FIPS 140-2 Security Policy

for

Motorola, Inc.

The Motorola EMS Cryptographic Library

Software Module

Software Component Version:
DAABES00-001-R00 – for MC9596 with Windows Mobile 6.5
DAABFS00-001-R00 – for MT2070/ MT2090 with Windows CE 5.0

Document Version Number: 1.5

1. Module Description

The Motorola EMS 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 retails and manufacturing.

For the purposes of FIPS 140-2 the module is classified as a software module.

The main purpose of the module is to encrypt/decrypt data.

FIPS 140-2 conformance testing of the module was performed at Security Level 1. The following configurations were tested by the lab:

Software Component Version	**Operating Systems**
Crypto.dll version DAABES00-001-R00	Windows Mobile 6.5
Crypto.dll version DAABFS00-001-R00	Windows CE 5.0

The following table summarizes FIPS 140-2 compliance claims

Security Requirements Section	Security 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

2. Cryptographic Boundary

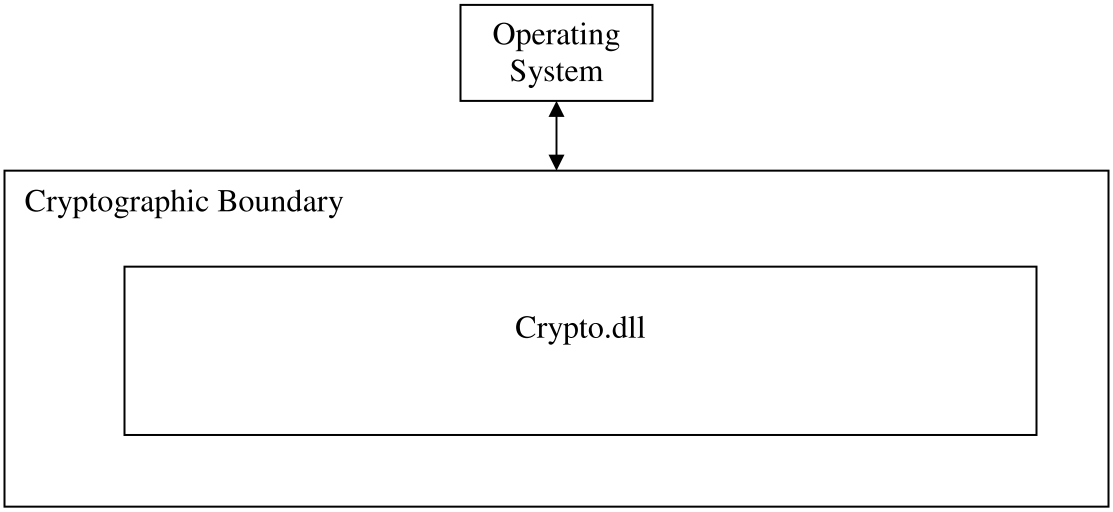
The cryptographic boundary of the module includes the software binary only.

The module includes the following logical interfaces:

- Control Input Interface: software API commands and command parameters used to control and configure module operation.
- Status Output Interface: return values from software API commands used to obtain information on the status of the module.
- Data Input Interface: data inputs to the software API commands
- Data Output Interface: data outputs of the software API commands

All module interfaces, inputs and outputs are provided by the software component.

Figure 1. Block Diagram



3. Roles and Services

The module provides the following roles:

1. User.
2. Crypto Officer.

The Crypto Officer configures the module and manages its cryptographic functionality. The User employs the cryptographic services provided by the module.

The module provides the following services to the User and Crypto Officer.

Service	Role	Access to Cryptographic Keys and CSPs R- read or use W – write or generate, Z – zeroize N/A – no CSPs are accessed by this service
Run-self tests	Crypto Officer/User	N/A
Get status of the module	Crypto Officer/User	N/A
Set AES key	Crypto Officer	W (sets AES encryption key)
Generate AES key	User, Crypto Officer	W (generates AES encryption key) R (uses RNG Seed Key and RNG Seed to generate the AES key)
Set shared encryption key	Crypto Officer	W (sets shared encryption key)
Encrypt/decrypt wireless data using the AES encryption key	User	R (uses the AES encryption key to encrypt/decrypt wireless data)
Encrypt/decrypt AES encryption key or the new shared encryption key using the current shared encryption key	User, Crypto Officer	R (uses the current shared encryption key to encrypt/decrypt the AES encryption key or the new shared encryption key)
Zeroize	Crypto Officer	Z (zeroizes all plaintext keys)

The module is always in FIPS mode of operation; non-FIPS mode is not applicable.

4. Security Functions

The table below lists approved cryptographic algorithms employed by the module

Algorithm	Certificate #
AES	1398 and 1396
HMAC	820 and 822
SHA-1	1267 and 1269
ANSI X9.31 RNG	764 and 765

5. Key Management

The following cryptographic keys are supported by the module

Name and Type	Generation or establishment	Usage
Access Key	Pre-set in the module binary	Read/Write AES and Shared keys
AES encryption key	Loaded encrypted with the access key, or the shared key. May also be generated using the ANSI X9.31 RNG	Encryption of the wireless data
Shared Key (Default)	Loaded encrypted with the access key.	Encryption of the AES key or the new Shared Key
Shared Key (Current)	Loaded encrypted with the previously established shared key.	Encryption of the AES key or the new Shared Key
HMAC SHA-1 integrity key	Pre-set in the module binary	Used to check integrity of the module at initialization
RNG Seed Key	Pre-set in the module binary	Used to initialize RNG
RNG Seed	Generated by the OS	Used to initialize RNG

To zeroize the keys inside the logical cryptographic boundary one shall execute Crypto_ClearAllKeys API function, which will zeroize all keys in RAM and in FLASH memory, and reboot the module. On Windows Mobile OS, a procedure is provided during which the FLASH is erased and over written by other data that does not contain the library file.

6. Self Tests.

The module runs a set of self-tests on execution of library load API call. If one of the self-tests fails, the module transitions into an error state, where all data output and cryptographic operations are disabled. The self-test success or failure is output as a return value of the library load API call.

The module runs self-tests for the following algorithms

Algorithm	Test
AES	Known Answer Test (encrypt/decrypt)
ANSI X9.31 RNG	Known Answer Test
ANSI X9.31 RNG	Conditional Test
SHA-1	Tested during the integrity check
HMAC SHA-1	Tested during the integrity check

Note: The integrity check is done by computing the HMAC SHA-1 signature on the module binary and comparing it to the previously computed value. Therefore the requirement to self test HMAC and SHA-1 is fulfilled.

7. Approved Mode of Operation

The module always runs in the Approved Mode of Operation and does not implement any Non-Approved Security Functions.