Zebra Technologies Corporation

Non-Proprietary FIPS 140-3 Security Policy for

Zebra DCS Cryptographic Library

Firmware Module

Firmware Versions:

DAACUS00-002-R00 on Zebra CR6080, and DAACWS00-002-R00 on Zebra CS6080

Documentation Version : 1.4
Last Update : March 5, 2024

Table of Contents

List of Figures

List of Tables

1. General

The Zebra DCS Cryptographic Library provides data encryption/decryption functionality to devices such as wireless barcode scanners and cradles. These devices are used in a variety of environments such as retail and manufacturing.

Figure 1. Zebra CR6080 Cradle

Figure 2. Zebra CS6080 Scanner

The module is a FIPS 140-3 compliance firmware module with a multi-chip standalone embodiment. The main purpose of the module is to encrypt/decrypt data.

The following table indicates the actual security levels for each area of the cryptographic module.

Table 1-1 Security Levels

ISO/IEC 24759 Section 6. [Number Below]	FIPS 140-3 Section Title	Security Level
1	General	1
2	Cryptographic Module Specification	1
3	Cryptographic Module Interfaces	1
4	Roles, Services and Authentication	1
5	Software/Firmware Security	1
6	Operational Environment	1
7	Physical Security	1
8	Non-invasive Security	N/A
9	Sensitive Security Parameter Management	1
10	Self-Tests	1
11	Life-cycle Assurance	1
12	Mitigation of other attacks	N/A

The module has an overall security level of 1.

2. Cryptographic Module Specification

The module was performed at Security Level 1. The following configurations were tested by the lab:

Table 2-1 Tested Operational Environments

Operating Systems	Tested Platform Hardware Versions	Processor on the Tested Platforms	PAA\Acceleration
ThreadX v6.5	Zebra CS6080	Faraday CortexA9 ICON-D	N/A
Micrium uC/OS-II v2.85	Zebra CR6080	STM32f427iih6tr	N/A

The table below lists approved cryptographic algorithms employed by the module:

Table 2-2 Approved Algorithms

CAVP Cert	Algorithm and Standard	Mode/ Method	Description/Key Size(s) / Key strength(s)	Use/Function
AES Certs. #A1639 and #A1640	AES [FIPS 197]	AES-CBC	256 bits	Data encryption/decryption
AES Certs. #A1639 and #A1640	AES [FIPS 197]	AES-ECB	256 bits	Prerequisite algorithm for AES-KWP
KTS (AES Certs. #A1639 and #A1640)	AES-KWP [SP800-38F]	AES-KWP	256 bits	Key wrapping
HMAC Certs. #A1639 and #A1640	HMAC [FIPS 198-1]	HMAC-SHA2-256 (MAC: 256 Key Length: 256)	256 bits	Firmware integrity and Firmware load test
SHS Certs. #A1639 and #A1640	SHS [FIPS 180-4]	SHA2-256 (Message Length: 0-65528 Increment 8)	N/A	Hash operation

Notes:

- There are some algorithm modes that were tested but not used by the module. Only the algorithms, modes, and key sizes that are implemented by the module are shown in this table.
- Algorithm Cert. #A1639 was tested for Zebra DCS Cryptographic Module running on Zebra Cradle (Zebra CR6080) tested platform.
- Algorithm Cert. #A1640 was tested for Zebra DCS Cryptographic Module running on Zebra Scanner (Zebra CS6080) tested platform.

Mode of Operation

The module can only be operated in Approved mode of operation. The module does not support non-Approved algorithms or services.

Block Diagram

Figure 3 below depicts the module's Block Diagram. Please note that the bold RED rectangle in the block diagram represents the Tested Operational Environment's Physical Perimeter (TOEPP) containing the Module (the thin RED rectangle).

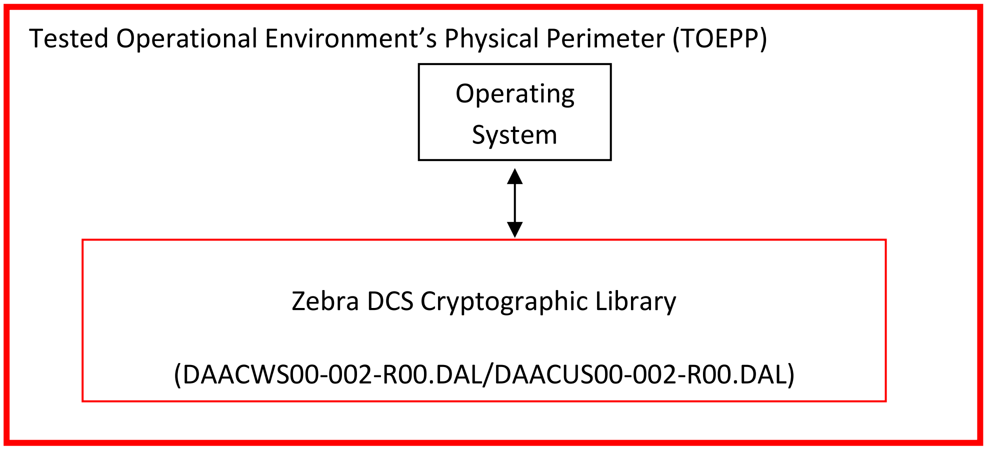
Tested Operational Environment's Physical Perimeter (TOEPP)
Operating System
Zebra DCS Cryptographic Library
(DAACWS00-002-R00.DAL/DAACUS00-002-R00.DAL)

