Motorola Mobility Linux Kernel Software Cryptographic Module

FIPS 140-2 Security Policy

Module Version 1.0

Document version 1.13

March 11, 2015

CHANGE RECORD

Revision	*Date*	*Author*	*Description of Change*
1.13	03/11/15	W. Ribeiro	Update to add new platforms
1.12	12/12/13	W. Ribeiro	Update to add new test platforms
1.11	09/16/13	W. Ribeiro	Update to add official name to new test platform
1.10	09/13/13	W. Ribeiro	Update based on CMVP comments
1.9	04/17/13	W. Ribeiro	Including new operation system version
1.8	01/07/13	J. Pinto	Update based on review comments
1.7	12/21/12	W. Ribeiro	Update based on review comments
1.6	12/19/2012	W. Ribeiro	Added CAVP numbers and updated based on review comments
1.5	11/12/2012	W. Ribeiro	Update based on review comments
1.4	10/31/2012	W. Ribeiro	Update based on review comments
1.3	10/30/2012	W. Ribeiro	Update based on review comments
1.2	10/11/2012	J. Pinto	Update with latest review comments
1.1	08/27/2012	J. Pinto	Include review comments
1.0	03/28/2012	J. Pinto	Initial version

Contents

Tables

Figures

1 Module Overview

This non-proprietary security policy describes the Motorola Mobility Linux Kernel Software Cryptographic Module (hereafter referred to as the module) and the FIPS Approved mode of operation per FIPS 140-2 Level-1 requirements.

The module is a software only Linux kernel cryptographic module intended to operate on a multi-chip standalone mobile device (physical boundary) running Android. The name of the module file in the mobile device file system is moto_crypto.ko. The module provides general-purpose cryptographic services to the remainder of the Linux kernel. As with all Linux kernel modules, this module is written in C.

The logical cryptographic boundary of the module is shown in Figure 1.

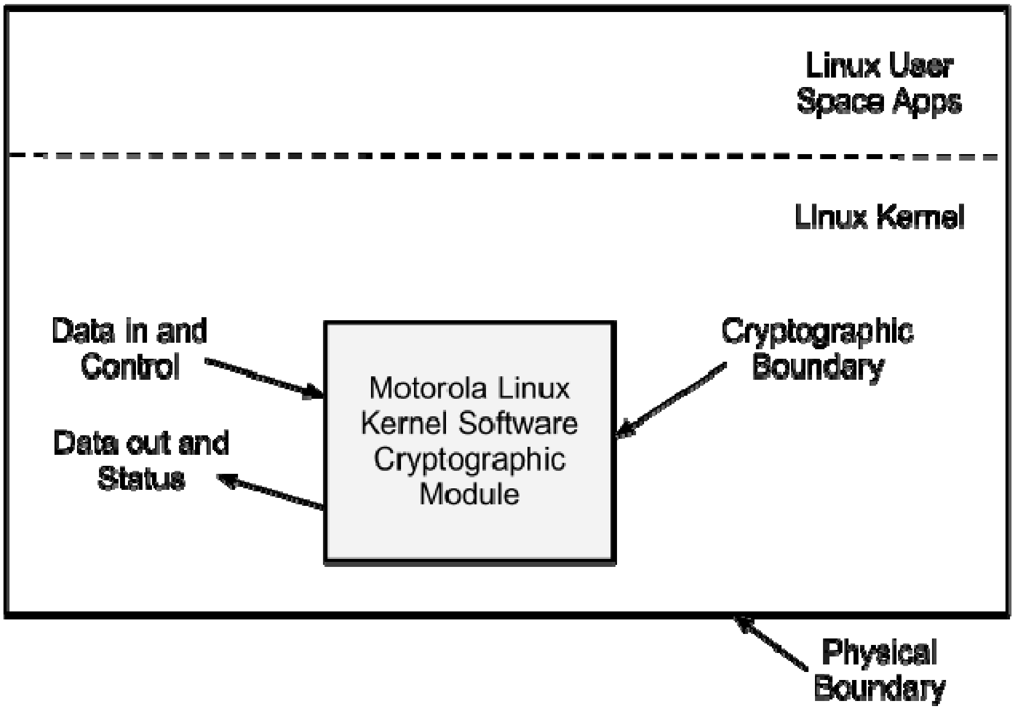

Figure 1 - Module Cryptographic Boundary – Software Block Diagram

The software configuration for this validation is Motorola Linux Kernel Software Cryptographic Module, Version 1.0. This module includes Power-On Self Tests.

Figure 2 depicts the hardware block diagram of a device that will run the module.

