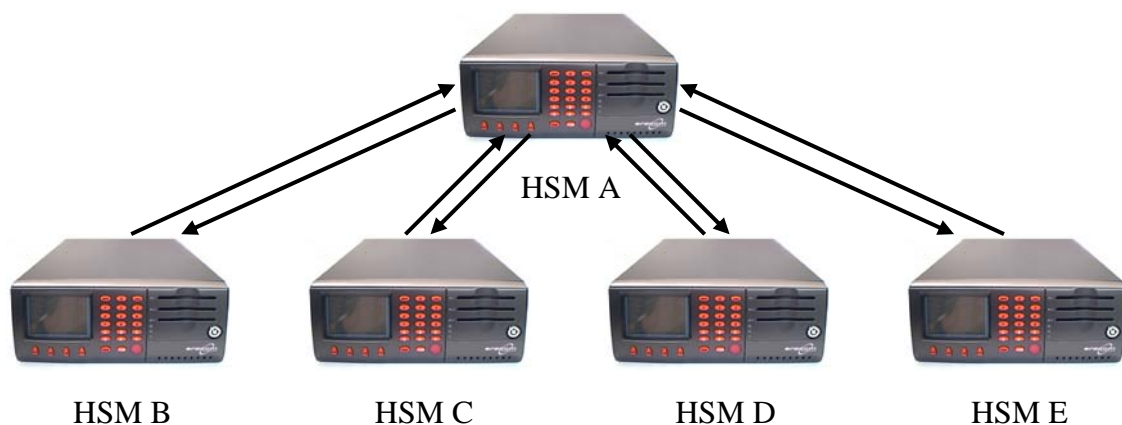


Figure 27 – Trust relationships in a typical WLD configuration

Token Replication

Token replication allows a user to replicate their tokens across one or more HSMs. Token replication is not a suitable mechanism to use in place of token export.

Token replication can only be performed on User Tokens (Administration Slots are not supported). Refer to the section entitled The ProtectToolkit C Model for a description of slot types. Token replication can occur from any User slot to any other User slot on the same HSM or a different HSM. During token replication, all the objects contained within the master token as well as the master token label are replicated. The latter is an important factor when a system is operating in WLD mode as the token label identifies which virtual WLD slot that the token is associated with.

Once a token has been replicated, any objects that are created or modified on that token will **not** be automatically replicated to those tokens replicated from the same token. If a token is modified, and there is the requirement for consistency amongst tokens, then the token replication process must be repeated. Please note: WLD requires token consistency,

so whenever a token is modified, replication to all participating WLD tokens is mandatory.

The *ctkmu* utility with the *rt* command is used to replicate tokens. The SO pin of token in the master slot and the SO pins of the tokens in the slots that the token is imported to must be the same. The User pin of the token in the master slot and the User pins of the tokens in the slots that the token is imported to must be the same. When replicating to an uninitialized token, the SO pin of the token is required to be entered. If the *No Clear PINs flag* is set, the User pin of the Administration token on the device importing the token is also required.

The *ctkmu rt* command utilizes slot positional numbers to identify the master slot and the destination slots. The slot positional numbers are dynamically assigned at the time that the command is invoked. If a device goes offline at the moment the command is invoked, the positional device number will be reassigned. This could result in the token being replicated to an incorrect slot. It is important that the system is stable when using this command.

Appendix F

Registry Configuration

The registry entries documented here are those created by default when ProtectToolkit M is installed unless otherwise noted.

Disclaimer

The ProtectToolkit M registry configuration as documented in this appendix, should only be modified by personnel who are competent at making changes to the Windows registry using the *regedit* utility. Changing the registry incorrectly can leave a system in an unrecoverable state and SafeNet cannot be held responsible should this occur. If you are unfamiliar with editing the registry it is strongly advised that you refer to your Windows documentation or seek help from a qualified systems administrator prior to attempting any changes.

ptkcRuntime

Key

HKEY_LOCAL_MACHINE\SOFTWARE\SAFENET\ ProtectToolkit M
- REG_SZ “ptkcRuntime”

Values

The ProtectToolkit M product relies on the ProtectToolkit C product. This is a string value which is used to record the version of the ProtectToolkit C runtime installed in the ProtectToolkit M installation directory.

CryptokiPath

Key

HKEY_LOCAL_MACHINE\SOFTWARE\SAFENET\ ProtectToolkit M
- REG_SZ “CryptokiPath”

Value

This string value is the path to where ProtectToolkit M is installed. This path is used to locate the required *cryptoki.dll* file.

Debug Level

Key

HKEY_LOCAL_MACHINE\SOFTWARE\SAFENET\ ProtectToolkit M
- REG_DWORD “debugLevel”

Values

Valid values are 0 through 5. If turned on, debug output is saved in the *ptkm.log* file in the root directory of the current drive. Each level includes the output of all preceding levels.