Figure 3. Module Block Diagram

3. Cryptographic Module Interfaces

The module's physical perimeter encompasses the case of the tested platform mentioned in Table 2-1. The module provides its logical interfaces via Application Programming Interface (API) calls. The logical interfaces provided by the module are mapped onto the FIPS 140-3 interfaces (data input, data output, control input, control output and status output) as follows.

Table 3-1 Ports and Interfaces

Physical Port	Logical Interface	Data that passes over port/interface
N/A	Data Input Interface	Arguments for an API call that provide the data to be used or processed by the module.
N/A	Data Output Interface	Arguments output from an API call.
N/A	Control Input Interface	Arguments for an API call used to control and configure module operation. The Control Input Interface also includes the registry values used to control module behavior.
N/A	Control Output Interface	N/A
N/A	Status Output Interface	Return values from firmware API commands used to obtain information on the status of the module. The Status Output Interface also includes the log file where the module messages are output.

4. Roles, Services and Authentication

The module supports Crypto Officer (CO). The cryptographic module does not provide any authentication methods. The module does not allow concurrent operators. The Crypto Officer is implicitly assumed based on the service requested.

The module provides the following services to the Crypto Officer.

Table 4-1 Roles, Service Commands, Input and Output

Role	Service	Input	Output
Crypto Officer	Run pre-operational and conditional self-tests	Commands to initiate the pre-operational or conditional self-tests	Self-Tests Pass/Fail status code
Crypto Officer	Show Status	Commands to check the module’s status	Status output code
Crypto Officer	Set AES Encryption Key	Commands to set AES Encryption Key	Success or error status code
Crypto Officer	Set Shared Key (Current)	Commands to set Shared Key (Current)	Success or error status code
Crypto Officer	Set Shared Key (Default)	Commands to set Shared Key (Default)	Success or error status code
Crypto Officer	Wrap/Unwrap AES Encryption Key	Commands to wrap/unwrap AES Encryption Key with Shared Key (Current) or Shared Key (Default)	Success or error status code
Crypto Officer	Wrap/Unwrap Shared Key (Current)	Commands to wrap/unwrap Shared Key (Current) with Shared Key (Default)	Success or error status code
Crypto Officer	Protect Data using AES Encryption Key	Commands to encrypt and decrypt the data with AES Encryption Key	Encrypted data or error status code
Crypto Officer	Zeroize	Commands to conduct zeroization	All SSPs were zeroized with “0”s
Crypto Officer	Conduct Firmware Integrity Test	Command to conduct the Firmware Integrity Test	Success or error
Crypto Officer	Conduct Firmware Load Test	Command to conduct the Firmware Load Test	Success or error
Crypto Officer	Show Version	Command to get Firmware version	Firmware version

Table 4-2 defines the relationship between access to CSPs and the different module services. The modes of access shown in the table are defined as:

- R = Read: The module reads the CSP. The read access is typically performed before the module uses the CSP.
- E = Execute: The module executes using the CSP.
- W = Write: The module writes the CSP. The write access is typically performed after a CSP is imported into the module, when the module generates a CSP, or when the module overwrites an existing CSP.
- Z = Zeroize: The module zeroizes the CSP.

Table 4-2 Approved Services

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access Rights to SSPs	Indicator

Run pre-operational and conditional self-tests	Run Pre-operational and Conditional Self-Tests	AES-CBC;
		AES-ECB;
		AES-KWP;
		HMAC-SHA2-256;
		SHA2-256
			N/A	Crypto Officer	N/A	Pass/fail

Show Status	Show status of the module state	N/A	N/A	Crypto Officer	N/A	N/A
Set AES Encryption Key	Encrypt or decrypt AES Encryption Key with Access Key	AES-CBC	Access Key, AES Encryption Key	Crypto Officer	R, W, E	Success or error code
Set Shared Key (Default)	Encrypt or decrypt Shared Key (Default) with Access Key	AES-CBC	Access Key, Shared Key (Default)	Crypto Officer	R, W, E	Success or error code
Set Shared Key (Current)	Encrypt or decrypt Shared Key (Current) with Access Key	AES-CBC	Access Key, Shared Key (Current)	Crypto Officer	R, W, E	Success or error code
Wrap/Unwrap AES Encryption Key	Wrap AES Encryption Key with Shared Key (Default) or Shared Key (Current)	AES-ECB; AES-KWP	AES Encryption Key, Shared Key (Default) or Shared Key (Current)	Crypto Officer	R, W, E	Success or error code
Wrap/Unwrap Shared Key (Current)	Wrap/unwrap Shared Key (Current) with Shared Key (Default)	AES-ECB; AES-KWP	Shared Key (Default), Shared Key (Current)	Crypto Officer	R, W, E	Success or error code
Protect Data using AES Encryption Key	Encrypt/decrypt data using AES Encryption Key	AES-CBC	AES Encryption Key	Crypto Officer	R, W, E	Success or error code
Zeroize	Zeroize all Keys and CSPs upon demand or request via the API Zeroization function.	N/A	All keys	Crypto Officer	Z	Zeroization status output

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access Rights to SSPs	Indicator
Conduct Firmware Integrity Test	Check Firmware Integrity Message Authentication Code (MAC) during the firmware integrity test	HMAC-SHA2-256	Firmware Integrity Test Key (non-SSP)	Crypto Officer	R, E	Success or error code
Conduct Firmware Load Test	Check MAC during the firmware load test	HMAC-SHA2-256	Firmware Load Test Key	Crypto Officer	R, E	Success or error code
Show Version	Get version of the current Firmware	N/A	N/A	Crypto Officer	N/A	N/A

5. Software/Firmware Security

Integrity Techniques

The module is provided in the form of binary executable code. To ensure the firmware security, the module is protected by HMAC-SHA2-256 (HMAC Certs. ##A1639 and #A1640) algorithm. The Firmware Integrity Test Key (non-SSP) was pre-loaded to the module's binary the factory and used for firmware integrity test only at the pre-operational self-test. At Module's initialization, the integrity of the runtime executable is verified using a HMAC-SHA2-256 digest which is compared to a value computed at build time. If at the load time the MAC does not match the stored, known MAC value, the module would enter to an Error state with all crypto functionality inhibited. The firmware module was saved in the Flash memory in the DAL format.

The module also supports the firmware load test by using HMAC-SHA2-256 (HMAC Certs. ##A1639 and #A1640) algorithm. The Firmware Load Test Key was pre-loaded to the module's binary the factory and used for firmware load test. The operator can update the module's firmware upon successful verification, the module will load the new update upon reboot. The update attempt will be rejected if the verification fails.

Integrity Test On-Demand

Integrity test is performed as part of the Pre-Operational Self-Tests. It is automatically executed at power-on. The operator can power-cycle or reboot the tested platform to initiate the firmware integrity test on-demand.

6. Operational Environment

The module is operated in limited operational environment per FIPS 140-3 level 1 specifications. The operating system is restricted to a single operator mode of operation (i.e., concurrent operators are explicitly excluded), no other settings or restrictions to the operational environment are required. The application that makes calls to the modules is the single user of the modules, even when the application is serving multiple clients.

The module's firmware version running on each tested platform is detailed below.

- DAACUS00-002-R00 on Zebra CR6080
- DAACWS00-002-R00 on Zebra CS6080

7. Physical Security

The module is running on the multi-chip standalone production grade platform to meet physical security requirements from FIPS 140-3 level 1. The module's Tested Operational Environment's Physical Perimeter (TEOPP) is drawn at the casing of the tested platform (Zebra CS6080 or Zebra CR6080). The module's tested platforms consist of production-grade components. All ICs are coated with industry standard passivation.

8. Non-invasive Security

The module does not support Non-invasive Security. Thus, the security requirements from Section Non-invasive Security in FIPS 140-3 are not applicable.

9. Sensitive Security Parameter Management

Table 9-1 SSPs

Key/SSP/ Name Type	Strength	Security Function and Cert.	Generation	Import/ Export	Establishment	Storage	Zeroization	Use & related keys
Access Key	256 bits	AES-CBC; Certs. # A1639 or #A1640	Pre-loaded at the factory (in the module's executable binary)	Import: No Export: No	N/A	Stored in tested platform's Flash (executable binary image) in plaintext	Assume the CO role, and call the zeroization API function	Used for protection of AES Encryption Key, Shared Key (Default) and Shared Key (Current) stored in the Flash memory
AES Encryption Key	256 bits	AES-CBC; Certs. # A1639 or #A1640	N/A	Imported to the module in ciphertext wrapped with Shared Key (Default) or Shared Key (Current);	MD/EE	Stored in the tested platform's Flash (key store) in ciphertext (encrypted by Access Key)	Assume the CO role, and call the zeroization API function	Used for Data protection

Key/SSP/ Name Type	Strength	Security Function and Cert.	Generation	Import/ Export	Establishment	Storage	Zeroization	Use & related keys
				Export: No				
Shared Key (Default)	256 bits	AES-ECB; AES-KWP; Certs. # A1639 or #A1640	N/A	Imported to the module in plaintext; Export: No	N/A	Stored in the tested platform's Flash (key store) in ciphertext (encrypted by Access Key)	Assume the CO role, and call the zeroization API function	Used for wrapping or unwrapping AES Encryption Key or Shared Key (Current)
Shared Key (Current)	256 bits	AES-ECB; AES-KWP; Certs. # A1639 or #A1640	N/A	Imported to the module in ciphertext wrapped with Shared Key (Default); Export: No	MD/EE	Stored in the tested platform's Flash (key store) in ciphertext (encrypted by Access Key)	Assume the CO role, and call the zeroization API function	Used for wrapping or unwrapping AES Encryption Key
Firmware Load Test Key	256 bits	HMAC-SHA2-256 Certs. #A1639 or A1640	Pre-loaded at the factory (in the module's executable binary)	Import: No Export: No	N/A	Stored in tested platform's Flash (executable binary image) in plaintext	N/A	User for Firmware load test

10. Self-Tests.

When the module is loaded or instantiated (after being powered off, rebooted, etc.), the module runs pre-operational self-tests. The operating system is responsible for the initialization process and loading of the library. The module is designed with a default entry point (DEP) which ensures that the self-tests are initiated automatically when the module is loaded.

Prior to the module providing any data output via the data output interface, the module would perform and pass the pre-operational self-tests. A firmware integrity test is performed on the runtime image of the module with HMAC-SHA2-256 algorithm. Prior to the firmware integrity test, the module conducts a HMAC-SHA2-256 Cryptographic Algorithm Self-test (CAST). If the CAST on the HMAC-SHA2-256 is successful, the HMAC value of the runtime image is recalculated and compared with the stored HMAC value pre-computed at compilation time. Following the successful pre-operational self-tests, the module would execute the Conditional Cryptographic Algorithm Self-tests (CASTs) for all approved cryptographic algorithms implemented by the module during power-up as well.

The self-test success (return code ‘0’) or failure (return code ‘-5’) is output as a return value of the library load API call, which is functioning as the self-test status indicator. If one of the self-tests fails, the module transitions into an error state with the return code ‘-5’ output from the module’s status output interface. While the module is in the error state, all data through the data output interface and all cryptographic operations are disabled. The error state can only be cleared by reloading the module. All self-tests must be completed successfully before the module transitions to the operational state.

Below are the details of the self-tests conducted by the module.

- ❖ Pre-Operational Self-Tests:
  - Pre-operational firmware integrity test
    - HMAC-SHA2-256 KAT
    - Firmware Integrity Test (using HMAC-SHA2-256)

- ❖ Conditional Self-Tests.
  - Conditional cryptographic algorithm tests
    - AES-CBC with 256 bits Encryption Know Answer Test (KAT)
    - AES-CBC with 256 bits Decryption KAT
    - AES-KWP with 256 bits Encryption KAT
    - AES-KWP with 256 bits Decryption KAT
    - HMAC-SHA2-256 with 256 bits KAT
    - SHA2-256 KAT

Please note that the module conducts all CASTs successfully prior to the first operational use of the cryptographic algorithm.

- Conditional firmware load test
  - Firmware Load Test (HMAC-SHA2-256)

11. Life-cycle Assurance

The module always runs in the Approved Mode of Operation and does not implement any Non-Approved Security Functions. The module design corresponds to the module security rules. This section documents the security rules enforced by the cryptographic module to implement the security requirements of this FIPS 140-3 Level 1 module.

General Guidance

1. The validated module’s firmware binary files, DAACUS00-002-R00.dal (on Zebra CS6080) or DAACWS00-002-R00.dal (on Zebra CR6080), was installed into the respective tested platform (Table 2-1) while being manufactured.
2. The module is provided directly to Zebra solution developers and is not available for direct download or purchase by the general public.
3. To initialize the module, the operator needs to power on the tested platform.
4. The module is operated in a limited Operational Environment and only supports single user operator.

The module provides one operator role: Crypto Officer

5. The module does not support concurrent operators.

6. The module does not support a maintenance interface or role.

7. The module does not have any external input/output devices used for entry/output of data.

8. The module does not output the plaintext CSPs.

9. The module does not output intermediate key values.

10. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.

11. The Crypto Officer shall load the FIPS 140-3 validated firmware only to maintain validation.

12. Please conduct the periodic self-tests no more than 30 days (i.e., once/month) in order to avoid any conditions that may result in the interruption of the module’s operations.

12. Mitigation of Other Attacks

The module does not support Mitigation of Other Attacks. Thus, the security requirements from Section Mitigation of Other Attacks in FIPS 140-3 are not applicable.