SafeNet ProtectToolkit C Key Management Utility

User Guide

gemalto
security to be free

Part Number 007-008394-006
Version 5.2

Trademarks

Gemalto Rebranding

In early 2015, Gemalto NV completed its acquisition of SafeNet, Inc. As part of the process of rationalizing the product portfolios between the two organizations, the HSM product portfolio has been streamlined under the SafeNet brand. As a result, the ProtectServer/ProtectToolkit product line has been rebranded as follows:

Old product name	New product name
Protect Server External 2 (PSE2)	SafeNet ProtectServer Network HSM
Protect Server Internal Express 2 (PSI-E2)	SafeNet ProtectServer PCIe HSM
ProtectToolkit	SafeNet ProtectToolkit

Disclaimer

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

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

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

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

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

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

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

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

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

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 Gemalto support. Gemalto 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 Gemalto and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Contact method	Contact	
Address	Gemalto NV 4690 Millennium Drive Belcamp, Maryland 21017 USA	
Phone	Global	+1 410-931-7520
	Australia	1800.020.183
	China	(86) 10 8851 9191
	France	0825 341000
	Germany	01803 7246269
	India	000.800.100.4290
	Netherlands	0800.022.2996
	New Zealand	0800.440.359
	Portugal	800.1302.029
	Singapore	800.863.499
	Spain	900.938.717
	Sweden	020.791.028
	Switzerland	0800.564.849
	United Kingdom	0800.056.3158
	United States	(800) 545-6608
Web	www.safenet-inc.com	

Support and Downloads	www.safenet-inc.com/support Provides access to the Gemalto Knowledge Base and quick downloads for various products.
Technical Support Customer Portal	https://serviceportal.safenet-inc.com Existing customers with a Technical Support Customer Portal account can log in to manage incidents, get the latest software upgrades, and access the Gemalto Knowledge Base.

Revision History

Revision	Date	Reason
A	14 March 2016	Release 5.2

Table of Contents

Glossary

PKCS#11	Public Key Cryptography Standard # 11. Cryptographic Token Interface Standard (Cryptoki). An RSA Laboratories Technical Note.
Cryptoki	Cryptographic Token Interface Standard. (aka PKCS#11).
Protecttoolkit C	SafeNet's implementation of PKCS#11. Protecttoolkit C represents a suite of products including various PKCS#11 runtimes including software only, hardware adapter, and host security module based variants. A Remote client and server are also available.
JCA	Java Cryptographic Architecture.
JCE	Java Cryptographic Extensions.
Protecttoolkit J	SafeNet's implementation of JCE. Runs on top of ProtectToolkit C
Slot	PKCS#11 slot which is capable of holding a token.
Token	PKCS#11 token that provides cryptographic services and access controlled secure key storage.
SO	Security Officer for a PKCS#11 token.

Introduction

The SafeNet Key Management Utility (KMU) provides functions that allow management of keys using a PKCS#11 sub-system. This manual provides details on how to correctly use these functions.

Installation

Before installing the KMU, you must have a ProtectToolkit C runtime installed on your system. Optionally a ProtectServer adapter may have to be installed depending on the version of ProtectToolkit C that you are running. For more information please refer to the relevant manuals that came with ProtectToolkit C or the ProtectServer Blue adapter.

To install the KMU:

Obtain the self-extracting installation file from the distribution media and run the sef-installation program. Follow the on-screen prompts to install the software.

The KMU needs to know where the 'cryptoki.dll' file is located in order to run. It will have been loaded when you installed the ProtectToolkit C runtime package. Make sure that you have this file in your path or that it is located in that same directory as the KMU executable.

Using CTBROWSE with ProtectToolkit J

ProtectToolkit J is SafeNet's Java Cryptography Architecture (JCA) and Java Cryptography Extension provider (JCE) software.

CTBROWSE may be used to set up tokens and keys for use with ProtectToolkit J. The tokens and keys that are managed with CTBROWSE are fully compatible and may be utilized by ProtectToolkit J. CTBROWSE may also be used to see and manipulate keys that have been created by ProtectToolkit J. For more information consult the Key Management section in the ProtectToolkit J Reference Manual.

Please contact SafeNet for further details on its ProtectToolkit J products.

Operation

To start the KMU, locate the program folder titled "SafeNet Key Management Utility" in the Windows Start menu and click on the on the appropriate shortcut. To exit the KMU select ***File | Exit*** from the menu bar. Selecting Help from the main menu can retrieve information about the current KMU version.

Main KMU Interface

When the KMU is first started, all toolbar functions are initially disabled. The user must first select a Token from the "Token Selection" dropdown box, which will list all available tokens. Initialised tokens are displayed by their assigned label name. Un-initialised tokens are displayed as **"<uninitialised token>**", and selection of such will prompt the user to first initialise that token. Refer to the section entitled **Initialising a Token**, for further details.

Once a token has been selected the user will be given the option to login (see below) and the PIN successfully authenticated, a list of keys contained within the token is displayed in the "Key Selection" box. Appropriate buttons on the toolbar will now also be enabled as shown in

Figure 1.

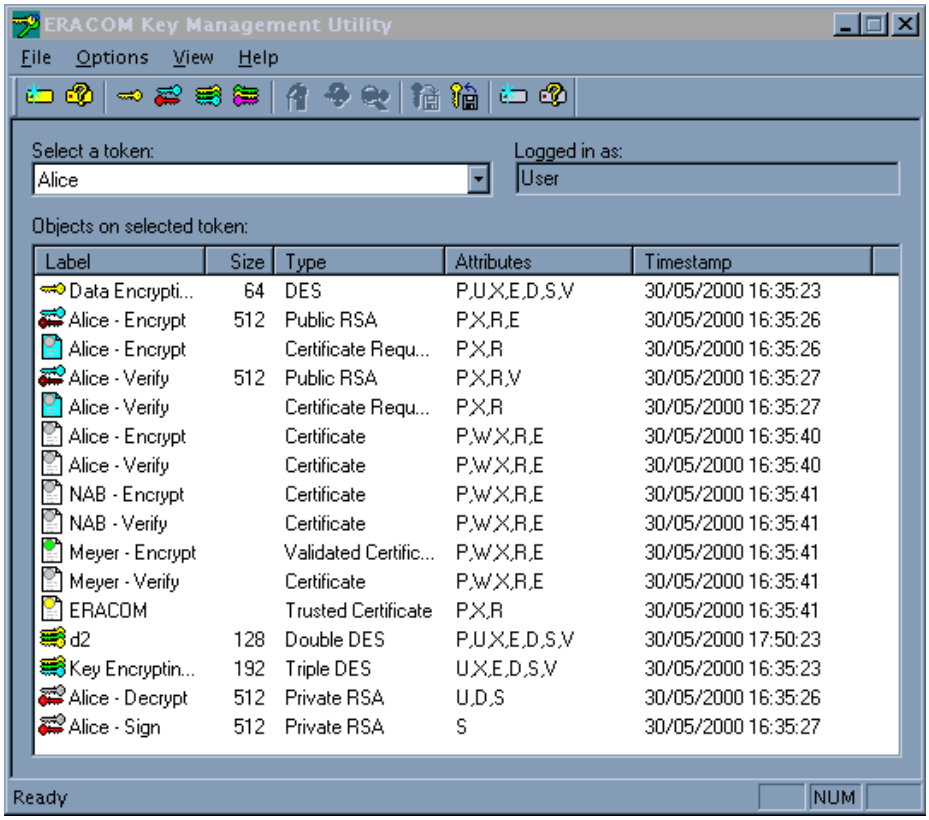
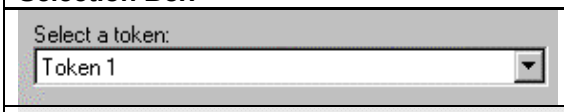
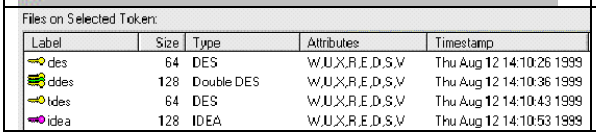


Figure 1: Main KMU Interface

Token and Key Selection

The selection boxes are as follows.

Selection Box	Description
	Token Selection Dropdown
	Key Selection Box

The 'Key Selection' box displays the keys label, the size of the key (in bits), the type of key, the attributes set for the key and the time the key was created. You can change the label of a key by selecting it and then clicking on the label (This is performed in the same manner as renaming files in Windows Explorer).

