PalmSource
Cryptographic Provider Manager and FIPS Provider (CPM+F)

FIPS 140-2 Non-Proprietary Security Policy
Version 1.8

July 2005

Version	Description
1.0	Initial
1.1	Added more detail to section 8.0
1.2	Cleaned up modifications from TechPubs
1.3	Added Cryptographic boundary diagram
1.4	More modifications from TechPubs
1.5	Removed unnecessary sections
1.6	Added HMAC-SHA1 certificate number
1.7	Updated version to 5.6.0
1.8	Updated due to comments

Table of Contents

1.0 Introduction

1.1 Purpose

This document specifies the security policy for the Cryptographic Provider Manager and FIPS Provider (CPM+F) as described in FIPS PUB 140-2. This security policy describes how the CPM+F meets the security requirements of FIPS 140-2 and how to place the module in a secure mode of operation.

1.2 Identification

Name: Cryptographic Provider Manager and FIPS Provider (CPM+F)
Software Version: 5.6.0

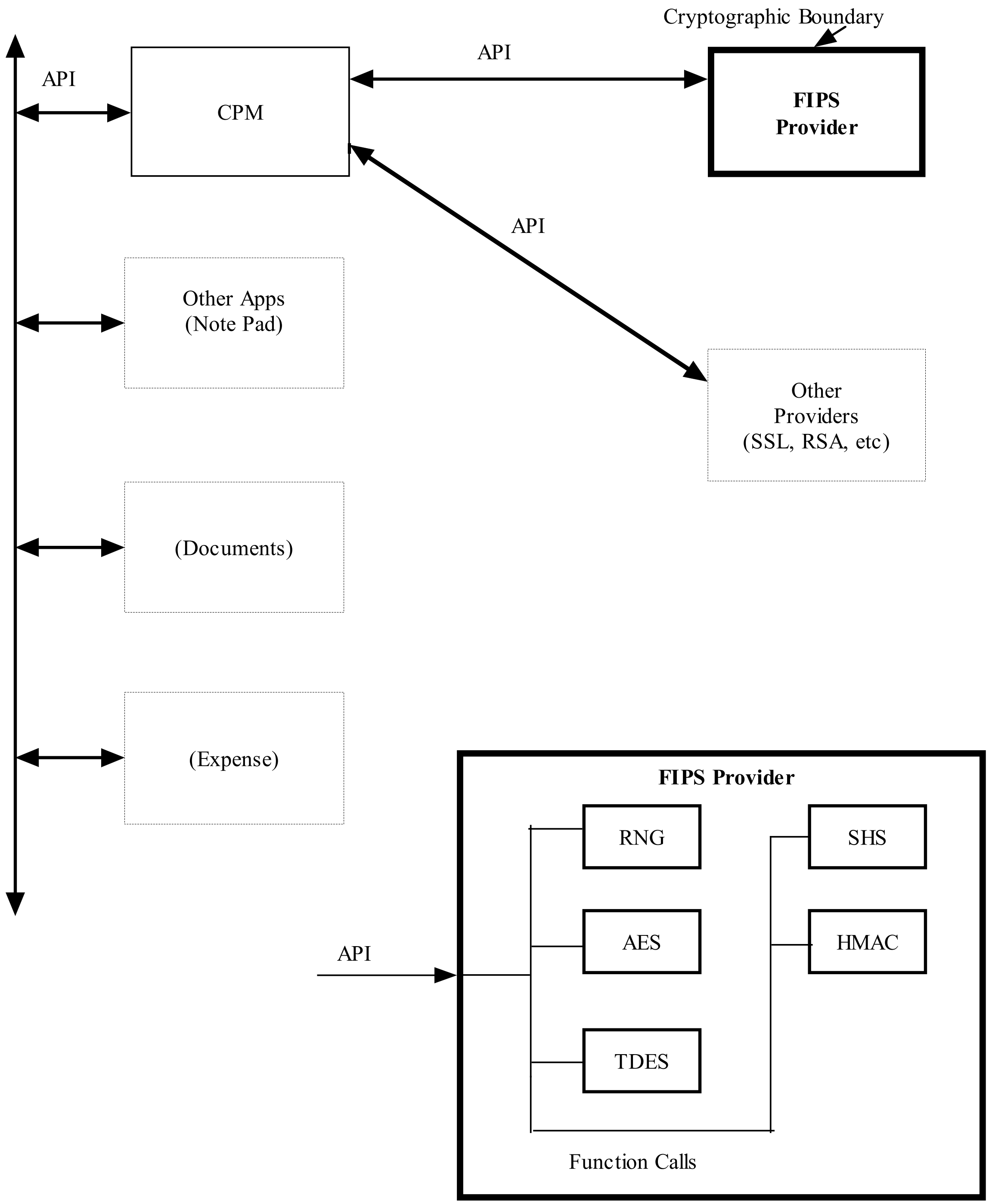
1.3 Design

The FIPS Provider for the Cryptographic Provider Manager (CPM) is a FIPS 140-2 Level 1 compliant, software-based, cryptographic module. The FIPS Provider encapsulates several different cryptographic algorithms accessible via the CPM API.

1.3.1 Cryptographic Boundary

The FIPS Provider consists of a single shared library named `FIPSProvider.prc` (software version 5.6.0), which comprises the module's logical boundary. The cryptographic boundary for the FIPS Provider is defined as the enclosure of the computer system on which the cryptographic module is to be executed. The physical configuration of the module, as defined in FIPS PUB 140-2, is Multi-Chip Standalone.

The following block diagram depicts the FIPS Provider cryptographic boundary:



PalmSource
CPM+F Block Diagram

1.3.2 Cryptographic Provider Manager

The Cryptographic Provider Manager (CPM) provides cryptographic functionality for applications. The CPM is designed in such a way as to allow applications to make use of cryptographic functionality without involving the complexity of cryptography. A good example is a diary application that wants to protect a diary entry with a password.

The CPM allows the application to perform some password-based encryption without specifying the details of the encryption itself. The CPM does not provide any cryptographic functionality by itself. The CPM provides a high-level interface for cryptography that is in turn handled by providers. The CPM loads various providers in the system. The providers implement the actual cryptographic functionality. Providers are free to allow arbitrary or unspecified parameters or require more stringent adherence to cryptographic inputs. The CPM provides the only interface to providers; there is no direct interface to any provider.

