



Australian Government

Department of Defence

**Contribution to PLAID reference source by the
Australian Department of Defence**

Physical Access Control Manager (PACMan)

User and Key Management Manual

(Unclassified)

Version 1.0 (Draft)

DOCUMENT INFORMATION

Version	Date	Author	Description
1.0	22/07/12	Graeme Freedman	Draft extract of unclassified material from Defence documentation and merge into single document

CONTENTS

Introduction	5
Application Functionality	5
References	6
Scope	6
Dependencies	6
Installation and configuration	8
Installation.....	8
Application Configuration.....	8
Operational Configuration	9
Administrative / Advanced Configuration	11
Create SAM and Keystore cards	13
Load Smart Card SAM & PLAID Applications	15
Generate Keys, Load SAM & Key Store cards	22
Securing the Key Archive files	26
Card Processing and Verification	30
Card Personalisation Steps	31
Card Verification Steps	31
Personalisation Journal (Logging)	32
Securing the Temporary Workstation	33
Security Notes and Considerations.....	34

List of Figures

Figure 1 - PAC Manager home screen	9
Figure 2- PACMan Configuration screen	10
Figure 3 - Setup PACMan CMD window.....	14
Figure 4 - Directory Listing	14
Figure 5 - Start Application Loader	16
Figure 6 - Configure Loader #1	17
Figure 7 - Configure Loader #2	17
Figure 8 - Configure Key Set.....	18
Figure 9 - Authenticate to card.....	19
Figure 10 - Load Applet #1.....	19
Figure 11 - Load Applet #2.....	20
Figure 12 - Load Applet #3.....	21
Figure 13 - Load Applet #4.....	21
Figure 14 - Load Applet #5.....	22
Figure 15 - Generate Keys	23
Figure 16 - Set Profile KA1/KA2.....	23
Figure 17 - Select Reader/ICC.....	24
Figure 18 - Encode KA1/KA2 card	24
Figure 19 - Configure Mifare	25
Figure 20 - Encode KS1 Cards	26
Figure 21 - Key Archive Files	27
Figure 22 - XML Key Format (example – file 1 of 3)	27
Figure 23 - Import Keys.....	28
Figure 24 - Importing keys from 3 part XML files	28
Figure 25 - Export Plain Text Keys	29
Figure 26 - Key Export warning.....	29
Figure 27 - Key Export Verification	29
Figure 28 - PAC Manager Processing screen	30

Figure 29 - Card verification screen	32
Figure 30 - Secured Disk	34

INTRODUCTION

Physical Access Control Manager (PACMan) has been developed by the Australian Department of Defence as a contribution to the Australian Commonwealth PLAID program managed by the Australian Department of Human Services. It is freely available under the Australian Commonwealth PLAID license available at www.plaid.gov.au. This version is the first public release.

PACMan is released as source and object code with the intention of generating critical review of both the architecture and the implementation, and the avoidance of "security by obscurity". This is achieved via an open, low cost and hopefully useful approach. Other agencies and corporations are encouraged to freely use and modify PACMan and to feed back their views.

PACMan is a simple utility, which may be deployed within secure environments for key management, PLAID card application management and card personalisation of PLAID on generic smartcards supporting the US NIST FIPS-201 suite of standards. PACMan uses and supports cards meeting GlobalPlatform Javacard standards with the addition of one or more Australian Standard 5185-2010 (PLAID) applets.

The capability of the solution is primarily targeted at contactless Physical Access Control Systems (PACS) and mobile solutions however it could be utilised for any strong authentication requiring high speed contactless support with key management.

PACMan also optionally supports the personalisation of Mifare Classic. This is supported for transition purposes and is probably not required for most implementations. Other transition strategies are likely and these can be accommodated by treating the Mifare capacity as a sample (or stub) and replacing it with the capability required for the particular implementation.

PACMan uses low cost smartcard security hardware in the form of Secure Access Management (SAM) smartcards and secure Keystore cards to backup, secure and distribute its keys. These cards are simply GlobalPlatform Javacard cards with a variant of the PLAID reference source applet (on-card Javacard application) used to store the key material and perform strong authentication.

Development continues on improvement of the PLAID reference source - new capability introduced in this release will likely be included in the next full release of the reference source. Further, new releases of this material will also likely include other reference source enhancements. Developers should therefore take particular care to read the comments and notes within the source code and to feed back any issues or bugs.

All comments and updates are coordinated using the PLAID web site www.plaid.gov.au and via the PLAID email address plaid@humanservices.gov.au

Application Functionality

PACMan is a standalone application supporting the following functionality;

1. Provides stateless Card Application Management System (CAMS) capability seeded on personalisation data loaded by any FIPS-201 CAMS and supporting GlobalPlatform CPLC (Card Production Life Cycle) capability.
2. Generates, loads and securely manages multiple PLAID key sets including supporting a three part secure import/load of keys
3. Makes use of hardware in the form of Secure Access Management (SAM) smartcards and secure Keystore cards to backup and secure all operational and administrative keys.
4. Loads end-user cards with Mifare Classic keys and sector locks unused Mifare Classic card sectors.
5. Personalises PLAID end-user applets with administrative and operational key sets.

6. Personalises Mifare Classic with stateless PACS data (Weigand record)
7. Personalises PLAID with stateless FIPS-201 and PACS data (multiple Weigand records, FIPS-201 PI (Personal Identifier) and FIPS-201 G/UUID)
8. Deployable both within the secure environment of a data centre for key management and card personalization, and in an off-line less secure environment for card personalization and read-only access to card data.
9. Generates XML records for every card personalized suitable for import to corporate directories in order to enable centralised PACS management.
10. Independently to the CMS, records all available card identification data in a log suitable for forensic analysis

References

The following may be referenced in this document:

1. Personal Computer / Smart Card (PC/SC) Specifications v2.01.10
2. TIG Smartcard Enabled Physical Access Control Systems v2.2 (FIPS 201 CHUID)

Scope

This document describes the operational procedures required to install, configure and operate the PACMan utility as well as the generation and secure management of master keys. Specifically, it details procedures to:

1. Install the PACMan utility
2. Configure the smartcard readers and their encoding parameters
3. Carry out card personalisation and verification
4. Interpret and use the card journal records
5. Generate and back up the PLAID administrative keyset (one keyset)
6. Generate and backup the PLAID operational keyset/s
7. Provision secure Keystore cards with keys
8. Provision the authentication cards (SAM cards) used by PACManager with both existing Mifare Classic keys and (newly generated) PLAID operational keysets
9. Additionally, advise on security assumptions and considerations and areas for future improvement of PACMan

Dependencies

PACMan requires the following software and hardware environment:

1. Microsoft Windows XP SP2 or Microsoft Windows 7 SP1
2. Microsoft .NET Framework 3.5 SP1
(<http://www.microsoft.com/download/en/details.aspx?id=22>)
3. 2 x Contact PC/SC readers
4. 1 x Contactless PC/SC reader
NOTE 1: The reader must have full support for 'Optional Functionality' as defined in Part 3 of the PC/SC specifications.
NOTE 2: The system has been tested with the ASK Logo and SCM Micro SCL3711 readers.
5. A number of GlobalPlatform Javacards for use as SAM and Keystore cards

6. A GlobalPlatform capable card application loader software suitable for the selected cards. This is normally available from the card manufacturer or the public domain GPShell loader is available at <http://sourceforge.net/projects/globalplatform/files/>.

INSTALLATION AND CONFIGURATION

Installation

1. Ensure that the operating system has the latest PC/SC drivers installed, as supplied by the vendor.
2. Ensure that the Microsoft .NET Framework v3.5 – Service Pack 1 (or later) runtime is installed
3. Ensure that the user is logged into the operating system with local administrative rights.
4. Ensure that all 3 PC/SC readers are plugged in and operating (check this via device manager)
5. Create a folder called 'PACMan' on the target PC and extract the contents of the PACMan package into it (by design, PACMan does not have an installation package, so the executable and its supporting files can simply be extracted and are ready-to-run)
6. Create a directory called 'Logs' somewhere on the target PC or on a shared drive (where appropriate). This will serve as the location for log files generated by the application and therefore it is important that this directory be writeable by the currently logged-in user of the application.
7. Optionally, create a shortcut to the application file 'PACManager.exe' on to your desktop, start menu or other desired location.

Installation is now complete.

Application Configuration

Configuration for PACMan is split into four modes:

1. The basic operational configuration, configured using the application itself, which is primarily for defining which smart card readers will be used.
2. Editing the configuration file which resides in the application directory and is called '*PACManager.exe.config*'. This is in XML format and can be modified by any XML or ASCII text editor. Parameters define the numbering schemes used by the application and other advanced parameters.
3. Load of the correct applets at the correct addresses to generate "blank" SAM and Keystore cards.
4. Key generation and load to SAM and Keystore cards

Operational Configuration

Run the application PACManager.exe. You should now be presented with the Home screen, which shows background and links to basic license information.



Figure 1 - PAC Manager home screen

Click 'Next', you are now presented with the application configuration screen.

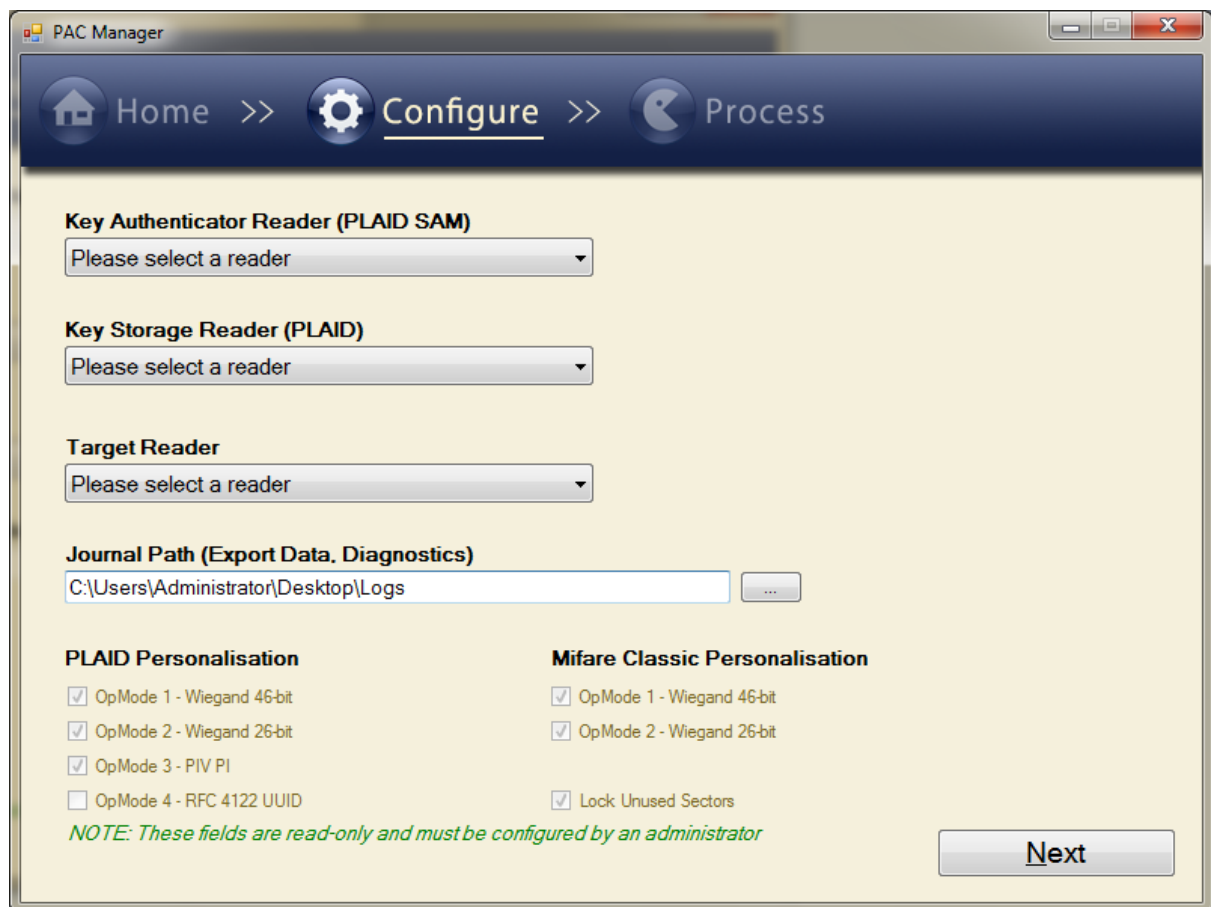


Figure 2- PACMan Configuration screen

The configuration parameters are as follows:

1. **Key Authentication Reader (PLAID SAM)**
This parameter selects which contact PC/SC reader will be used to interface with the Key Authentication (KA) card.
Note: A valid 'Key Authentication' card must be inserted in the reader before you will be allowed to proceed.
2. **Key Storage Reader (PLAID)**
This parameter selects which contact PC/SC reader will be used to interface with the Key Storage (KS) card.
Note: A valid 'Key Storage' card must be inserted in the reader before you will be allowed to proceed.
3. **Target Reader**
This parameter selects which contactless PC/SC reader will be used to encode target (patron) cards. This must be a contactless reader since the Mifare Classic application is only able to be personalized over the card contactless interface.
4. **Journal Path**
This parameter points to the writeable directory that will be used to store the personalisation logs. This may be a directory accessible to the Corporate directory services, or a directory which is backed up regularly for transfer to the corporate directory.
5. **PLAID Personalisation**
This section shows the currently configured personalisation parameters for the PLAID application (these parameters are read-only within the application).

