Samsung FIPS BC for Mobile Phone and Tablet

FIPS 140-2 Security Policy

Version 1.4

Last Update: 2013-06-26

1. Introduction

This document is a non-proprietary FIPS 140-2 Security Policy for the Samsung FIPS BC for Mobile Phone and Tablet. It contains a specification of the rules under which the module must operate and describes how this module meets the requirements as specified in FIPS PUB 140-2 (Federal Information Processing Standards Publication 140-2) for a Security Level 1 multi-chip standalone software module.

1.1. Purpose of the Security Policy

There are three major reasons that a security policy is required:

- it is required for FIPS 140-2 validation,
- it allows individuals and organizations to determine whether the cryptographic module, as implemented, satisfies the stated security policy, and
- it describes the capabilities, protection, and access rights provided by the cryptographic module, allowing individuals and organizations to determine whether it will meet their security requirements.

1.2. Target Audience

This document is intended to be part of the package of documents that are submitted for FIPS validation. It is intended for the following people:

- Developers working on the release
- FIPS 140-2 testing lab
- Crypto Module Validation Program (CMVP)
- Consumers

2. Cryptographic Module Specification

This document is the non-proprietary security policy for the Samsung FIPS BC for Mobile Phone and Tablet, and was prepared as part of the requirements for conformance to Federal Information Processing Standard (FIPS) 140-2, Level 1.

The following section describes the module and how it complies with the FIPS 140-2 standard in each of the required areas.

2.1. Description of Module

The Samsung FIPS BC for Mobile Phone and Tablet is a software only security level 1 cryptographic module that provides general-purpose cryptographic services to the applications. The crypto module runs on an ARM processor.

The following table shows the overview of the security level for each of the eleven sections of the validation.

Security Component	Security Level
Cryptographic Module Specification	1
Cryptographic 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	3
Self Tests	1
Design Assurance	3
Mitigation of Other Attacks	N/A

Table 1: Security Levels

The module has been tested on the following platform:

Module/Implementation	Device	O/S & Ver.
Samsung FIPS BC for Mobile Phone and Tablet (SBC1.45_1.1)	Galaxy S3	Android Ice-cream Sandwich 4.0

Table 2: Tested Platform

2.2. Description of Approved Mode

The Module can be initialized in one mode, FIPS mode. In FIPS mode, the module will be initialized with symmetric algorithms, digest algorithms, HMAC and random generators. In this mode, asymmetric algorithms will be available without key generation functionality.

In the Approved mode the module provides the following approved functions:

- AES (CBC, ECB, CFB, OFB)
- SHA-1, SHA-224, SHA-256, SHA-384, SHA-512
- RNG (ANSI X9.31)
- Triple-DES (CBC, ECB, CFB, OFB)

- HMAC (with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512)
- RSA (sign/verify)
- DSA (sign/verify)

The module implements the following Non-Approved algorithms, which shall not be used in the FIPS 140-2 approved mode of operation:

- MD2
- MD4
- MD5
- IES
- ISSAC
- BLOWFISH
- TWOFISH
- RC2
- RC4
- RC5
- RC6
- RSA (encrypt/decrypt)
- Noekeon
- SALSA20
- HC128
- HC256
- VMPC
- SERPENT
- RIJNDAEL
- CAST5
- CAST6
- GOST28147
- GOST3411
- TEA
- XTEA
- ELGAMAL
- IDEA
- Tiger
- RIPEMD

- WHIRLPOOL
- ISO9797ALG3MAC
- GOST28147MAC
- GOST3410
- VMPCMAC
- ECGOST3410
- Grain
- Camellia
- SEED
- Direct random generator (non-approved RNG)
- Thread-based seed generator (non-approved RNG)
- Reverse window generator (non-approved RNG)

The above three non-approved random number generators will not be available to the user via Bouncycastle Provider. Please see Table 5, “Services” in Section 4.2 for the CAVP certificate numbers.

The module implements the following Non-compliant algorithms, which shall not be used in the FIPS 140-2 approved mode of operation:

- ECDSA (non-compliant)
- AES CMAC (non-compliant)
- Triple-DES-CMAC (non-compliant)
- SKIPJACK (non-compliant)
- Diffie-Hellman (non-compliant)
- EC Diffie-Hellman (non-compliant)

CAVEAT: The true cryptographic strength of AES encryption keys is 128 bits.

2.3. Cryptographic Module Boundary

2.3.1. Software Block Diagram

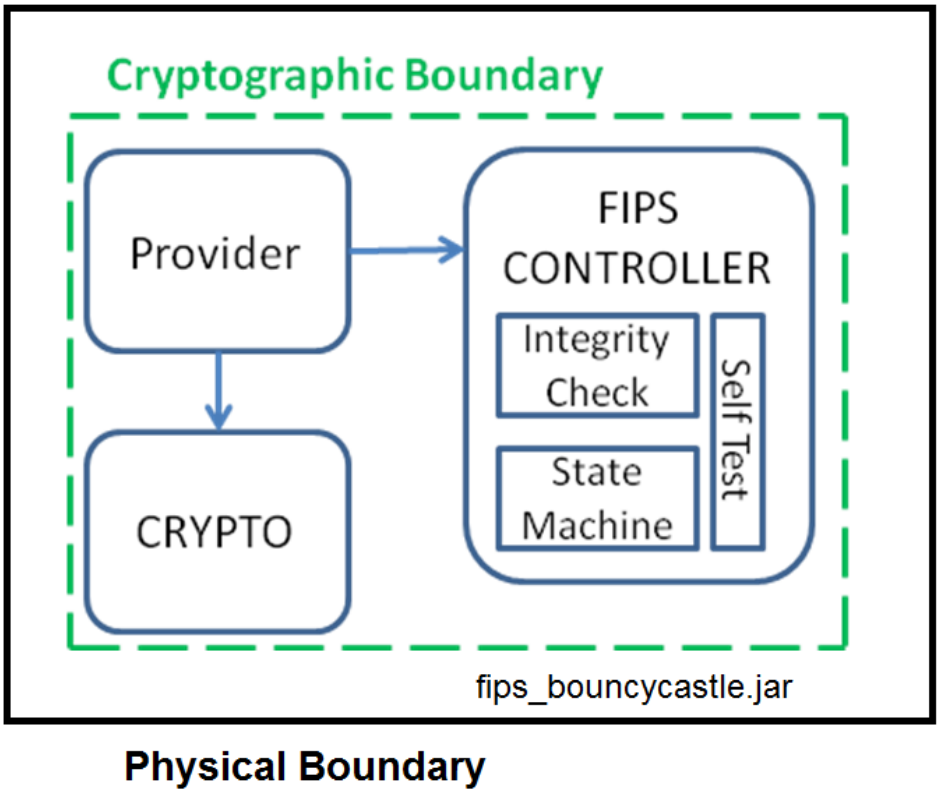

Figure 1: Software Block Diagram

The executable for the Bouncycastle module is fips_bouncycastle.jar

Related documentation:

- Bouncycastle FIPS certification High Level Design (Bouncycastle_FIPS_HLD.doc) version 1.1
- Samsung Bouncycastle Cryptographic Module (Samsung_Bouncycastle_SPv1.2.doc)

Note: The master component list is provided in Section 7.1 of the High Level Design document.

2.3.2. Hardware Block Diagram

This figure illustrates the various data, status and control paths through the cryptographic module. Inside, the physical boundary of the module, the mobile device consists of standard integrated circuits, including processors and memory. These do not include any security-relevant, semi- or custom integrated circuits or other active electronic circuit elements. The physical boundary includes power inputs and outputs, and internal power supplies. The logical boundary of the cryptographic module contains only the security-relevant software elements that comprise the module.

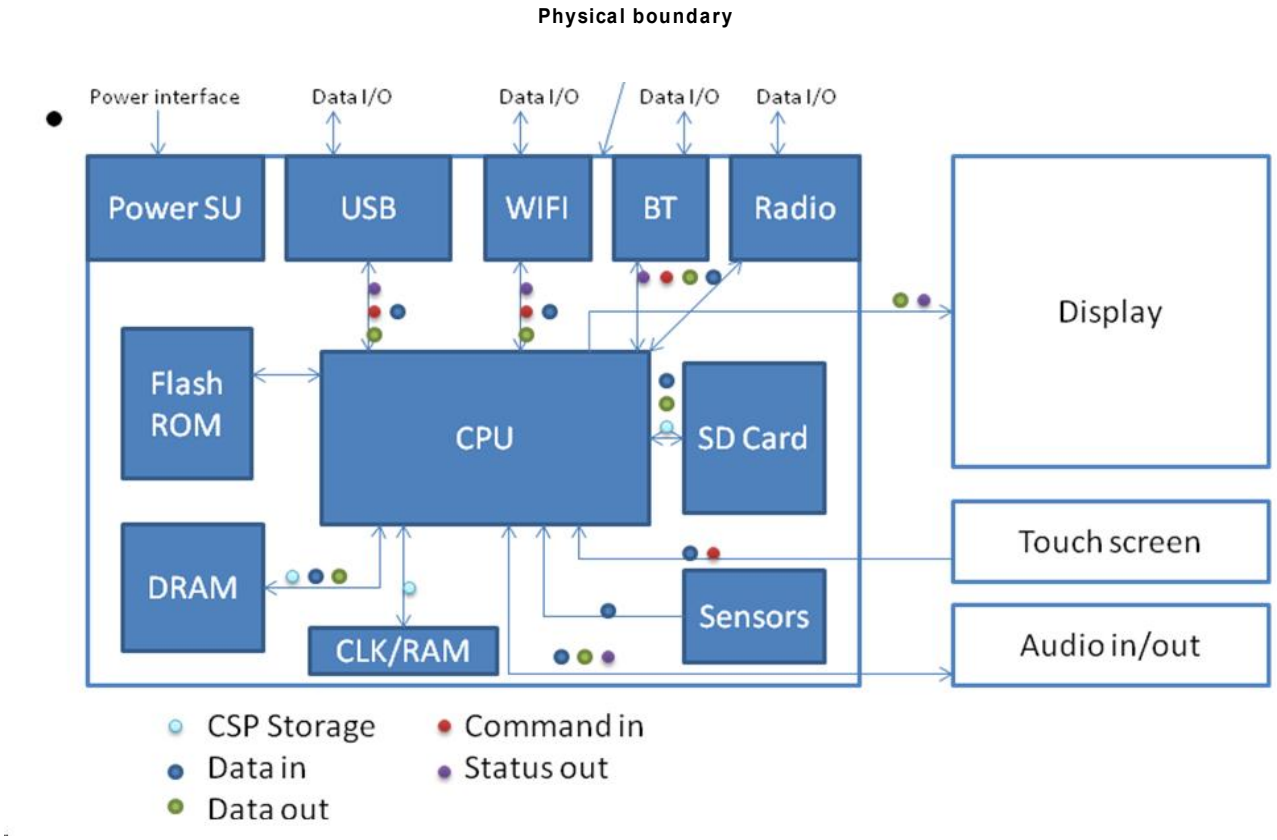

Figure 2: Hardware Block Diagram

3. Cryptographic Module Ports and Interfaces

FIPS Interface	Ports
Data Input	API input parameters
Data Output	API output parameters
Control Input	API function calls
Status Output	API function calls, or configuration files on filesystem, UI of the device application
Power Input	Physical power connector

Table 3: Ports and Interfaces

4. Roles, Services and Authentication

4.1. Roles

Role	Services (see list below)
User	Encryption, Decryption, Random Numbers, Digest Creation, Signature Generation, Signature Verification
Crypto Officer	Configuration, Encryption, Decryption, Random Numbers, Initialization of Module, Digest Creation, Signature Generation, Signature Verification

Table 4: Roles

The Module meets all FIPS 140-2 level 1 requirements for Roles and Services, implementing both User and Crypto Officer roles. The Module does not allow concurrent operators.

The User and Crypto Officer roles are implicitly assumed by the entity accessing services implemented by the Module. No further authentication is required. The Crypto Officer can initialize the Module.

4.2. Services

Role	Service	CSP	Modes	FIPS Approved (Cert #)	Access (Read, Write, Execute)
User, Crypto Officer	AES (encryption and decryption)	128, 192, 256 bit keys	ECB, CBC, CFB, OFB	Cert #2124	R, W, EX
Crypto Officer, User	HMAC with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 (key message digest)	HMAC Key	N/A	Cert #1295	R, W, EX
User, Crypto Officer	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512 (message digest creation)	N/A	N/A	Cert #1848	R, W, EX
User, Crypto Officer	Triple-DES (encryption/decryption)	2 Key & 3 Key	CBC, ECB, CFB, OFB	Cert #1350	R, W, EX
User, Crypto Officer	RSA (signature generation/ verification)	1024, 2048 bit keys	N/A	Cert #1093	R, W. EX
User, Crypto Officer	DSA (signature generation/ verification)	1024, 2048 bit keys	N/A	Cert #665	R, W, EX
User, Crypto Officer	Random Number Generator	Seed Key	AES-128, AES-192, AES-256	Cert #1090	R, W, EX

Role	Service	CSP	Modes	FIPS Approved (Cert #)	Access (Read, Write, Execute)
	ANSI X9.31				
User, Crypto Officer	Initialization	N/A	N/A	N/A	N/A
User, Crypto Officer (self tests are executed upon module initialization)	Execute Self Test	N/A	N/A	N/A	N/A
User, Crypto Officer	Check Status/Get State of the Module	N/A	N/A	N/A	R
Crypto Officer	Configuration	N/A	N/A	N/A	R, W, EX
User, Crypto Officer	Zeroization	RSA/DSA Keys	N/A	N/A	R, W, EX

Table 5: Services

4.3. Operator Authentication

There is no operator authentication; assumption of role is implicit by action.

4.4. Mechanism and Strength of Authentication

No authentication is required at security level 1; authentication is implicit by assumption of the role.

5. Finite State Machine

For information pertaining to the Finite State Model, please refer to the Functional Design document.

6. Physical Security

The Module is comprised of software only and thus does not claim any physical security.

7. Operational Environment

This module will operate in a modifiable operational environment per the FIPS 140-2 definition.

7.1. Policy

The operating system shall be restricted to a single operator mode of operation (i.e., concurrent operators are explicitly excluded).

The external application that makes calls to the cryptographic module is the single user of the cryptographic module, even when the application is serving multiple clients.

8. Cryptographic Key Management

8.1. Random Number Generation

The Module employs an ANSI X9.31 compliant random number generator for creation of keys. Note: the RNG seed is the tuple {V key DT}, where those values are defined in ANSI X9.31 Appendix A.2.4.

The Module's RNG algorithm is seeded and keyed using data from /dev/random. Environmental noises are fed into the entropy pool. When random is read, /dev/random provides data only if the entropy pool has enough data based on the estimate of the randomness generated from the environmental noises. If there is not enough data to be provided, the random read is blocked after entropy is exhausted. SHA digest is applied on the entropy collected before the data is given as output. This assures the unpredictability of the entropy collection itself.

The RNG provides the seed of size 128 bits from dev/random. Therefore, the RNG provides entropy of 128 bits.

The Module performs continual tests on the random numbers it uses, to ensure that the seed and seed key input to the approved RNG do not have the same value. The Module also performs continual tests on the output of the approved RNG to ensure that consecutive random numbers do not repeat.

Caveat: The module generates cryptographic keys whose strengths are modified by available entropy.

8.2. Key Entry and Output

The module does not support manual key entry or key output. Keys or other CSPs can only be exchanged between the module and the calling application using appropriate API calls.

8.3. Key Storage

No keys are stored in the Crypto module.

8.4. Zeroization Procedure

Bouncycastle algorithms considered for FIPS validation gets keys as input parameters. Algorithms do not make any local copies of the keys and are used for reference only.

Any internal and intermediate keys that algorithms generate are zeroed by calling the reset API available for all validated algorithms. All references to external CSPs are nullified after use.

9. Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)

Lab Name: PCTEST Engineering Laboratory, Inc
FCC Registration: #90864

For information related to FCC ID of the devices, please refer to the Functional Design document.

10. Self Tests

As per FIPS 140-2 requirements, self tests must be conducted up on initialization of the module and before the module becomes usable. Whenever an application invokes the module, a set of self tests executes automatically.

Self test consists of the following tests:

10.1. Power-Up Tests

10.1.1. Cryptographic algorithm tests (Known Answer Tests)

Cryptographic algorithm test using a known answer will be conducted for all cryptographic functions (e.g., encryption, decryption, authentication, and random number generation) of each Approved cryptographic algorithm implemented by the Bouncycastle Crypto module in FIPS mode.

Algorithm	**Test**
AES (encryption/decryption)	KAT
Triple-DES (encryption/decryption)	KAT
RSA	KAT
DSA	Pair-wise consistency test
PRNG	KAT
HMAC-SHA-1	KAT
HMAC-SHA-224	KAT
HMAC-SHA-256	KAT
HMAC-SHA-384	KAT
HMAC-SHA-512	KAT
SHA-1	KAT
SHA-224	KAT
SHA-256	KAT
SHA-384	KAT
SHA-512	KAT

Table 6: Power-Up Tests

10.1.2. Integrity Check

Integrity tests ensure that the Bouncycastle module is same as that was certified for FIPS compliance. This prevents malicious code to perform masquerading attacks, by replacing the FIPS certified Bouncycastle with another tainted implementation of Bouncycastle.

A digest will be calculated of the fips_bouncycastle.jar file that is the target of FIPS 140-2 validation and

certification.

During power up, a digest will be calculated of the available fips_bouncycastle.jar file, using one of the approved algorithms, and will be compared against the known expected digest value.

If the two digests match, it shows that fips_bouncycastle.jar has not been modified and is the one that was FIPS certified.

10.2. Conditional Tests

10.2.1. Continuous Random Number Generator (RNG) test

The Module currently uses PRNG based on ANSI X9.31 for all random number requirements. The PRNG is implemented as defined in NIST's document, Recommended Random Number Generator Based on ANSI X9.31, Appendix A.2.4.

The module ensures that the values of the seed and seed key are not the same. A Continuous Random Number Generator (CRNG) test is implemented for the RNG as well as for NDRNG (/dev/random).

11. Design Assurance

11.1. Configuration Management

All source code is maintained in internal source code servers and the tool, Perforce, is used as code control. Release is based on the Change List number maintained by Perforce, which is auto-generated. Every check-in process creates a new change list number.

Versions of controlled items include information about each version. For documentation, revision history inside the document provides the current version of the document. Version control maintains the all the previous version and the version control system automatically numbers revisions.

For source code, unique information is associated with each version such that source code versions can be associated with binary versions of the final product.

All documents are maintained in an internal document server per project. The versioning tool used is Sub version (svn). The version number is auto generated by the tool and version is controlled by a check-in and check-out mechanism.

In the development team, only authorized developers verified by login/password is allowed to access permitted documents in version control system.

11.2. Delivery and Operation

The Crypto module is never released as Source code. It may be released as Source for internal purposes based on Change List number generated by perforce. The module sources are stored and maintained at a secure development facility with controlled access.

This crypto module is built-in as a separate shared Java library, which can be used by any application. Currently it is used by the Email application and Exchange service in Ice-Cream Sandwich and later versions of Android devices. Once the device enters manufacturing phase, the source code branch is locked. Once it is locked, the source control system provides only read access. This ensures that no one can modify the source code in the Perforce depot.

The final binary is registered with its hash value to the internal system, which is not connected to any other network. Only authorized personnel through VPN can register the binary to automated manufacturing system, so that it can be downloaded to hardware without any manual intervention. The factory is also a secure site with strict access control to the manufacturing facilities. Employees are not allowed to bring in any personal belongings to the manufacturing facility and the entrance is controlled with employee ID-based badge access and monitored using CCTV.

The binary is released only by a SAMSUNG released tool and OTA (over the air). Over the air mechanism is controlled by service providers. If the binary is modified by an unauthorized entity, the device has a feature to detect the change and does not accept the binary changes.

11.3. User and Crypto Officer Guidance

Applications can get access to the crypto module by invoking an instance of the library.

A valid instance is returned only if all the self tests pass successfully. Otherwise, a null is returned.

Once an application has a valid Bouncycastle provider instance, it needs to register the Bouncycastle provider with JCE framework by adding it to the top of the JCE provider list.

This is so the Bouncycastle provider is accessible through the standard Java JCE interface. Please refer to the High Level Design document for more information on APIs.

12. Mitigation of Other Attacks

No other attacks are mitigated.

13. Glossary and Abbreviations

AES	Advanced Encryption Specification
CAVP	Cryptographic Algorithm Validation Program
CBC	Cypher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cypher Feedback
CC	Common Criteria
CMT	Cryptographic Module Testing
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CVT	Component Verification Testing
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
EAL	Evaluation Assurance Level
FSM	Finite State Model
HMAC	Hash Message Authentication Code
KAT	Known Answer Test
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
NVLAP	National Voluntary Laboratory Accreditation Program
OFB	Output Feedback
O/S	Operating System
PP	Protection Profile
RNG	Random Number Generator
RSA	Rivest, Shamir, Addleman
SDK	Software Development Kit
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SLA	Service Level Agreement
SOF	Strength of Function
SSH	Secure Shell
SVT	Scenario Verification Testing
TDES	Triple DES
TOE	Target of Evaluation

UI	User Interface

14. References

[1] FIPS 140-2 Standard,**Error! Hyperlink reference not valid.**

[2] FIPS 140-2 Implementation Guidance, http://csrc.nist.gov/groups/STM/cmvp/standards.html

[3] FIPS 140-2 Derived Test Requirements, http://csrc.nist.gov/groups/STM/cmvp/standards.html

[4] FIPS 197 Advanced Encryption Standard, http://csrc.nist.gov/publications/PubsFIPS.html

[5] FIPS 180-3 Secure Hash Standard, http://csrc.nist.gov/publications/PubsFIPS.html

[6] FIPS 198-1 The Keyed-Hash Message Authentication Code (HMAC), http://csrc.nist.gov/publications/PubsFIPS.html

[7] FIPS 186-3 Digital Signature Standard (DSS), http://csrc.nist.gov/publications/PubsFIPS.html