The CPM loads the providers found on the system in an arbitrary order. The list of providers, therefore, is not ordered in any particular way. An application may order the list in a particular way by calling a series of functions to enumerate the current list of providers and set providers to the default. A default provider is the "first" provider in the list of providers.

As cryptographic calls are made by the application, the CPM routes the calls to the list of providers sequentially, one after another, until one provider reports that it handled the call. For initial cryptographic contexts, the CPM evaluates whether the provider supports the basic operation (Encrypt, Message Digest, etc.) before allowing the provider a chance to handle the call. If the provider does handle the call, the rest of the provider list is skipped. If the provider does not handle the call, the next provider in the list is allowed a chance to handle the call. All subsequent calls into the CPM within the same cryptographic context are routed to the provider that handled the initial context. There is no direct interface into the cryptographic boundary of the FIPS provider. The only interface is through the CPM.

1.3.3 Approved Mode of Operation

The FIPS provider itself only has one mode of operation and it is the approved mode of operation. Using the FIPS Provider is an implicit approved mode of operation.

The CPM interface allows multiple providers for cryptographic algorithms. The CPM maintains a list of providers and iterates through the list to determine which provider will satisfy the cryptographic operation. Once a provider satisfies an initial cryptographic operation within a particular cryptographic context (e.g. Init), that provider is the only provider that handles subsequent cryptographic operations within that same cryptographic context. To ensure that the FIPS

Provider is the first handler in the list and therefore gets the first chance to handle all the cryptographic operations, an application must call:

```
CPMLibSetDefaultProvider((UInt32)'fips');
```

To verify that the FIPS Provider is the first provider in the list, and therefore the provider that will handle all operations first, enumerate the providers and check that the first provider in the enumeration is the provider with id of `'fips'`.
The following code shows how to do this:

```
UInt16 numProviders = 0;
UInt32 pIDs[];

CPMLibEnumerateProviders(NULL, &numProviders);
/* now numProviders contains the number of providers */
pIDs = MemPtrNew(sizeof(UInt32) * numProviders);
CPMLibEnumerateProviders(pIDs, &numProviders);
if (pIDs[0] == ((UInt32)'fips')) {
  /* FIPS provider is first */
}
```

Once the FIPS provider is set as the default and determined to be the first provider in the list of providers, cryptographic operations can begin. To ensure that the approved mode of operation is maintained, only the approved cryptographic algorithms are allowed. These approved cryptographic algorithms include and are limited to: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, AES, 3DES and HMAC-SHA-1.

