Digital Defence Secure Mobile FIPS 140-2 Validation Security Policy Level 1 Validation

Document Version 1.1

NON-PROPRIETARY

Contents

1 Introduction

1.1 Purpose

This non-proprietary Cryptographic Module Security Policy for Secure Mobile Version 11.1.0.0 describes how Secure Mobile Version 11.1.0.0 meets the Level 1 security requirements of FIPS 140-2. Validation testing for the module was completed on Windows Mobile 6.5 running on a Motorola MC65 mobile handset.

FIPS 140-2 (Federal Information Processing Standards Publication 140-2 Security Requirements for Cryptographic Modules) details the U.S. Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the NIST website at http://csrc.nist.gov/groups/STM/cmvp/.

1.2 References

This document deals only with operations and capabilities of the Secure Mobile Cryptographic Module Version 11.1.0.0 in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on Secure Mobile Version 11.1.0.0 from the following source:

Refer to: http://www.digital-defence.com for information on Digital Defence products and services as well as answers to technical or sales related questions.

2 Product Description

The Secure Mobile Cryptographic Module (SMCM) provides core cryptographic functionality for software applications in the Windows Embedded Handheld (WEH) environment. It supports functions XTS-AES-128 cipher mode for storage encryption, a KDF according to NIST SP 800-108 for derivation of the storage encryption key and HMAC-SHA-256 to provide integrity protection for its binaries an settings.

The cryptographic module is comprised of:

- Device authentication to authorize against credentials which are stored in an SHA-256 hash.
- Real-time encryption of persistent data using the AES-128 algorithm in XTS cipher mode whereby the Encryption Key is derived from the 128-bit Product Universal Key (PUK)

The cryptographic module is implemented as a File System Filter driver (FSFilter.*dll*) for real-time encryption and a control panel applet (*Password.cpl*) to provide key management and self-test features.

The module is a multi-chip standalone cryptographic module consisting of software that executes on the Windows Mobile computing platform.

The product meets the overall requirements according to Security Level 1 of FIPS 140-2. As a software module at Security Level 1, Physical Security requirements of section 4.5 of FIPS 140-2 do not apply. The security levels for the individual sections are as follows:

Table 1- Achieved FIPS 140-2 security levels

FIPS 140-2 Section	Security Level
1. Cryptographic Module Specification	1
2. Cryptographic Module Ports and Interfaces	1
3. Roles, Services and Authentication	1
4. Finite State Model	1
5. Physical Security	N/A
6. Operational Environment	1
7.Cryptographic Key Management	1
8. Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)	1
9. Self-Tests	1
10. Design Assurance	1
11.Mitigation of Other Attacks	1

2.1 High Level Block Diagram

Figure 1 shows the logical boundary of the cryptographic module and how it fits in with the Windows Mobile application/driver architecture model. The cryptographic module consists of two binary files, *FSFilter.dll* and *Password.cpl*. That is, the File System Filter driver for real-time encryption of persistent data and the control panel applet dynamic link library which is used as a cryptographic interface to the Secure Mobile product.