Note that more than one key may be selected by using SHIFT-LBUTTON to choose a range or SHIFT-RBUTTON to add items to a selection. Operations that can accept more than one key will process all selected keys.

Toolbar Buttons

The buttons on the toolbar correspond to the following commands.

Button	Description	Button	Description
	New Token		Initialise Smartcard
	Token Info		Smartcard Info
	Create Random Secret Key		Delete Key
	Create Key Pair		Edit Key Attributes
	Create Key & Components		Display Key Verification Code
	Enter Key from Components		Import Key
			Export Key

The toolbar and statusbar can be enabled or disabled from within the View menu. A check mark beside the selected toolbar item indicates the current view status. For example, to hide the toolbar menu from view, select ***View | Toolbar***, and make sure that there is no check mark beside the toolbar name. Follow the same procedure when wanting to show the toolbar, although this time ensuring that there is a check mark shown.

Logging into a Token

When an initialised token is selected, you will be prompted to select a user type and to enter the PIN corresponding to the selected token (See Figure 2). PIN entry is masked so only the '*' character will be displayed as characters are typed. Some operations require the Security Officer (SO) to be logged in while other operations (private object operations) require the user to be logged in. It is also possible to open the token without logging in however only public objects will be visible if this option is used.

Warning: Make sure that the CAPS lock is not on if the password contains lower case characters.

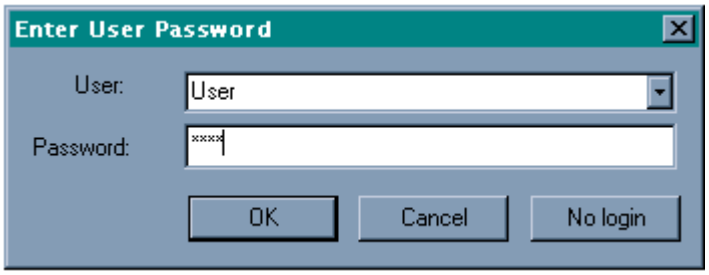


Figure 2: Token Password Entry screen.

Logging Out from a Token

Users can log out from the current token by selecting ***File | Logout From Token*** from the File menu bar.

Initializing a Token

Initialising a token enables information to be stored (e.g. keys) on that token and sets both the Security Officer PIN and User PIN.

If you have any uninitialized tokens on your ProtectServer Blue, you must initialise them before you can create any keys. Unitialized tokens appear as "<uninitialized token>" in the "Token Selection" dropdown box.

When a token is initialised for the first time, a new Security Officer (SO) password may be selected however when a token is re-initialised then the tokens current SO password must be entered correctly before the re-initialisation will succeed.

Warning: Re-initialising a token removes all keys and other components you have stored on that token.

To initialize a token, select a token from the "Token Selection" dropdown and select ***File | Initialise Token*** from the menu bar. The "Initialise Token" dialog will be displayed (See Figure 3).

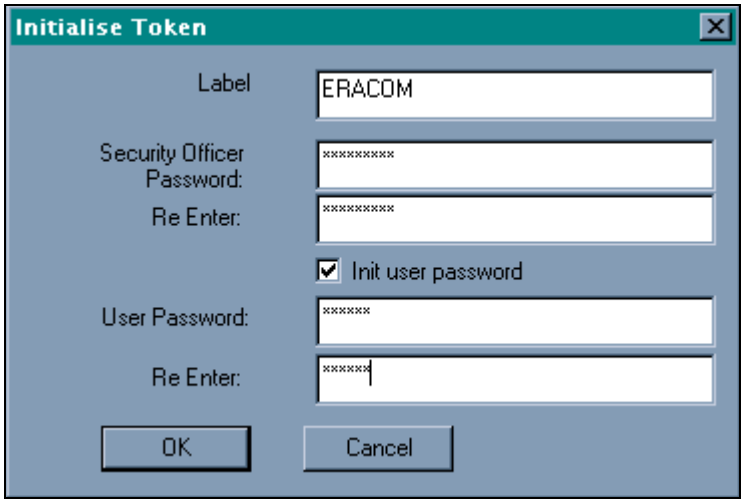


Figure 3: Init Token Dialog

Enter values for the label, security officer password and user password for the selected token. Input to the two password fields will display only asterisks (*) for security reasons.

NOTE: It is possible to change the security officer and user PINs via the ProtectToolkit C or ProtectToolkit J browser. The token label however cannot be changed without re-initialising the token.

Resetting the User's Password

The password for the user of a token can be set or reset by the security officer. Choose ***File | Set User Password*** from the File menu bar. The Set User Password dialog appears (see Figure 4).Enter the Security Officer and User Password into the appropriate fields. The User Password must be re-entered for validation. Press **OK** to confirm your entry or **Cancel** to reject your input.

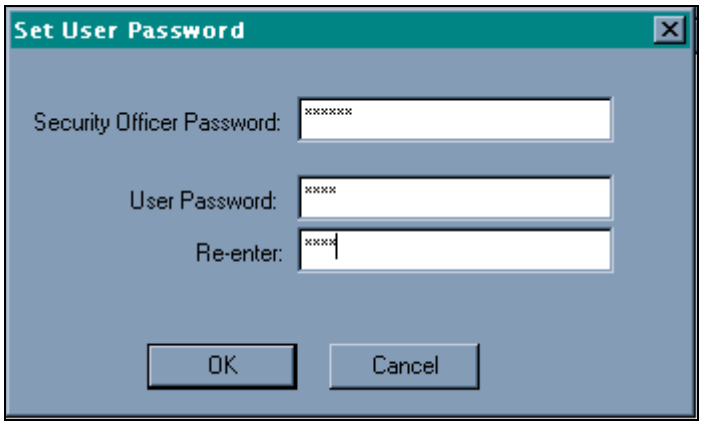


Figure 4: Set User Password

The values for each password must be between 4 and 32 characters long and alphanumeric. All input fields echo characters with an asterisk (*).

Changing the Logged on User's Password

The password of the logged on user for a token can be changed by choosing ***File | Change Logged on User Password*** from the menu bar. The Change Logged in User's Password dialog appears (see Figure 5).Enter the Old User Password and the New User

Password into the appropriate entry fields. The new User Password must be re-entered for validation. Press **OK** to accept or **Cancel** to reject your input.

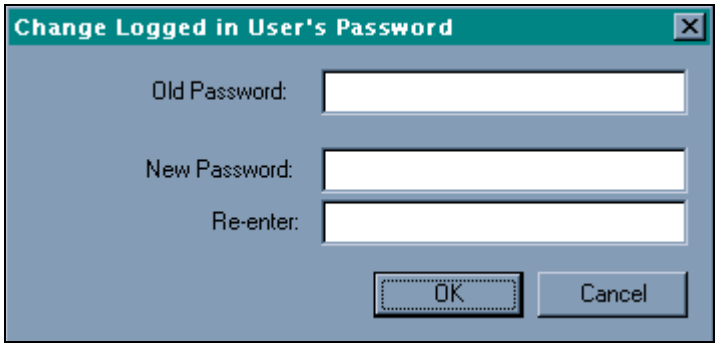


Figure 5: Change Token User PIN Dialog

The values for each password must be between 4 and 32 characters long and alphanumeric. All input fields echo characters with an asterisk (*).

Retrieving Information about a Token

Various information about a token may be viewed by clicking on the "Token Info" button found on the toolbar or by choosing ***File | Get Token Info*** from the menu bar. The Token Info dialog will be shown (See Figure 6).