6. Mifare Classic Personalisation

This section shows the currently configured personalisation parameters for the Mifare Classic application (these parameters are read-only within the application).

Administrative / Advanced Configuration

WARNING: Exercise extreme caution in modifying any of these parameters. Most are provided to future-proof the application against changes in numbering schemes, but would normally not require any changes to the default values provided.

To edit the advanced parameters, open the file 'PACManager.exe.config' files in an XML or ASCII text editor. The parameters are defined below and are located in the 'application Settings' xml node.

Name	Type	Description	Default Value
PlaidOpMode1	Boolean	Indicates whether PACMan will personalise the PLAID application with Op Mode 1	True
PlaidOpMode2	Boolean	Indicates whether PACMan will personalise the PLAID application with Op Mode 2	True
PlaidOpMode3	Boolean	Indicates whether PACMan will personalise the PLAID application with Op Mode 3	True
PlaidOpMode4	Boolean	Indicates whether PACMan will personalise the PLAID application with Op Mode 4	False
MifareOpMode1	Boolean	Indicates whether PACMan will personalise the Mifare Classic application with Op Mode 1	True
MifareOpMode2	Boolean	Indicates whether PACMan will personalise the Mifare Classic application with Op Mode 2	True
MifareLockUnused	Boolean	If set to 'True', all unused Mifare sectors will be locked with the production Mifare keys to prevent unauthorised read/writes.	True
AcI SiteCode	UInt16	Specifies the PACS site code that will be used for calculating Op Mode's 1 and 2 actual weigand number.	12345
AcI Seed	UInt32	Specifies that seed value for generating the serial number in Op Mode's 1 and 2. Whatever value this is set to will be added to the CPLC 'IC Serial No' field.	0
Keyset0	UInt16	Allows you to override the PLAID keyset identifier used for the 'Admin' keyset (this should never need to	0

		change).	
Keyset1	UInt16	Allows you to override the PLAID keyset identifier used for the 'Target' or first operational keyset.	4369 (1111h)
OpMode1Id	UInt16	Allows you to set the PLAID identifier for Op Mode 1.	1
OpMode2Id	UInt16	Allows you to set the PLAID identifier for Op Mode 2.	2
OpMode3Id	UInt16	Allows you to set the PLAID identifier for Op Mode 3.	3
OpMode4Id	UInt16	Allows you to set the PLAID identifier for Op Mode 4.	4
OpModeKSIAAdmin	UInt16	Specifies the PLAID OpModeID identifier used for storing the 'Initial Authenticate' Admin key in the Key Storage card.	1
OpModeKSFAAdmin	UInt16	Specifies the PLAID OpModeID identifier used for storing the 'Final Authenticate' Admin key in the Key Storage card.	2
OpModeKSIATarget	UInt16	Specifies the PLAID OpModeID identifier used for storing the 'Initial Authenticate' Target key in the Key Storage card.	3
OpModeKSFATarget	UInt16	Specifies the PLAID OpModeID identifier used for storing the 'Final Authenticate' Target key in the Key Storage card.	4
OpModeKSMifare	UInt16	Specifies the PLAID OpModeID identifier used for storing the PACS Mifare KeyA and KeyB values in the Key Storage card.	5
MifareOpMode1Sector	Byte	Allows you to set the Mifare sector to be used for Mifare personalisation of Op Mode 1.	12
MifareOpMode1Block	Byte	Allows you to set the Mifare block to be used for Mifare personalisation of Op Mode 1.	48
MifareOpMode2Sector	Byte	Allows you to set the Mifare sector to be used for Mifare personalisation of Op Mode 2.	11
MifareOpMode2Block	Byte	Allows you to set the Mifare block to be used for Mifare personalisation of Op Mode 2.	44

KeyStoragePath	String	<i>Key Injection Mode Only</i> – Specifies the path where generated keys will be stored and loaded from.	“.\KeyStorage”
JournalRecoveryPath	String	Specifies the path where journal records will be written to, if for some reason they cannot be written to the user-specified path.	“.\Recovery”
SoundIntroPath	String	Specifies the path to the WAV audio file to be played when entering encode mode.	[Empty]
SoundPassPath	String	Specifies the path to the WAV audio file to be played when a card encode succeeds.	“.\Media\mmpass.wav”
SoundFailPath	String	Specifies the path to the WAV audio file to be played when a card encode fails.	“.\Media\mmfail.wav”
SoundSkipPath	String	Specifies the path to the WAV audio file to be played when a card encode is skipped.	“.\Media\mmskip.wav”

Create SAM and Keystore cards

Configure a stand-alone workstation with the same drivers and configuration as the previous steps.

For this operation there is no need for network access of any kind from the workstation, and the workstation should NOT be connected to any network. The workstation should be re-imaged or securely wiped or destroyed at the completion of this procedure.

Create two directories C:\PACMan and C:\Loader (or whatever names the applet loader you will use requires).

Set up a command prompt screen in the C:\PACMan directory Start>Run>Cmd to obtain a command prompt and change directory to C:\PACMan.

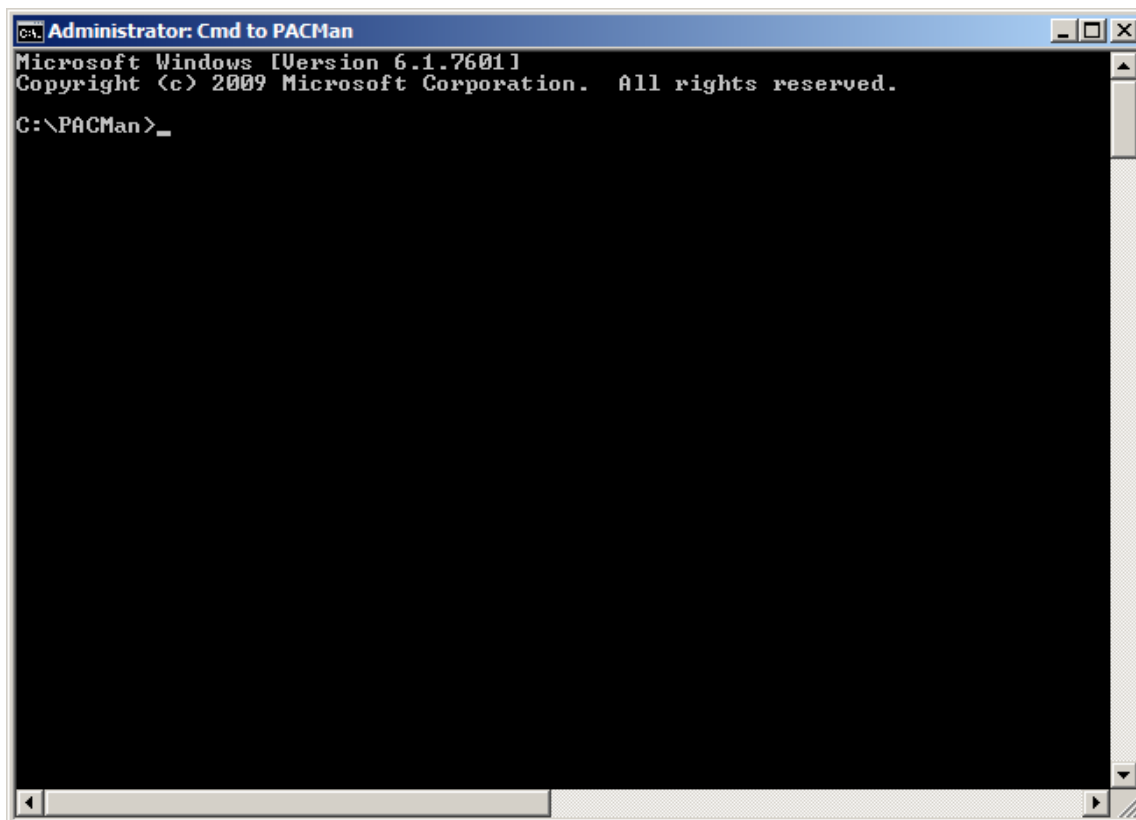


Figure 3 - Setup PACMan CMD window

Now copy the PACMan version 1.0 files and directories into the C:\PACMan directory on the workstation and unzip into the current directory.

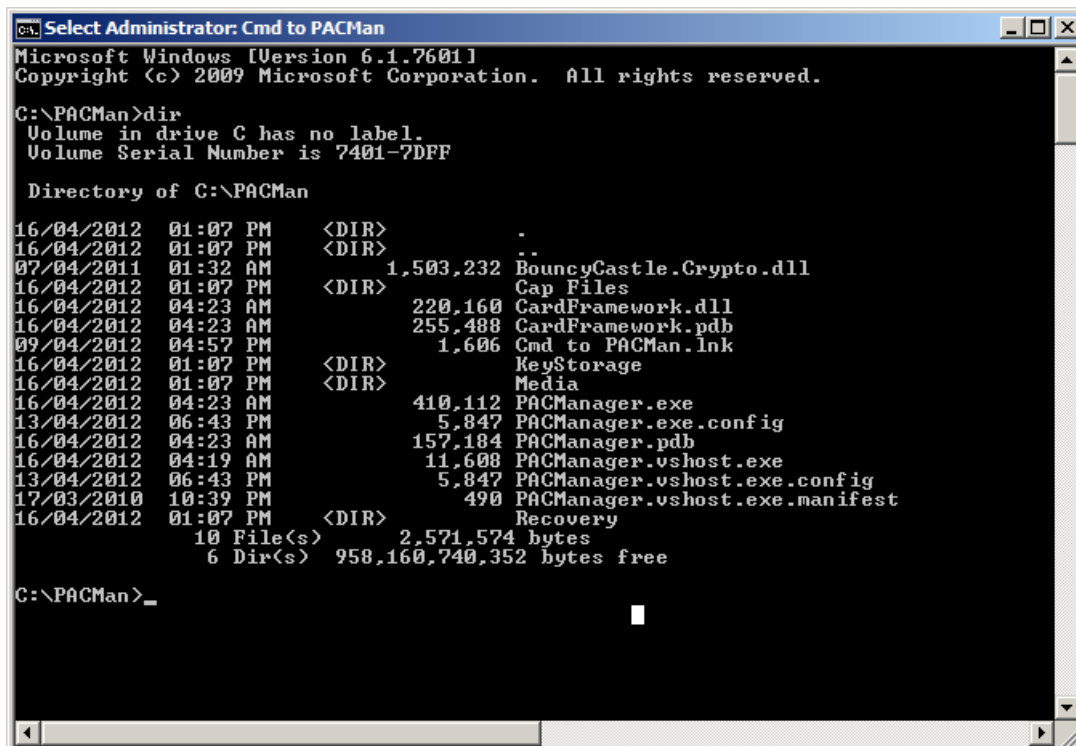


Figure 4 - Directory Listing

Install the card application (applet) loader you will use, either from the card vendor, or you may use the GPShell public domain loader <http://sourceforge.net/projects/globalplatform/files/>. Follow the card manufacturers instructions including obtaining the keys and configuration details for the

GlobalPlatform environment. Depending on the card selected and the vendor it may be necessary to recompile the supplied applets with specific compile time options.

Load Smart Card SAM & PLAID Applications

Obtain sufficient GlobalPlatform Javacards for the current operational requirement. The number required can be estimated from the recommendations in the following table:

Label	Card Type	Applet Type	Min No	Purpose	Operational Requirement	Package, Class & Instance AID respectively in Hexadecimal
KS1-XX	KeyStore	PLAID Default	3	Secure storage and recovery of PLAID and MiFare Keys	1 in off-site safe, 1 in local safe and 1 for each operational issuance station. Store in separate safe to KA1 or physically secure adjacent to workstation.	A0 00 00 04 64 64 A0 00 00 04 64 A0 00 00 04 64
KA1-XX	Key Authenticator - Admin	PLAID SAM	3	Secure Authentication to KS1 in order to access keys for personalisation operations	1 in off-site safe, 1 in local safe and 1 for each operational issuance station. Store in separate safe to KS1	A0 00 77 6B 67 67 A0 00 77 6B 67 60 A0 00 77 6B 67 60
KA2-XX	Key Authenticator - Read Only	PLAID SAM	0	Read/check card reference data such as PACS Weigand numbers and UUID in the field. Primarily needed by PACS administrators on non corporate systems.	1 for each read-only field station. Make secure in safe when not in use or otherwise secure physically. Loss of these only risks privacy of credential values like weigand numbering. Loss does not risk the KeySet itself.	A0 00 77 6B 67 67 A0 00 77 6B 67 66 A0 00 77 6B 67 66
NA	End User Cards	PLAID Default	0		1 for each end user of the system	A0 00 00 04 64 64 A0 00 00 04 64 A0 00 00 04 64

Table 1 – Administrative Card Types and AIDs

The cards used for these administrative tokens can be the same as those used in normal production. Alternatively other Global Platform JavaCards could be utilised. The only difference is that they are not personalised in the normal way (i.e. do not personalise these cards in the normal CMS). The cards should be marked indelibly with the label and sequence number (XX) in column one of Table 1 (above) and a manual log entry should be created for each card produced to indicate the location where the card is stored along with the officers who controls the card.

The following is an **example only** of the type of manual log that could be kept. Normally this would be customised to fit in with existing security procedures for the storage of key material:

Label	Card Type	Storage Safe Location	Operations Location	Role Responsible	Officer Responsible	Signature	Date
KS1-01	KeyStore	CDMCS03	NA	H Trusted A	ACH	xxxxxxxxxx	
KS1-02	KeyStore	CDMCS04	NA	H Trusted B	DGR	xxxxxxxxxx	
KS1-03	KeyStore	CDMOS01	NA	H Trusted C	HSL	xxxxxxxxxx	
KS1-04	KeyStore	CDMCO1	CDMCP1	Trusted D	SDH	xxxxxxxxxx	
KS1-05	KeyStore	CDMCO2	CDMCP2	Trusted E	DFG	xxxxxxxxxx	
KA1-01	Auth Admin	CDMCS08	NA	H Trusted A	ACH	xxxxxxxxxx	
KA1-02	Auth Admin	CDMCS09	NA	H Trusted B	DGR	xxxxxxxxxx	
KA1-03	Auth Admin	CDMCS02	NA	H Trusted C	HSL	xxxxxxxxxx	
KA1-04	Auth Admin	CDMCO1	CDMCP1	Trusted D	SDH	xxxxxxxxxx	
KA1-05	Auth Admin	CDMCO2	CDMCP2	Trusted E	DFG	xxxxxxxxxx	
KA2-01	Auth Read Only	DSG-APW	DSG-APW	SI Trusted	CH	xxxxxxxxxx	
KA2-02	Auth Read Only	DSG-CP	DSG-CP	SI Trusted	GH	xxxxxxxxxx	
KA2-03	Auth Read Only	DSG-CH	DSG-CH	SI Trusted	JK	xxxxxxxxxx	
KA2-04	Auth Read Only	DSG-FW	DSG-FW	SI Trusted	SD	xxxxxxxxxx	
KA2-05	Auth Read Only	DSG-TEST	DSG-TEST	SI Trusted	KL	xxxxxxxxxx	
KA2-06	Auth Read Only	DSG-FG	DSG-FG	SI Trusted	SD	xxxxxxxxxx	

Table 2 - Manual Log File – Example Only

Once the full batch of administrative cards has been determined, label each card in the batch clearly with the correct label using an indelible method.

Start the application loader. The following is an example of the use of a loader from one vendor. Others are similar but will differ in the interface. GPshell is CMD prompt script based

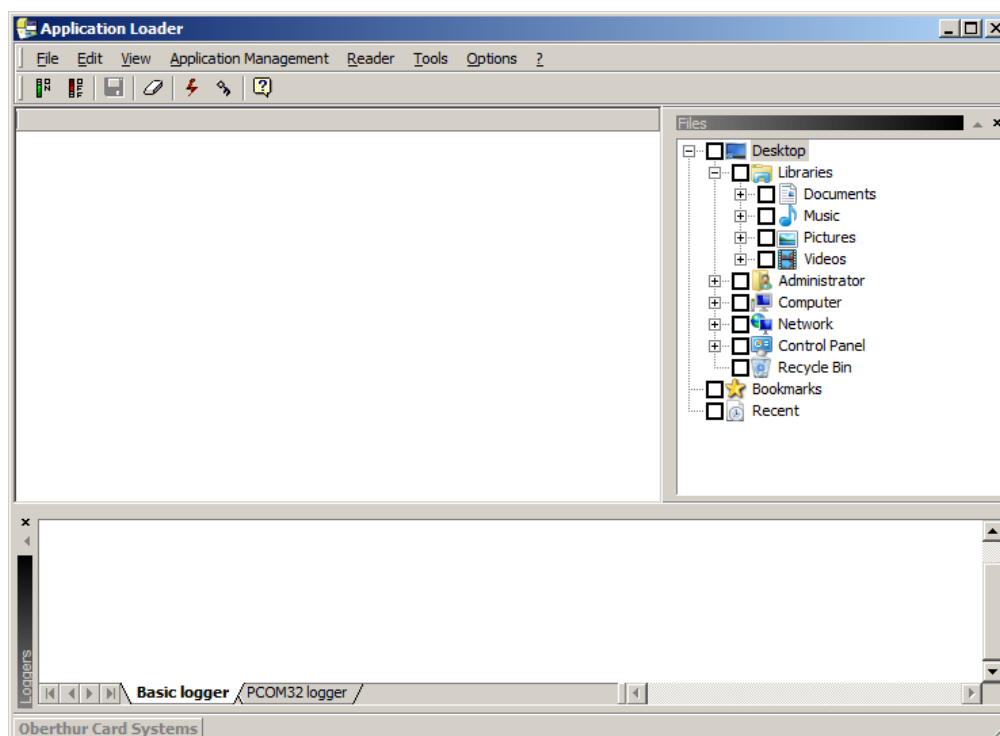


Figure 5 - Start Application Loader

Insert the first blank card and Click the “On” button:

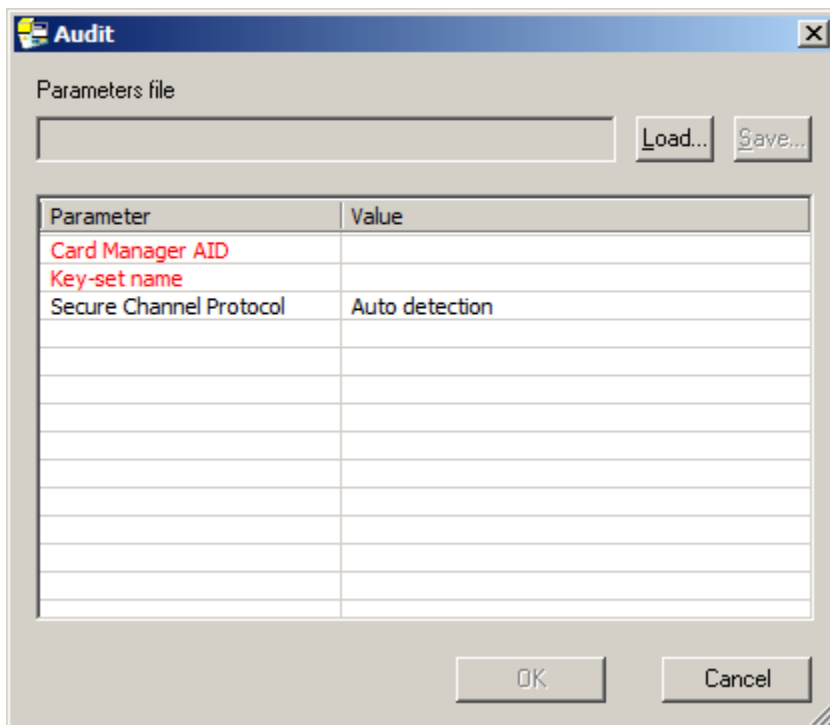


Figure 6 - Configure Loader #1

Assuming the card is an Oberthur ID One-PIV version 7, fill in the following values (Card Manager AID is "A0000001510000"):

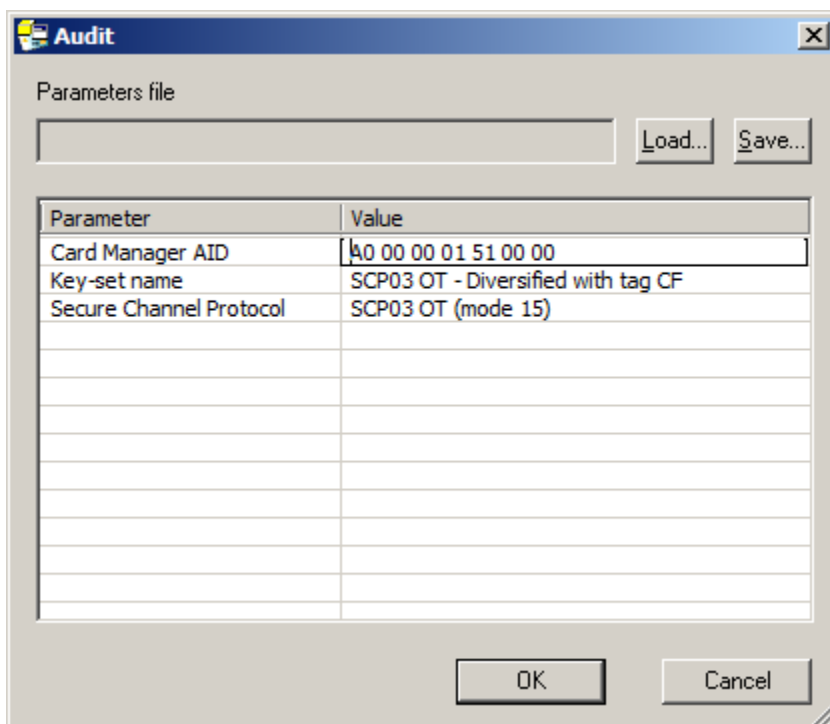


Figure 7 - Configure Loader #2

The key set values will need to be updated with the transport keyset (Master key) agreed between the vendor and the scheme: In this example Select Key-set-name > "SCP03 OT – Diversified with tag CF" and enter the correct value for the Master Key. Select "Confidentiality and Integrity"

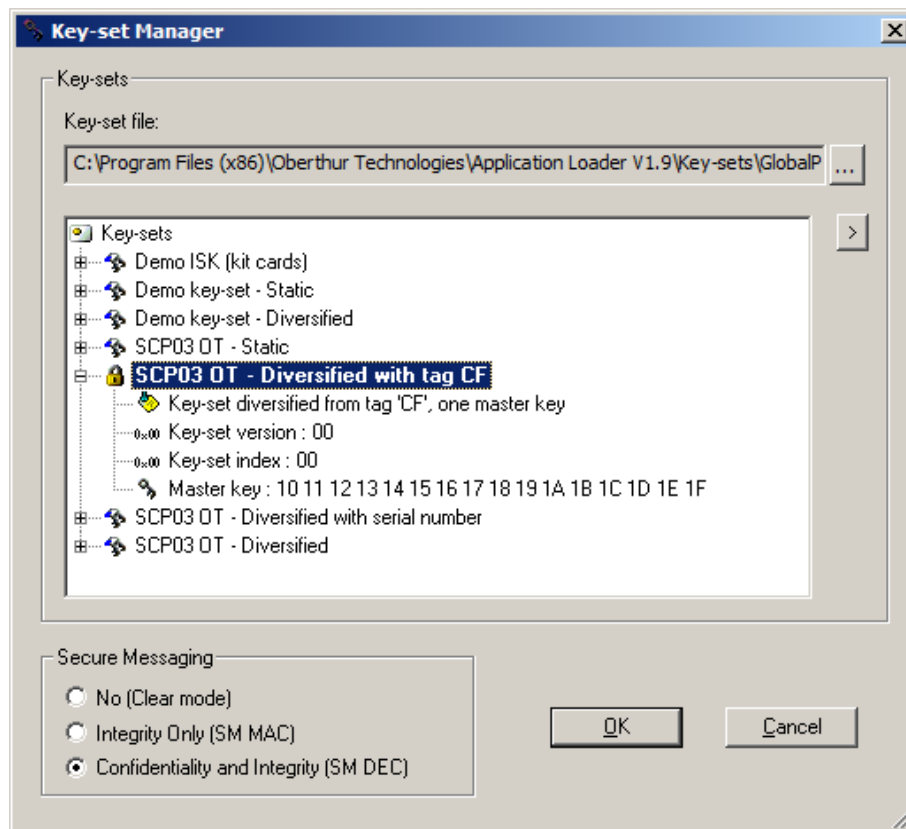


Figure 8 - Configure Key Set

Click “OK” and an authentication should take place followed by an “Operation succeeded” message .

If this is not successful, then debug the problem before proceeding. Any problems are most likely from selecting the wrong keyset or the wrong Globalplatform Secure Channel protocol options. The screen should now list all applications on the card. In this example the existing applications are the default FIPS-201 applications.

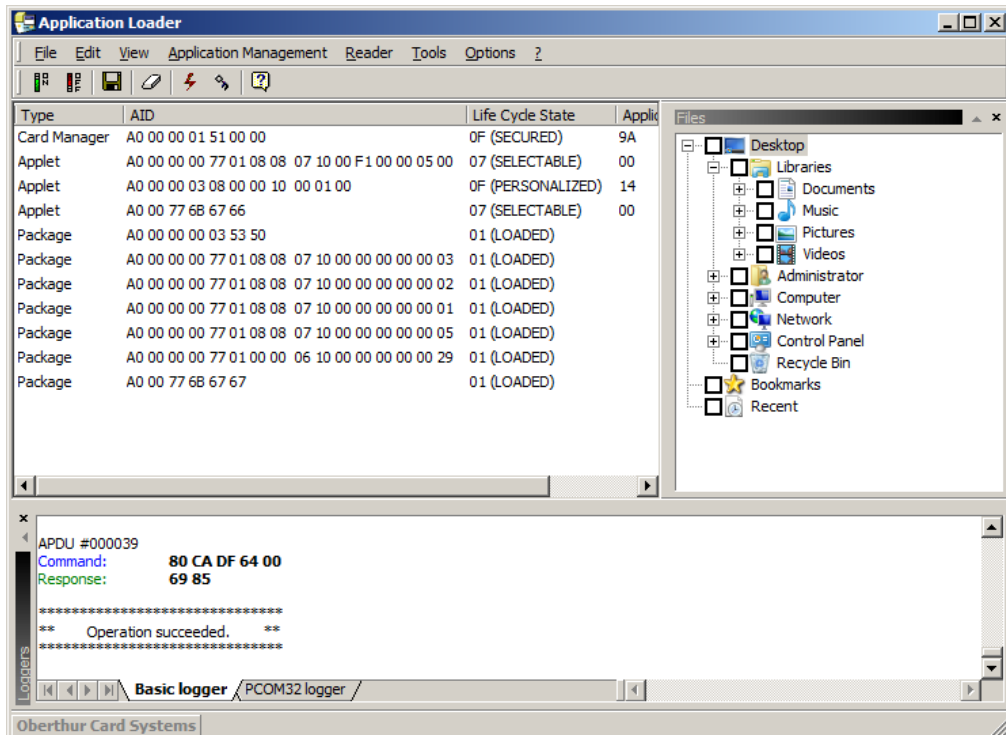


Figure 9 - Authenticate to card

For each card you are now going to load a new application on the card using the address and application type listed in table 1.

Go to “Application Management” > Load and Install”

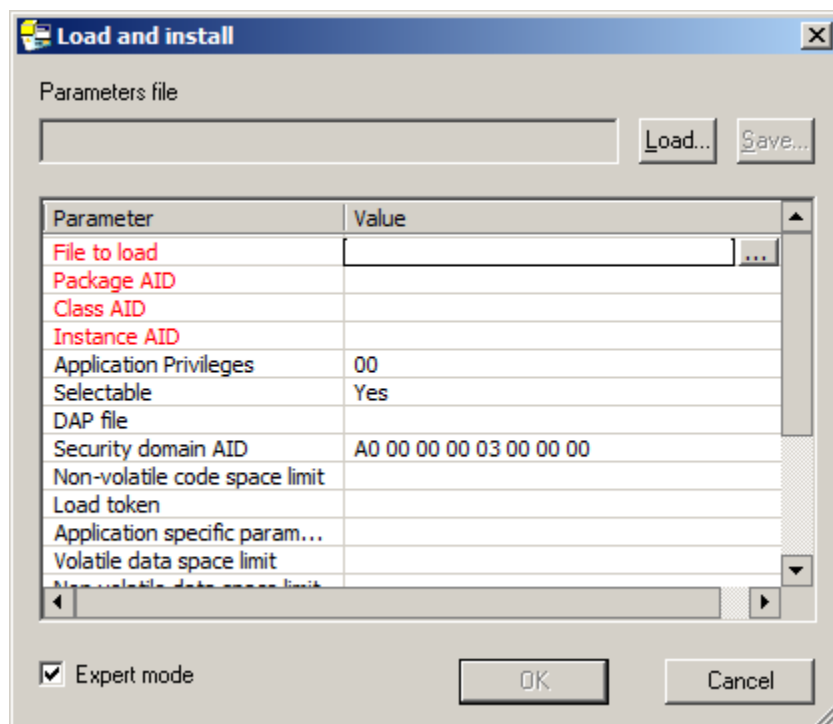


Figure 10 - Load Applet #1

Select “File to Load”

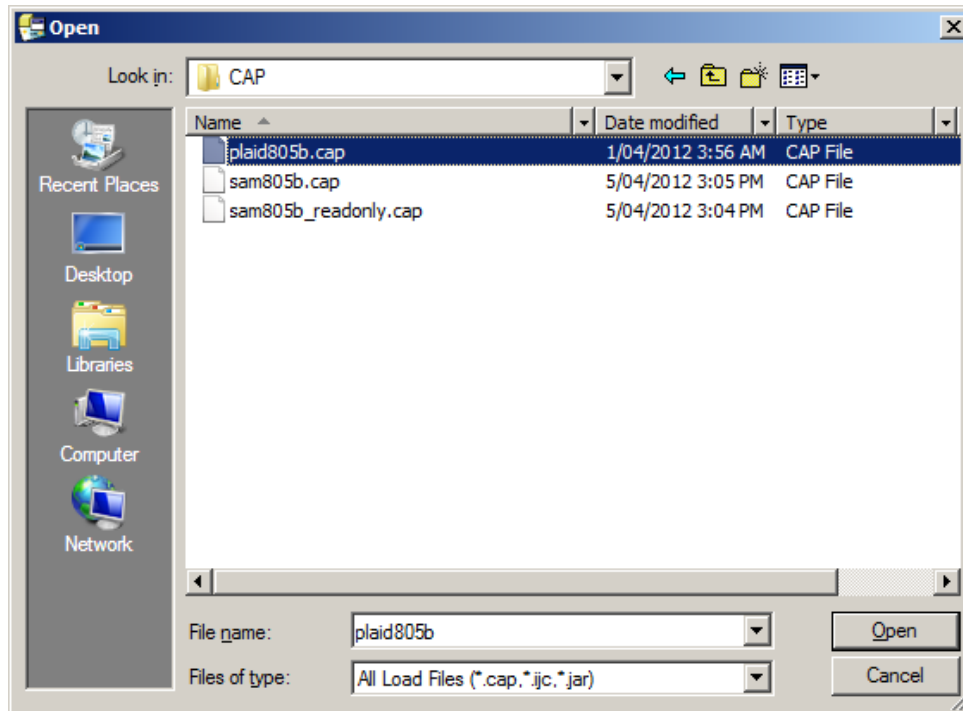


Figure 11 - Load Applet #2

Look in the C:\PACMan\CAP directory and select the correct file based on Table 1. There are only three options:

1. PLAID Client – the same application is used for all PLAID use-cases\
2. SAM – Administrative version
3. SAM – Read only version

Once selected, tab to the next field and the addresses other than “security domain AID” should be filled in correctly for you. Change the security domain AID to: “a0000001510000”

Check every AID field against table 1, and correct if necessary

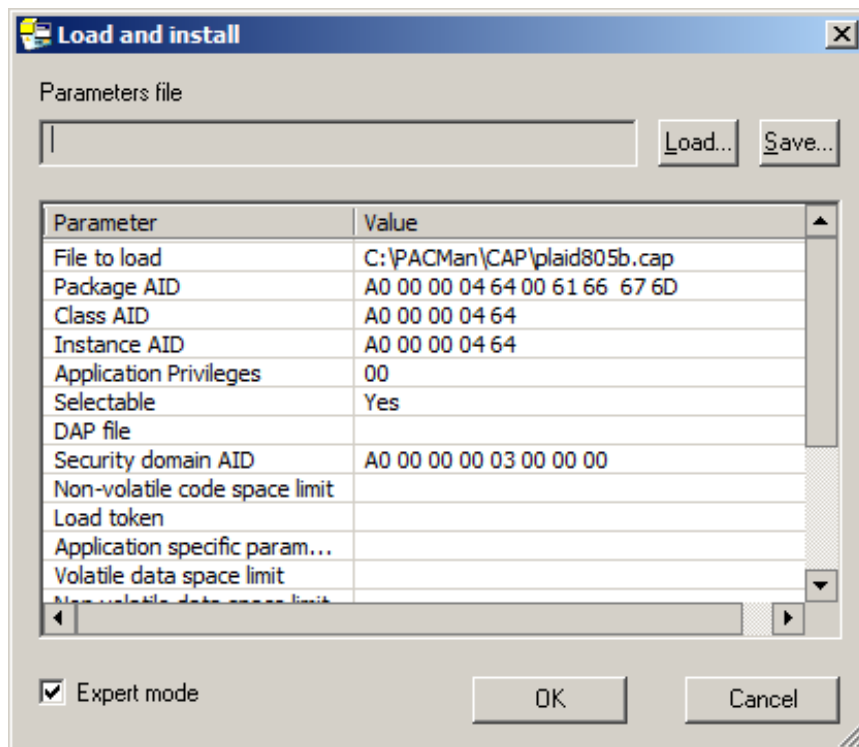


Figure 12 - Load Applet #3

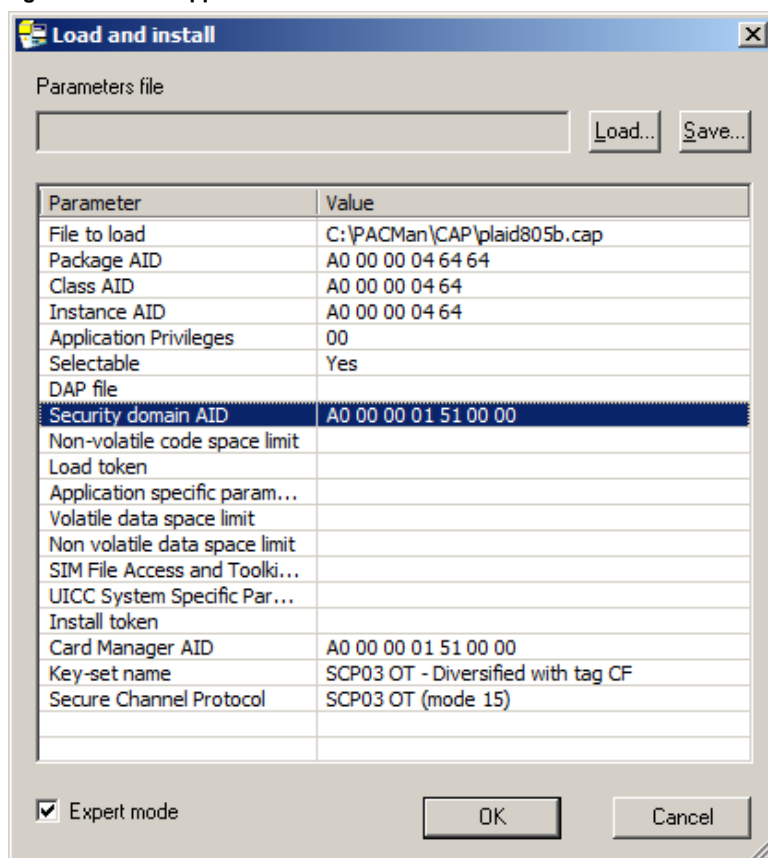


Figure 13 - Load Applet #4

Click OK and the application will be loaded and should now appear in the list on-screen along with "operation succeeded" in the log:

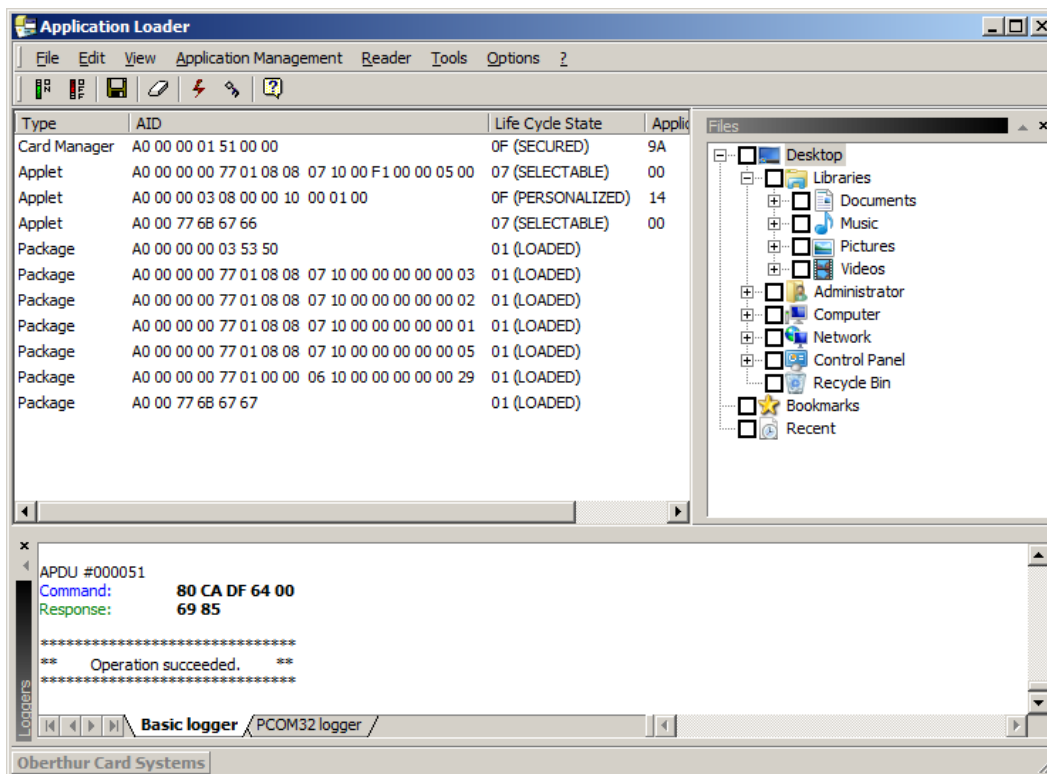


Figure 14 - Load Applet #5

If desired the transport keyset can be replaced with an operational keyset via the “replace a keyset in a security domain” option at this point.

Remove the card and repeat the process for all cards using table 1 and the draft log (table 2) as the guide for applications to be loaded.

Your cards are now loaded with the correct applets but not “keyed” or personalised with key material.

Generate Keys, Load SAM & Key Store cards

Using the same PC as in the previous step, go to the Cmd prompt in the C:\PACMan directory.

Double check the machine used is not connected to ANY network and is in an adequately secure area.

If this is the first time keys are to be generated or imported, delete the various KeyArchive-Xof3.xml files. They only contain default test key values as distributed.

Run the PACManager utility from that directory with the special option –k. This starts the utility in the non-default key management mode. “PACManager.exe –k”

Select either to generate keys or to import keys. Keys are generated to or imported from the KeyArchive-1of3.XML, KeyArchive-2of3.XML and KeyArchive-3of3.XML files.

A separate process is used to secure these files – see “Securing the Key Archive files”

If this is the first time this utility is run, the “generate keys” option is the only option possible. If this utility is being run to create additional administrative or key store cards then use the “Import Keys” option.

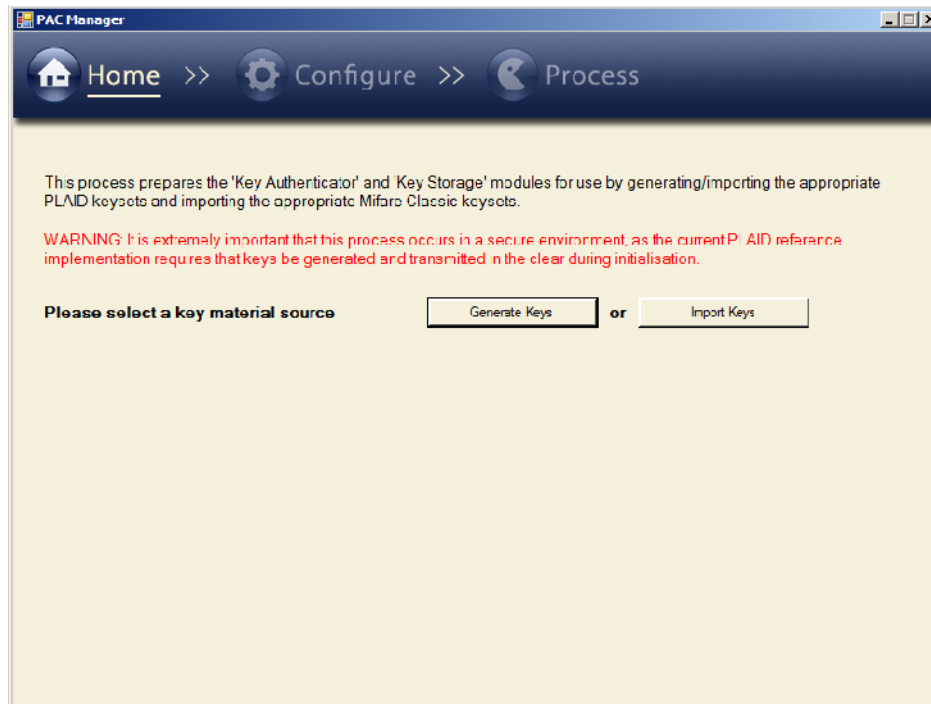


Figure 15 - Generate Keys

Now select an encoding profile from Table 1 and your draft log (table 2) and select the equivalent card profile using the PACMan utility.

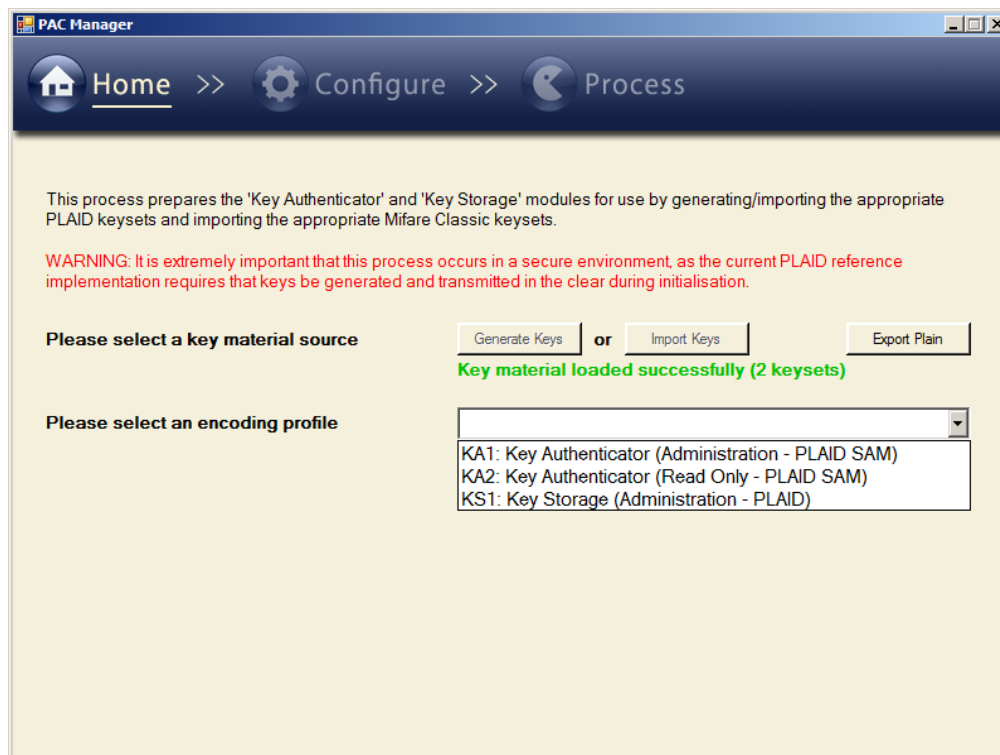


Figure 16 - Set Profile KA1/KA2

Select a contact only card reader and insert the correct card for the profile. If you insert a card in the reader with the wrong applet loaded you will get a warning "Incorrect ICC detected". Change either the profile or the card until you have the correct applet and profile.

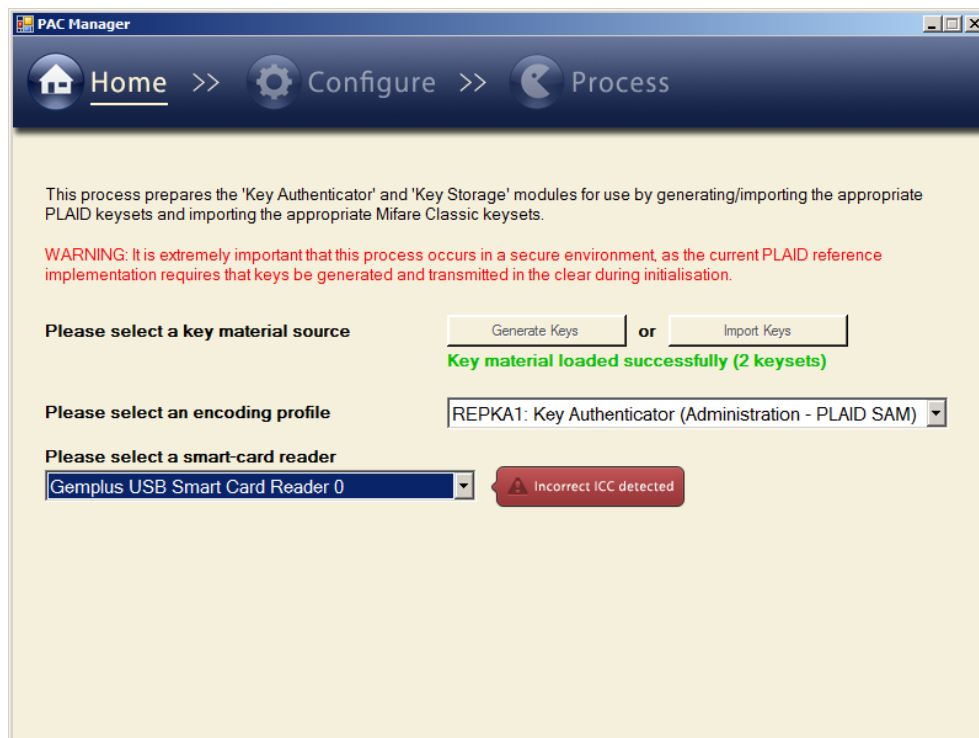


Figure 17 - Select Reader/ICC

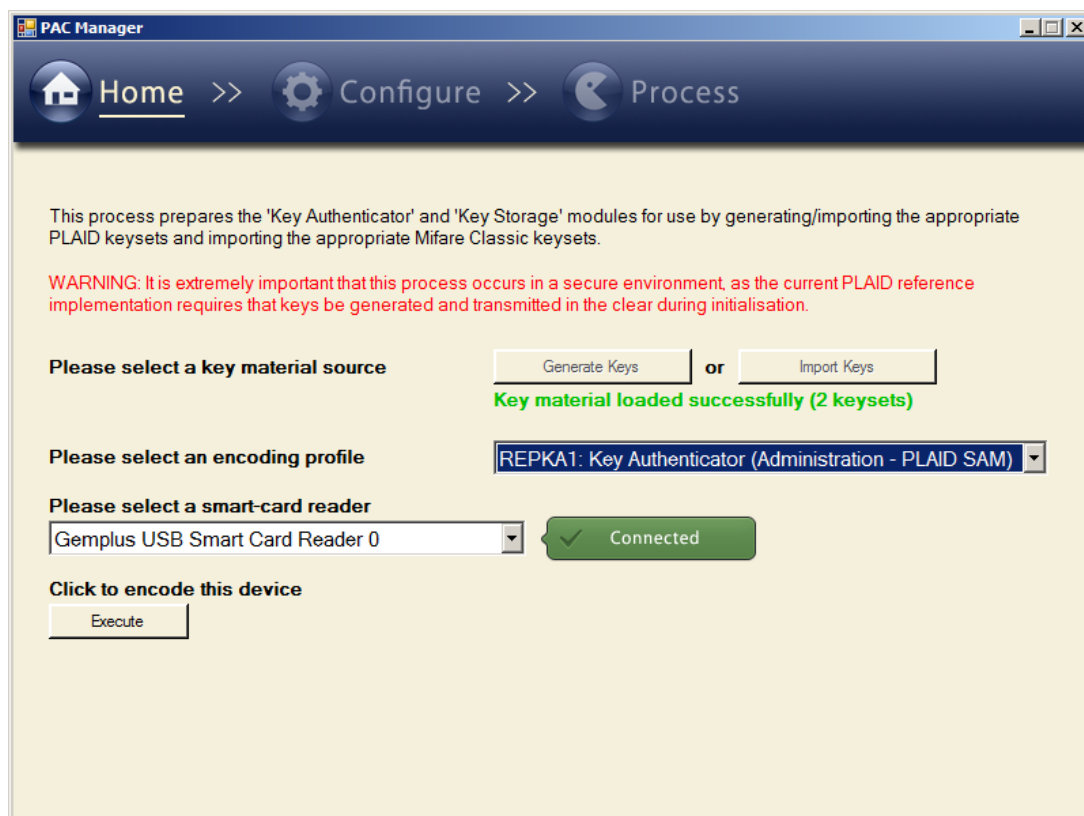


Figure 18 - Encode KA1/KA2 card

You may now encode (initialise keys) to all cards with the KA1 and KA2 profiles by clicking “execute”. This only takes a second or so for each card and is irreversible. A message appears stating that the process has completed successfully. If there are any errors the issue should be investigated and if the card cannot be encoded successfully it should be physically destroyed.