CPM cryptographic functions that use one of the approved algorithms are within the approved mode of operation.

2.0 Interfaces

Interfaces to the CPM and FIPS Provider take the form of APIs.

3.0 Roles and Services

3.1 Operator Roles

The only roles supported by the CPM and FIPS Provider are the User Role and the Crypto Officer Role. The User Role allows cryptographic operations to be performed using the FIPS Provider. The Crypto Officer Role can perform all of the User Role operations and can manage cryptographic parameters and the cryptographic module itself (including self tests and reload of the cryptographic module). Operators assume roles implicitly by the services they access via the cryptographic module. The CPM and FIPS Provider does not support a maintenance role.

The services available to operators are shown in the table below:

Service Class	Role
Key Generation	User, Crypto Officer
Message Digest	User, Crypto Officer
Encryption/Decryption	User, Crypto Officer
Message Authentication	User, Crypto Officer
Module Load/Unload	Crypto Officer
Module Configuration	Crypto Officer
Provider Management	Crypto Officer
Import/Export Cryptographic Parameters	Crypto Officer
Self-Test	Crypto Officer

3.2 Access Control Policy

Access to services is controlled as shown in the table below.

Service Class	CSP	Type of Access
Key Generation	3DES, AES, SHS	Read/Write
Message Digest	SHS	Read/Write
Encryption/Decryption	3DES, AES	Read
Message Authentication	HMAC	Read
Module Load/Unload	N/A	N/A
Module Configuration	N/A	N/A
Provider Management	N/A	N/A
Import/Export Cryptographic Parameters	3DES, AES, SHS, HMAC	Read/Write
Self-Test	N/A	N/A

4.0 Physical Security

The CPM and FIPS Provider is a software module. Physical security is the responsibility of the operator of the device upon which the module is loaded. The FIPS 140-2 physical security requirements do not apply to this module.

5.0 Operational Environment

The operating environment for the CPM and FIPS Provider is Palm OS version 5.2.1 running on a Palm Tungsten™ C platform.”.

6.0 Cryptographic Key Management

The CPM and FIPS Provider support the following FIPS Approved algorithms:

Algorithm	Modes	Certificate Number
3DES	ECB, CBC, CFB, OFB,	226
SHS	SHA-1	202
SHS	224, 256, 384, 512	303
AES	ECB, CBC, CFB	114
HMAC-SHA-1	FIPS 198	46
PRNG	X9.31 A.2.4	63

Key Storage
The CPM and FIPS Provider does not store any keys in the module. Keys are generated for each session and then destroyed.

Key Zeroization
There are no restrictions on when plaintext secret and private cryptographic keys and CSP's can be zeroized. Any time any of the `Release` functions are called, the associated cryptographic keys and CSP's are zeroized.

7.0 Self-Tests

The module performs self-tests upon power-up.

1. 3DES Known Answer Test
2. SHA-1 Known Answer Test
3. SHA-2 Known Answer Tests
4. AES Known Answer Tests
5. HMAC-SHA-1 Known Answer Test
6. RNG Known Answer Test

The module performs a continuous random number and a software integrity conditional test.

8.0 Secure Operation

Due to the sensitive nature of the cryptographic module, care must be taken to ensure the secure initialization and start-up of the module.

- **Initialization**
  To initialize the CPM+F library, call the standard Palm OS API for loading a shared library (`SysLibLoad`). This returns a reference number to be used in all subsequent calls into the CPM+F.

- **Start-up**
  To start using the cryptographic module, a call to `CPMLibOpen` is required. This initializes the list of providers.

  After calling `CPMLibOpen` the FIPS provider must be set as the default provider by calling `CPMLibSetDefaultProvider` and specifying the FIPS provider (as `'fips'`). If the call to `CPMLibSetDefaultProvider` returns `cpmErrProviderNotFound`, start-up failed and no further operations should be attempted.

- **Shutdown**
  All successful calls to `CPMLibOpen` require a call to `CPMLibClose` to finalize the cryptographic module library and allow cryptographic contexts to zeroized and freed.

9.0 Mitigation Against Other Attacks

The module does not mitigate against any other attacks.

Acronyms

API	Application Programmer Interface
CPM	Cryptographic Provider Manager
CSP	Critical Security Parameter