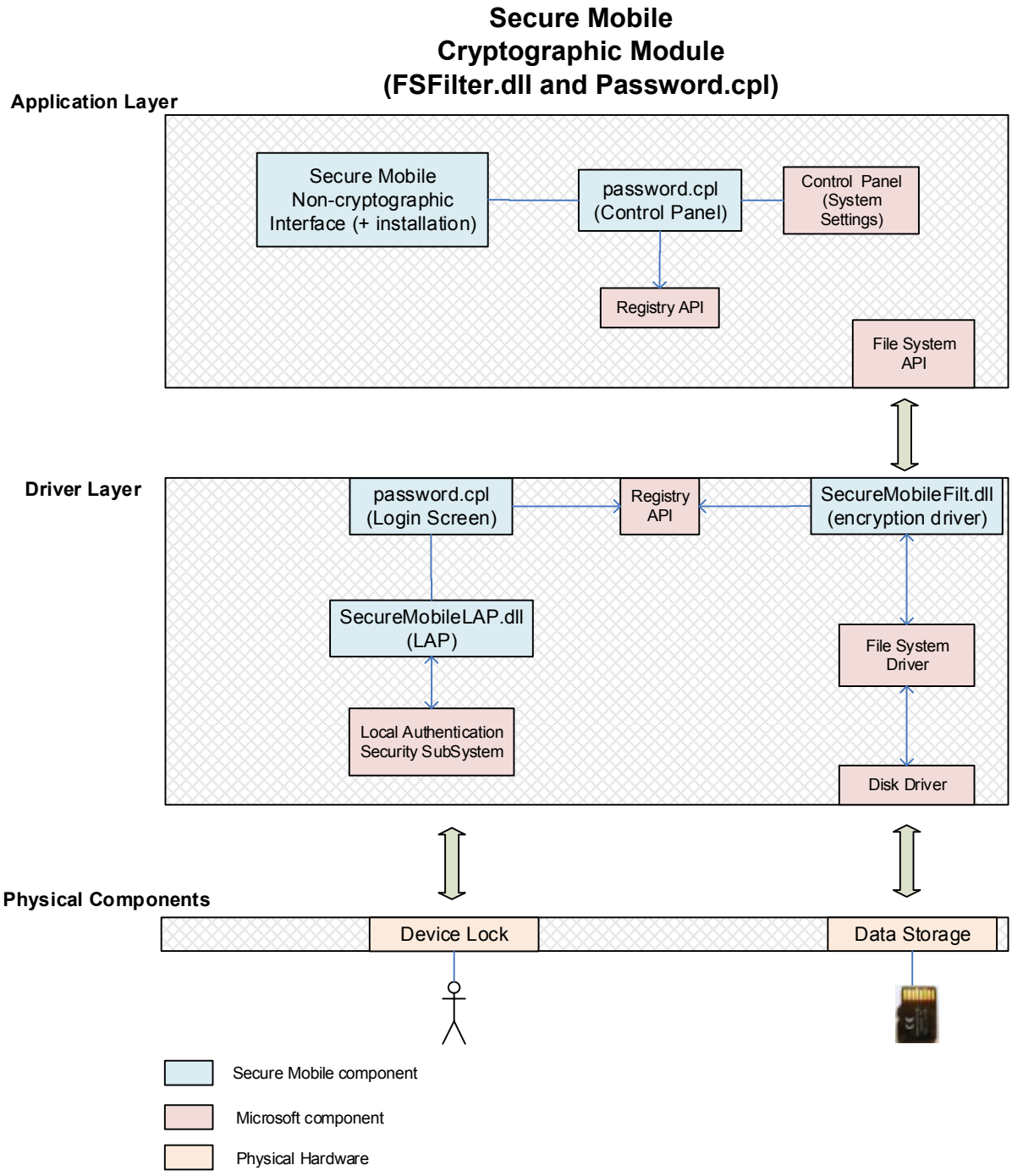

Figure 1- Secure Mobile Cryptographic Module (SMCM)

Figure 2 shows the physical boundary of the SMCM physical boundary, which includes the case of the mobile handset. The system's internal components which affect the security of the cryptographic module are considered inside the physical boundary of the cryptographic module along with the internal file data. It should be noted that the physical storage media is not protected by the SMCM however the logical port connecting the external media is included in the physical boundary so the data contained in external media is protected via this logical interface.

The cryptographic boundary of the SMCM is identical with the physical boundary of the mobile handset.