There is an additional step for the KS1 (key store) profile.

PAC Manager

Home >> Configure >> Process

This process prepares the 'Key Authenticator' and 'Key Storage' modules for use by generating/importing the appropriate PLAID keysets and importing the appropriate Mifare Classic keysets.

WARNING: It is extremely important that this process occurs in a secure environment, as the current PLAID reference implementation requires that keys be generated and transmitted in the clear during initialisation.

Please select a key material source or

Key material loaded successfully (2 keysets)

Please select an encoding profile

Please enter the Mifare Classic key values KEY A KEY B ☐ Show

Please select a smart-card reader

Click to encode this device

Figure 19 - Configure Mifare

For this profile the Mifare Classic keys need to be imported from the secure record kept in the respective CDMC safe. This is a two part record which should be entered by two separate trusted operators.

The record expected is a hex value in the form Key A="FFFFFFFFFFFFFF" Key B="FFFFFFFFFFFFFF".

NOTE - PACMan does not backup or manage the Mifare Keys in any recoverable form, so existing facilities for backing up the Mifarre keys should be retained.

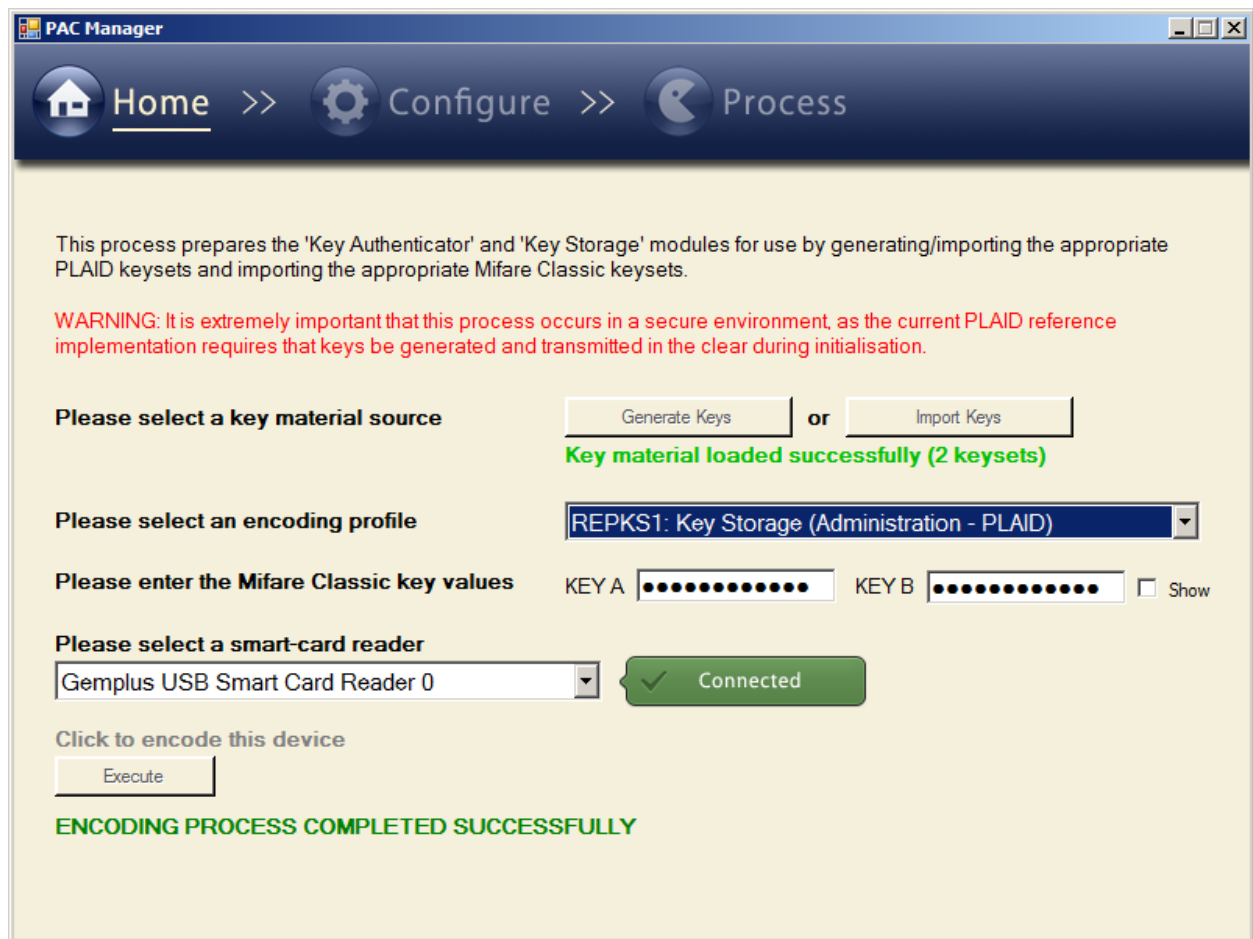


Figure 20 - Encode KS1 Cards

You may now encode (store keys) to all cards with the KS1 profile by clicking “execute”. This only takes a second or so for each card and is irreversible. A message appears stating that the process has completed successfully. If there are any errors the issue should be investigated and if the card cannot be encoded successfully it should be physically destroyed.

The cards created should now be tested using the operational mode of PACManager (see separate operations document) and then stored in the appropriate safes whilst not in use.

Securing the Key Archive files

PACMan by default stores PLAID operational and Administrative keys in a 3 part form in the KeyStorage directory in three separate XML files as shown here:

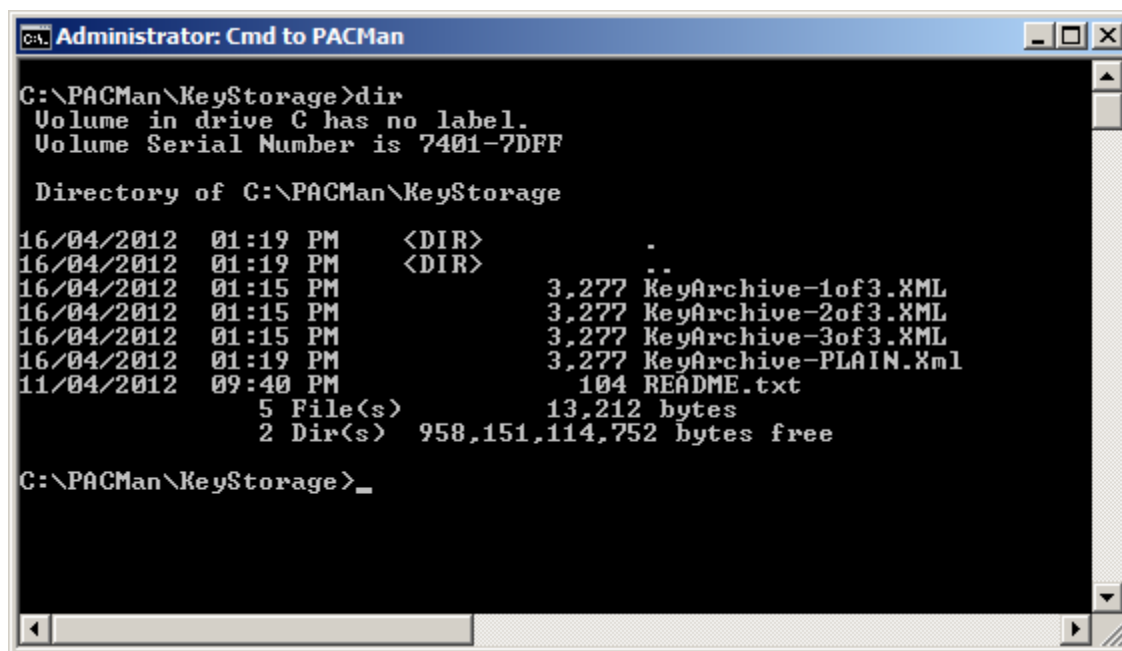


Figure 21 - Key Archive Files

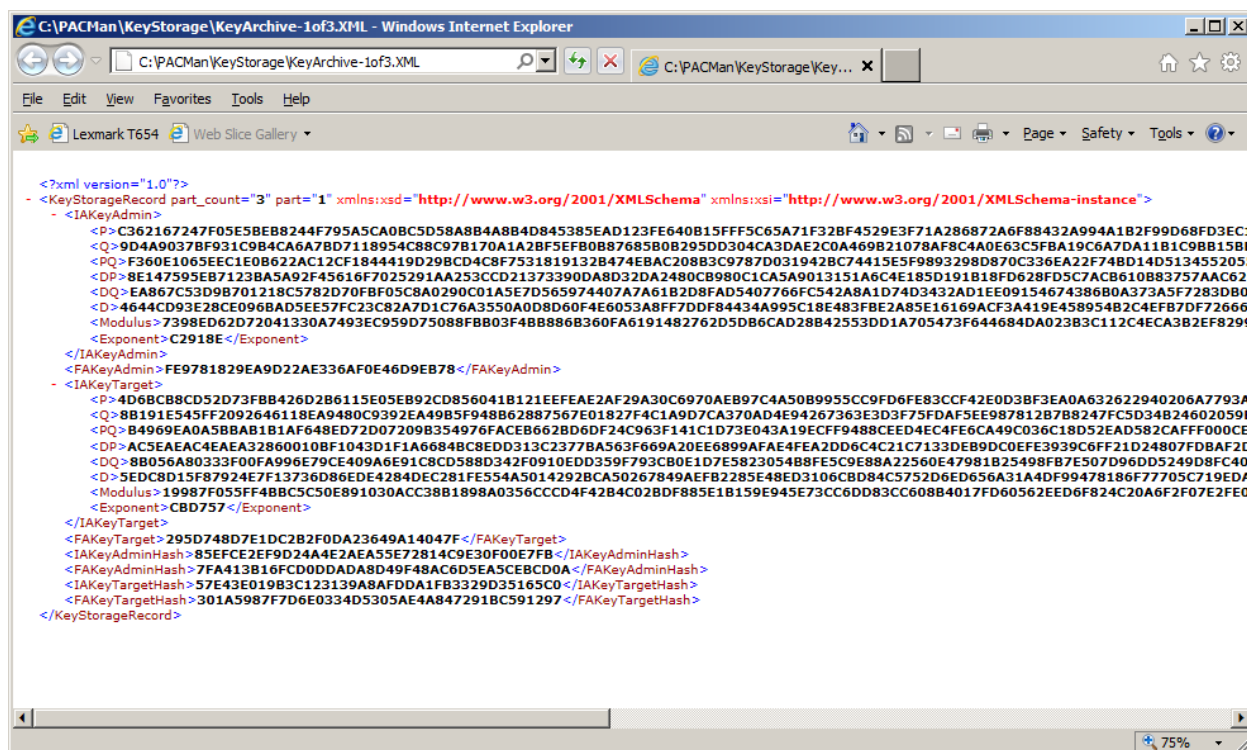


Figure 22 - XML Key Format (example – file 1 of 3)

These files are created at the time of keyset generation, and must be manually deleted should a new keyset be created.

PACMan uses random seeds and strong cryptography to secure these files. In order to establish the original keys, all three files must be available and un-modified, Any single file or pair of these files are useless to an attacker.

Once generated, a copy of each of these files should be placed in three separate safes or secure systems designed for this purpose. Each file should be under the control of different trusted people. Additionally a backup set should be kept with similar three part controls. The original keyset can be re-generated at any time by placing the files in a common keystorage directory and running PACManager –k and clicking "Import Keys".

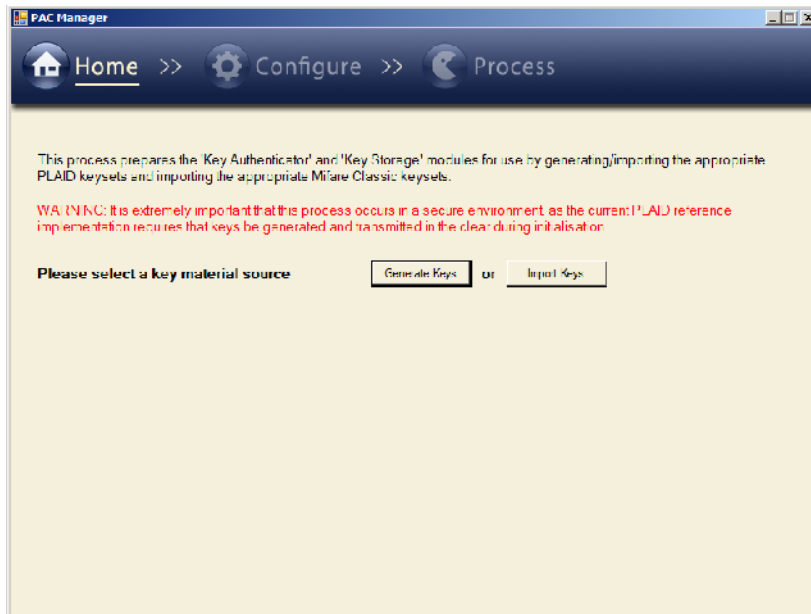


Figure 23 - Import Keys

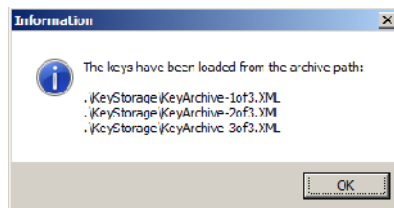


Figure 24 - Importing keys from 3 part XML files

It is not necessary for the administrator to know or to generate the clear text version of the actual keys, however many administrators will not be comfortable with this, or may wish to generate multiple keysets and prove to themselves the key management works before fully trusting the method used to secure the PLAID master keys. To facilitate this an option is available to write out (export) the plain text keys as a separate file "KeyArchive-PLAIN.xml". It is recommended that any file generated by this method should be secured in the most secure facility possible. And preferable this option should never be utilised, or only be utilised in testing so as to provide confidence to the administrator that the keysets are recoverable.

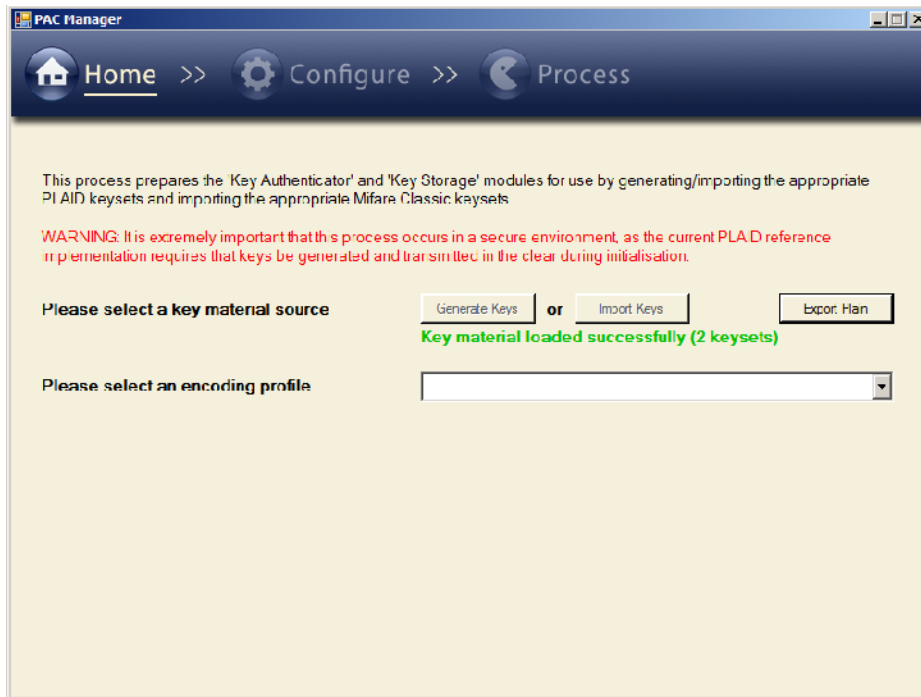


Figure 25 - Export Plain Text Keys

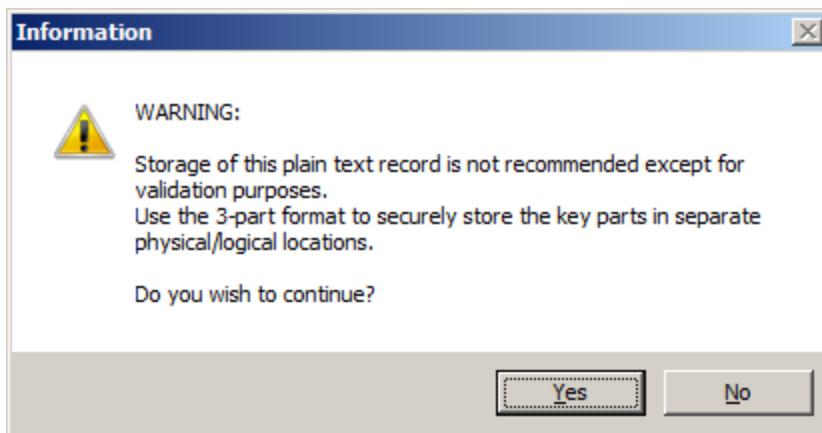


Figure 26 - Key Export warning

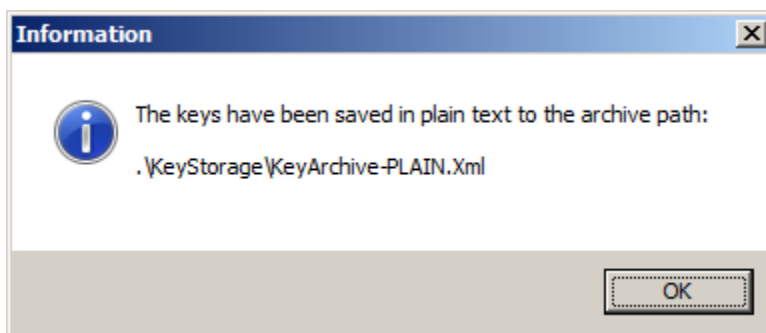


Figure 27 - Key Export Verification

Card Processing and Verification

Once you have configured PAC Manager, click 'Next' to proceed to the 'Process' screen.

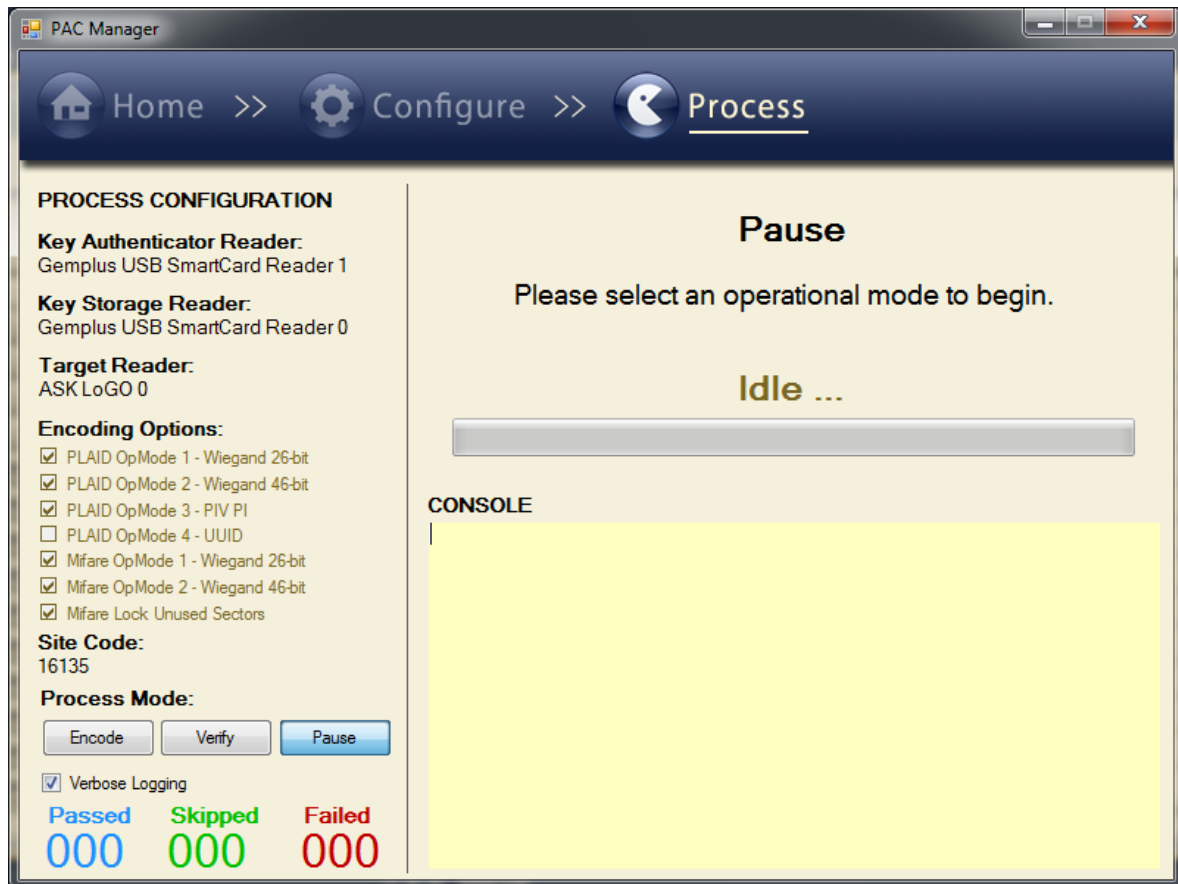


Figure 28 - PAC Manager Processing screen

The process screen contains the following user interface elements:

- Upper Left – Personalisation Options**
This section lists the options that have been configured for this session. All values here are read-only and are only used for reference.
- Lower Left – Process Mode**
This section allows the operator to select whether the application is in personalisation mode (Encode), verification mode (Verify) or is Idle (Pause).
- Lower Left – Verbose Option**
If this is selected, a greater amount of processing information will appear in the console, allowing quick diagnosis of any operating errors.
- Lower Left – Personalisation Counters**
These three coloured counters provide a quick reference of the count of cards personalised during any given session. These values are not saved and only exist to assist in bulk personalisation where a higher number of cards are being encoded at once.
- Upper Right – Status panel**
This section displays a simply update of the current session status, as well as an indicator of encoding progress.
- Lower Right – Console**
This shows personalisation, verification, diagnostic and error information to provide basic, or detailed feedback to the operator.

Card Personalisation Steps

To enter personalisation mode, click the 'Encode' button. The following steps will occur:

- a. The Key Authentication and Key Storage devices (cards) will be checked
- b. The cryptographic material will be extracted from the Key Storage device and stored for this session (See 'Security Considerations' below for more information).
- c. The target reader will be monitored for cards
- d. Once a card is placed on the reader;
 - a. The card will be validated
 - b. The GlobalPlatform CPLC data element will be read and validated
 - c. The FIPS 201 CHUID data element will be read and validated
 - d. The Op-Mode values will be generated.
 - e. The PLAID application will be personalised with the selected Op Modes and placed in the 'Secured' state.
 - f. The Mifare Classic application will be personalised and (optionally) all unused sectors will be changed from the factory keys to the production keys. All writes to the Mifare Classic data blocks will be verified by a subsequent read.
 - g. The PLAID application will be validated with a subsequent authentication.
 - h. The card journal will be written
 - i. The card will be deselected and the target reader will wait for the card to be removed
 - j. The target reader will wait for the next card.

If the 'Pause' button is clicked, the application will immediately clear all cryptographic data and deselect all cards.

CARD TEARING

The target card must not be removed from the field during the personalisation process. This process is particularly sensitive to tearing due to the fact that multiple applications are involved and the cards are being handled manually, rather than in an automated fashion. A tear will leave the card in an unknown state that potentially will require manual intervention to correct.

Card Verification Steps

To enter verification mode, click the 'Verify' button. The following steps will occur:

- a. The Key Authentication device will be checked (the Key Storage module is not used in this case).
- b. The target reader will be monitored for card.
- c. Once a card is placed on the reader;
 - a. The GlobalPlatform CPLC data element will be read and validated
 - b. The FIPS 201 CHUID data element will be read and validated
 - c. All available PACS records will be read from the PLAID application
 - d. The Mifare application will be checked for its authentication status (NOTE: Because this process does not have access to the production keys, it will only validate that the Mifare application is not in its factory state.
 - e. The verification screen will be displayed.

The Card Viewer dialog box displays the following information:

- Card Status:**
 - PLAID: [Yellow field]
 - Mifare: [Yellow field]
- CPLC:**
 - SERIAL: [Yellow field] [...]
- CHUID:**
 - FASCN PI: [Yellow field] [...]
 - GUID: [Yellow field] [...]
- ACS RECORDS:**
 - OpMode 1: [Yellow field] [...]
 - OpMode 2: [Yellow field] [...]
 - OpMode 3: [Yellow field] [...]
 - OpMode 4: [Yellow field] [...]

A **Close** button is located at the bottom of the dialog.

Figure 29 - Card verification screen

Personalisation Journal (Logging)

For each card that is personalised using the PACMan utility, a corresponding log file, or 'journal' entry is created and stored to facilitate further back-office processing of the card, or integration into third-party PACS/LACS control systems.

Upon successful personalisation, the journal entry is saved in the directory specified on the configuration screen. In the event that the specified directory is not accessible or writeable, the journal will be saved to a secondary directory (by default, in the directory 'Recovery' underneath the application folder) and an error message will be displayed on screen.

The card journal is an XML formatted text document and the schema (XSD) is supplied with the application in the file 'journal.xsd' for type and structural definitions.

There are 7 primary elements in the journal:

- 1) 'ATR' – The ISO 7816 ATR value returned by the reader
- 2) 'CPLC' – The GlobalPlatform CPLC data element in its parsed form
- 3) 'CHUID' – The FIPS 201 CHUID data element in its parsed form
- 4) 'OpMode1' – The value of OpModeID 1 in hexadecimal form as written to the card
- 5) 'OpMode2' – The value of OpModeID 2 in hexadecimal form as written to the card
- 6) 'OpMode3' – The value of OpModeID 3 in hexadecimal form as written to the card
- 7) 'OpMode4' – The value of OpModeID 4 in hexadecimal form as written to the card

An example of the log file is displayed below:

```
<?xml version="1.0"?>
<EncodeJournal xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  <ATR>3B8F80018073CC91CBF9A0000003080000100029</ATR>
  <CPLC>
    <ICFabricator>1234</ICFabricator>
    <ICType>12345</ICType>
    <OSProviderId>12345</OSProviderId>
    <OSReleaseDate>12345</OSReleaseDate>
    <OSReleaseLevel>12</OSReleaseLevel>
    <ICFabricationDate>12345</ICFabricationDate>
    <ICSerialNo>1234567890</ICSerialNo>
    <ICBatchId>1234</ICBatchId>
    <ICModuleFabricator>1234</ICModuleFabricator>
    <ICModulePackagingDate>1234</ICModulePackagingDate>
    <ICCManufacturer>1234</ICCManufacturer>
    <ICEmbeddingDate>1234</ICEmbeddingDate>
    <PrePersoId>1234</PrePersoId>
    <PrePersoDate>1234</PrePersoDate>
    <PrePersoEquipment>1234</PrePersoEquipment>
    <PersoId>1</PersoId>
    <PersoDate>1</PersoDate>
    <PersoEquipment>1</PersoEquipment>
  </CPLC>
  <CHUID>
    <FASCN>
      <AgencyCode>9999</AgencyCode>
      <CredentialNumber>999999</CredentialNumber>
      <SystemCode>9999</SystemCode>
      <CS>1</CS>
      <ICI>1</ICI>
      <PI>123456789</PI>
      <OC>1</OC>
      <OI>9999</OI>
      <POA>1</POA>
    </FASCN>
    <GUID>00000000000000000000000000000000</GUID>
    <ExpirationDate>2013-03-02T00:00:00</ExpirationDate>
    <IssuerSignature>308207207E43082[....]</IssuerSignature>
  </CHUID>
  <OpMode1>0123456789ABCDEF</OpMode1>
  <OpMode2>01234567</OpMode2>
  <OpMode3>0123456789</OpMode3>
  <OpMode4>00000000000000000000000000000000</OpMode4>
</EncodeJournal>
```

Securing the Temporary Workstation

The previous procedures should have been implemented on a stand-alone temporary workstation within a secure area. Since it is possible that keys could be saved to disk and be recoverable, it is important to ensure any data written to disk is destroyed at the end of the procedure according to the agency standard practice.

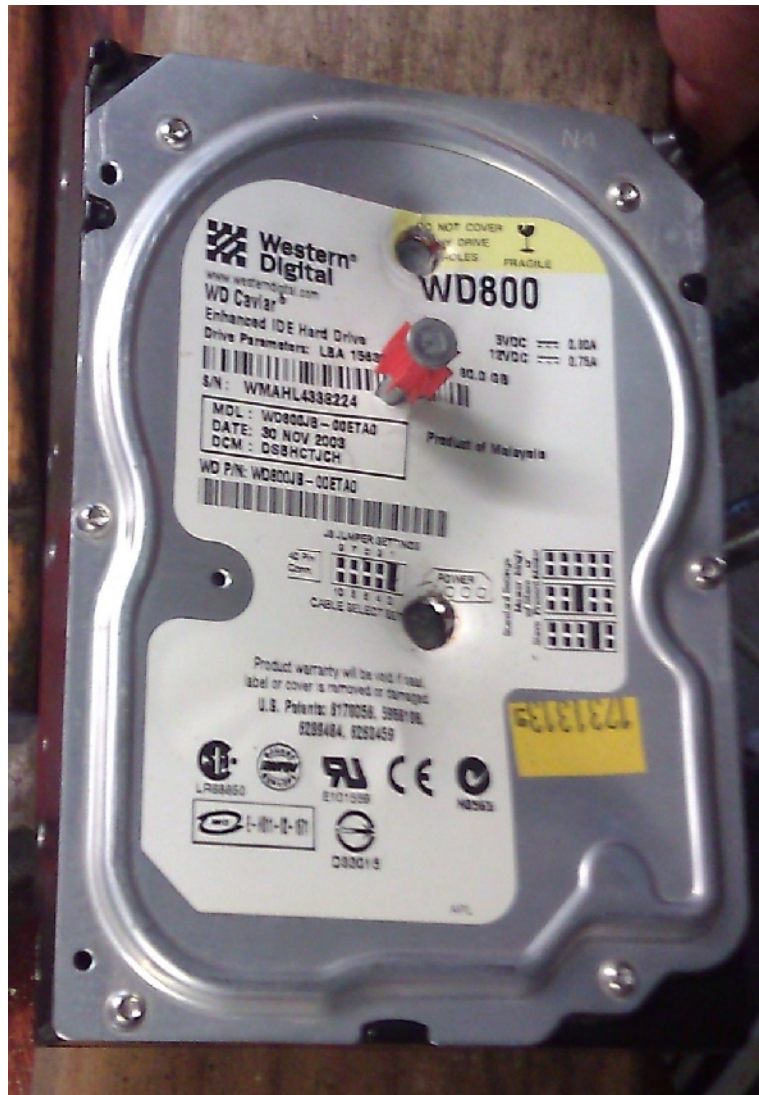


Figure 30 - Secured Disk

Security Notes and Considerations

The PACMan utility implements a number of security elements that relate to newly-developed technologies, as well as legacy technologies that have known attacks available in the public domain.

This section provides a list of the some of the design decisions, assumptions and restrictions that must be considered when utilising the PACMan utility in a production environment, to minimise the risk of a security compromise and to guide the development of operating policies.

NXP Mifare Classic

Mifare Classic (MFIC S50/S70) is a card technology developed by NXP and has been used in PACS implementations. In the last few years, numerous theoretical and practical attacks have emerged where it has been demonstrated that keys and therefore protected information can be recovered with a relatively low technical and cost barrier.

The PACMan utility uses Mifare Classic functionality to personalise cards for use with legacy PACS installations, so it must be noted that these security flaws continue to exist in cards personalized by PACMan. PACMan only implements Mifare as a transition methodology which can be switched off following transition. This security flaw does not affect the PLAID application, or PACS implementations that utilise PLAID as a means of authentication.

Key Storage device based on PLAID reference code

There is currently no off-the-shelf HSM or SAM module that supports the secure personalisation of production PLAID cards.

In order to reduce development time, and cost in the provision of personalisation functionality for the PLAID protocol, a design decision was made to use a PLAID instance as the key storage container in place of a dedicated Hardware Security Module (HSM) or Secure Access Module (SAM). This allowed rapid development of a personalisation application for PLAID, whilst maintaining secure storage of the production keys, whilst strongly securing the keys within a PLAID instance.

One implication of this however is that **once a successful authentication** has been made against the PLAID key storage device, keys are then exposed as ACSRecords (see PLAID documentation) and as a result are available in plain text form to a PLAID authenticated terminal. (In this case the personalization station PC running PACMan)

During a session the PACMan application therefore holds cryptographic keys in-memory in their plain form. This means that a person with reverse engineering skills would be able to extract the keys while the application is running.

Additionally, Microsoft Windows uses 'virtual memory', where the contents of memory are stored on the hard drive temporarily to increase amount of memory available to applications. Because this functionality is outside the control of the application, it could result in cryptographic keys being stored in non-volatile memory.

Although it is possible to hold the keys in encrypted form until they are being used, it provides very little benefit as they must still be decrypted at some point during the personalisation process and consequently still vulnerable to extraction.

Although this is not an unusual practice, this issue is on the list for potential rectification as part of the PLAID reference source project. For now the vulnerability can be minimised by ensuring cards are always personalised within a secure area on a separate and secure pc (such as using a dedicated PC within the secure area with no network access)

Key generation based on Microsoft .NET Framework CSP

PACManager utilises the Microsoft .NET Framework classes RNGCryptoServiceProvider, AesCryptoServiceProvider and RSACryptoServiceProvider for generation of keys. Microsoft has stated that these libraries are actively maintained for FIPS 140 compliance (Security requirements for Cryptographic Modules).

3-Part Key Generation

The Key Management functionality of this utility generates a keyset, which is split into three parts, using a simple key splitting protocol. With the assumption of a sufficiently random number generator, this provides an absolutely secure method of ensuring that all three parts are required before the key is recovered and is comparable to the security of two consecutive one-time pad operations in this regard.

The algorithm is as follows for each key in the keyset:

- 1) Generate key K which is to be split
- 2) Generate 2 random bit-strings X and Y using a FIPS-140-2 certified RNG , each having the same length as K
- 3) Calculate $Z = K \oplus X \oplus Y$
- 4) X, Y and Z are the three key parts which can be distributed to separate parties
- 5) To combine again, simply calculate $K = X \oplus Y \oplus Z$

Lack of authentication on Key Authentication card

The Key Authentication device is based on the PLAID SAM reference code, and so lacks a number of features available on a production HSM. One of these is the lack of an operator PIN code or other authentication mechanism to allow authentication to proceed.

This means that an assumption must be made that the possession of both the Key Authentication and Key Storage devices presumes the authority to use them.

This also means that there is no mechanism to prevent repudiation (i.e. the denial of card personalisation after the fact).

Recommendation 1: Any pilot or production use of this application must be performed in a trusted environment, by operators who would also be authorised to handle key material in its plain form.

Recommendation 2: The Key Authentication device and Key Storage device should be stored separately, to provide a lower risk of an unauthorised user possessing both

Recommendation 3: For PACMan to be used in a non or semi-trusted environment, the PLAID SAM should be updated to provide an acceptable level of security regarding authentication, non-repudiation and key injection at a minimum. N.B a proposal is in place to do these using Global Platform standards as part of the existing reference implementation, so a later version of PACMan will likely include this feature.