- 0 no debug output
- 1 log the invocation of CSP related methods and if an error occurs, their return code
- 2 Internal errors are converted to NTE_SYS_FAIL. This debug level logs the internal error at the point of conversion.
- 3 CSP function input and output parameters - but NOT user related data (such as encrypted or clear text, or data to hash).
- 4 Other information not covered by the preceding levels - but NOT user related data.
- 5 Lists the Cryptoki function calls; by name only.

Note that the debug level is read when a process loads the ProtectToolkit M library file. This means that in order to change the debug level, you must first stop any ProtectToolkit M applications.

The log file *ptkm.log* is cleared during initialization of the ProtectToolkit M library.

Safenet RSA Full Cryptographic Provider

Key

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\
Provider\Safenet RSA Full Cryptographic Provider

Description

This is the registry key (and contained values) which defines one of the CSPs installed by the ProtectToolkit M product.

Safenet RSA SChannel Cryptographic Provider

Key

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Safenet RSA SChannel Cryptographic Provider

Description

This is the registry key (and contained values) which defines one of the CSPs installed by the ProtectToolkit M product.

Default RSA SChannel Cryptographic Provider Type

Key

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider Types\Type 012

Description

This is the registry key which defines the default “RSA SChannel” provider. This provider is used by applications which request RSA SChannel services, but do not specify which provider (such as IIS).

NOTE: This registry entry is required for Windows 2000 but not for Windows 2003 as Windows 2003 provides for provider selection.

Value

Name of the default provider, after installing ProtectToolkit M. This should be “Safenet RSA SChannel Cryptographic Provider”.

Default RSA Full Cryptographic Provider Type

Key

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider Types\Type 001

Description

This is the registry key which defines the default “RSA Full” provider. During the logon process, this provider is used to validate the entered password.

NOTE: The logon process requires a particular key pair to exist. This key pair does not exist in the “Safenet RSA Full” provider. Therefore, the default should **NOT** be set to “Safenet RSA Full Cryptographic Provider”

Value

Name of the default provider. This should **NOT** be “Safenet RSA Full Cryptographic Provider”

Silent User Keyset Login Password

Key

HKEY_CURRENT_USER\Software\Safenet\ProtectToolkit M

Description

This entry is NOT created by default. Create it manually if silent User keyset login is required. See the section *Silent User Keyset Login* in *Chapter 3* for further information if required.

Value

Enter the password as clear text for key entry *UserKeysetPassword*.

Appendix G

Event Log Error Types

The following table lists the error entries that may be generated by SafeNet hardware security module firmware and written to the HSM's event log.

Event records are written sequentially and chronologically. If the date and time of a later entry in the log is stating an earlier time than an entry preceding it, it indicates that the real time clock or audit information has been altered.

Name	Description
POST_ERR_SRAM_WRITE	POST Error: Cannot write to SRAM
POST_ERR_SRAM_READ	POST Error: Cannot read from SRAM
POST_ERR_SDRAM_DATA_STUCK	POST Error: SDRAM, bit stuck
POST_ERR_SDRAM_DATA_SHORT	POST Error: SDRAM data bits short Param 1. Bit number Param 2. Value
POST_ERR_SDRAM_ADDR_STUCK	POST Error: SDRAM address bit stuck
POST_ERR_SDRAM_ADDR_SHORT	POST Error: SDRAM address bits short Param 1. Bit number
POST_ERR_SDRAM_BAD_BYTESEL	POST Error: SDRAM bad bytes select
POST_ERR_BAD_SECTOR0	POST Error: POST Sector checksum is not correct
POST_ERR_NOMEM	Cannot allocate memory
POST_ERR_OS_HASH	The OS hash value is incorrect
POST_ERR_KAT	Known answer test failed Param 1. Algorithm Identifier Param 2. Error Code
POST_ERR_RNG	RNG did not pass chi-squared test
POST_ERR_NO_THREAD	Unable to start POST Thread
POST_ERR_SMFS	Secure memory file system error Param 1. Error Number
POST_ERR_RTC	Unable to access RTC
POST_ERR_SER	Unable to access UART
EXCEPT_UNDEF	An undefined instruction has been executed Param 1. Address Param 2. Instruction
EXCEPT_SWI	A software interrupt generated Param 1. Address Param 2. Instruction
EXCEPT_PREFETCH	A Prefetch abort generated Param 1. Address
EXCEPT_DATA	A Data abort generated Param 1. Address
EXCEPT_IRQ	An unhandled IRQ received Param 1. Identifier
ERR_HOT_TAMPER	Hot tamper detected
LOG_FIRST_ENTRY	Initial event entry
LOG_INITIALIZING_SRAM	Initialising the SRAM after a tamper
LOG_EVENT_LOG_PURGED	Event log has been purged
ERROR_ASSERT	Runtime Assertion Param 1. File Param 2. Line