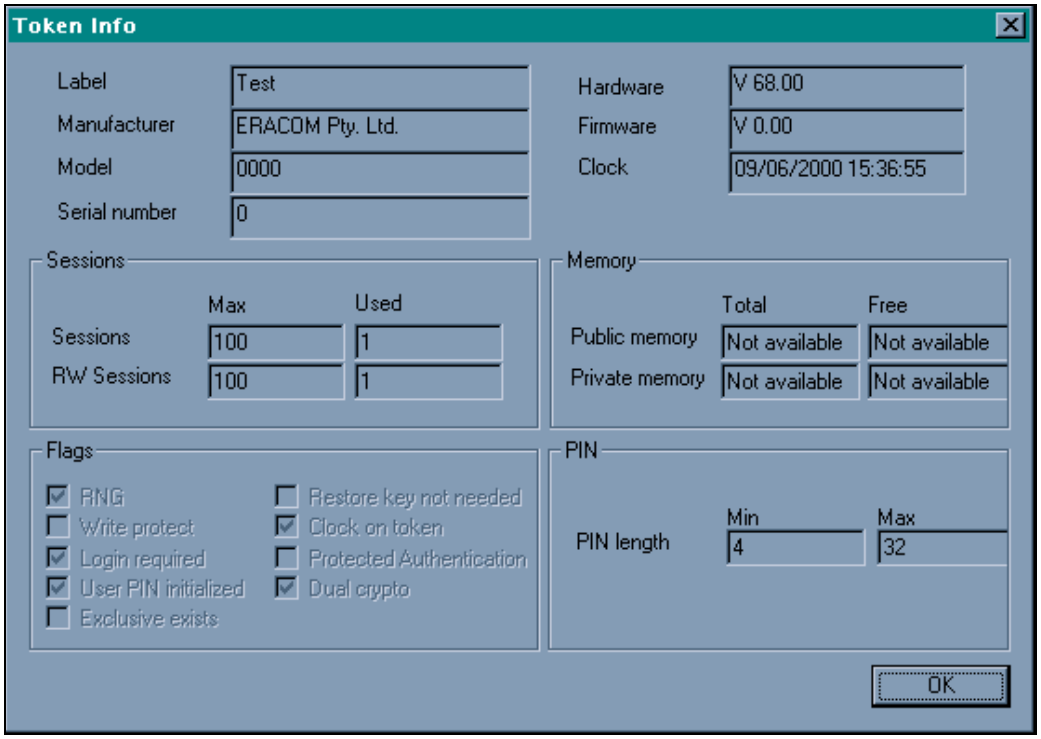


Figure 6 – Token Info dialog.

For a more information on the items shown in this dialog, please refer to the PKCS#11 standard document.

Smart Card Operations

As explained in the previous sections, the KMU allows for the export and import of keys via smart cards.

NOTE: At present the only type of supported smart card that may be used is the Gemplus GPK-4000 model card.

Before smart card key export or import can be performed, a smart card reader must be attached to the lower serial port of the ProtectServer Blue. Smart cards must be initialised before they can be used to store keys.

Smart Card Initialisation

To intialise a smart card, click on the "Initialise Smart Card" button on the toolbar. Alternatively select ***File*** **|** ***Initialise Smart Card*** from the menu bar.

The smart card initialisation dialog is displayed (see Figure 7).

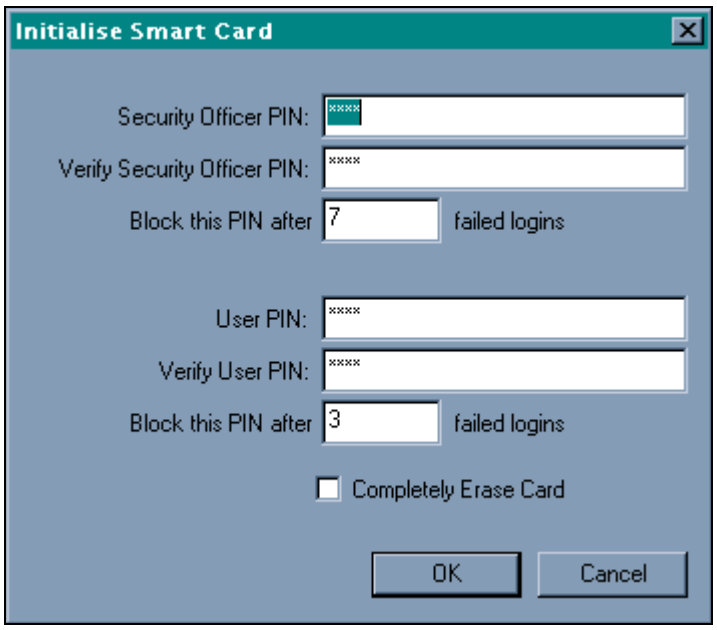


Figure 7 – Initialise Smart Card dialog.

The entry fields allow for the input of two PIN types:

- **User PIN:** The user PIN is prompted for when attempting to keyload to or from the smart card
- **Security Officer PIN:** The Security Officer PIN is utilised when needing to unlock a blocked smart card.

Enter the Security Officer PIN and User PIN into the appropriate entry fields. PINs may contain any combination of up to eight ASCII characters.

You can specify the number of incorrect PIN entries that can be made before a PIN is blocked via the "**Block this PIN after** n **failed logins**" field.

Should the user enter the incorrect User PIN more than the number of times specified, the card will be blocked. In such a case, only entry of the Security Officer PIN will unlock the card. Should the wrong security officer PIN be entered more than the number specified, the data stored on the card will be permanently blocked and unrecoverable. In this instance the card will need to be re-initialised and erased before it can be used again.

The **Completely Erase Card** checkbox may be ticked when wanting to remove any unwanted data that is already stored on the smart card.

Changing a Smart Card's User PIN

Once a smart card has been initialised, the user PIN may be changed by selecting *File | Change Smart Card User PIN* from the menu bar. The smart card Change Smart Card User PIN dialog is displayed

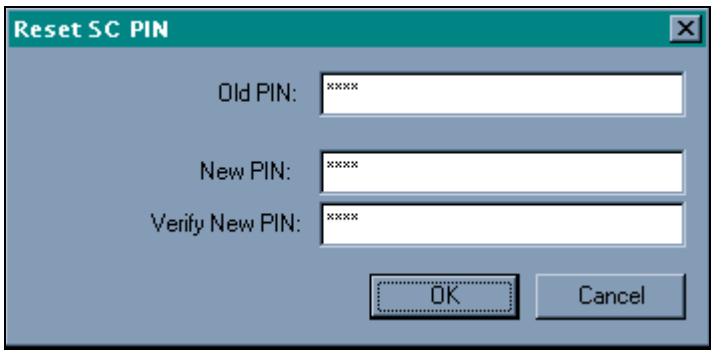


Figure 8 – Change Smart Card User PIN dialog.

Enter the existing user PIN into the **Old User PIN** field.
Enter the new user PIN into the **New User PIN** field.
Enter the new user PIN into the **Verify New User PIN** field.

Press **OK** to store the new PIN or **Cancel** to reject your input and return to the main interface screen.

Retrieving Smart Card Information

The batch name, date/time when the card was created, batch card number and number of custodians can be retrieved from a smart card by choosing ***File | Get Smart Card Info*** from the menu .

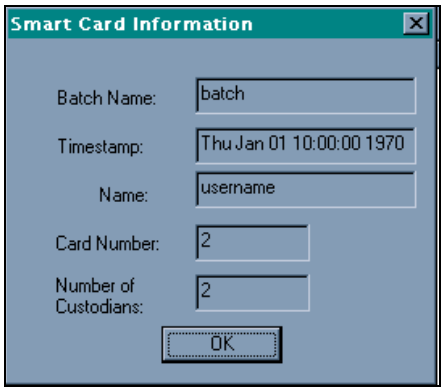


Figure 9: Smart Card Information Dialog

Unblocking a User PIN

If an incorrect user PIN has been entered more than the number of times specified when the smart card was initialised (see section “Initialising Smart Cards” for details), the PIN becomes blocked and hence unusable. To unblock the User PIN, select the menu option ***File | Unblock User PIN***, which displays the Unblock User PIN dialog (See Figure 10).

Figure 10: Unblock User PIN

Enter the SO PIN for the smart card and press **OK**.

A new User PIN of 1357 is set on the smart card after it has been unblocked. It is recommended that users change this PIN immediately using the Change Smart Card User PIN dialog described previously.

Key Management Operations

Creating Keys

The KMU supports four key creation functions. These are:

- Creating a Random Secret Key
- Creating a Random Key Pair (e.g. RSA public and private keys)
- Creating Key Components
- Entering a Key from Components

Note : To refresh the key information that is displayed on screen at any time, select ***Options*** **|** ***Refresh*** from the menu bar. The display is just a representation of what KMU has found on that token. If the token is modified by any other process or if for any reason the KMU is out of sync with the token. In such a case the display can be refreshed simply by choosing this menu option.

The KMU also supports key export and import for the purposes of key backup and / or key escrow. This feature uses the PKCS#11 concept of key wrapping using high security key encryption keys (KEK) to wrap other key encrypting keys and / or data keys. The KEK is a special key that is created with the "wrap" attribute allowing it to be used for this purpose. KEKs are usually created as split custodian keys because of their high security nature.

Note: Only keys marked for export may be wrapped in this way so it is possible to create keys that can never be extracted from the secure key storage.

