Samsung FIPS BC for Mobile Phone and Tablet

FIPS 140-2 Security Policy

Version 1.6

Last Update: 2014-02-11

Trademarks

The Android™ name is property of Google Inc.

The Samsung™ name, "Samsung.com" and "Samsung DIGITall Everyone's invited" are trademarks of Samsung in the United States, other countries, or both. Unauthorized use or duplication of these marks is strictly prohibited by law.

1. Introduction

This document is the non-proprietary FIPS 140-2 Security Policy for 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 CHANGE NOTICES (12-03-2002)) for a Security Level 1 multi-chip standalone software cryptographic module. This security policy, v1.6, is for the revalidation of the Samsung FIPS BC for Mobile Phone and Tablet to include information about new test platforms and minor code changes. The security policy from the previous validated version of the module can be found on CMVP validation website under certificate #1985.

1.1. Purpose of the Security Policy

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

- To provide a specification of the cryptographic security that will allow individuals and organizations to determine whether a cryptographic module, as implemented, satisfies a stated security policy.
- To describe to individuals and organizations the capabilities, protection, and access rights provided by the cryptographic module, thereby allowing an assessment of whether the module will adequately serve the individual or organizational security requirements.

1.2. Target Audience

This document has the following audience:

- Those specifying cryptographic modules
- Administrators of the cryptographic module(s)
- Users of the cryptographic module(s)

2. Cryptographic Module Specification

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 (versions: SBC1.45_2.0 on Galaxy Note II, SBC1.45_2.1 on Galaxy S4) 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/ Name and Version	Device	O/S & Ver.
Samsung FIPS BC for Mobile Phone and Tablet (SBC1.45_2.0)	Samsung Galaxy Note II	Android Jelly Bean 4.1
Samsung FIPS BC for Mobile Phone and Tablet (SBC1.45_2.1)	Samsung Galaxy S4	Android Jelly Bean 4.2

Table 2: Tested Platform

2.2. Description of Approved Mode

The module implements a FIPS mode and a non-FIPS mode. The power-up self-test function is called in the constructor of Bouncycastle Provider which automatically performs a set of power up self-tests including the integrity test and the Known-Answer Tests for all Approved algorithms when the module is initialized. On successful completion of self-tests, the module enters in **non-FIPS mode** of operation by default. All services are available in the non-FIPS mode.

In order to transition to the FIPS mode of operation, it needs to invoke setFIPSMode (true) which calls the power-up self-test function again. On successful completion of self-tests, the module will enter the FIPS mode of operation.

In the FIPS-approved mode of operation 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 (2048-, 3072- and 4096-bit modulus size for Key generation and Signature Generation with all SHAs except SHA-1; and 1024- to 4096-bit modulus size for Signature Verification with all acceptable SHAs)
- DSA (1024 bits Domain Parameter Verification and Signature Verification)

Caveat 1: Per the SP 800-131A Transition on 2014-01-01, the key lengths providing less than 112 bits of security strength are disallowed. Any use of the non-Approved key size will cause the module to operate in ***the non-FIPS approved mode implicitly****.*

Please see Table 5, “Services” in Section 4.2 for details and the CAVP certificate numbers.

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

- DSA (1024 bits Domain Parameter Generation, Key Generation and Signature Generation)
- RSA (1024- and 1536-bit modulus size for Key Generation and Signature Generation, 2048- to 4096-bit modulus size with SHA-1 for Signature Generation)
- Blowfish
- Camellia
- Camellia Light
- CAST 5/ CAST 128
- CAST 6/ CAST 256
- DES
- GOST 28147-89
- IDEA
- Integrated Encryption Scheme(IES)
- Rijndael
- RC2
- RC4
- RC5
- RC6
- SEED
- Serpent
- Tiny Encryption Algorithm(TEA)