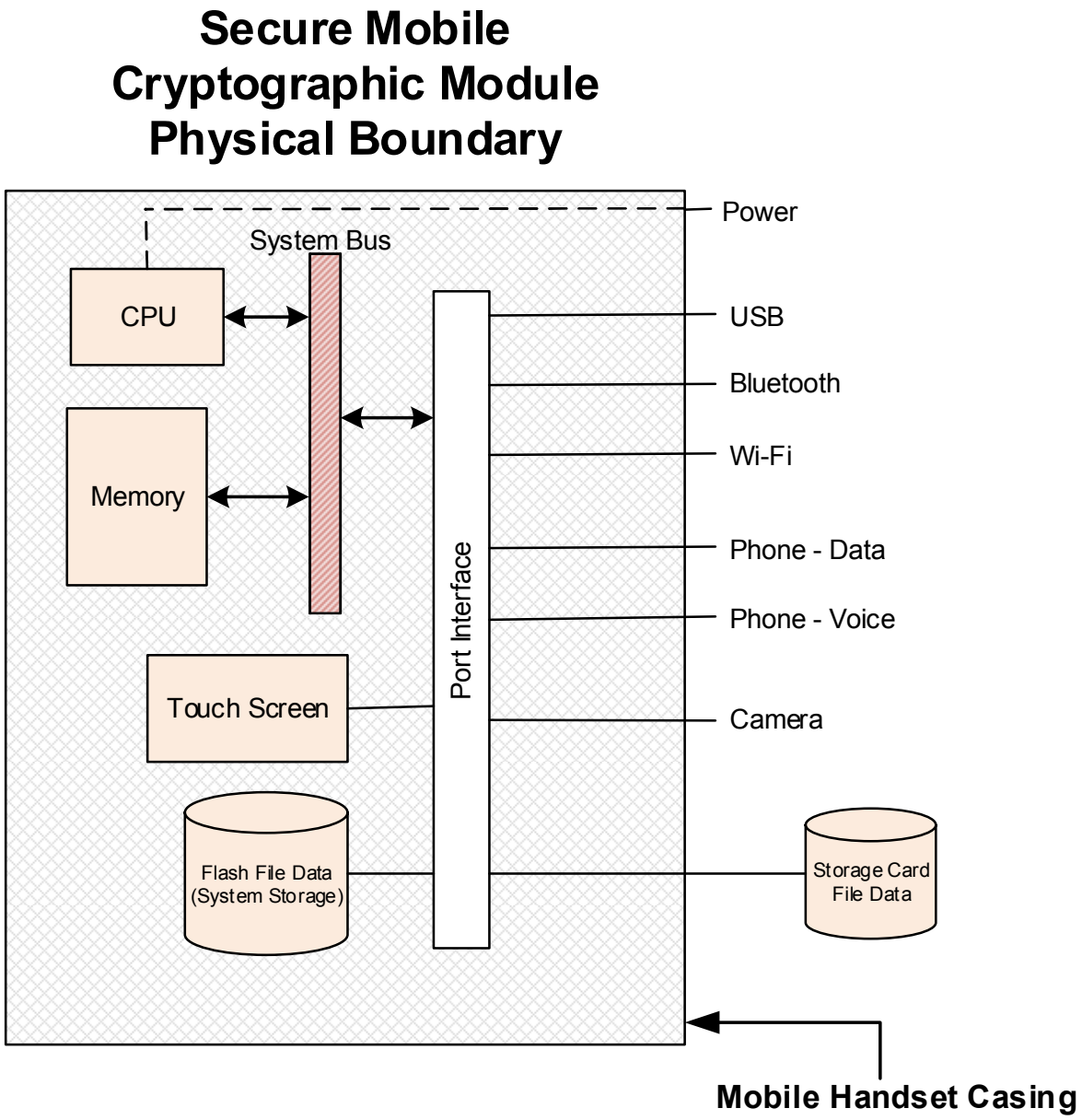

Figure 2- SMCM Physical Boundary

2.2 Finite State Machine

The SMCM uses a finite state model (FSM) to ensure the module is performing cryptographic operations only in the correct state. The FSM acts as a layer between the SMCM and the logical boundary, running various initialization and self-tests before allowing cryptographic functions to operate.

The FSM has the states shown in the following Table 2.