Key Component entry is an important feature of this software since it allows the distribution of key material, in parts, across multiple trusted custodians for the highest level of security assurance where keys must be managed this way. To reconstruct any of the key material, all custodians must combine their components together so that the key parts may be recombined into the original key(s). Key custodians may use smart cards for key component and authentication PIN data storage, or just use a disk file for key component storage.

Available Keys

There are nine different key types available when selecting a key operation. A list of available key types are as follows:

Single Key Types	Key Pair Types
DES	RSA (Public)
Double DES	RSA (Private)
Triple DES	DSA (Public)
IDEA	DSA (Private)
CAST128 (1 to 16 bytes)	DH (Public)
RC2 (1 to 128 bytes)	DH (Private)
RC4 (1 to 256 bytes)	

Key Attribute Types

You can specify what attributes a key will have when it is created. The following table describes the attributes which you can set when creating a key using the KMU.

Attribute	Description
Persistent	Stores the object on non-volatile memory. Persistent objects can be accessed after session termination
Private	Defines whether the object is protected by the user PIN. A private object is only accessible to an application that has supplied the user PIN.
Sensitive	If a key is sensitive, the key's value may not be revealed in plain text. Once a key becomes Sensitive it cannot be modified to be non-sensitive.
Modifiable	Indicates whether or not the object is modifiable, that is, if the object's attributes may be modified after creation.
Wrap (W)	Indicates that the key may be used to wrap (i.e extract) other keys.
Unwrap (U)	Indicates that the key may be used to unwrap keys.
Extractable (X)	An extractable key can be wrapped (encrypted with another key) and extracted from the adapter.
Derive (R)	Indicates that the key can be used in key derivation functions.
Encrypt (E)	Indicates that the key may be used for encryption.
Decrypt (D)	Indicates that the key may be used for decryption.
Sign (S)	Indicates that the key may be used for signing.
Verify (V)	Indicates that the key may be used for verifying signatures or MAC values.

Creating a Random Secret Key

To create a random secret key, select an initialised token from the "Token Selection" dropdown box and click on the "Generate Secret Key" button in the toolbar.
Alternatively select ***Options*** **|** ***Create*** **|** ***Secret Key*** from the menu bar.

The "Generate Secret Key" dialog is displayed (See Figure 11).

Figure 11: Generate Secret Key Dialog.

Choose the type of key you wish to generate from the "Key Type" dropdown box. If you are generating a CAST, RC2 or RC4 key, you must specify a Key Size.
Enter a Key label for the key into the Label input field.

The group of checkboxes shown in this dialog represent the various attributes which can be set for the selected key. There will be a default set of attributes checked for the selected key.

Click on the **OK** button to generate the secret key, or **Cancel** to reject your input and return to the previous menu.

The generated key will be displayed in the "Key Selection" box on the main KMU user interface.

NOTE: Random Key generation is handled by the underlying ProtectToolkit C functionality, which assures cryptographically strong keys.

Creating a Random Key Pair

To create a random key pair, select an initialised token from the "Token Selection" dropdown box and click on the "Generate Key Pair" button in the toolbar. Alternatively select ***Options*** **|** ***Create*** **|** ***Key Pair*** from the menu bar.

The "Generate Key Pair" dialog is displayed (See Figure 12).

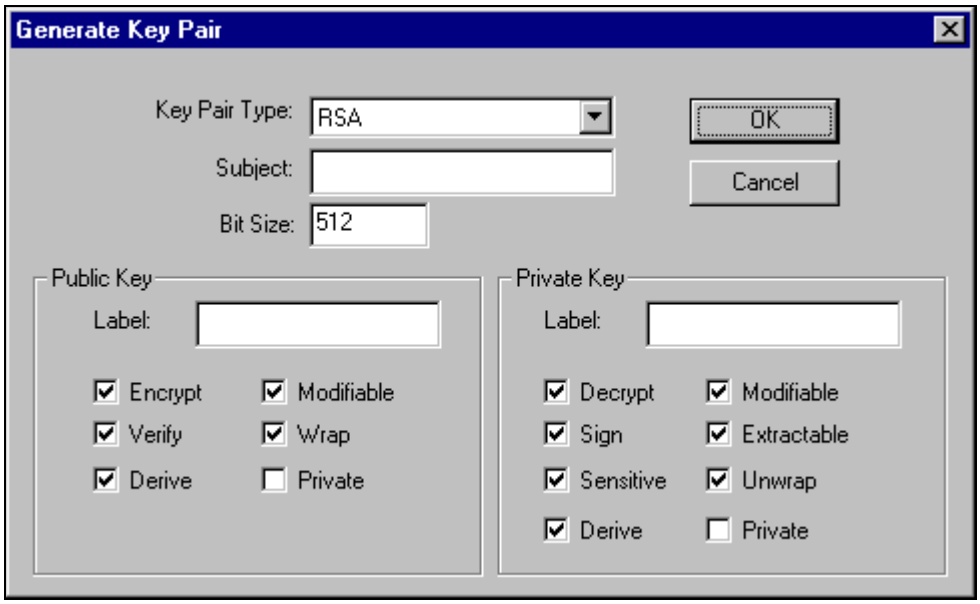


Figure 12: Generate Key Pair

Select the type of key pair you wish to generate from the "Key Pair Type" dropdown box.

Specify the "Bit Size" and give both the public key and private key labels.

You can optionally specify a "Subject", which must be set according to X.500 distinguished name syntax rules. (e.g. C=AU, O=SafeNet, CN=Alice). The subject fields can be any of the following, and may be input in any order. :

C= C*ountry code*
O= *Organisation*
CN= *Common Name*
OU= *Organisational Unit*
L= *Locality name*
S= *State name*

This information will be stored with the public and private key objects in the CKA_SUBJECT_STR attribute and also DER encoded and stored in the CKA_SUBJECT attribute. This attribute will be propagated into PKCS#10 and X.509 certificates which are derived from these keys.

The field can be left blank in which case there will be no X500 certificate information attached to the key pair.

Press the **OK** button to generate the keys, or press **Cancel** to discard your input and return to the previous menu.

Generated keys will be displayed under the "Key Selection" list on the main KMU user interface.

To refresh the key information that is displayed on screen, select ***Options | Refresh*** from the menu bar.

Creating Key Components

This function will create a random key and split it into a number components which you can record and enter on another machine using the "Enter Key" function.

This is useful for the creation and distribution of Key Encryption Keys (KEK's) with multiple custodians. With this function it is possible to create a key whose value is unknown to any single party. However, by combining the components known by each custodian the key may be regenerated. Each component that is generated is random and in itself does not expose any portion of the final key value.

To create a random key and display its components, select an initialised token from the "Token Selection" dropdown box and click on the "Split Key" button in the toolbar. Alternatively choose ***Options | Create | Create Key Components*** from the menu bar.

The "Create Key Components" dialog will be displayed (See Figure 13).

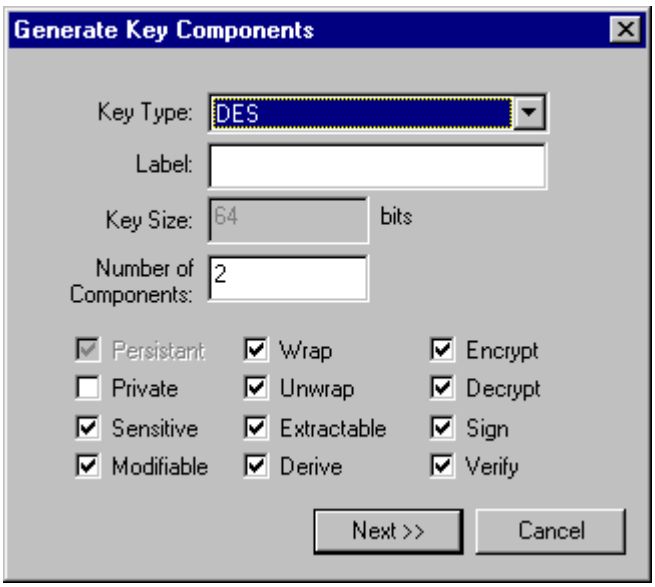


Figure 13: Create Key Components Dialog.

Specify a key type from the Key Type drop down list and enter a key label into the label field.

Specify the number of components you wish the key to be split into by entering a value into the Number of Components field. The KMU places no limit on the number of components that are allowed.

When selecting CAST, RC2 or RC4 as a key type, make sure that you specify the size of the key you wish to generate in the Key Size dialog.

Set the required attributes for the key with the checkboxes that are provided.