Name	Description
ERROR_INIT_RESOURCE	Out of resources in initialisation Param 1. File Param 2. Line
HEAP_INVALID_ADDRESS	Heap Invalid block address Param 1. Heap number Param 2. Address
HEAP_MEM_FREED_TWICE	Heap: Memory Freed twice Param 1. Address
PCCISES_TIMEOUT	PCCISES: Timeout error on device Param 1. Error
PCCISES_BAD_STAT	PCCISES: Bad device status Param 1. Status
PCCISES_BAD_DATA	PCCISES: Bad input data
PCCISES_RNG_STUCK	PCCISES: Continuous RNG test error Param 1. Value
PCCISES_LNAU_EXCEPTION	PCCISES: Large Number Arith Hardware exception (Unit,0)
PCCISES_FAILED_RESET	PCCISES: Failed to reset
PCCISES_RESOURCES	PCCISES: Insufficient resources to start driver
CPROV_OS_UPGRADED	OS Upgrade performed Param 1. Mod Param 2. Version
CPROV_OS_UPGRADE_FAILED	OS Upgrade failed
PROT_NO_SMPR	PROTECTION: Adapter SMPR not found
PROT_CIPHER_ERROR	PROTECTION: Cipher operation failed
KEYGEN_ERR_PAIRWISE	Key generation: Pair-wise consistency failure

Glossary

CA	Certification Authority.
MSCAPI	Microsoft Cryptographic API
CSA	Cryptographic Services Adapter.
CSPs	Microsoft Cryptographic Service Providers
DES	Cryptographic algorithm named as the Data Encryption Standard.
HSM	Hardware Security Module. This may take the form of a PCI adapter card installed in a host PC or it may be a standalone unit accessed across a network.
IIS	Microsoft Internet Information Services
Keyset	A keyset is the definition given to an allocated memory space on the HSM. It contains the key information for a specific user.
MSDN	Microsoft Developer Network
PCI	Peripheral Component Interconnect.
PIN	Personal Identification Number.
PKCS	Public Key Cryptographic Standard. A set of standards developed by RSA Laboratories for Public Key Cryptographic processing.
PKCS#11	Cryptographic Token Interface Standard developed by RSA Laboratories.
PKI	Public Key Infrastructure.
RSA	Refers to the algorithm invented by Ron Rivest, Adi Shamir and Leonard Adleman.
RTC	Real Time Clock.

THIS PAGE INTENTIONALLY LEFT BLANK

Index

A	
Administration Utility.....	43
Administrative Tasks.....	25
C	
CA server.....	90
Certificate	
Creating.....	89
Installing.....	90, 95
Clock Drift Correction.....	29
CREATECERT.....	60
CSP.....	9
D	
Debug Level.....	115, 116
Device Administrator.....	15
E	
Error Types.....	119
Errors	
Application.....	66
Duplicate Key Objects.....	65
Event Log.....	119
Keyset Restore.....	63
Session Exists.....	64
Event Log.....	29
Purging.....	29
Viewing.....	29
F	
Firmware Upgrade.....	30
G	
Glossary.....	121
H	
HSM	
Initializing.....	16
I	
IIS.....	13, 89, 90, 95, 96, 97, 99, 117
Installation.....	13
K	
Key Container	
Adding.....	38
Deleting.....	38
Key Pair	
Deleting.....	40
Generating.....	39
Properties.....	40
Key Usage.....	39
Keys.....	9
Keyset	
Backup.....	31
Deleting.....	27
Restore.....	33
Keyset Management Utility.....	52
Keyset Password	
Changing.....	37
Keyset Space	
Allocating.....	21, 25
De-allocating.....	26
Keysets.....	10
M	
MACHINE keyset.....	11
Microsoft Cryptographic API.....	9
Microsoft Cryptographic Service Provider.....	9
Microsoft Internet Explorer.....	63
MSCAPI.....	9
N	
Netscape	
Self-signed Certificates.....	63
P	
Password change.....	25
R	
Registry Configuration.....	22
RSA Full.....	9, 11, 68, 116, 117, 118
RSA SChannel.....	9, 11, 66, 93, 117
S	
SYSTEM keyset.....	11
T	
Tamper	
Forcing.....	31
PCI.....	28, 48, 50
Soft.....	30
Transport Mode.....	28, 48, 50
Troubleshooting.....	63
U	
Uninstallation.....	14
User.....	15
User Keysets	
Creating.....	21, 27, 37
User Roles.....	15
User Tasks.....	37