- Twofish
- Extended Tiny Encryption Algorithm (XTEA)
- Grain 218
- Grain V1
- HC 128
- HC 256
- ISAAC
- Salsa 20
- VMPC
- Elgamal
- Naccache-Stern
- MD2
- MD4
- MD5
- RIPEMD-128
- RIPEMD-160
- RIPEMD-256
- RIPEMD-320
- Tiger
- Whirlpool
- GOST 3411
- ISO9797
- HMAC (based on RFC 2104)
- VMPC-MAC
- Secure Remote Password Protocol SRP 6
- Elliptic Curve Menezes-Qu-Vanstone (ECMQV)
- Digest random generator (non-approved RNG)
- VMPC random number generator (non-approved RNG)
- Thread-based seed generator (non-approved RNG)
- Reverse window generator (non-approved RNG)

The above four non-approved random number generators will not be available to the user via the Bouncycastle Provider in the FIPS mode.

The module implements the following FIPS approved algorithms/protocol which are not CAVS tested, and shall not be used in the FIPS 140-2 approved mode of operation:

- AES light (non-compliant)
- ECDSA (non-compliant)
- AES CMAC (non-compliant)

- Triple-DES-CMAC (non-compliant)
- Skipjack (non-compliant)
- Diffie-Hellman (non-compliant)
- EC Diffie-Hellman (non-compliant)
- TLS 1.0 (non-compliant)

If the User uses any of the non-complaint or non-approved algorithms/protocol listed above, the module will operate in **the non-FIPS approved mode implicitly**.

2.3. Cryptographic Module Boundary

2.3.1. Software Block Diagram

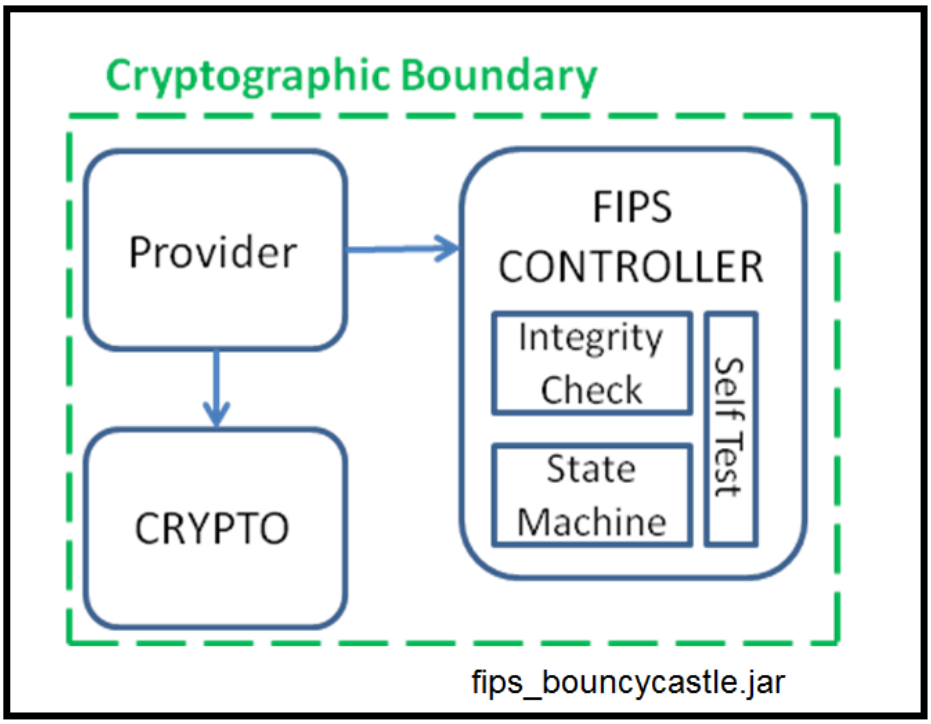

Figure 1: Software Block Diagram

The executable for the Samsung FIPS BC for Mobile Phone and Tablet is fips_bouncycastle.jar

Related documentation:

- Bouncycastle FIPS certification High Level Design (Bouncycastle_FIPS_HLD.doc) version 1.4
- Samsung FIPS BC for Mobile Phone and Tablet (Samsung_Bouncycastle_SPv1.6.doc)

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.

