MiniCrypt

Firmware Version: 1.6
Document Version: 1.29
15 June 2020

Teledyne Webb Research
49 Edgerton Drive
North Falmouth, MA 02556
www.teledynemarine.com/webb-research/

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

Table of Contents

1 Introduction

This document specifies the Security Policy for the Teledyne Webb Research MiniCrypt module (MiniCrypt) designed for use with the APF-11 (Autonomous Profiling Float version 11). This Security Policy was produced as part of the Federal Information Processing Standard (FIPS) 140-2 Level 2 validation of the MiniCrypt library, firmware version 1.6.
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 signed library image (minicrypt.img.signed). The module, together with its associated application file, are programmed into, and executed directly from, flash memory on the target microcontroller (STM32F103ZGT6). The “minicrypt.img.signed” image must be programmed to a specific memory address: 0x080FB400. As part of self-test processing before any cryptographic operations can be used, MiniCrypt will use HMAC to generate message authentication code given a key and its own code image and compare the generated message authentication code to the message authentication code stored after the code image in flash. If the message authentication code comparison fails, the module cannot be used.

MiniCrypt contains only FIPS approved cryptographic algorithms listed in Table 2. Host application developers may call the APIs defined in the “mc_local.h” header file to use the approved cryptographic algorithms.

Per FIPS 140-2, the MiniCrypt cryptographic module is classified as an embedded multi-chip firmware cryptographic module.

MiniCrypt runs directly on the hardware and any RTOS that might be running on the device is not part of the module’s operational environment. MiniCrypt does not use any OS resources.

1.2 Boundary

The physical boundary for the Module (shown in Figure 1) is defined as the APF-11 printed circuit board (PCB) on which the functions of the Module execute. The cryptographic boundary (shown in Figure 2) contains only the MiniCrypt software module, “minicrypt.img.signed”.

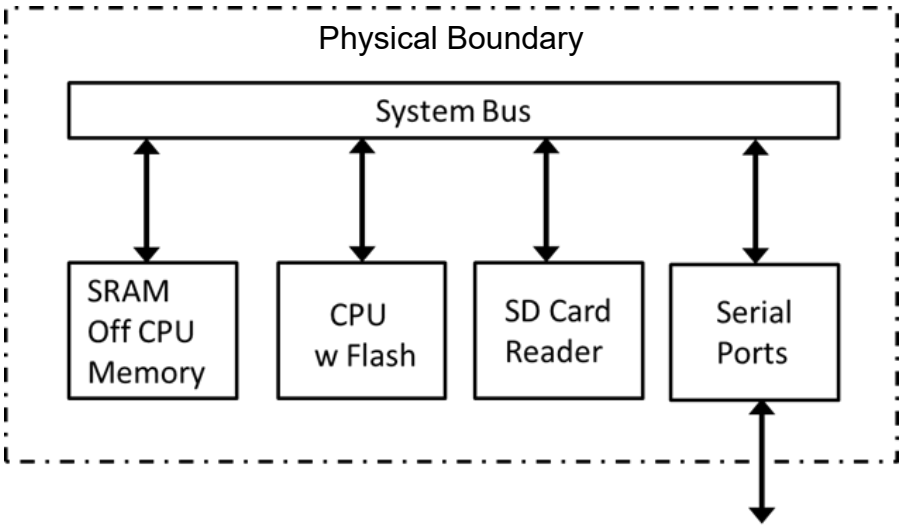

Figure 1 – Hardware Diagram Showing Controller Containing Cryptographic Module

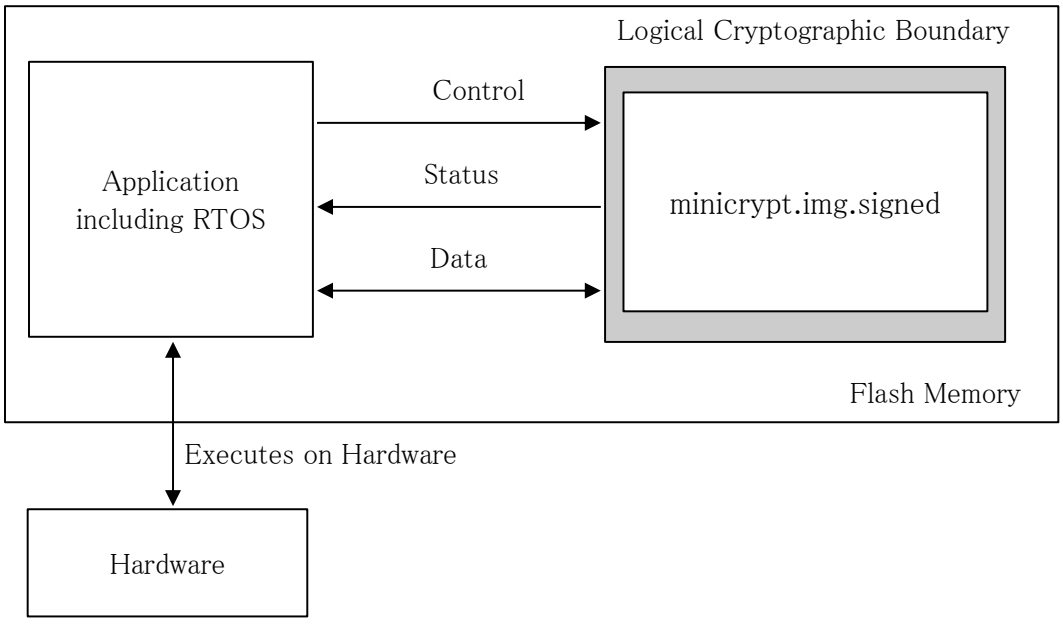

Figure 2 – Firmware Diagram Showing Logical Cryptographic Boundary

2 Security Level

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

Security Requirements Section	Level
Cryptographic Module Specification	2
Cryptographic Module Ports and Interfaces	2
Roles, Services and Authentication	2
Finite State Model	2
Physical Security	2
Operational Environment	N/A
Cryptographic Key Management	2
EMI/EMC	2
Self-Tests	2
Design Assurance	2
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. In this mode, the module can only determine the current mode or enter Approved mode; no cryptographic operations are allowed.
- Approved mode. In this module, the module can perform any cryptographic operations.

The following FIPS approved algorithms are supported in Approved mode:

CAVP Cert.	Algorithm	Standard	Mode / Method	Key Lengths	Use
#C897	AES	FIPS 197, SP 800-38A	CBC, ECB	128, 192, 256	Data Encryption and Decryption
#C897	HMAC	FIPS 198-1	HMAC-SHA-256	112-512	Message Authentication
#C897	SHS	FIPS 180-4	SHA-256	N/A	Message Digest

Table 2 – Approved Algorithms

Performing a successful role based authentication via the MC_set_role function is required in order to use any MiniCrypt cryptographic functions or MiniCrypt state altering functions like MC_set_mode. Once a successful role based authentication is achieved, all MiniCrypt API functions can be used.

Once authenticated, the API function MC_set_mode can be called to explicitly put the module in and out of Approved mode; this operation executes the power on self-test routines when putting the module in Approved mode. No cryptographic functionality is available unless MiniCrypt is in the Approved mode.

With or without authenticating, an application can 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 Approved mode.

The module does not implement any non-approved cryptographic algorithms.

4 Logical Interfaces

All FIPS 140-2 Logical 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. As a firmware-only module, the module does not have physical ports. For the purpose of the FIPS 140-2 validation, the physical ports are interpreted to be, the physical ports of the hardware platform on which the firmware application, containing the module, runs.

FIPS Interface	Logical Interface
Data Input	API input parameters
Data Output	API output parameters
Control Input	API input parameters
Status Output	API return codes

Table 3 – Logical Interfaces

5 Access Control Policy

5.1 Roles and Services

MiniCrypt supports two authenticated roles: a “Crypto Officer” and a “User” role. Only one role can be active at a time and MiniCrypt does not allow concurrent operators.

5.1.1 User Role

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

5.1.3 Role Authentication

MiniCrypt requires a password be provided with the selected role. There is a password associated with the “User” role and another password associated with the “Crypto Officer” role. The application using MiniCrypt provides the plaintext password. It is the application’s responsibility to clear the memory that held any password. The known SHA256 hashed password for each role is stored statically in the MiniCrypt code library.

Role	Authentication Type and Data	Strength of Authentication
User	Role based password	25 characters (Combination of lower, upper, special and number characters) The probability of guessing the password is greater than or equal to P = 1/94^25 = 1/2.129e49, which is less than 1 in 1,000,000. The MC_set_role command used to perform the User role authentication can be executed approximately 300 times within 1 minute. This means that an attacker has the probability of guessing the User password in one minute as 300/2.129e49 which is much less than the requirement of 1/100,000.
Crypto Officer	Role based password	28 characters (Combination of lower, upper, special and number characters) The probability of guessing the password is greater than or equal to P = 1/94^28 = 1/1.768e55, which is less than 1 in 1,000,000. The MC_set_role command used to perform the Crypto Officer role authentication can be executed approximately 300 times within 1 minute. This means that an attacker has the probability of guessing the Crypto Officer password in one minute as 300/1.768e55 which is much less than the requirement of 1/100,000.

Table 4 – Role Authentication Details

5.2 Definition of Critical Security Parameters (CSPs)

The Critical Security Parameters (CSPs) defined for MiniCrypt consists of cryptographic keys and passwords.

The module does not persistently store keys. The following keys 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.

There are two SHA-256 password hashes stored persistently within the logical boundary and plaintext inputs are compared to them.

5.2.1 Definition of CSPs Access Types

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

Services	Keys / CSPs	Authorized Roles	Type of Access
AES Encrypt[1]	AES Key	User / Crypto Officer	Execute
AES Decrypt[1]	AES Key	User / Crypto Officer	Execute
Set AES Key[1]	AES Key	User / Crypto Officer	Write
HMAC-SHA-256[1]	HMAC Key	User / Crypto Officer	Write/Execute
SHA-256[1]	N/A	User / Crypto Officer	Execute
Get Mode	N/A	User / Crypto Officer / Unauthenticated	N/A
Set Mode	N/A	User / Crypto Officer	N/A
Get Role	N/A	User / Crypto Officer / Unauthenticated	N/A
Set Role	Password and SHA-256 password hash	User / Crypto Officer	Read/Execute
Get Status	N/A	User / Crypto Officer / Unauthenticated	N/A
Run Self-tests[1]	N/A	Crypto Officer	N/A

Table 5 – Key and CSP Access Rights within Services

[1] Available in Approved mode only

6 Key Management

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.

MiniCrypt does not currently support key generation.

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.

MiniCrypt accepts key data input from calling application 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.

CSP Name	CSP Type	Generation/Input	Storage	Zeroization
AES Key	128, 192, 256 bit AES key	Input	Volatile memory	Cleared upon conclusion of usage
HMAC Key	112-512 bit key	Input	Volatile memory	Cleared upon conclusion of usage
Crypto Officer Password	ASCII password	Input	Volatile memory	Cleared upon conclusion of usage
User Password	ASCII password	Input	Volatile memory	Cleared upon conclusion of usage
SHA-256 Password Hash	SHA-256 hash	N/A	Non-Volatile memory	Never cleared

Table 6 – Life Cycle of CSPs

7 Operational Environment

This module operates in a non-modiable operational environment under the FIPS 140-2 Section 4.6 definitions.

7.1 Hardware Platform

For FIPS 140-2 testing, the library was installed into flash memory on STM32F103ZGT6 microcontroller with 1 M Bytes of flash program memory and 96 K Bytes of SRAM data memory using a serial port for human computer interaction.

8 Self Tests

The cryptographic module performs the following power up self-tests:

A. Integrity Test (HMAC-SHA-256)
B. Known Answer tests:
    a. AES Encrypt KAT
    b. AES Decrypt KAT
    c. SHA-256 KAT
    d. HMAC-SHA-256 KAT

9 Physical Security

Since the MiniCrypt module is entirely firmware, the MiniCrypt module depends on protections of the APF-11 printed circuit board (PCB) on which MiniCrypt is installed. The PCB includes an opaque conformal coating used to protect against visual identification of hardware components and allows for detection of any tamper of the underlying circuitry.

Figure 3 –APF-11 PCB Front with Conformal Coating

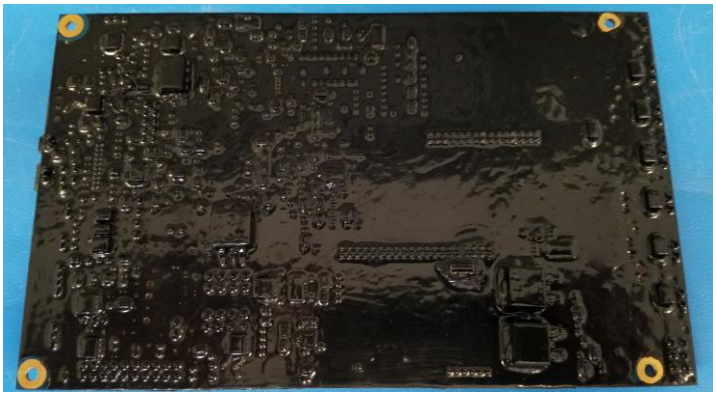

Figure 4 – APF-11 PCB Back with Conformal Coating

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 Design Assurance

11.1 Delivery 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.

One must verify the authenticity of the “minicrypt.img.signed” file using an external SHA256 application. The correct SHA256 hash for MiniCrypt 1.6 “minicrypt.img.signed” file is:

d83cfd7f9bb29c14c51a40435bc307bfcae74253ed9cab2062a9c9ce5718d7df

11.2 Operator Guidance

Guidance for the operator to bring the module into operational state is as follows:

- Program the module into the target system flash memory at address 0x080FB400.
- Call API functions per the MiniCrypt Users Guide.
- To bring the module into an operational state:
    - The API function MC_set_role must be called with correct role ID and password for the desired role. Upon a valid role and password combination, the module will enter Approved mode.

Reference the MiniCrypt Users Guide for API information and additional details.

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.
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 180-4.

Table 7 – Acronyms and Definitions

13 References

1. [FIPS 140-2] NIST, Security Requirements for Cryptographic Modules, May 25, 2001
2. [FIPS 180-4] NIST, Secure Hash Standard (SHS), August 2015
3. [FIPS 197] NIST, Advanced Encryption Standard (AES), November 26, 2001
4. [FIPS 198-1] NIST, The Keyed-Hash Message Authentication Code (HMAC), July 2008
5. [SP 800-38A] NIST, Recommendation for Block Cipher Modes of Operation: Methods and Techniques, December 2001
6. Teledyne Webb Research, MiniCrypt Users Guide, revision 1.6