Click on the "**Next >>**" button to start displaying the key components, or **Cancel** to quit this operation and return to the previous menu.

The number of component screens that will be displayed corresponds to the number of components that were specified in the Create Key Components dialog (See Figure 13).

An example component generation dialog is shown in Figure 14.

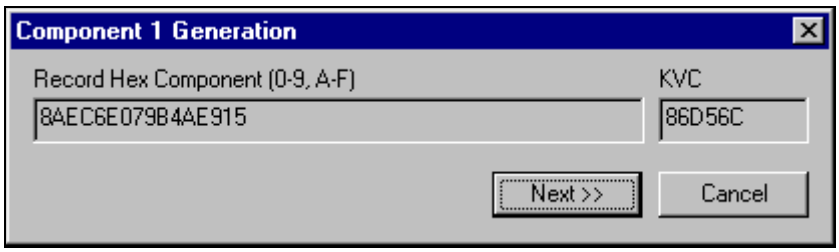


Figure 14: Component Generation Dialog

You must record the Component Value and Key Verification Code (KVC), both given in hexadecimal, that is displayed in these dialogs. The KVC for the generated component may be used to verify correct entry of the component when performing manual key component entry.

Entering a Key from Components

This function enables a key to be entered from one or more components.

To Enter a key select an initialised token from the "Token Selection" dropdown box and click the "Enter Key" button on the toolbar. Alternatively select ***Options | Create | Enter Key From Components*** from the menu bar.

The "Enter Key" dialog will be displayed (See Figure 15).

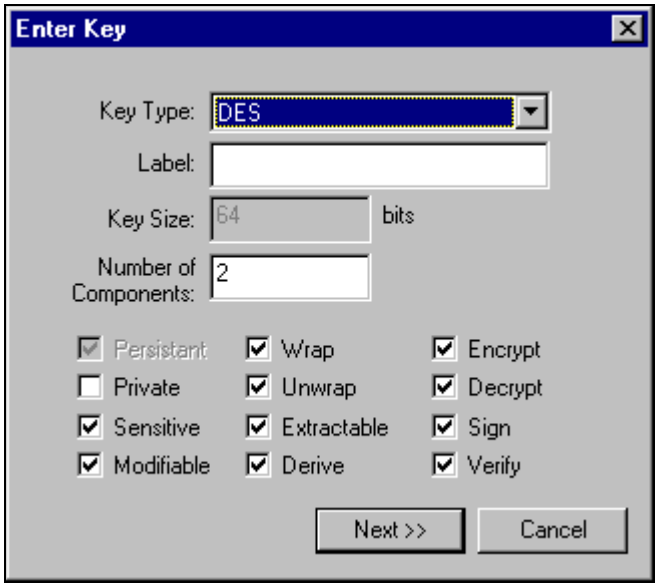


Figure 15: Enter Key Dialog

Select the key type from the drop down list, and enter a key label into the **Label** input field.

Specify the number of components that you wish to enter by entering the corresponding number into the **Number of Components** field. The KMU places no limit on the number of components that are allowed.

When selecting CAST, RC2 or RC4 as a key type, make sure that you specify the size of the key you wish to enter in the Key Size dialog.

Select the attributes for the key from the available checkboxes.

Press the ***"Next >>"*** button to display the ***"Component Entry"*** dialog (See Figure 16), or ***Cancel*** to quit this operation and return to the previous menu.

The number of components screens requiring input, corresponds to the number of components specified in the Enter Key dialog (See Figure 15).

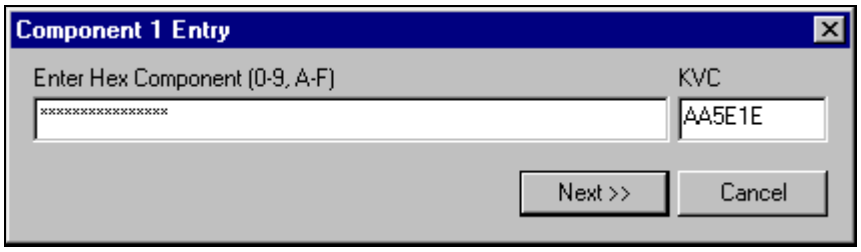


Figure 16: Component Entry Dialog

As you enter the component, the input will be masked by a '*' and is limited to the size of the key being entered.

NOTE: After the last hex digit of the component is entered into the component-entry field, the KVC is automatically generated and displayed in the KVC field for validation. The operator entering in the component should check, by inspection, that the KVC shown matches the KVC for the component being entered.

Key verification codes (KVC) of symmetric keys can be displayed by selecting a key and by clicking the "Show KVC" button on the toolbar. Alternatively you may also select ***Options | Show KVC*** from the menubar.

Refer to Appendix B for details on how the KVC is calculated.

Editing Key Attributes

You can change certain attributes of a key after the key has been created. Double click on the key you want to edit and in the Edit Attributes Dialog set or unset the checkboxes corresponding to the attributes you want to change (see Figure 17).

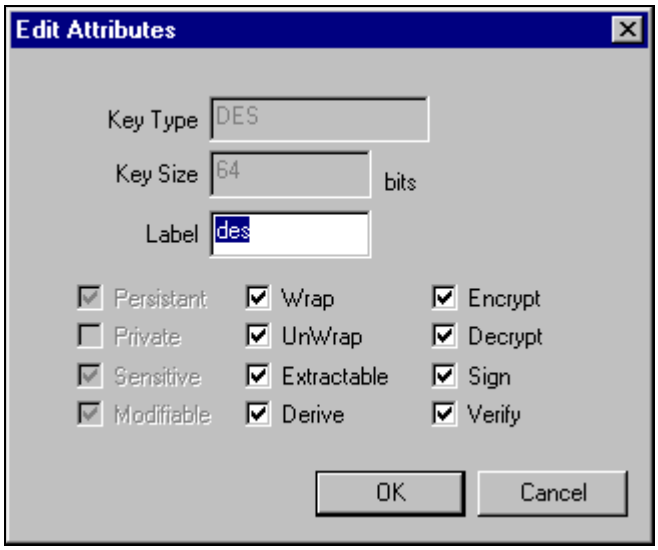


Figure 17: Edit Attributes Dialog

The attributes you can edit depend on what attributes were set when the key was created. The 'Edit Attributes Dialog' will only display the attributes that can be changed. Unavailable attributes will be grayed out.

Deleting a Key

To delete a key, select an initialised token from the "Token Selection" dropdown box. Then select the key or set of keys that you want to delete from the "Key Selection box", and click on the "Delete Key" button on the toolbar. Alternatively select ***Options | Delete Key*** from the menu bar.

Display Key Verification Code

You can check that a key matches an expected key value, without having to reveal anything about the actual key value by checking the Key Verification Code (KVC) for the key.

The KVC is a standard technique for obtaining a fingerprint from a key for identification purposes. The mechanisms used is compatible with AS2805 and is simply the first 3 hex digits obtained by encrypting binary zeros with the key. Please refer to Appendix B for details of the KVC generation.

The KVC for a key can be displayed (see Figure 21) by selecting a key and clicking the "Show KVC" button on the toolbar. Alternatively you may also select ***Options | Show KVC*** from the menubar.

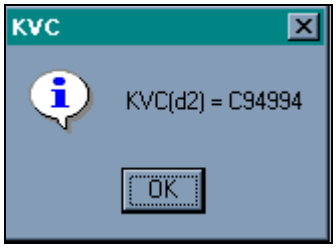


Figure 21: Display KVC Dialog

Exporting Keys

This function allows keys to be encrypted, written to smart cards or files, and then transferred to other machines.

To export a key (or set of keys), select an initialised token from the "Token Selection" dropdown box. Then select the key you want to export from the "Key Selection box", and click on the "Export Key" button on the toolbar. Alternatively select ***Options | Export Key*** from the menu bar.

NOTE: Only previously initialized and erased smart cards can be used for key storage

The "Export Keys" dialog is displayed (See Figure 18).