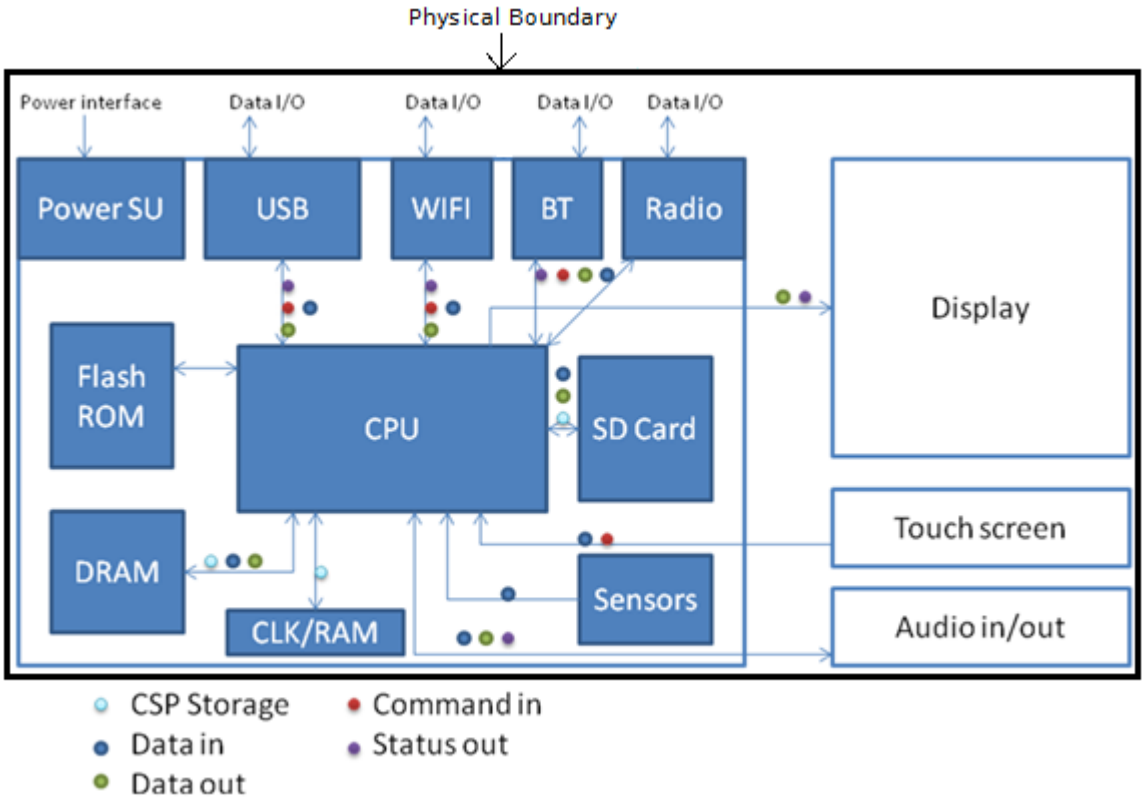

Figure 2: Hardware Block Diagram

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	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	Initialization of Module, Encryption, Decryption, Random Numbers, Digest Creation, Key Generation, Signature Generation, Signature Verification, Zeroization
Crypto Officer	Configuration, Initialization of Module, Encryption, Decryption, Random Numbers, Digest Creation, Key Generation, Signature Generation, Signature Verification, Zeroization

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.