Table 2- SMCM Finite State Model

State	Description
POWER_ON	Initial (startup) state - self tests not yet run.
SELF_TESTS	Self-tests are running. This includes both cryptographic algorithm and binary and settings integrity tests.
READY	Self-tests have successfully passed. Ready to accept requests for cryptographic function processing. Background threads are in action to be notified when binary files are altered to determine if any self-tests should be performed to determine a binary error (resulting in transition to ERROR_BINARIES state).
ERROR_BINARIES	Self-test failed one or more binary file integrity checks.
ERROR_SETTINGS	Self-test failed the stored settings integrity check. The Secure Mobile product will need to be re-installed.
UNRECOVERABLE_BINARIES	The binary data cannot be recovered via embedded file backup data. The Secure Mobile product will need to be re-installed.
RECOVERY_BINARIES	One or more SMCM binary file(s) are tried to be recovered, via embedded file backup data.
POWER_OFF	SMCM has been installed but is not running.

2.3 Approved mode of operation

The SMCM has got only one mode of operation, which is the Approved mode of operation as defined by FIPS PUB 140-2.

3 Module Ports and Interfaces

Table 3 maps the Logical Interfaces to the software interfaces and the equivalent physical ports.

The Data Input and Output operations perform the work of the cryptographic module, such as:

- Cryptographic key derivation, storage, and retrieval.
- Data cipher operations. Cipher text to plain text and vice versa.
- Authentication storage and authorization.

Table 3- SMCM Module Interface

Logical Interface	API Interface	Physical Port
Data Input	Input of Authentication, Encryption, and Cryptographic Key operations.	Standard Input Ports (e.g. USB, Touch Screen, Memory Card).
Data Output	Output of Authentication, Encryption, and Cryptographic Key operations.	Standard Output Ports (e.g. USB, Touch Screen, Memory Card).
Control Input	Requests to initialize, run self-tests, and power down the module.	N/A
Status Output	Output of requests to retrieve the status of the module.	Standard Output Ports (e.g. Touch Screen).
Power	N/A	Supplied by the device.

4 Roles, Services, and Authentication

The Secure Mobile Cryptographic Module supports a Crypto Officer (CO) Role and a User Role. The CO is in charge of installation and uninstallation of Secure Mobile, and therefore performing key management (key derivation during installation, zeroization during uninstallation). The User is in charge of actually using Secure Mobile. Doing so, the User will use the key established by the CO, but the user is not performing key management.

The module is only certified to Security Level 1, therefore it does not require identification and authentication of the operator to assume certain roles (Table 5).

The types of services are shown in Table 4 with their corresponding required access (Role) and the cryptographic module's associated operations.

Table 4- SMCM Services

Service	Role	SMCM Operations	Input/Output
Show Status	User	ShowStatus	O: status of the module
Perform Self-Tests	User	RunSelfTests	O: success of the test
Initialize	User	Initialize	-
Key Derivation (KDF using HMAC-SHA-256); this is a one-time process during installation of SMCM	CO	SetupEncKeyFromPUK SetupConnKeyFromPUK SetupCRStringFromPUK IsConnKeyEqual CreateFileHandle	I: PUK (128 bits) O: success (internally the derived DWEK, a XTS-AES-128 key, is set) I: PUK (128 bits) O: success (internally the derived connection password PCCP, 128 bits, is set) I: PUK (128 bits) O: success (internally the derived challenge/response string CRString, 80 bits,, is set) I: Connection Key (128 bits) O: success or failure I: file details (name, access etc.) O: FileHandle (pointer to an obscured structure to be used for cipher operations)
XTS-AES-128 storage encryption/ decryption	User	ReadEncryptedData (decrypt) WriteEncryptedData (encrypt)	I: FileHandle, Data, Size, Offset O: data, which have been read from file and decrypted I: FileHandle, Data, Size, Offset O: : success or failure
Zeroization	CO	Uninstallation	-

Table 5- SMCM Identification and Authentication

Role	Identification	Authentication
User	None	None
Crypto Officer	None	None

It should be noted that the User and Crypto Officer roles are implicitly selected based on the service being invoked.

5 Physical Security

The Secure Mobile Cryptographic Module is implemented entirely as software, therefore the physical security requirements of FIPS 140-2 are not applicable.

6 Cryptographic Key Management

The Secure Mobile Cryptographic Module derives the so-called Deployment Wide Encryption Key (DWEK), the PC Connection Password (PCCP), and the Challenge Response String (CRString) from the Product Universal Key (PUK) provided during installation of SMCM. Key Derivation is performed using the Key Derivation Function (KDF) in Counter Mode from *NIST Special Publication 800-108: Recommendation for Key Derivation Using Pseudorandom Functions*.

The SMCM provides file encryption using the XTS cipher mode of the AES algorithm; the DWEK is the corresponding XTS-AES-128 key and a 128 bit value is randomly generated (using a FIPS 140-2 approved RNG provided by Windows Mobile platform) at file creation time as the start value for the XTS tweak value i.

The DWEK, the PCCP, and CRString values are stored in the system registry, encrypted using AES-128 encryption with a hard coded key value (see section "Mitigation of other attacks"). The encrypted values of DWEK, PCCP, and CRString are zeroized when the Secure Mobile Cryptographic Module is uninstalled.

Table 6- SMCM Cryptographic Key Management

Key	Strength	Purpose	Establishment	Storage	Zeroization
Product Universal Key (PUK)	128 bits	Deployment Wide key containing 128 bits of entropy, to derive DWEK, PCCP and CRString values.	Provided by the installer program when SMCM is installed.	Not stored (only used for one-time key derivation during installation of SMCM)	N/A
Deployment Wide Encryption Key (DWEK)	128 bits (XTS-AES-128 key)	Encryption Key for all files for 128 bit AES-XTS encryption.	Derived from PUK using KDF from NIST SP 800-108 in Counter Mode.	Device Registry Encrypted with hard-coded key using AES-128 in ECB mode, see Mitigation of other Attacks	DWEK (its encrypted value) is zeroized during uninstallation of SMCM.
PC Connection Password (PCCP) and **Challenge/Response String (CRString)**	128 bits 80 bits	Authentication data used for establishing a connection between SMCM and a PC (to make sure that both, SMCM and Secure Mobile PC software were installed using the same PUK value).	Derived from PUK using KDF from NIST SP 800-108 in Counter Mode.	Device Registry Encrypted with hard-coded key using AES-128 in ECB mode, see Mitigation of other Attacks	PCCP and CRString (their encrypted values) are zeroized during uninstallation of SMCM.
File Encryption Tweak value (FETV)	128 bits	Base value to determine AES-XTS tweak value i.	Generated by RNG as provided by Windows Mobile	Header of encrypted file, as plaintext	Zeroized upon file deletion.

Approved cryptographic functions implemented in SMCM:

- Key derivation according to NIST SP 800-108 (KDF cert. #26)
- Encryption/decryption using AES-128 in ECB and XTS modes (AES certs. #2851 and #2852)
- Secure hashing using SHA-256 (SHA certs. #2394 and #2395)
- Keyed hashing HMAC-SHA-256 (HMAC certs. #1792 and #1793)

Vendor-affirmed cryptographic functions implemented in SMCM:

None.

Non-approved cryptographic functions implemented in SMCM:

None.

Cryptographic functions implemented in the platform of SMCM:

- RNG according to FIPS 186-2 (RNG cert. #286) as part of "Windows CE and Windows Mobile Enhanced Cryptographic Provider (RSAENH)" which is a FIPS 140-2 certified cryptographic software module contained in Windows Mobile (CMVP cert. #560).

7 Self-Tests

The Secure Mobile Cryptography Module performs a series of self-tests to ensure the integrity of the module's software components as well as the cryptographic functions it implements (KAT denotes to known answer test).

Power-up self-tests:

- AES-128 block cipher primitive encryption KAT
- AES-128 block cipher primitive decryption KAT
- AES-128 ECB multi-block encryption KAT
- AES-128 ECB multi-block decryption KAT
- AES-128 XTS encryption KAT
- AES-128 XTS decryption KAT
- SHA-256 KAT
- HMAC-SHA-256 KAT
- Software/firmware integrity test (HMAC-SHA-256 verification for the binaries and settings of SMCM)

Conditional self-tests:

- Software/firmware integrity test (HMAC-SHA-256 verification for the settings of SMCM each time the settings are read from the registry)

Power-up test can be executed on demand by power-cycling the mobile handset or restarting Windows Mobile SMCM is installed on (then a regular power-up is performed and corresponding self-tests are automatically executed.

If any of the self-test fails, Secure Mobile informs the operator via a modal dialog on the display of the handset about the integrity error. Detail information about the error (e.g., which particular self-test failed in which of the binaries) can be retrieved from the log file "SMCMLog.txt" written by Secure Mobile. This log file also contains explicit information about self-tests that passed successfully.

The log file, SMCMLog.txt, is updated in the \Program Files\SecureMobile\ folder of the mobile handset. Hence, the FIPS 140-2 log file can be viewed by reading the file \Program Files\SecureMobile\SMCMLog.txt.

As soon as any aspect of the Secure Mobile FIPS 140-2 Cryptographic Module becomes corrupt (binary file or storage of cryptographic information), the following message is displayed to the user.

Figure 3 - Screenshot of corruption in Secure Mobile FIPS 140-2 Cryptographic Module

At this point Secure Mobile is no longer operational and encrypted files are not accessible. Upon restart of the device, the following screen is displayed to the user.

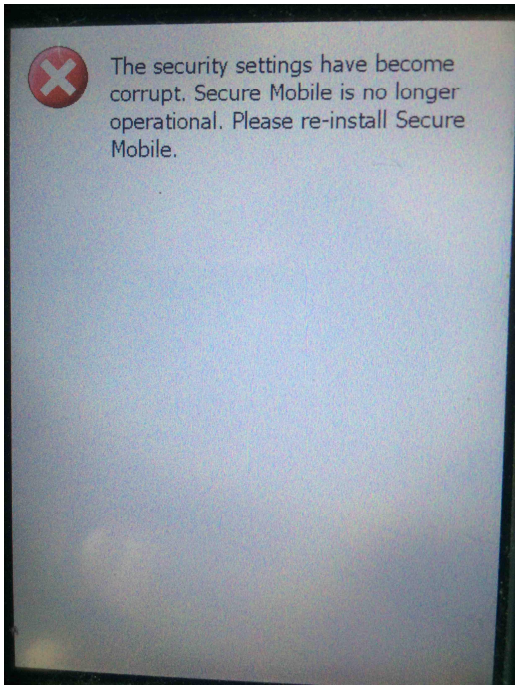

Figure 4 - Screenshot after device restart from a corruption in FIPS 140-2 SMCM

8 Design Assurance

The implementation of the Secure Mobile Cryptographic Module is controlled using Concurrent Versions System (CVS). This is a version control system which allows multiple developers to implement components of the overall software module concurrently and can precisely re-generate source code for each version that has been tagged. All released versions of Secure Mobile are tagged using CVS so any released version can be re-built at any time.

The Secure Mobile Cryptographic Module is installed onto the target device when the Secure Mobile product is installed. Installation can be performed either by:

- Using a PC installation which installs Secure Mobile onto a connected handheld device. The PUK value is entered into the installer program during this process.
  Creating a Mass Deployment Kit which is a .cab file for direct handheld installation. A Mass Deployment Kit contains a pre-packed PUK value inside the .cab file.

9 Mitigation of Other Attacks

The Secure Mobile Cryptographic Module implements the following measure to mitigate attacks other than those already addressed by functionality required by FIPS PUB 140-2 Security Level 1:

Other Attack	Mitigation Mechanism	Specific Limitations
Disclosure of DWEK, PCCP, or CRString values from their storage in the registry.	Values of DWEK, PCCP, or CRString are stored in the registry encrypted by an AES-128 ECB mode encryption; the corresponding key is hard-coded in SMCM.	The hard-coded encryption key cannot be zeroized (nevertheless, the encrypted values of DWEK, PCCP, and CRString can be zeroized by uninstallation of the Secure Mobile Cryptographic Module). By reengineering of the software of the Secure Mobile Cryptographic Module and attacker could disclose the used encryption key.