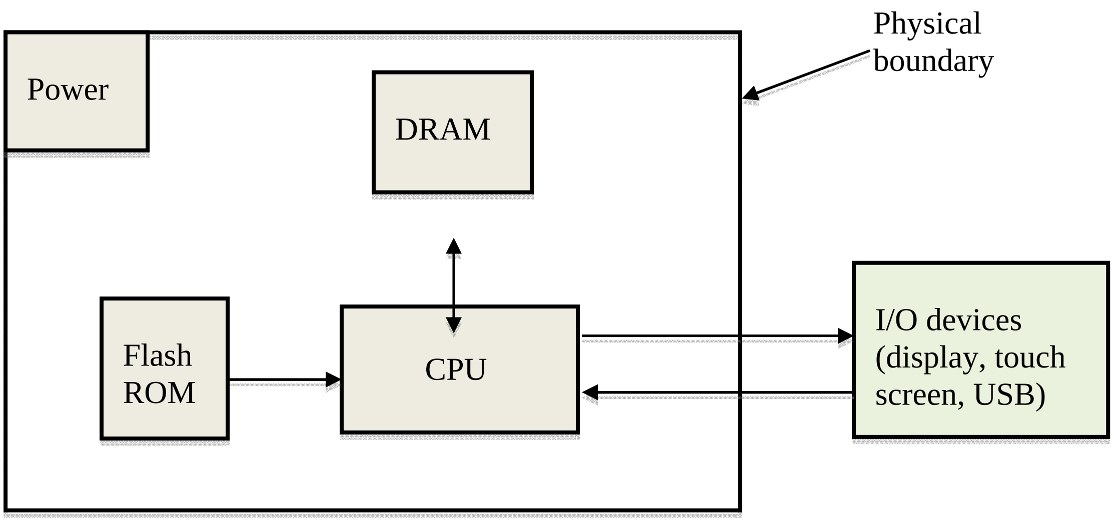

Figure 2 - Module Cryptographic Boundary – Hardware Block Diagram

2 Security Level

The cryptographic module is designed to operate at FIPS 140-2 overall security level 1. Table 1 below shows the security level met for each of the eleven areas specified within the FIPS 140-2 security requirements.

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 Specification

3 Modes of Operation

3.1 FIPS Approved Mode of Operation

The module always operates in a FIPS Approved mode of operation and provides all of the algorithms described in Table 2.

The module will enter FIPS Approved mode following successful power up initialization.

The module supports the following FIPS Approved algorithms. The module does not provide non-approved security functions.

FIPS Approved Algorithm	Key / block size	CAVP Cert.
AES (CBC/ECB/CTR) Encryption and Decryption	128, 192, 256 bits	#2287
Triple DES (CBC/ECB) Encryption and Decryption	192 bits (3-key only)	#1435
SHA	160, 224, 256, 384, 512 bits	#1968
HMAC SHA	160, 224, 256, 384, 512 bits	#1403
ANSI X 9.31 PRNG	AES-128	#1138

Table 2 - FIPS Approved Algorithms Used in Current Module

4 Ports and Interfaces

FIPS Interface	Port
Data Input	API input parameters
Data Output	API return values
Control Input	API function calls
Status Output	API return codes; Kernel log file
Power Input	Physical power connector

Table 3 – Ports and Interfaces

5 Identification and Authentication Policy

5.1 Assumption of Roles

The module supports two distinct operator roles, User and Cryptographic Officer (CO). The operators assume their roles implicitly based on the services being accessed. The module does not support authentication of roles.

6 Access Control Policy

6.1 Roles and Services

Table 4 shows the services available on the module, which roles are enabled to access the service, the API calls which support the service, the CSPs involved, if any, and the access mode to them.

Role	Service	Description	CSP	API Calls	Access (Read, Write, Execute)
User	Encrypt/Decrypt with AES	Encrypt/Decrypt data using AES in CBC/ECB/CTR modes	AES keys	moto_aes_set_key moto_ecb_aes_encrypt moto_ecb_aes_decrypt moto_cbc_aes_encrypt moto_cbc_aes_decrypt moto_ctr_aes_encrypt moto_ctr_aes_decrypt	R, W, EX
User	Encrypt/Decrypt with Triple DES	Encrypt/Decrypt data using Triple DES	Triple DES 3 keys	moto_des3_ede_set_key moto_ecb_des3_ede_encrypt moto_ecb_des3_ede_decrypt moto_cbc_des3_ede_encrypt moto_cbc_des3_ede_decrypt	R, W, EX
User	SHA	Use hash functions to perform SHA-1, SHA-224, SHA-256, SHA-384 or SHA-512	N/A	moto_sha1_init moto_sha1_update moto_sha1_final moto_sha224_init moto_sha224_final moto_sha256_update moto_sha256_init moto_sha256_final moto_sha384_init moto_sha512_update moto_sha384_final moto_sha512_init moto_sha512_update moto_sha512_final	R, W, EX
User	HMAC SHA	Perform HMAC SHA-1, HMAC SHA-224, HMAC SHA-256, HMAC SHA-384 or HMAC SHA-512	HMAC key	moto_hmac_create moto_hmac_init moto_hmac_setkey moto_hmac_update moto_hmac_final	R, W, EX

Role	Service	Description	CSP	API Calls	Access (Read, Write, Execute)
User	PRNG[1]	Generate pseudo random number using ANSI X 9.31 PRNG with AES-128	Seed, Seed key	moto_cprng_init moto_fips_cprng_reset moto_fips_cprng_get_random moto_cprng_exit	R, W, EX
User	Show status	Provides information about the status of the module	N/A	Kernel log and return codes from API calls	R
User	Perform Self-Test	Power cycle the module to perform self-test on demand	N/A	N/A	N/A
Crypto Officer	Initialization	Module initialization	N/A	moto_crypto_init	N/A
Crypto Officer	Finalization	Module finalization	N/A	moto_crypto_fini	N/A

Table 4 - Services

The table below details the usage of the CSPs involved in the module.

CSP	Description / Usage
AES keys	Passed in moto_aes_set_key API call to be used as the key on the AES encrpyt/decrypt process. Supported key sizes are 128, 192 and 256 bits.
Triple DES keys	Passed in moto_des3_ede_set_key API call to be used as the key on the triple DES encrpyt/decrypt process. Total key size is 192 bits (3 keys of 64 bits each).
HMAC key	Passed in moto_hmac_setkey API call to be used as the key on the HMAC generation process.
Seed key[1]	Passed in moto_fips_cprng_reset API call to be used as the key for the PRNG underlying AES encryption. Key size is 128 bits.
Seed[1]	Passed in moto_fips_cprng_reset API call to be used as the PRNG seed.

Table 5 – CSPs description

Note: The module does not use public keys.

[1] There is no assurance of the strength of the externally provided entropy in the seed and seed key.

7 Operational Environment

The module was operational tested on the following platforms:

- Motorola Droid Razr HD (XT926) device running Android 4.1.2
- Motorola Droid Ultra (XT1080) device running Android 4.2.2
- Motorola Moto G (XT1028) device running Android 4.3
- Motorola Moto X (XT1060) device running Android 4.4
- Motorola Droid Turbo (XT1254) device running Android 5.0.2.

The module is intended for use on a personal mobile device using the Android operating system. For FIPS 140-2 compliance this is considered to be a single-user operational environment due to the fact that only one operator is in possession of the device at a time. Also, the module implementation is single-threaded.

All CSPs in memory remain in the process space of the operator using the module. The Android operating system uses its memory management and process separation mechanisms to ensure that outside processes cannot access the process memory used by the module.

8 Security Rules

The module design corresponds to the FIPS 140-2 Level 1 security rules. This section documents the security rules enforced by the cryptographic module to implement the security requirements of this FIPS 140-2 Level 1 module.

1. The cryptographic module shall provide two distinct operator roles. These are the User role, and the Cryptographic Officer role.
2. The cryptographic module does not support authentication; roles are implicitly assumed depending on the service accessed.
3. The cryptographic module shall perform the following tests:

   A. Power on Self-Tests

      1. Cryptographic algorithm tests
         a. SHA-1 Known Answer Test
         b. SHA-224 Known Answer Test
         c. SHA-256 Known Answer Test
         d. SHA-384 Known Answer Test
         e. SHA-512 Known Answer Test
         f. HMAC-SHA-1 Known Answer Test
         g. HMAC-SHA-224 Known Answer Test
         h. HMAC-SHA-256 Known Answer Test
         i. HMAC-SHA-384 Known Answer Test
         j. HMAC-SHA-512 Known Answer Test
         k. AES Encrypt Known Answer Test
         l. AES Decrypt Known Answer Test
         m. TDES Encrypt Known Answer Test
         n. TDES Decrypt Known Answer Test
         o. ANSI X 9.31 RNG Known Answer Test

      2. Software Integrity Test (HMAC SHA-256)

   B. Conditional Tests
   A continuous RNG test is performed during each use of the RNG service. If values of two consecutive random numbers are the same, the module goes into error state.

4. The operator shall be capable of commanding the module to perform the power-up self-test by cycling power or resetting the module.
5. Power-up self tests do not require any operator intervention.
6. Data output is inhibited during self-tests and error states. Only status information is output when in those states.
7. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
8. The module does not support concurrent operators.

9. The module does not support a maintenance interface or role.
10. The module does not support manual key entry.
11. The module does not have any external input/output devices used for entry/output of data.
12. The module does not output plaintext CSPs.
13. The module does not generate keys.
14. The module zeroizes memory used by CSPs prior to deallocation.
15. The RNG service returns an error if initialized with a seed and seed key with the same value.

9 Physical Security Policy

This is a software module therefore the FIPS 140-2 physical security requirements are not applicable.

10 Mitigation of Other Attacks Policy

The module has not been designed to mitigate attacks outside of the scope of FIPS 140-2. Therefore this section is not applicable.

11 References

[FIPS 140-2] FIPS Publication 140-2 *Security Requirements for Cryptographic Modules.*