4.2. Services

The following table identifies the FIPS-approved services:

Role	Service (Description)	Standard	CSP	Modes	FIPS Approved (Cert #)	Access (Read, Write, Execute)
User, Crypto Officer	AES (encryption and decryption)	FIPS 197	128, 192, 256 bit keys	ECB, CBC, CFB, OFB	Cert # 2353, 2409	R, W, EX
User, Crypto Officer	HMAC with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 (key message digest)	FIPS 198	At least 112 bits HMAC Key	N/A	Cert # 1459, 1494	R, W, EX
User, Crypto Officer	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 (message digest creation)	FIPS 180-4	N/A	N/A	Cert # 2027, 2067	R, W, EX
User, Crypto Officer	Triple-DES (encryption/ decryption)	SP 800-67	2 Key & 3 Key	CBC, ECB, CFB, OFB	Cert # 1472, 1499	R, W, EX

Role	Service (Description)	Standard	CSP	Modes	FIPS Approved (Cert #)	Access (Read, Write, Execute)
User, Crypto Officer	RSA (key generation)	FIPS 186-2	2048, 3072, 4096 bit keys	N/A	Cert # 1213, 1243	R, W, EX
User, Crypto Officer	RSA (signature generation) with SHA-224, SHA-256, SHA-384, and SHA-512	FIPS 186-2	2048, 3072, 4096 bit keys	N/A	Cert # 1213, 1243	R, W, EX
User, Crypto Officer	RSA (signature verification) with SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512	FIPS 186-2	1024, 1536, 2048, 3072, 4096 bit keys	N/A	Cert # 1213, 1243	R, W, EX
User, Crypto Officer	DSA (domain parameter verification)	FIPS 186-2	1024 bit keys	N/A	Cert # 736, 751	R, W, EX
User, Crypto Officer	DSA (signature verification)	FIPS 186-2	1024 bit keys	N/A	Cert # 736, 751	R, W, EX
User, Crypto Officer	Random Number Generator	ANSI X9.31	Seed, Seed Key	AES-128, AES-192, AES-256	Cert # 1172, 1189	R, W, EX
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

The following table identifies the non- FIPS-approved services:

Role	Service (Description)	CSP	Access (Read, Write, Execute)
Block Ciphers			
User, Crypto Officer	RSA (key generation)	1024, 1536 bits modulus size	R, W, EX
User, Crypto Officer	RSA (signature generation)	1024, 1536 bits modulus size, or using SHA-1 for any modulus size	R, W, EX
User, Crypto Officer	DSA (domain parameter generation, key generation and signature generation)	1024 bits modulus size	R, W, EX
User, Crypto Officer	AES Light (encryption and decryption)	128, 192, 256 bit Symmetric keys	R, W, EX
User, Crypto Officer	Blowfish (encryption and decryption)	variable key length from 32 bits up to 448 bits	R, W, EX
User, Crypto Officer	Camellia (encryption and decryption)	128, 192, 256 bit Symmetric keys	R, W, EX
User, Crypto Officer	Camellia Light (encryption and decryption)	128, 192, 256 bit Symmetric keys	R, W, EX
User, Crypto Officer	Cast 5/CAST 128 (encryption and decryption)	Key size of between 40 to 128 bits (but only in 8-bit increments)	R, W, EX
User, Crypto Officer	CAST 6/CAST 256 (encryption and decryption)	128, 160, 192, 224 or 256 bits Symmetric key	R, W, EX
User, Crypto Officer	DES (encryption and decryption)	56 bits Symmetric key	R, W, EX
User, Crypto Officer	GOST 28147-89 (encryption and decryption)	256 bits Symmetric key	R, W, EX
User, Crypto Officer	International Data Encryption Algorithm(IDEA) (encryption and decryption)	128 bit Symmetric key	R, W, EX
User, Crypto Officer	Integrated Encryption Scheme(IES) (hybrid encryption scheme used for MAC, KDF and encryption and decryption,)	Symmetric key, asymmetric key pair	R, W, EX
User, Crypto Officer	Rijndael (encryption and decryption)	Variable key sizes between 128 -256 bits (in the multiples of 32 bits) Symmetric key	R, W, EX
User, Crypto Officer	RC2 (encryption and decryption)	Variable key size	R, W, EX
User, Crypto Officer	RC5 (encryption and decryption)	Variable key size	R, W, EX
User, Crypto	RC6	128, 192 or 256 bits	R, W, EX