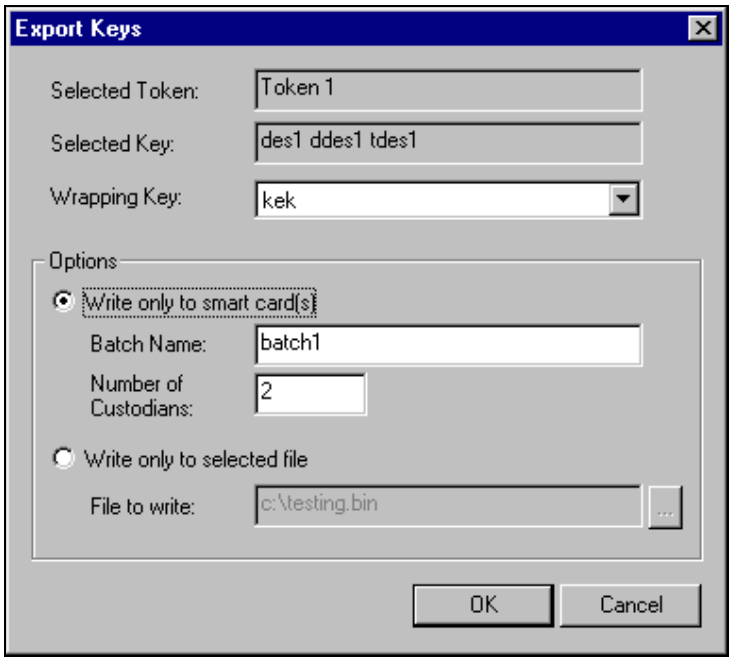


Figure 18: Export Keys Dialog

The "Export Keys" dialog displays the token and key (or set of keys), you have selected.

Select a wrapping key from the "**Wrapping key**" list box. The wrapping key will be used to encrypt the key (or set of keys), that you are exporting.

The Options area allows you to select your destination media. Files can either be written to a smart card, or to a selected file on disk.

To export the selected key to smart cards:

- In the Options area of the Export Keys dialog, select **Write only to smart card(s)**.
- Enter an identifying name for the smart card set into the **Batch Name** field.
- If you selected the <random key> option as the wrapping key, specify the number of custodians (between 1 and 20) that will own the encrypted key set.

Note: The <random key> feature will only function with ProtectToolkit C V2.0 runtime or greater.

To export the selected key to a file:

- In the Options area of the Export Keys dialog, select **Write only to selected file**.
- Enter the path and filename of the file you wish to write to into the **File to Write** field. You can also browse to a file by clicking on the "**…**" button next to the input field. If the file and directory you specify does not exist, it will be created.

Press **OK** to export the selected key, or **Cancel** to abort this operation.

Note: Wrapping keys must be created before you can export keys. Refer to the section entitled "Creating Keys" for details on how to create keys.

Note: If there are no custodians (the key set was exported without the <random key> option) the unwrapping key you use to import a key with must be the same as the wrapping key used to export the key. If this is not followed the error message, "Key used to import was not the same as the key used to export", will be displayed.

When the key set is being exported to smart cards, a dialog will be displayed showing the batch name, a User Name entry field and a PIN entry field (see Figure 19).

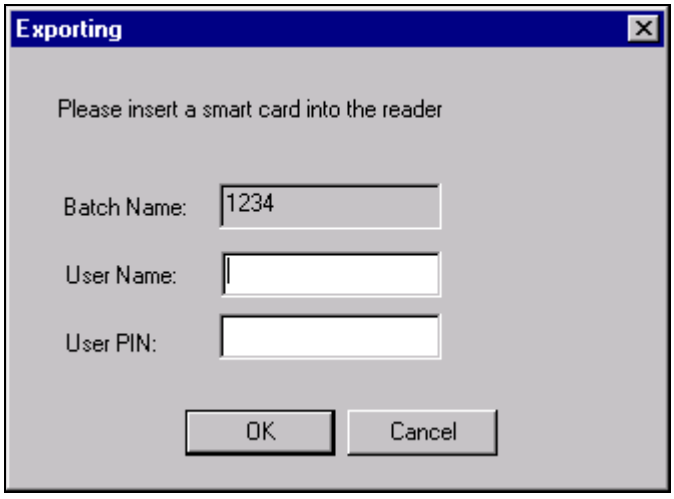


Figure 19 – Smart Card User PIN entry dialog.

Press **OK** to proceed with the keyload operation, and follow the above steps until the key export operation has completed.

NOTE: If an incorrect user PIN is entered more times than the number specified for the card during its initialisation, the smart card will become blocked. The card may then only be un-locked by entering the Security Officer PIN. Refer to the smart card initialisation section for further details.

Importing Keys

Importing allows keys stored on smart cards and in files to be unwrapped and read back into a token.

To Import a key, select the token you want to import the keys to from the "Token Selection" dropdown box and click on the "Import Keys" button on the toolbar. Alternatively select ***Options | Import Key*** from the menu bar.

The "Import Keys" dialog will be displayed (See Figure 20).

Figure 20: Import Keys Dialog.

In the **Options** area of the Import Keys dialog, choose either "**Read from smart card(s)**" or "**Read from selected file**", depending on what media your encrypted key set is stored on.

When choosing to read from a selected file:

- Specify the unwrapping key from the drop down list, and enter the filename for the encrypted key file into the **File to Read** field. The "**…**" button will allow you to find and select a file via an explorer type window.
- Press **OK** to import the selected key, or **Cancel** to abort this operation.

When choosing to read from smart card(s):

Importing from smart cards with no custodian:

- You must specify an unwrapping key from the token if no custodians have been selected since a randomly generated key must be split among 1 or more custodians. Note "no custodians" simply means that the export key is selected and not randomly generated as part of the export process.
- Press **OK** to start the import operation, or **Cancel** to abort.

After a valid card is inserted, a PIN entry dialog will be displayed.

- Enter the PIN for the smart card and press **OK** to load the keys.

The above procedure is repeated until all cards from the batch have been read.

Note: If there are no custodians (the key set was exported without the <random key> option) the unwrapping key you use to import a key with must be the same as the wrapping key used to export the key. If this is not followed the error message, "Key used to import was not the same as the key used to export", will be displayed.

Importing keys from a batch of two or more smart cards with multiple custodians:

There is no need to specify the unwrapping key when there are multiple custodians.

- Press **OK** to start the import operation, or **Cancel** to abort.

A dialog will be displayed showing the current user's name, asking you to insert a smart card from the current batch. If no cards have already been read as part of this import operation, a card from any batch may be inserted.

After entering a card and pressing OK, a dialog displaying the current card number and batch name will be shown prompting for the User PIN. (see Figure 21).

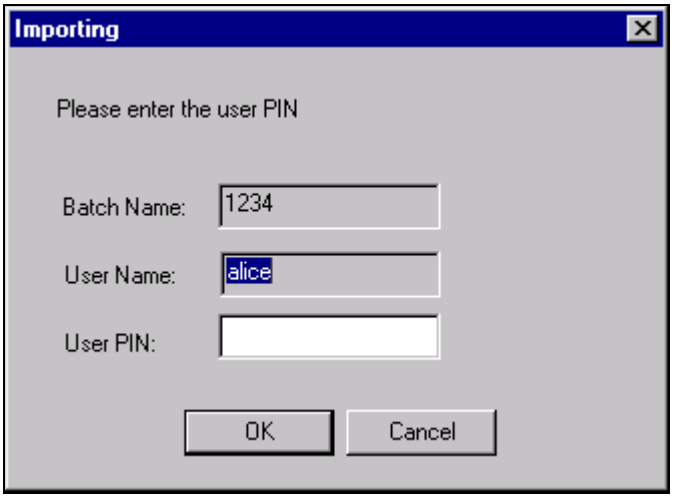


Figure 21 Smart Card Key Import – Card request dialog.

- Press **OK** to read from the inserted card. If a Smart Card is from a different batch or if the card has already been read it will be rejected, and the operator will be prompted to insert another card. If the wrong PIN was entered, it will be prompted for again.

The above procedure is repeated until all cards from the batch have been read.

NOTE: If an incorrect user PIN is entered more times than the number specified for the card during its initialisation, the smart card will become blocked. The card may then only be un-locked by entering the Security Officer PIN. Refer to the smart card initialisation section for further details.

If the import key operation is a success, a message dialog showing "The import command succeeded" will be displayed. The newly imported key will also be displayed in the "Key Selection" box.

Appendix A

Error Messages and Warnings

KMU specific error messages

Error Message	Explanation	Action
Couldn't download SAM	the Software Application Module (SAM) that is used by the KMU when performing smart card operations was not set correctly	Check that the smart card reader is securely attached to the lower port of the ProtectServer Blue . After having verified the connection, open a DOS command window and change into the directory where the KMU is installed. Type: "SAMDEVL" and press Enter
Couldn't get smart card to respond		Check the connection between the smart card reader and the ProtectServer Blue
Not enough room left on smart card for export		Re-initialize the card and erase it's contents, and repeat the export operation
Smart card was not initialized		Re-initialize the smart card and retrieve information about the card
The key used to import was not the same as the key used to export		Retry the export/import operation using the same key for wrapping and unwrapping
Smart card was not initialized. Export Failed	An attempt was made to export keys on a non-initialized smart card	
Smart card has already been processed		Retry the import operation using another card from the same batch
Smart card is from a different batch		Retry the import operation using another card from the given batch
The card inserted has already been written as part of this batch. Please insert another card		Insert a new card
There are no keys stored on the inserted card	An attempt was made to read a key from an empty card	
Please select a file to read	An attempt was made to import a key from a file without specifying the file containing the key	
Couldn't read the selected file		Check the file's read access rights
Unable to open selected file		Check the file's path and access rights
An error occurred when writing file		Retry the export to file operation making sure that the path is correct

Error Message	Explanation	Action
		and adequate access to the file is provided
Chunk MAC's do not match. Import Failed	The MAC retrieved from the different data chunks written on smart cards doesn't match with the initial one	
No wrapping key selected	An attempt was made to write a key to a file without specifying a wrapping key	
An unwrapping key is required for this batch		Retry the import operation after making sure that the correct unwrapping key was selected
An unwrapping key is not required for this batch	The keys were exported using a random wrapping key, so that no unwrapping key must be specified for import operations	
Could not generate KVC on a key with Encrypt(E) attribute set to false	Only KVC's of keys enabled for encryption can be calculated	
No KVC entered		Re-enter the key with a KVC
KVC mismatch	The KVC of the key components and of the entered key do not match	
Not a valid hex digit	An attempt was made to enter a wrong character in a key	
No component entered	An attempt was made to store a key without having entered any key components	
Wrong key size		When generating keys use the key ranges specified on page 17
User PIN's don't match		Make sure that you use the same user and user verification PINs.
SO PIN's don't match		Make sure that you use the same SO and SO verification PIN
Incorrect PIN length		Re-enter the PIN using a PIN length between 4 and 32
Incorrect PIN	An attempt was made to enter an incorrect character in the PIN string	
User PIN is blocked		Unblock the PIN as SO, and then retry the operation

For other error messages generated by Cryptoki functions see the PKCS#11 documentation

KMU specific warning messages:

Warning Message	Explanation	Action
Incompatible Cryptoki Version	An attempt was made to use a feature of another Cryptoki version	
Token not initialized	tokens must be initialized before being accessed	
Re-initializing the token will erase all currently stored keys		Store all further needed objects on cards/files before initailizing the token
No SO PIN entered		You must enter a SO PIN when initializing a token
No User PIN entered		You must enter a user PIN when initializing a token
No label entered		A key label should be entered when generating a key

Appendix B

Key Verification Code (KVC) Calculation

The KMU calculates and displays keys according to AS 2805.6.3

Single Length Key KVC

The single length key verification code is a one way cryptographic function of a key which is used to verify that the key has been correctly entered.

The KVC is calculated by taking an input of constant D (64 Zero bits) and encrypting it with key K (64 bit). The 64 bit output is truncated to the most significant 24 bits which is reported as the keys KVC (See Figure 22).

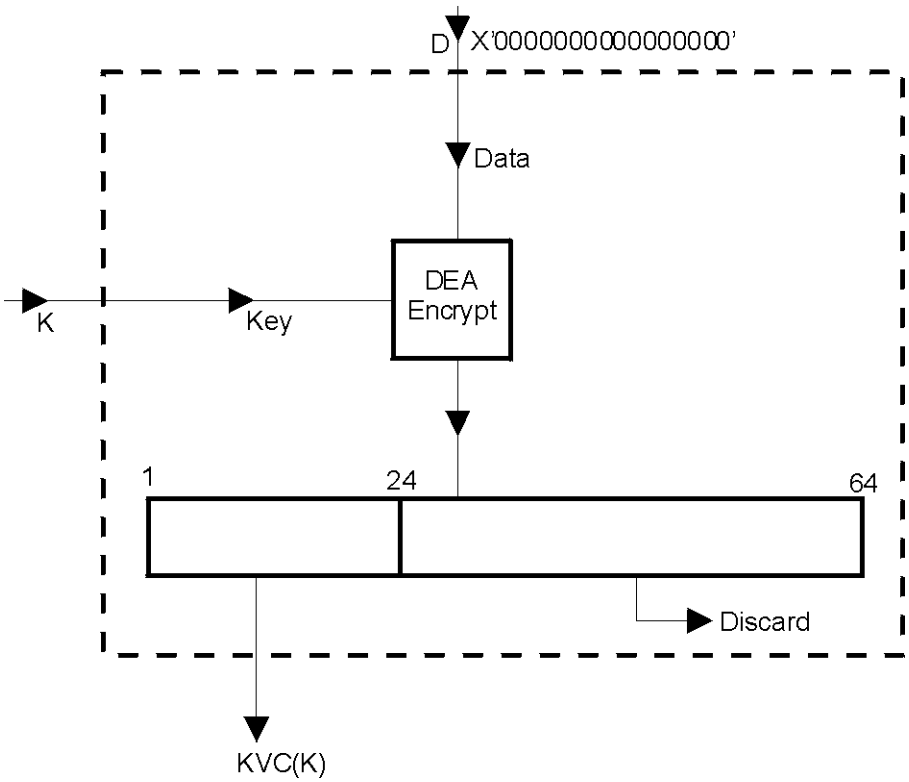


Figure 22 - Single length Key Verification Code KVC(K).

Double Length Key KVC

The single length key verification code is a one way cryptographic function of a key which is used to verify that the key has been correctly entered.

The KVC is calculated by taking an input of constant D (64 Zero bits) and key *K (128 bit string made up of two 64 bit values KL and KR). Data value D is encrypted with KL as the key. The result is decrypted with KR as the key. The result is then encrypted with KL as the key. The 64 bit output is truncated to the most significant 24 bits which is reported as the double length keys *KVC (See Figure 23).

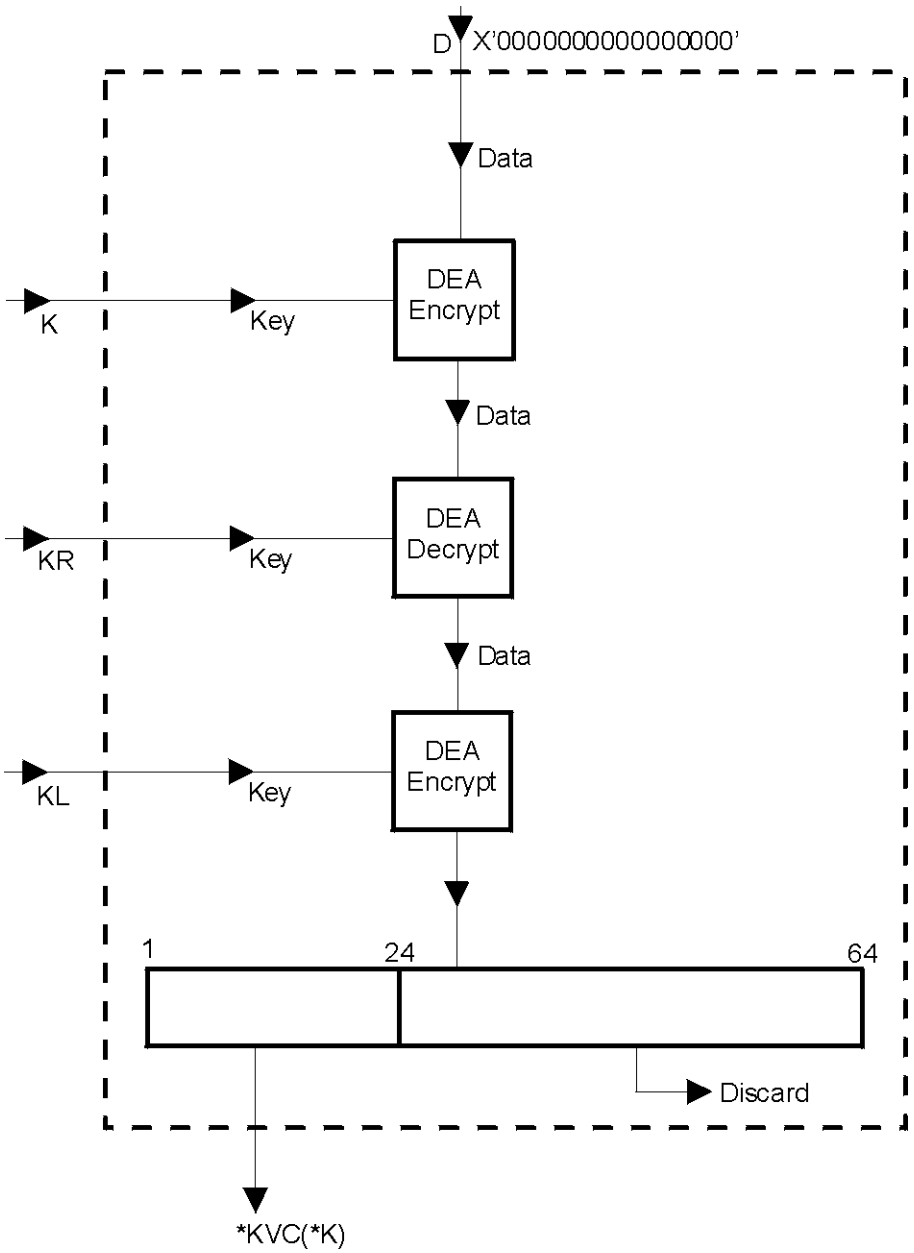


Figure 23 - Double length Key Verification Code *KVC(*K)

END

Appendix C

Summary of Key Backup Feature

This appendix illustrates the use of KMU for Key Backup, which can be used to ensure keys, certificate objects and other PKCS#11 objects can be recovered after a failure or tamper. There are two storage media options available, smart card and file (hard disk drive or floppy diskette). For smart card media, there are two modes available, single custodian and multiple custodian.

For any key/object, all the PKCS#11 attributes, including the security attributes, as well as the key/object's value are backed up.

When backing up to smart card, the utility will automatically prompt for additional smart cards, if the size of the backup is larger than one smart card.

The security officer and user PINs for a token are **not** capable of being backed up. Before a restore operation, the destination token must be already initialized and the security officer and user PINs set.

There are a number of additional keys that are generated, used and then deleted during the backup process.

Definitions:

wK Wrapping key. The top-level key for the backup process. This key must be valid for the operation $E2_x$. When performing a backup to file or single custodian to smart card, the custodian **must** provide this key. It is recommended that this be a triple length DES key.

For the multiple Custodian backup, this key is created from the randomly generated split components for each custodian.

tK A randomly generated transport key, which is a triple length DES key, using CKM_DES3_KEY_GEN. This is the key that the keys/objects to be backed up will be wrapped under. This key is used with W_x.

mK A randomly generated MAC key, which is a triple length DES key, using CKM_DES3_KEY_GEN. This key is used with M_x.

E_x Encryption using CKM_DES3_ECB_PAD with key 'x'.

$E2_x$ Encryption using CKM_(based on key type of 'x') with key 'x', e.g. CKM_DES3_ECB.

W_x C_WrapKey() operation using CKM_WRAPKEY_DES3_CBC with key 'x'.

R_x C_DeriveKey() operation using CKM_XOR_BASE_AND_DATA with key 'x' and provided data.

M_x MAC generation, using CKM_DES3_MAC (4 byte MAC result) with key 'x'.

Creation of Encrypted Key Set to Backup (Payload)

The creation of the encoded payload to backup is common to all storage options. The payload can contain one or more keys/objects.

Step 1

Generate tK.

Step 2

For each key/object to be backed up:
$w = W_{tK}(\text{Key/Object})$

The format of the resulting Payload is as follows:
$p = Nl_1w_1[l_2w_2[l_3w_3[\ldots l_Nw_N]]]$

Where:

N	Number of keys/objects in Payload
l_i	length of w_i
w_i	The i'th wrapped key data, i.e. W_{tK}(Key/Object)

Step 3

Generate mK.

Step 4

Calculate the MAC for the Payload, $m = M_{mK}(p)$.

Backup to File

This is the simplest form of backup. The only limitation is that the wrapping key **must** already exist. This key must be able to be re-created after a tamper/failure before a restore can be performed. It may be entered in components, have a known value, or be backed up using the multiple custodian backup mode (described below).

Step 1

Encode mK with tK, $emK = E_{tK}(mK)$

Step 2

Encode tK with wK, $etK = E_{wK}(tK)$

Step 3

Write the binary file containing the backed up Payload. The format of the file is:

Header	Contains the version of the Backup Feature
length p	Length of the encoded Payload
p	Encoded Payload
m	MAC of the Payload
length emK	Length of the Encoded MAC key
emK	Encoded MAC key
length etK	Length of the Encoded Transport key
etK	Encoded Transport key

Step 4

Delete mK and tK.

Backup to Smart Card – Single Custodian Mode

This backup mode has more security than the backup to file mode because the payload is stored on a smart card instead of in a file. The payload data on the smart card is also protected by the custodian's PIN, i.e. the PIN must be presented and authenticated to the smart card before the data can be read.

The only limitation is that the wrapping key **must** already exist. This key must be able to be re-created after a tamper/failure before a restore can be performed. It may be entered in components, have a known value, or be backed up using the multiple custodian backup mode (described below).

If the payload cannot fit on one smart card, then the backup process will prompt the custodian to continue entering new smart cards, until the entire payload has been exported.

Step 1

Encode mK with tK, emK = E_{tK}(mK)

Step 2

Encode tK with wK, etK = E_{wK}(tK)

Step 3

Write the following data files to the smart card:

Header	**Not** protected by custodian's PIN. Contains the following information about the payload: Contains the version of the backup feature Name of this backup payload MAC of the complete payload MAC of the payload component on this smart card, i.e. M_{mK}(p') Timestamp of payload creation Total number of custodians Number of the custodian who owns this smart card Number of the current card being written Flag to indicate if encoded transport key (etK) is on this smart card Flag to indicate if encoded MAC key (emK) is on this smart card Size of the complete payload Size of the payload component on this smart card Offset of this payload component in the complete payload Name of custodian who owns this smart card
Payload	Protected by the custodian's PIN. The component of the payload contained on this smart card. This may be the entire payload.
etK	Protected by the custodian's PIN. Encoded transport key This data file will only be located on the last smart card of the backup set.
emK	Protected by the custodian's PIN. Encoded MAC key This data file will only be located on the last smart card of the backup set.

Step 4

Delete mK and tK.

Backup to Smart Card – Multiple Custodian Mode

This backup mode has the most security. This is because the payload is stored on smart cards and the payload is split between a number of custodians. Also, the payload data on the smart card is protected by the custodian's PIN, i.e. the PIN must be presented and authenticated to the smart card before the data can be read.

The top level wrapping key (wK) is randomly generated, and each custodian has a component of this key. The entire set of smart cards is needed before the wrapping key can be successfully re-created.

If each custodian's payload component cannot fit on one smart card, then the backup process will prompt the custodian to continue entering new smart cards, until their payload component has been exported.

Step 1

Create an initial intermediate wrapping key, which is a triple length DES key, wK', with a value of zero.

For each custodian:

Step 2

Generate random wrapping key component (24 bytes), wC
Derive new intermediate wrapping key wK' = $R_{wK'}$(wC)
Delete the previous intermediate wrapping key (wK'-1)

Step 3

Write the following data files to the smart card:

Header	**Not** protected by custodian's PIN. Contains the following information about the payload: Contains the version of the backup feature Name of this backup payload MAC of the complete payload MAC of the payload component on this smart card, i.e. M_{mK}(p') Timestamp of payload creation Total number of custodians Number of the custodian who owns this smart card Number of the current card being written Flag to indicate if encoded transport key (etK) is on this smart card Flag to indicate if encoded MAC key (emK) is on this smart card Size of the complete payload Size of the payload component on this smart card Offset of this payload component in the complete payload Name of custodian who owns this smart card
wC	Protected by the custodian's PIN. The wrapping key component for this custodian.
Payload	Protected by the custodian's PIN. The component of the payload contained on this smart card.

If last custodian:

Step 4

Encode mK with tK, emK = E_{tK}(mK)

Step 5

Encode tK with the final wrapping key (wK = wK'), etK = E_{wK}(tK)

Step 6

Write the following data files to the smart card:

etK	Protected by the custodian's PIN. Encoded transport key This data file will only be located on the last smart card of the last custodian of the backup set.
emK	Protected by the custodian's PIN. Encoded MAC key This data file will only be located on the last smart card of the last custodian of the backup set.

Step 7

Delete mK, tK and wK.