MiniCrypt

Module Version: 1.2
Document Version: 1.8

5 July 2011

Teledyne Webb Research
82 Technology Park Drive
East Falmouth, MA 02536
www.webbresearch.com

Non-Proprietary Security Policy
FIPS 140-2 Level 1 Validation

Table of Contents

1 Introduction....3
1.1 Module Description....3
1.2 Boundary....3
2 Security Level....4
3 Modes of Operation....4
4 Ports and Interfaces....5
5 Access Control Policy....6
5.1 Roles and Services....6
5.1.1 User Role....6
5.1.2 Crypto Officer Role....6
5.2 Definition of Critical Security Parameters (CSPs)....7
5.2.1 Definition of CSPs Access Modes....7
6 Identification and Authentication Policy....7
6.1 Key Generation....7
6.2 Key Storage....8
6.3 Key Protection/Zeroization....8
7 Operational Environment....8
7.1 Software Environment....8
7.2 Hardware Platform....8
8 Self Tests....8
9 Physical Security....8
10 Mitigation of Other Attacks Policy....9
11 Operator Guidance....9
12 Acronyms and Definitions....9
13 References....10

1 Introduction

This document specifies the Security Policy for the Teledyne Webb Research MiniCrypt module (MiniCrypt). This Security Policy was produced as part of the Federal Information Processing Standard (FIPS) 140-2 Level 1 validation of the MiniCrypt library, version 1.2. MiniCrypt is a small, low resource utilization library for use in embedded systems. It is intended to provide a secure cryptographic infrastructure for a group of remote data acquisition products offered by Teledyne Webb Research.
MiniCrypt provides the following FIPS 140-2 validated cryptographic services:

- Encrypt and decrypt application data using Advanced Encryption Standard (AES)
- Ensure message authentication and integrity using Keyed-Hash Message Authentication Code (HMAC)-Secure Hash Algorithm (SHA)-256

1.1 Module Description

MiniCrypt is a standalone Application Programming Interface (API), distributed as a C object library (minicrypt.lib). The module, together with its associated application file, are programmed into, and executed directly from, flash memory on the target system hardware (Persistor CF1 HW system). MiniCrypt contains only FIPS approved cryptographic algorithms listed in Table 2. Host application developers may call the APIs defined in the minicrypt.h header file to use the approved cryptographic algorithms.

1.2 Boundary

The physical boundary for the Module (shown in Figure 1) is defined as the processor module on which the functions of the Module execute. The cryptographic boundary (shown in Figure 2) contains only the MiniCrypt software module, minicrypt.lib.

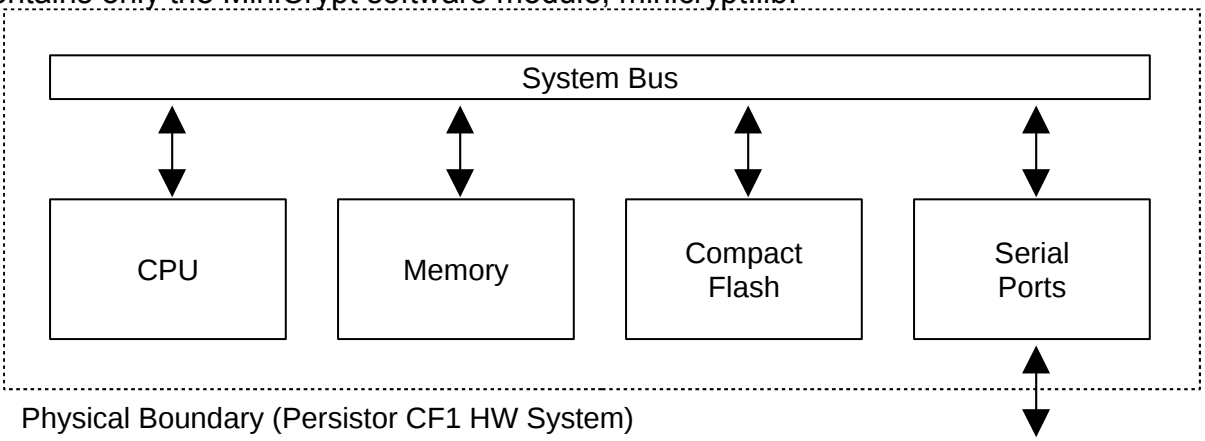

Figure 1 – Hardware Diagram Showing Controller Containing Cryptographic Module

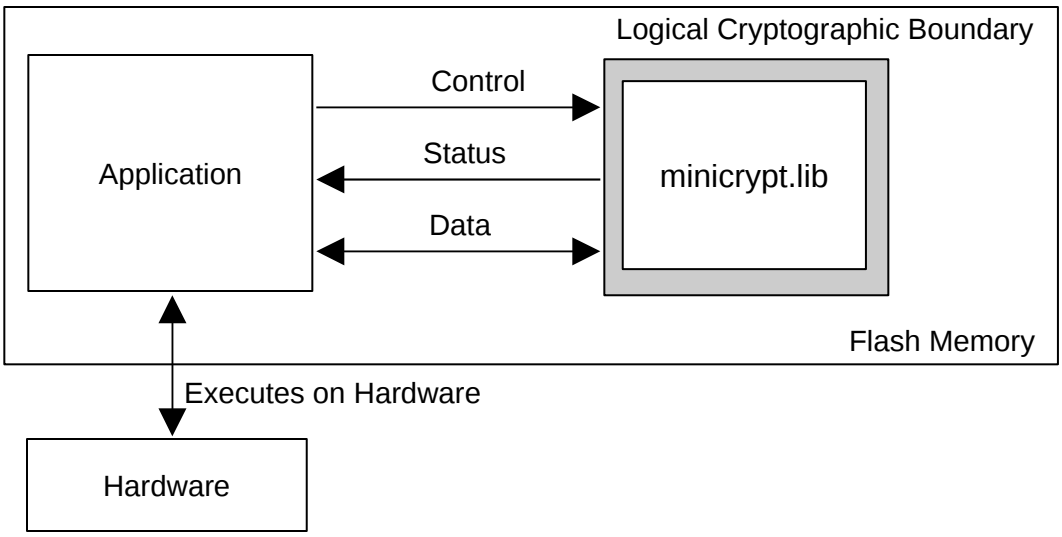

Figure 2 – Software Diagram Showing Logical Cryptographic Boundary

2 Security Level

MiniCrypt meets the overall requirements applicable to Level 1 security of FIPS 140-2.

Security Requirements Section	Level
Cryptographic Module Specification	1
Module Ports and Interfaces	1
Roles, Services and Authentication	1
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A

Table 1 – Module Security Level Specifications

3 Modes of Operation

MiniCrypt supports two modes of operation:

- Non-approved mode. The module is in this mode by default at startup. In this mode, the module can only determine the current mode or enter Approved mode; no

cryptographic operations are allowed.

- Approved mode, required to perform any cryptographic operations.

The following FIPS approved algorithms are supported in Approved mode:

Service Type	Algorithm	API Functions
Symmetric Cipher	AES Certificate # 1268 (ECB (e/d; 128, 192, 256); CBC (e/d; 128, 192, 256))	AES_cbc_encrypt AES_ecb_encrypt AES_final AES_set_decrypt_key AES_set_encrypt_key
Message Digest	SHS Certificate # 1168 (BYTE-only; SHA-256)	SHA256 SHA256_final SHA256_init SHA256_update SHA256_vector
Message Authentication	HMAC Certificate # 738 (HMAC-SHA-256)	HMAC HMAC_vector

Table 2 – Approved Algorithms

Calling any cryptographic function for the first time executes the power on self-test routines and will put the module into Approved mode. The API function MC_set_mode(MC_MODE_FIPS) can be called to explicitly put the module into Approved mode; this operation executes the power on self-test routines. No cryptographic functionality is available to the user until the power on self-test has been completed. Upon passing the self-test, all authorized functions are available.
An application may determine the current operating mode of the module by calling the MC_get_mode function. A return value of MC_MODE_FIPS indicates that the module is in FIPS mode.
Calling the API function MC_set_mode(MC_MODE_INACTIVE) will put the module into non-approved mode, and disable further access to cryptographic functions.
The module does not implement any non-approved cryptographic algorithms.

4 Ports and Interfaces

All FIPS ports and interfaces are defined as the API of the cryptographic module. Control Input to MiniCrypt is through the API function calls. Data Input and Output are provided in the variables passed with the API calls, and Status Output is provided through the returns and error codes that are documented for each call.

FIPS Interface	Physical Interface	Logical Interface
Data Input	Serial ports	API input parameters
Data Output	Serial ports	API output parameters
Control Input	Serial ports	API function calls
Status Output	Serial ports	API return codes

Table 3 – Ports and Interfaces

5 Access Control Policy

5.1 Roles and Services

As allowed by FIPS 140-2 Level 1, MiniCrypt does not support user identification or authentication for user roles. Only one role can be active at a time and MiniCrypt does not allow concurrent operators. The following table describes the services accessible by the two roles.

Role	Services
User and Crypto Officer	AES Encrypt / Decrypt SHA-256 HMAC-SHA-256 Get Mode Set Mode Get Role Set Role Get Status
Crypto Officer	Run self-tests

Table 4 – MiniCrypt Roles and Services

5.1.1 User Role

An operator assuming the User Role can access the AES Encrypt / Decrypt, SHA-256, HMAC-SHA-256, Get Mode, Set Mode, Get Role, Set Role, and Get Status services.

5.1.2 Crypto Officer Role

An operator assuming the Crypto Officer Role can access the AES Encrypt/Decrypt, SHA-256, HMAC-SHA-256, Get Mode, Set Mode, Get Role, Set Role, Get Status, and Run Self-tests services.

5.2 Definition of Critical Security Parameters (CSPs)

The Critical Security Parameters (CSPs) defined for MiniCrypt consist of cryptographic keys. The module does not persistently store any CSPs within the logical cryptographic boundary. The following CSPs are supported by the module:

- AES Keys: 128, 192 and 256 bit keys used to AES encrypt/decrypt data.
- HMAC Keys: For use during HMAC operations.

5.2.1 Definition of CSPs Access Modes

Table 4 defines the relationship between access to CSPs and the different module services. The modes of access shown in the table are defined as follows: Read, Write, or Execute.

Services	**Keys / CSPs**	**Authorized Roles**	**Type of Access**
AES Encrypt / Decrypt	AES Key	User / Crypto Officer	Read
HMAC-SHA-256	HMAC Key	User / Crypto Officer	Read
SHA-256	N/A	User / Crypto Officer	N/A
Get Mode	N/A	User / Crypto Officer	N/A
Set Mode	N/A	User / Crypto Officer	N/A
Get Role	N/A	User / Crypto Officer	N/A
Set Role	N/A	User / Crypto Officer	N/A
Get Status	N/A	User / Crypto Officer	N/A
Run Self-tests	N/A	Crypto Officer	N/A

Table 5 – Key and CSP Access Rights within Services

6 Identification and Authentication Policy

MiniCrypt meets all FIPS 140-2 Level 1 requirements for roles and services, implementing both a User role and a Crypto Officer role.

Role	**Type of Authentication**	**Authentication Data**
User	N/A	N/A
Crypto Officer	N/A	N/A

Table 6 – Roles and Required Identification and Authentication

Cryptographic key management is concerned with generating and storing keys, managing access to keys, protecting keys during use, and zeroizing keys when they are no longer required.

6.1 Key Generation

MiniCrypt does not currently support key generation.

6.2 Key Storage

MiniCrypt does not provide long-term cryptographic key storage. Keys are provided through a defined API and stored in volatile (short term) memory in plain text.

6.3 Key Protection/Zeroization

MiniCrypt accepts key data input from calling applications for its cipher and HMAC operations. Key data is supplied as a pointer to a caller-managed data buffer. The calling application maintains responsibility for managing the key buffer memory throughout the operation. MiniCrypt uses context data structures to store cipher and digest state information across multiple API calls. All context data is zeroized when the context structures are destroyed at the conclusion of the cipher or digest operations.
The HMAC key used by the module integrity check, and the AES, SHA-256, and HMAC Known Answer Test (KAT) keys are compiled into the cryptographic module binary. If these keys are modified, the module will either fail the integrity check, or will fail the KATs, and transition to the Module Invalid state.

7 Operational Environment

7.1 Software Environment

MiniCrypt executes directly from flash memory on the target system. It calls the memset and memcpy functions from the C standard library. It does not call or depend on any operating system calls for any purpose. Testing was performed running under PicoDOS version 2.26.

7.2 Hardware Platform

For FIPS 140-2 testing, the library was installed and tested on a Persistor CF1 HW system with: a 68338 microprocessor operating at 14.77 MHz, 1 M Bytes flash program memory, 1 M Bytes RAM data memory, 1 G Bytes compact flash file storage, and a single serial port for I/O.

8 Self Tests

The cryptographic module shall perform the following power up self-tests:

A. Known Answer tests:
   a. AES KAT
   b. HMAC-SHA-256 KAT
   c. SHA-256 KAT

B. Software Integrity Test (HMAC-SHA-256)

9 Physical Security

The FIPS 140-2 Physical Security requirements are not applicable because the device is a software only module.

10 Mitigation of Other Attacks Policy

The module has not been designed to mitigate any specific attacks outside the scope of FIPS 140-2 requirements.

11 Operator Guidance

In order to ensure correct operation, the application developer and/or Crypto Officer is responsible for integrating the MiniCrypt library into the target application. The following steps must be taken:

- Verify the authenticity of the minicrypt.lib file, using an external SHA256 application. The correct message digest for MiniCrypt 1.2 is: 6118d24ee6ff68b2f8ef8aa9d3f991ccf529ee93e657ec39c4e3f2f43e911e0a
- Program the module into the target system flash memory at the correct address.
- Call API functions per the MiniCrypt Users Guide.

Detailed instructions for these steps can be found in section 2 of the MiniCrypt Users Guide.

12 Acronyms and Definitions

The following table lists and describes the acronyms and definitions used throughout this FIPS submission documentation.

Term	Definition
AES	Advanced Encryption Standard. Specified in FIPS 197.
API	Application Programming Interface.
CBC	Cipher Block Chaining. A mode of encryption in which each encrypted block depends upon previous output data.
Decryption	The restoration of the original plaintext data from a ciphertext.
ECB	Electronic Codebook. A mode of encryption that divides a message into blocks and encrypts each block separately.
Encryption	The transformation of input data (called plaintext) into a less intelligible form (called ciphertext) through a mathematical process.
FIPS	Federal Information Processing Standards.
HMAC	Hashed Message Authentication Code.
IV	Initialization Vector. Used as a seed value for an encryption operation.
KAT	Known Answer Test.
Key	A string of bits used in cryptography, to encrypt and decrypt data. Can be used to perform other mathematical operations as well.
NIST	National Institute of Standards and Technology.
SHA-256	Secure Hash Algorithm, 256 bit. Specified by FIPS 180-2.

Table 6 – Acronyms and Definitions

13 References

1. National Institute of Standards and Technology, Security Requirements for Cryptographic Modules. FIPS 140-2, May 25, 2001.
2. National Institute of Standards and Technology, Derived Test Requirements for FIPS PUB 140-2, Security Requirements for Cryptographic Modules, Draft, March 24, 2004.
3. National Institute of Standards and Technology, Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program. Updated October 22, 2009
4. National Institute of Standards and Technology, Advanced Encryption Standard (AES), FIPS 197, November 26, 2001.
5. National Institute of Standards and Technology, Secure Hash Standard, FIPS 180-3, October 2008.
6. National Institute of Standards and Technology, Keyed-Hash Message Authentication Code (HMAC) FIPS 198-1, July 2008.
7. Teledyne Webb Research, MiniCrypt Users Guide, revision 1.2