Role	Service (Description)	CSP	Access (Read, Write, Execute)
Officer	(encryption and decryption)	Symmetric key	
User, Crypto Officer	SEED (encryption and decryption)	128 bit Symmetric key	R, W, EX
User, Crypto Officer	Serpent (encryption and decryption)	128, 192 or 256 bits Symmetric key	R, W, EX
User, Crypto Officer	Skipjack (non-complaint) (encryption and decryption)	80 bits Symmetric key	R, W, EX
User, Crypto Officer	Tiny Encryption Algorithm (TEA) (encryption and decryption)	128 bits Symmetric key	R, W, EX
User, Crypto Officer	Twofish (encryption and decryption)	128, 192 or 256 bits Symmetric key	R, W, EX
User, Crypto Officer	Extended Tiny Encryption Algorithm (XTEA) (encryption and decryption)	128 bits Symmetric key	R, W, EX
Steam Ciphers			
User, Crypto Officer	Grain 128 (encryption and decryption)	128 bits key	R, W, EX
User, Crypto Officer	Gain V1 (encryption and decryption)	80 bits key	R, W, EX
User, Crypto Officer	HC 128 (encryption and decryption)	128 bits key	R, W, EX
User, Crypto Officer	HC 256 (encryption and decryption)	256 bits key	R, W, EX
User, Crypto Officer	ISAAC (encryption and decryption)	Variable key size	R, W, EX
User, Crypto Officer	RC4 (encryption and decryption)	Variable key size between 40 and 256 bits	R, W, EX
User, Crypto Officer	Salsa20 (encryption and decryption)	128, 256 bits key	R, W, EX
User, Crypto Officer	VMPC (encryption and decryption)	Up to 512 bits key	R, W, EX
Asymmetric Ciphers			
User, Crypto Officer	Elgamal (asymmetric encryption block cipher)	Asymmetric key pair	R, W, EX
User, Crypto Officer	Naccache–Stern	Asymmetric key pair	R, W, EX
User, Crypto Officer	ECDSA (non-complaint)	Asymmetric key pair	R, W, EX
Message Digest/Message Authentication Code (MAC)			

Role	Service (Description)	CSP	Access (Read, Write, Execute)
User, Crypto Officer	MD2 (cryptographic hash)	N/A	R, W, EX
User, Crypto Officer	MD4 (cryptographic hash)	N/A	R, W, EX
User, Crypto Officer	MD5 (cryptographic hash)	N/A	R, W, EX
User, Crypto Officer	RIPEMD-128 (cryptographic hash)	N/A	R, W, EX
User, Crypto Officer	RIPEMD-160 (cryptographic hash)	N/A	R, W, EX
User, Crypto Officer	RIPEMD-256 (cryptographic hash)	N/A	R, W, EX
User, Crypto Officer	RIPEMD-320 (cryptographic hash)	N/A	R, W, EX
User, Crypto Officer	Tiger (cryptographic hash)	N/A	R, W, EX
User, Crypto Officer	Whirlpool (cryptographic hash)	N/A	R, W, EX
User, Crypto Officer	CMAC (Message Authentication Code)	Symmetric key	R, W, EX
User, Crypto Officer	GOST 3411 (Message Authentication Code)	256 bits Symmetric key	R, W, EX
User, Crypto Officer	ISO9797 (DES based CBC Block Cipher Message Authentication Code according to ISO9797, algorithm 3 (ANSI X9.19 * Retail MAC))	Symmetric key	R, W, EX
User, Crypto Officer	HMAC (Keyed Hashed Message Authentication Code implementation based on RFC2104)	HMAC key	R, W, EX
User, Crypto Officer	VMPC-MAC (Message Authentication Code)	VMPC key	R, W, EX
Key Establishment			
User, Crypto Officer	Diffie-Hellman (non-compliant) (Key Agreement)	Asymmetric key pair, secret key	R, W, EX
User, Crypto Officer	EC Diffie-Hellman (non-compliant)	Asymmetric key pair, secret key	R, W, EX

Role	Service (Description)	CSP	Access (Read, Write, Execute)
	(Key Agreement)		
User, Crypto Officer	Secure Remote password Protocol SRP6 (password-authenticated Key Agreement)	Asymmetric key pair, secret key, password	R, W, EX
User, Crypto Officer	Elliptic Curve Menezes–Qu–Vanstone (ECMQV) (Key Agreement)	Asymmetric key pair, secret key	R, W, EX
User, Crypto Officer	RC2 (Key Wrapping)	Variable RC2 key size, secret key	R, W, EX
User, Crypto Officer	AES (non-compliant) (Key Wrapping)	128, 192, 256 bits AES key, secret key	R, W, EX
User, Crypto Officer	TLS 1.0 protocol (non-compliant)	Asymmetric key pair, secret key	R, W, EX
Non-approved RNGs (The following RNGs are available only in Non-approved mode)			
User, Crypto Officer	Digest Random Generator (Non-approved RNG)	Seed	R, W, EX
User, Crypto Officer	Thread-based seed generator (Non-approved RNG)	Seed	R, W, EX
User, Crypto Officer	Reserve window generator (Non-approved RNG)	Seed	R, W, EX
User, Crypto Officer	VMPC Random Number Generator	Seed	R, W, EX

Table 6: Non-Approved 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. Physical Security

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

6. Operational Environment

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

6.1. 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.

7. Cryptographic Key Management

7.1. Random Number Generation

The module employs an Approved ANSI X9.31 compliant random number generator for the creation of symmetric keys and HMAC keys as well as for the inputs to the FIPS 186-2 compliant RSA key generation algorithms.

Caveat 2: Per the algorithms and key sizes transition standard SP 800-131A, RSA/DSA key pairs with less than 112 bits of security strength are not allowed to be generated in the FIPS mode.

Note: the RNG seed is the tuple {V key DT}, where those values are defined in ANSI X9.31 Appendix A.2.4.

The ANSI X9.31 RNG is seeded and keyed using data from /dev/random which is provided by the underlying Operating System. 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 module performs continual tests on the random numbers it uses, to ensure that the seed and seed key input to the ANSI X9.31 RNG do not have the same value. The module also performs continuous random number generation test on the output of the ANSI X9.31 RNG to ensure that consecutive random numbers do not repeat.

Caveat 3: The module generates cryptographic keys whose strengths are modified by available entropy.

7.2. Key Entry and Output

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

7.3. Key Storage

No keys are stored in the cryptographic module.

7.4. Zeroization Procedure

The zeroization mechanism is implemented using the reset API.

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 zeroized after use.

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

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

This module was tested on both hardware test platforms, Samsung Galaxy Note II and Samsung Galaxy S4, which are compliant with FCC 47 Code of Regulations, part 15, Subpart B, Unintentional Radiators, Digital Devices, Class B.

9. Self Tests

As per FIPS 140-2 requirements, self tests must be conducted during initialization of the module and before the module becomes usable. Whenever an application invokes the module, a set of power-up self-tests executes automatically without any operator intervention. If any of the self tests fails, the module enters an error state. The error is indicated on the UI of the module or calling application.

Self test consists of the following tests:

9.1. Power-Up Self-tests

9.1.1. Cryptographic algorithm tests (Known Answer Tests)

A 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 Samsung FIPS BC for Mobile Phone and Tablet in FIPS mode.

Algorithm	**Test**
AES (encryption/decryption tested separately)	KAT
Triple-DES (encryption/decryption tested separately)	KAT
RSA (signature generation/verification tested separately)	KAT
DSA	Pair-wise consistency test
ANSI X9.31RNG	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

9.1.2. Integrity Check

Integrity tests ensure that the Samsung FIPS BC for Mobile Phone and Tablet module is same as that which was validated for FIPS compliance. This prevents malicious code to perform masquerading attacks, by replacing the FIPS 140-2 validated Samsung FIPS BC for Mobile Phone and Tablet module with another tainted implementation of Bouncycastle.

A HMAC-SHA-256 digest is calculated for the fips_bouncycastle.jar file that is the target of FIPS 140-2 validation and certification. This pre-calculated digest is appended to the fips_bouncycastle.jar file.

During power up, a new HMAC-SHA-256 digest will be calculated for the fips_bouncycastle.jar file to be loaded for use, using the approved HMAC-SHA-256 algorithm implemented within the module. The newly calculated digest will then be compared against the pre-calculated digest.

If the two digests match, it shows that fips_bouncycastle.jar has not been modified and is the one that was FIPS 140-2 validated.

9.2. Conditional Tests

9.2.1. Continuous Random Number Generator (RNG) Test

The module currently uses RNG based on ANSI X9.31 for all random number requirements. The RNG 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. Continuous Random Number Generation Test is implemented in the random number generation function to check the output of the random value generated by ANSI X9.31 RNG.

9.2.2. Pair-wise Consistency Test

The module performs a pair-wise consistency test whenever asymmetric key generation service for RSA/DSA is requested.

10. Design Assurance

10.1. Configuration Management

All source code is maintained in internal source code servers and the tools, Subversion and Mercurial, are used as code control. Release is based on the Change List number maintained by Subversion and Mercurial, 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 Subversion (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.

10.2. 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, Jelly Bean 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). The OTA 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.

10.3. User and Crypto Officer Guidance

Applications can get access to the crypto module by requesting a new instance of the Bouncycastle Provider.

A valid provider, with all services available, is returned only if all the self tests pass successfully. Otherwise, the module goes into error state and none of the services are available for use.

Applications must call setFIPSMode(true) for the provider to transition the module into FIPS mode.

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

This is how the Bouncycastle provider is accessible through the standard Java JCE interface.

11. Mitigation of Other Attacks

There is no description for the mitigation of other attacks.

12. Glossary and Abbreviations

AES	Advanced Encryption Specification
ANSI	American National Standards Institute
API	Application Programming Interface
ARM	Advanced RISC Machines
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CMT	Cryptographic Module Testing
CMVP	Cryptographic Module Validation Program
CRNG	Congruential Random Number Generator
CSP	Critical Security Parameter
CVT	Component Verification Testing
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interface
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standard
FSM	Finite State Model
HMAC	Hash Message Authentication Code
IES	Integrated Encryption Scheme
JCE	Java Cryptography Extension
KAT	Known Answer Test
MAC	Message Authentication Code
NDRNG	Non-Deterministic Random Number Generator
NIST	National Institute of Science and Technology
NVLAP	National Voluntary Laboratory Accreditation Program
O/S	Operating System
OTA	Over-The-Air
RNG	Random Number Generator
RSA	Rivest, Shamir, Addleman

SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SLA	Service Level Agreement
SSH	Secure Shell
SVN	Subversion
TDES	Triple Data Encryption Standard
TEA	Tiny Encryption Algorithm
TLS	Transport Layer Security
VMPC	Variably Modified Permutation Composition

13. References

[1] FIPS 140-2 Standard, http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

[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 140-2 Annex A: Approved Security Functions, http://csrc.nist.gov/publications/fips/fips140-2/fips1402annexa.pdf

[5] FIPS 140-2 Annex C: Approved Random Number Generators, http://csrc.nist.gov/publications/fips/fips140-2/fips1402annexc.pdf

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

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

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

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

[10] NIST Recommendation for Block Cipher Modes of Operation, http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

[11] NIST Special Publication 800-131A, Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths, http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf