BCM58200 Series: BCM58201, BCM58202 Cryptographic Module *Non-Proprietary Security Policy*

Document *Version 0.3*

Broadcom Ltd.

Revision Date: 2019-04-16

TABLE OF CONTENTS

1. MODULE OVERVIEW 3

2. SECURITY LEVEL 5

3. MODES OF OPERATION 6

4. PORTS AND INTERFACES 8

5. IDENTIFICATION AND AUTHENTICATION POLICY 10

6. ACCESS CONTROL POLICY 11

DEFINITION OF SERVICES 11
DEFINITION OF CRITICAL SECURITY PARAMETERS (CSPS) 16
DEFINITION OF CSPS MODES OF ACCESS 22

7. OPERATIONAL ENVIRONMENT 25

8. SECURITY RULES 25

9. PHYSICAL SECURITY POLICY 28

PHYSICAL SECURITY MECHANISMS 28

10. MITIGATION OF OTHER ATTACKS POLICY 28

11. REFERENCES 28

12. DEFINITIONS AND ACRONYMS 29

1. Module Overview

The BCM58200 Series: BCM58201, BCM58202 Cryptographic Module, a single-chip encased in hard opaque tamper evident IC packaging, is a highly integrated system on a chip. It is marketed in two part numbers.

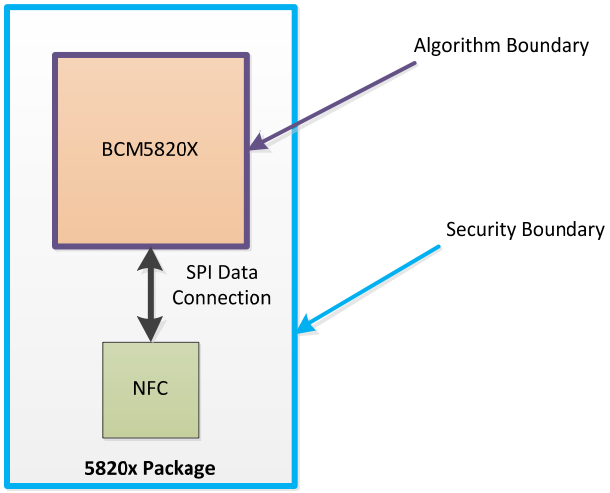
- BCM58201A0KFBG : integrated system on a chip with no NFC capabilities
- BCM58202PA0KFBG : integrated system on a chip with NFC capabilities

All devices use the same physical package.

The module runs firmware version `8bc25ceb540a57ed8fdbb2104b6751c6b1d0450f of June 27, 2018`

Figure 1 shows that the BCM58200 Series is composed of two components, the BCM5820X component and the optional NFC component. These modules are interconnected with a SPI (Serial Peripheral Interface bus) connection. The NFC component is purely a peripheral block to BCM5820X for NFC communication. No cryptographic implementation is included in this component; all cryptographic capabilities are encapsulated in the BCM5820X component. The interconnect between BCM5820X and NFC is for data communication only, no cryptographic material or key is passed between the modules.

Figure 1 - BCM58200 Top Level Blocks



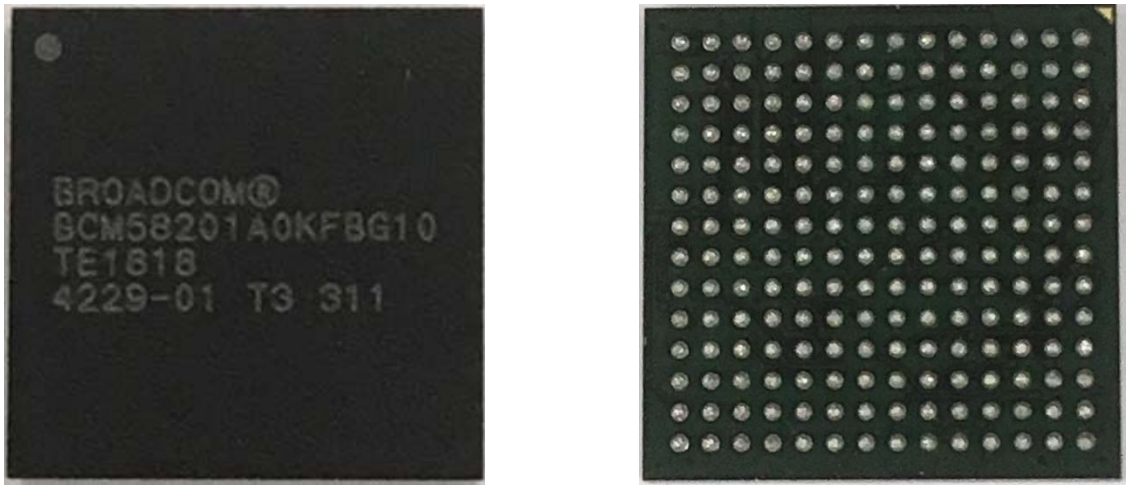
For the purpose of FIPS140-2 validation, the physical boundary of the chip is used as the security boundary of the cryptographic module.
The BCM58200 Series Cryptographic Module's FIPS boundary is defined as:

- The external surface of the BCM58200 chip including the hard opaque encapsulating material that physically protects all module components.

The algorithm boundary is defined as the BCM5820X component.

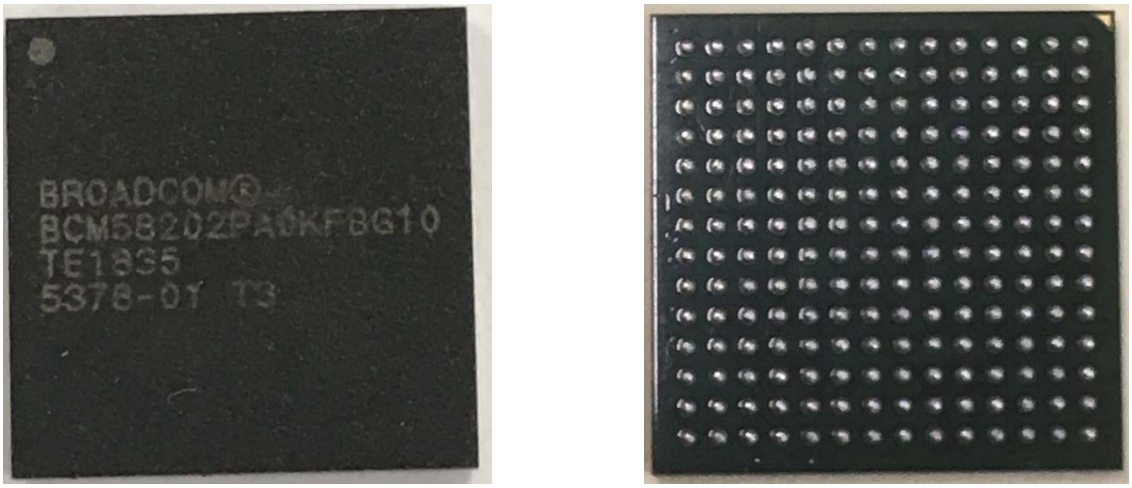
The figures below illustrate the cryptographic module's physical boundary, interfaces, and logical software execution contexts within the physical boundary.

Figure 2 - Pictures of the Cryptographic Module Physical Boundary: BCM58201



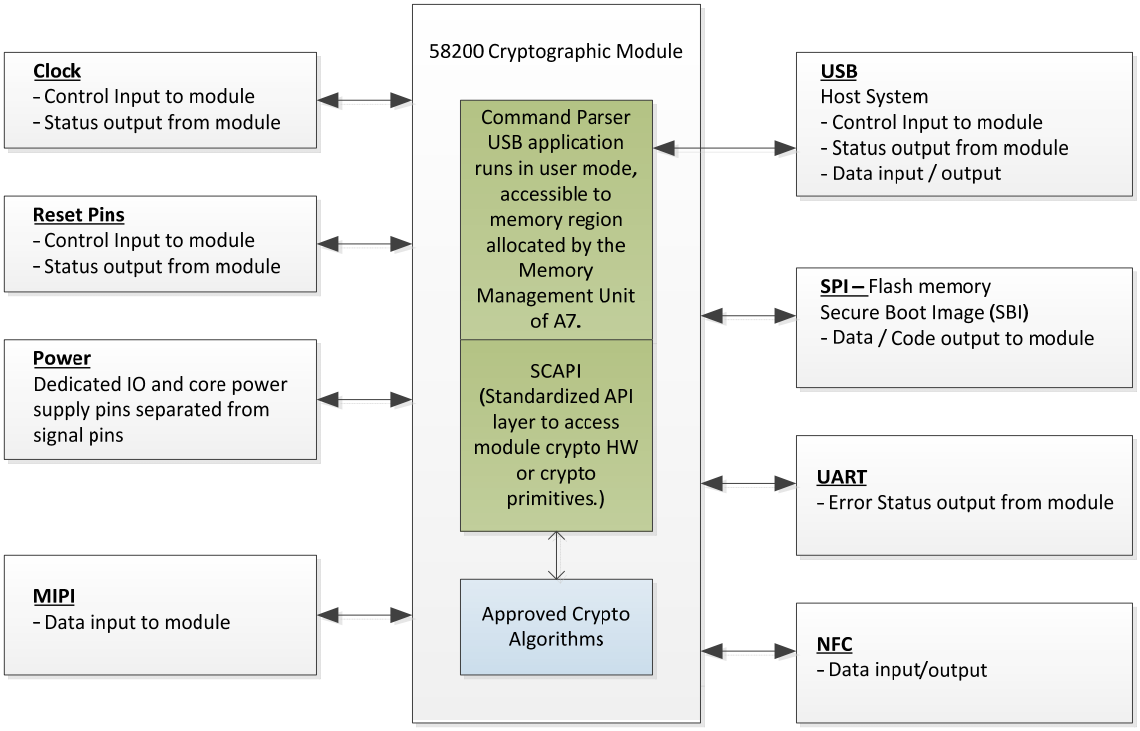
(Top) **(Bottom)**

Figure 3 - Pictures of the Cryptographic Module Physical Boundary: BCM58202



(Top) **(Bottom)**

Figure 44 - Block Diagram of module Interfaces & logical Software Execution Contexts



2. Security Level

The cryptographic module meets the overall requirements applicable to Level 3 security of FIPS 140-2.

Table 1 - Module Security Level Specification

Security Requirements Section	Level
Cryptographic Module Specification	3
Module Ports and Interfaces	3
Roles, Services and Authentication	3
Finite State Model	3
Physical Security	3
Operational Environment	N/A
Cryptographic Key Management	3

EMI/EMC	3
Self-Tests	3
Design Assurance	3
Mitigation of Other Attacks	NA

3. Modes of Operation

FIPS Approved mode of operation

The BCM58200 Series cryptographic module supports a single FIPS Approved mode of operation. The user can determine that the cryptographic module is running in FIPS Approved mode of operation when the status output RESET_OUT_L is high. The module does not support a non-Approved mode of operation.

Approved Algorithms

The module implements the following approved and allowed cryptographic algorithms using a hardware crypto engine called [SMAU - Crypto/Auth] block. The same hardware block is used twice in the Secure Memory Access Unit or SMAU. One instance is being used for offloading generic cryptographic operations. The other instance is being used to support secure caching of instruction and data stored externally in encrypted and integrity-protected format. Individual self-tests are conducted after power-on to test the instantiation for generic cryptographic operations. Each algorithm implementation is used during different scenarios. They are never used simultaneously for the same operation. Each algorithm implementation has its own algorithm certificate and has its own power-on Self-test.

Table 2 – Approved Algorithms

Cryptographic Algorithm	Description	Certificate Number
AES	[SMAU – Crypto/Auth] block, ECB, CBC, CTR Encryption and decryption Key size : 128, 192, 256	5895
AES CCM	[SMAU – Crypto/Auth] block Encryption and decryption Key size : 128, Nonce Len 12, Tag Len 4, 8, 12, 16	5896

	Note: Key sizes 192 and 256 were tested but are not implemented.	
Cryptographic Key Generation based on DRBG output	DRBG output is used for the following keys. 256bit ephemeral key pair for ECDH session; output of DRBG used as key. 128/192/256bit AES key for host requested services. 160/256bit HMAC key for host requested service Keys are generated according to section 5 of SP 800-133 All keys are from an unmodified output.	Vendor Affirmed
DRBG	SP800-90A DRBG, SHA-256 is the HASH functions used	2452
DSA	Signature generation, signature verification 2048-bit key, with SHA-256 for signature generation	1486
ECDSA	Signature verification 256-bit key, curve P-256	1593
ECDSA	Component signature generation 256-bit key, curve P-256	C 222
HMAC- SHA256	[SMAU – Crypto/Auth] block	3870
KTS	SP800-38F – Key Transport based on AES-CCM with 128bit of security Key establishment methodology provides 128 bits of encryption strength	5896
RSA	Signature generation, signature verification 2048-bit key with SHA-256	3087
SHA-3	[SMAU – Crypto/Auth] block	60

	Digest size : 224bit, 256bit, 384bit, 512bit	
SHA256	[SMAU – Crypto/Auth] block	4646

Allowed Non-Approved Algorithms

The module implements the following non-approved cryptographic algorithms

EC Diffie-Hellman: Non-SP800-56A Compliant ECDH allowed per FIPS 140-2 Implementation Guidance D.8 (key agreement; key establishment methodology provides 128-bits of encryption strength)

NDRNG: Internal module source utilizing free running oscillators to capture thermal noise as the source of randomness. The NDRNG is used to collect entropy to be fed to the FIPS SP800-90A DRBG. The module claims a key security strength of 128-bits from the output of the NDRNG. The module generates cryptographic keys whose strengths are modified by available entropy.

4. Ports and Interfaces

The BCM58200 Series Cryptographic Module provides physical ports as listed in Table 3 below.

Table 3 – Physical Ports

Note: the BCM5820X chip has a total of 141 signal pins. Each BCM5820X Interface Group listed in Table 3 contains several BCM58200 pins. Unused Interface Groups will be marked as "Non-Available" because they are currently disabled by the cryptographic module.

Clock group	Control Input Status Output	Clock - 26MHz clock - 32KHz clock Clock output - 26MHz clock output
Reset group	Control input Status output	One reset input Reset output: Indicates that system power supply is stable.
Secure boot	Control Input Status Output	- one key zeroization request input (MANU_DEBUG) - Ten external tamper detection (e.g., can be hooked up to a temperature sensor or a voltage sensor. No claims made for FIPS mode).

		- One ERROR status.
SPI group: All Code/Data Input is authenticated by the module.	Data input (code and data)	Code and data from SPI flash (clock, device select, and four data I/O)
USB group: Device interface used by the module's operators to make service requests. Requests are authenticated via the ECDH secure session.	Data input Data output Control input Status output	Service request input Service response output (USB differential data bus)
UART group: UART0 port is enabled as error status output. Other UART ports are disabled and logic is put in reset state.	Status output Other UART ports are intended for future use: Data input Data output	Status output (Four UART ports of four signals each.) Intended use in the future: Data received or transmitted for UART console application
Static Memory Interface group: Clock to the group block is disabled and logic is put in reset state.	Non-available Intended use in the future: Data input Data output	Non-available (chip select, read/write control, 8 data bit, 20 address bit) Intended use in the future : Code and data from SRAM or flash memory.
NFC group	Non-available; BCM58202: Data input Data output	(Four antenna connections, two SWP interface ports.): Data received or transmitted for contactless smart card applications.
Smart Card group: Clock to the group block is disabled and logic is put in reset state.	Non-available Intended use in the future: Data input Data output	Non-available (Seven interface signals) Intended use in the future: Data received or transmitted For contacted Smart Card applications.
MIPI group	Non-available; BCM58201: Data input Data output	(One differential pair for data, one differential pair for clock.): Data and clock signals for connecting to external CSI-2 compliant camera.
SDIO group	Non-available; Data input Data output	(Clock, reset, status, command, 8 data bit.): SDIO interface signals
JTAG group: Completely disabled by HW in FIPS mode. Module HW\FW\SW	Non-available	Non-available

enforces that non-volatile plaintext critical security parameters cannot be shared, used, or viewed in FIPS mode.		
Power group	Power is distributed to the chip using designated IO and core power pins that are completely separated from any signal pin groups. Power pins are only connected to the internal power planes of the silicon chip.	Over 50 power and ground pins.

5. Identification and Authentication Policy

Assumption of roles

The BCM58200 Series Cryptographic Module supports two operator roles, User and Cryptographic-Officer. Only the authorized user (in either role) could establish a secure session with the cryptographic module. The module is designed to operate with a single entity that is assigned the User and Cryptographic-Officer roles. The user identity is embedded in the module's SBI during manufacturing (Secure Boot Image: an authenticated software extension of the module's BOOT ROM. SBI software is part of the BCM58200 Series Cryptographic Module). The cryptographic module implements identity-based operator authentication to allow only the authorized user to access cryptographic services.

Authentication is accomplished via a 256-bit ECDSA-based signature verification process. A single 256-bit ECDSA public key is embedded in the SBI. The 256-bit ECDSA public key is used to authenticate the operator during the establishment of an ECDH secure session between the module and the operator on the external host system.

After an operator is authenticated successfully, the operator can assume either the role of the Cryptographic Officer or the role of the User. The module allows the operator to perform both CO and User services.

Table 4 - Roles and Required Identification and Authentication

Role	Type of Authentication	Authentication Data
User	Identity-based operator authentication	• 256-bit ECDSA signature verification

Cryptographic-Officer	Identity-based operator authentication	• 256-bit ECDSA signature verification

Table 5 - Strengths of Authentication Mechanisms

Authentication Mechanism	Strength of Mechanism
ECDSA Signature Verification (256 bit)	The probability that a random attempt will succeed or a false acceptance will occur is $1/2^{128}$ which is less than 1/1,000,000. The probability of successfully authenticating to the module within one minute is $3{,}750/2^{128}$ which is less than 1/100,000. The module will only allow one attempt to verify the operator – if that attempt fails the module will be in an error state and must be rebooted to try and become operational again. Please see Section "8. Security Rules" below (security rules imposed by the vendor) for the detail supporting this calculation.

6. Access Control Policy

Definition of Services

The cryptographic module supports the following authenticated services defined in Table 6:

Table 6 - Authenticated Services

Name of Service	Description of Service
Generate Key	This service generates an AES or HMAC key to be used during operator requested services.
AES Encrypt	This service encrypts bulk operator supplied data using a previously generated AES key.
AES Decrypt	This service decrypts bulk operator supplied data using a previously generated AES key.
SHA-256 Hashing	This service generates a SHA-256 digest on supplied data.

SHA-3 Hashing	This service generates a SHA-3 digest on supplied data. Digest lengths of 224, 256, 384, and 512 bits are supported.
Load Key	This service allows an operator to load a key into the module's key cache. The key being loaded can be a private key or a public key of an asymmetrical key pair, or a symmetrical key for AES or HMAC. All keys loaded via this service are being protected by the ECDH established session providing128-bit AES-CCM encryption and integrity protection.
RSA Signature Verification	This service performs RSA Signature Verification on operator supplied data with a previously loaded public key (see service "Load Key").
DSA Signature Verification	This service performs DSA Signature Verification on operator supplied data with a previously loaded public key (see service "Load Key").
ECDSA Signature Verification	This service performs ECDSA Signature Verification on operator supplied data with a previously loaded public key (see service "Load Key").
RSA Signature Generation	This service performs RSA Signature Generation on operator supplied data with a previously loaded private key (see service "Load Key").
DSA Signature Generation	This service performs DSA Signature Generation (2048 bit key) on operator supplied data with a previously loaded private key (see service "Load Key").
ECDSA Signature Generation	This service performs ECDSA Component Signature Generation on operator supplied data with a previously loaded private key (see service "Load Key").
Generate Random Number	This service generates a random number with the module's FIPS 800-90A DRBG and outputs the generated random number to the requesting operator.
EC Diffie-Hellman Key Exchange	This service is comprised of several steps which establish the AES-CCM key used for data encryption between the module and an external entity.
HMAC Request	Compute an HMAC on an operator supplied blob of data.

The cryptographic module supports the following unauthenticated services defined in Table 7:

Table 7 - Unauthenticated Services

Name of Service	Description of Service
Self-Test	This service executes the suite of self-tests required by FIPS 140-2. Self-tests are invoked by power cycling the module.
Show Status	This service provides the current status of the cryptographic module.
Get Info	This service computes and outputs the ECDSA device public key of the cryptographic module
Get Version	This service returns the version/revision information of the cryptographic module
Zeroize	• Power-cycle or hard reset will zeroize all volatile critical security parameters including internally generated CSPs or loaded keys. • When the MANU_DEBUG pin within the Secure Boot group physical interface is turned high all volatile and non-volatile plaintext critical security parameters will be zeroized – after this the module will not boot again.

Table 8 - Specification of Service Inputs & Outputs

Service	Control Input	Data Input	Data Output	Status Output
Generate Key	Key Type	N/A	Key Handle	Success/fail
AES Encrypt	Length Key Handle	Plaintext	Ciphertext	Success/fail
AES Decrypt	Length Key Handle	Ciphertext	Plaintext	Success/fail
SHA-256 Hashing	Hash Type	Data Blob	Digest	Success/fail
SHA-3Hashing	Hash Type	Data Blob	Digest	Success/fail
Load Key	Key Type Key Handle	Key	N/A	Success/fail
RSA Signature Verification	Hash Length Key Handle	Input is hashed, then a signature is generated for	N/A	Success/fail

Service	Control Input	Data Input	Data Output	Status Output
		validation.		
DSA Signature Verification	Hash Length Key Handle	Input is hashed, then a signature is generated for validation.	N/A	Success/fail
ECDSA Signature Verification	Hash Length Key Handle	Input message is the hashed message. Signature is generated using the inputs for validation.	N/A	Success/fail
RSA Signature Generation	Hash Length Key Handle	Input is hashed, then a signature is generated	Signature	Success/fail
DSA Signature Generation	Hash Length Key Handle	Input is hashed, then a signature is generated	Signature	Success/fail
ECDSA Signature Generation	Hash Length Key Handle	Input is the hashed message	Signature	Success/fail
Generate Random Number	DRBG Type Length	N/A	Random Number	Success/fail
ECDiffie-Hellman Key Exchange (comprised of two steps)	Header info.	ECDiffie-Hellman key establishment data received from Host System	ECDiffie-Hellman key establishment data sent to Host System	Success/fail
HMAC Request	Length Hash Type Key Handle	Data Blob	MAC	Success/fail
Self Test	N/A (Power cycle)	N/A	N/A	Success/fail
Show Status	N/A	N/A	N/A	All the above Status Output (Table–8 Specification of Service Inputs & Outputs) Status Output of Interface groups (Table 3 Physical Ports)

Service	Control Input	Data Input	Data Output	Status Output
Get Info	N/A	N/A	Cryptographic Module device public key $K_{DI\text{-}EC\text{-}PUB}$	Success/fail
Get Version	N/A	N/A	Version and Revision information of the Cryptographic Module	Success/fail
Zeroize	Power-cycle, hard reset, or set MANU_DEBUG pin	N/A	N/A	N/A

Definition of Critical Security Parameters (CSPs)

The following are the CSPs contained in the module.

Table 9 - Secret and Private Keys

Key	Description/Usage	Generation	Storage	Entry/Output	Destruction
$K_{ECDH-PRIV}$ 256 bit random number used for ephemeral ECDH key.	Used to establish an ECDH based session key.	Ephemeral key generated internally via DRBG per SP800-90A.	**Stored** in plaintext internally in the module's [Scratch RAM] block. **Key-to-entity association**: associated with a session ID during the ECDH secure session establishment.	**Entry***:* N/A **Entry Key-to-entity association**: N/A **Output***:* N/A **Output Key-to-entity association**: N/A.	Zeroize service. Additionally always destroyed after the symmetrical session key is established.
K_{AES} 128 bit AES key. A unique value for each module.	Used to encrypt and decrypt the Secure Boot Image (SBI) when the SBI is loaded (symmetrically).	Generated internally during manufacturing via DRBG per SP800-90A.	**Stored** in plaintext internally in OTP. When in use it is temporality copied to the [Scratch RAM] block. **Key-to-entity association**: Key index = 2 in OTP.	**Entry***:* N/A **Entry Key-to-entity association**: N/A **Output***:* N/A **Output Key-to-entity association**: N/A.	Zeroize service. Temporary copy in [Scratch RAM] block always destroyed after each reset cycle.
K_{HMAC} 256 bit HMAC-SHA-256 key. A unique value for each module.	Used to protect and verify the SBI.	Generated internally during manufacturing via DRBG per SP800-90A.	**Stored** in plaintext internally in OTP. When in use it is temporality copied to the [Scratch RAM] block. **Key-to-entity association**: Key index = 3 in OTP.	**Entry***:* N/A **Entry Key-to-entity association**: N/A **Output***:* N/A **Output Key-to-entity association**: N/A.	Zeroize service. Temporary copy in [Scratch RAM] block always destroyed after each reset cycle.

Key	Description/Usage	Generation	Storage	Entry/Output	Destruction
$K_{DI\text{-}EC\text{-}PRIV}$ 256 bit ECDSA private key. A unique value for each module.	Used to establish the mutually authenticated ECDH secure session communication channel between the module and an external entity. Used as the identity key of the module in these authenticated communications.	Generated internally during manufacturing via DRBG per SP800-90A.	**Stored** in plaintext internally in OTP. When in use it is temporality copied to the [Scratch RAM] block. **Key-to-entity association**: Key index = 4 in OTP.	**Entry***:* N/A **Entry Key-to-entity association**: N/A **Output***:* N/A **Output Key-to-entity association**: N/A.	Zeroize service. Temporary copy in [Scratch RAM] block always destroyed after each reset cycle.
$K_{APP\text{-}AES}$ 128, 192 or 256 bit AES keys.	Used to encrypt/decrypt application data when external applications issue encrypt or decrypt service requests.	Generated internally during operation via DRBG per SP800-90A. See Generate Key service.	**Stored** in the volatile "key cache" within the [Scratch RAM] block. **Key-to-entity association:** "key cache" handle. Note this handle is given by the application that requested the creation of the key so that application can request encryption/ decryption with the key at a later point in time.	**Entry***:* Entered into the module by Load Key service[1] **Entry Key-to-entity association**: Session key derived during the ECDiffie-Hellman Key Exchange service. **Output***:* N/A **Output Key-to-entity association**: N/A.	Zeroize service. Temporary copy in [Scratch RAM] block always destroyed after each reset cycle.

[1] 192 and 256-bit keys entered using the Load Key service only provide 128-bits of security strength.

Key	Description/Usage	Generation	Storage	Entry/Output	Destruction
$K_{APP-HMAC}$ 256 bit HMAC keys (SHA-256).	Used to protect and verify application data when external applications issue protection or verification service requests.	Generated internally during operation via DRBG per SP800-90A. See Generate Key service.	**Stored** in the volatile "key cache" within the [Scratch RAM] block. **Key-to-entity association**: "key cache" handle. Note this handle is given by the application that requested the creation of the key so that application can request protection/ verification with the key at a later point in time.	**Entry***:* Entered into the module by Load Key service **Entry Key-to-entity association**: Session key derived during the ECDiffie-Hellman Key Exchange service. **Output***:* N/A **Output Key-to-entity association**: N/A.	Zeroize service. Temporary copy in [Scratch RAM] block always destroyed after each reset cycle.
$K_{APP-PRIV}$ 2048 bit DSA 2048 bit RSA 256 bit ECDSA	Used to perform signature generation during the RSA, DSA or ECDSA Signature services.	N/A	Multiple instances. **Stored** in the volatile "key cache" within the [Scratch RAM] block. **Key-to-entity association**: "key cache" handle. Note this handle is given by the application that requested the entry of the key so that the application can request signature generation with the key at a later point in time.	**Entry***:* Entered into the module by Load Key service **Entry Key-to-entity association**: This is a private key that is associated with the public key member of a key-pair. **Output***:* N/A **Output Key-to-entity association**: N/A.	When the zeroize service is requested. Always destroyed after each reset cycle.

Key	Description/Usage	Generation	Storage	Entry/Output	Destruction
$K_{ECDH-SS}$ 256 bit ephemeral ECDH shared secret.	Used to derive the session key **Kss**	Derived using ECDH key exchange algorithm based on **$K_{ECDH-PRIV}$** and **$K_{ECDH-OP-PUB}$**	**Stored** only temporarily stored in the scratch RAM, erased after **Kss** is derived **Key-to-entity association**: associated with a session ID during the ECDH secure session establishment.	**Entry***:* N/A **Entry Key-to-entity association**: N/A **Output***:* N/A **Output Key-to-entity association**: N/A.	Zeroize service. Additionally always destroyed after the symmetrical session key is established.
Kss 128 bit AES-CCM key.	Session key derived during the ECDiffie-Hellman Key Exchange service. The module will use this key for secure communications to/from the external host system. This key is not loaded with the Load Key service.	Generated during the ECDiffie-Hellman Key Exchange service via SHA256-based KDF function. This is a part of the ECDH component.	**Stored** in the volatile "key cache" within the [Scratch RAM] block. **Key-to-entity association**: Only one session key exists at any given point in time.	**Entry***:* N/A **Entry Key-to-entity association**: N/A **Output***:* N/A **Output Key-to-entity association**: N/A.	Zeroize service. Temporary copy in [Scratch RAM] block always destroyed after each reset cycle.
DRBG Seed 20,000 bits	Entropy value fed to the SP800-90A.	Gathered from internal module NDRNG utilizing free running oscillators to capture thermal noise.	Generated via NDRNG and stored in DRBG registers **Key-to-entity association**: Only one DRBG seed key exists at any given point in time.	**Entry***:* N/A **Entry Key-to-entity association**: N/A **Output***:* N/A **Output Key-to-entity association**: N/A.	Reset DRBG or power cycle the chip.
DRBG State (values V and C)	State of the module's SP800-90A.	Generated within the module's SP800-90A DRBG.	**Stored** in DRBG registers. **Key-to-entity association**: The DRBG maintains one state at a given time.	**Entry***:* N/A **Entry Key-to-entity association**: N/A **Output***:* N/A **Output Key-to-entity association**: N/A.	Reset DRBG or power cycle the chip.

Definition of Public Keys:

The following are public keys contained in the module.

Table 10 - Public Keys

Key	Description/Usage	Generation	Storage	Entry/Output
$K_{DI-EC-PUB}$ 256 bit ECDSA public key. A unique value for each module.	Used by the operator to authenticate the cryptographic module in a mutually authenticated secure session	Computed internally upon each get_info request per ECDSA algorithm	**Stored** only stored temporarily in the scratch RAM during the processing of the get_info service **Key-to-entity association**: Public part of the device identity key.	**Entry***:* N/A **Entry Key-to-entity association**: N/A **Output***:* as the result of get_info service **Output Key-to-entity association**: embedded in the get_info command response.
$K_{ECDH-PUB}$ 256 bit public ephemeral ECDH key of the cryptographic module	Used to establish an ECDH based session.	Ephemeral public key generated internally for on non- SP800-56A compliant ECDH	**Stored** only stored temporarily in the scratch RAM during the process of establishing the ECDH session, erased after the session key is established **Key-to-entity association**: Public key of the ephemeral ECDH key pair.	**Entry***:* N/A **Entry Key-to-entity association**: N/A **Output***:* as the result of the ECDH key exchange **Output Key-to-entity association**: embedded in the command response for ECDH key exchange.

Key	Description/Usage	Generation	Storage	Entry/Output
$K_{ECDH-OP-PUB}$ 256 bit public ephemeral ECDH key of the operator	Used to establish an ECDH based session.	Ephemeral public key generated and signed by the operator, pass into the cryptographic module during ECDH session key exchange	**Stored** only stored temporarily in the scratch RAM during the process of establishing the ECDH session, erased after the session key is established **Key-to-entity association**: Associated with the authentication session. Only one session is active.	**Entry***:* input of the ECDH key exchange **Entry Key-to-entity association**: embedded in the command for ECDH key exchange. **Output***:* NA **Output Key-to-entity association**: NA
$K_{APP-PUB}$ 2048 bit DSA 2048 bit RSA 256 bit ECDSA	Used to perform signature verification during the RSA, DSA or ECDSA Signature Verification services.	N/A	Stored in the volatile "key cache" within the [Scratch RAM] block on the block diagram. **Key-to-entity association**: "key cache" handle. Note this handle is passed back to the application that requested the entry of the key so that the application can request signature verification with the key at a later point in time.	**Entry***:* Entered into the module by the Load Key service. **Entry Key-to-entity association**: This is a public key that is associated with the private key member of a key-pair. **Output***:* N/A **Output Key-to-entity association**: N/A.

Key	Description/Usage	Generation	Storage	Entry/Output
K_{OP-PUB} 256 bit ECDSA	Operator's public key Used to authenticate the operator during an ECDH secure session.	N/A	Stored in the on-chip RAM. **Key-to-entity association**: This key is located at a fixed offset of the SBI image known to the implementation of the cryptographic module.	**Entry***:* Embedded in the SBI during the manufacturing process. **Entry Key-to-entity association**: This is a public key that is associated with the private key member of a key-pair. **Output***:* N/A **Output Key-to-entity association**: N/A.

Definition of CSPs Modes of Access

Table 11 defines the relationship between access to CSPs and the different module services. The modes of access shown in the table are defined as:

- **G** = Generate: The module generates the CSP.
- **R** = Input / 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** = Output / Write: The module writes the CSP. The write access is typically performed after a CSP is imported into the module, or the module generates a CSP, or the module overwrites an existing CSP.
- **Z** = Zeroize: The module zeroizes the CSP.

Table 11 - CSP Access Rights within Roles & Services

Role		Service	Cryptographic Keys and CSPs Access Operation
C.O.	**User**		
X	X	Generate Key	**G** $K_{APP-AES}$ **G** $K_{APP-HMAC}$ **R** DRBG internal state For each service call a handle to the generated key will be passed back to the operator.

X	X	AES Encrypt	**R** $K_{APP-AES}$ **E** For each service request the operator will indicate which $K_{APP-AES}$ key to use by passing in the key's handle as input.
X	X	AES Decrypt	**R** $K_{APP-AES}$ **E** For each service request the operator will indicate which $K_{APP-AES}$ key to use by passing in the key's handle as input.
X	X	SHA-256 Hashing	**N/A**
X	X	SHA-3 Hashing	**N/A**
X	X	Load Key	**W** $K_{APP-PUB}$ **W** $K_{APP-PRIV}$ **W** $K_{APP-AES}$ **W** $K_{APP-HMAC}$ For each service request a handle to the loaded key will be passed back to the operator.
X	X	RSA Signature Verification	**R** $K_{APP-PUB}$ **E** For each service request the operator will indicate which $K_{APP-PUB}$ RSA key to use by passing in the key's handle as input.
X	X	DSA Signature Verification	**R** $K_{APP-PUB}$ **E** For each service request the operator will indicated which $K_{APP-PUB}$ DSA key to use by passing in the key's handle as input.
X	X	ECDSA Signature Verification	**R** $K_{APP-PUB}$ **E** For each service request the operator will indicated which $K_{APP-PUB}$ ECDSA key to use by passing in the key's handle as input.
X	X	RSA Signature Generation	**R** $K_{APP-PRIV}$ **E** For each service request the operator will indicated which $K_{APP-PRIV}$ RSA key to use by passing in the key's handle as input.
X	X	DSA Signature	**R** $K_{APP-PRIV}$

		Generation	**E** For each service request the operator will indicated which $K_{APP-PRIV}$ DSA key to use by passing in the key's handle as input.
X	X	ECDSA Signature Generation	**R** $K_{APP-PRIV}$ **E** For each service request the operator will indicated which $K_{APP-PRIV}$ ECDSA key to use by passing in the key's handle as input.
X	X	Generate Random Number	**R** DRBG Seed (note: a new Seed is generated for each call to service Generate Random Number). **R** DRBG Internal State **E** The DRBG is seeded with the Seed. The random number generated by the DRBG is returned to the operator requesting the service.
X	X	EC Diffie-Hellman Key Exchange	**R** $K_{DI-EC-PRIV}$ **R** $K_{ECDH-PRIV}$ **R** K_{OP-PUB} **G** $K_{ECDH-PUB}$ **R** $K_{ECDH-OP-PUB}$ **G** $K_{ECDH-SS}$ **G** Kss **R** DRBG internal states **E** The operator establishes a secure ECDH key exchange session with a derived session key Kss
X	X	HMAC Request	**R** $K_{APP-HMAC}$ **E** For each service request the operator will indicated which key to use by passing in key handles as input.
X	X	Self Test	**R** K_{AES} **R** K_{HMAC}
X	X	Show Status	**N/A**
X	X	Get Info	**R** $K_{DI-EC-PUB}$
X	X	Get Version	**NA**
X	X	Assert MANU_DEBUG pin	**Z zeroize all volatile and non-volatile CSP**
X	X	Assert hardware reset	**Z** zeroize all volatile CSP

		pin, or software reset	

7. Operational Environment

The FIPS 140-2 Area 6 Operational Environment requirements are not applicable because the module does not contain a modifiable operational environment.

8. Security Rules

This section documents the security rules enforced by the BCM58200 Series Cryptographic Module to implement the security requirements for a FIPS 140-2 Level 3 module.

1. The module indicates when the device is in the Approved mode of operation.
2. The module implements one approved mode of operation. Power-cycling zeroizes all volatile plaintext critical security parameters.
3. Prior to completion of all FIPS power-on self-tests, the module performs several special *initialization period* functions (e.g., RAM Memory BIST Read/Write, and OTP Checksum). Failure during these special *initialization period* functions causes a chip reset. Subsequent to the special *initialization period* functions, any failure in a FIPS power-on self-test cause the ERROR status to be issued followed by a chip reset.
4. No hardware, software, or firmware components of the cryptographic module are excluded from the security requirements of FIPS 140-2.
5. The module restricts all information flow and physical access points to physical ports and logical interfaces that define all entry and exit points to and from the module.
6. All data output via the data output interface are inhibited when an error state exists and during self-tests.
7. The output data paths are logically disconnected from the circuitry and processes that perform key generation, and key zeroization.
8. The module never outputs plaintext cryptographic keys or CSPs or sensitive data.
9. Status information never contains CSPs or sensitive data that if misused could lead to a compromise of the module
10. The module provides two operator roles; these are the User role, and the Cryptographic-Officer role.
11. The module does not support concurrent operators.
12. The module does not support a maintenance role.
13. The module does not support a bypass capability.
14. The module supports identity-based authentication.
15. When the module is powered off and subsequently powered on, the results of previous

authentications are not retained and the module requires the operator to be re-authenticated.

16. Authentication data within the module is protected against unauthorized disclosure, modification, and substitution.
17. The module contains the authentication data required to authenticate the operator for the first time.
18. For each attempt to use the authentication mechanism, the probability is less than one in 1,000,000 that a random attempt will succeed or a false acceptance will occur.
19. For multiple attempts to use the authentication mechanism during a one-minute period, the probability is less than one in 100,000 that a random attempt will succeed or a false acceptance will occur.
20. The module's authentication mechanism does not supply any feedback information to the operator.
21. Recovery from "soft" error states is possible via power-cycling. Recovery from "hard" error states is not possible.
22. The module is physically protected with a production-grade hard opaque tamper evident encapsulating material.
23. The module does not contain any doors or removable covers.
24. Secret keys, private keys, and CSPs within the module are protected from unauthorized disclosure, modification, and substitution.
25. Public keys within the module are protected against unauthorized modification and substitution.
26. Compromising the security of the key generation methods requires as least as many operations as determining the value of the generated keys.
27. Entropy with nonce and personalization string are gathered internally for DRBG initialization and reseeding.
28. Intermediate key generation values are not output from the module.
29. Key agreement is performed via non-SP 800-56A ECDH (allowed as per FIPS 140-2 Implementation Guidance D.8).
30. The ECDSA key for host session establishment provides 128 bits of security, same as the AES-CCM key agreed upon for the session.
31. The module does not support manual key entry.
32. All secret and private keys entered into the module must be encrypted with an ECDH session key, 128-bit AES-CCM mode key.
33. The module does not support key entry via split knowledge procedures.
34. The module does not support a SW/FW Load service from the operator (host).
35. The module provides a method to zeroize all plaintext secret and private cryptographic keys and CSPs within the module (MANU_DEBUG PIN within the

Secure Boot group physical interface turned high).

36. The module conforms to the EMI/EMC requirements specified by 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class B (i.e., for home use).
37. The module performs the following self-tests:

a. <u>Power up Self-Tests:</u>

i. Cryptographic algorithm tests:
- AES [SMAU – Generic Crypto/Auth] block KAT; encryption and decryption are done separately. (CBC with 128, 192, 256b keys)
- HMAC SHA256 [SMAU – Generic Crypto/Auth] block KAT, covers SHA256.
- DRBG SP800-90A KAT, covering instantiation, generation, and reseeding functions
- RSA, signature generation and signature verification KATs. Key size of 2048 bit; hash size of 256 bit.
- DSA, signature generation and signature verification PCT. Key size of 2048 bit; hash size of 256 bit.
- ECDSA signature generation and signature verification KATs. Key size of 256 bit.
- Non-SP SP800-56A ECDH:
  - DLC primitives KAT.
  - Key Agreement KAT.
  - Key Derivation Function KAT.
- SHA-3 hashing for digest lengths of 224, 256, 384, 512

ii. Firmware Integrity Test:
- BootROM: 32 bit checksum.
- Secure Boot Image, SBI the authenticated software extension of the module's Secure Boot Loader, is authenticated by Secure Boot Loader code when Secure Boot Loader code loads the SBI. Authentication is accomplished via 256 HMAC verification (the module also decrypts the SBI image with its 128 bit AES CSP, K_{AES}).

iii. Critical Functions Tests:
- Memory BIST (Read/Write)
- OTP Checksum Verification

b. <u>Conditional Self-Tests:</u>

i. Continuous Random Number Generator test – performed on NDRNG and DRBG. Block size for NDRNG and DRBG tests is 32 bytes.
ii. Pair-wise consistency test for generated ECDH key pair.

38. The operator is capable of commanding the module to perform the power-up self-test via power cycling.
39. After a secure session is established, all data transfer between the operator and the cryptographic module is encrypted. Any key and secure material that enters and exits the cryptographic module is encrypted.

This section documents the security rules imposed by the vendor:

1. The module does not support the update of the logical serial number or vendor ID.
2. Each 256-bit ECDSA operation takes > 8ms to perform. For each authentication attempt, the cryptographic module has to perform two ECDSA operations, one for ECDSA signature generation and the other for ECDSA signature verification before the operator can be authenticated. The operator can make no more than 3750 attempts in every minute even if attempts were made continuously.

9. Physical Security Policy

Physical Security Mechanisms

The BCM58200 Series Cryptographic Module includes the following physical security mechanisms:

- The module is production-grade and uses standard passivation techniques.
- The module is enclosed in a hard opaque tamper evident enclosure.
- User can periodically, depending on application, examine the device package for visual evidence of tampering like, scratch marks.

10. Mitigation of Other Attacks Policy

The module has not been designed to mitigate any specific attack beyond the requirements of FIPS 140-2.

11. References

- National Institute of Standards and Technology, Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-4 July 2013
- NIST Special Publication 800-90A, Revision 1, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, June 2015
- National Institute of Standards and Technology, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, Special Publication 800-56A, May 2013.
- NIST Special Publication 800-57 Part 1, January 2016

12. Definitions and Acronyms

AES: ECB, CBC, CTR, CCM	Advanced Encryption Standard as defined by FIPS197 and SP800-38A to SP800-38D
API	Application Programming Interface
BIST	Built-In Self-Test
CSP	A FIPS Critical Security Parameter
DLC	Discrete Logarithm Cryptography
DSA	Digital Signature Algorithm as defined by FIPS186-4
DRBG	Deterministic Random Bit Generator
ECDH	Elliptic-curve Diffie-Hellman algorithm
ECDSA	Elliptic-curve Digital Signature Algorithm as defined by FIPS186-4
EMI/EMC	Electromagnetic Interference/Electromagnetic Compatibility
FIPS	Federal Information Processing Standard
FW	Firmware
HMAC	A keyed-Hash Message Authentication Code
HW	Hardware
JTAG	Joint Test Action Group – refer to the test interface standard as defined by IEEE 1149.1 Standard
LPC	Low Pin Count interface
MIPI	Mobile Industry Processor Interface
NFC	Near Field Communication
OTP	One Time Programmable memory.
RAM	Random Access Memory
ROM	Read Only Memory
RSA	Rivest, Shamir, and Adleman algorithm for public key encryption

SBI	Secure Boot Image. Authenticated software extension of the module's BOOT ROM (note: SBI software is part of the BCM5880 Cryptographic Module).
SCAPI	Simple Cryptographic Application Programming Interface (refer to the crypto library of BCM5880 firmware that utilizes the cryptographic hardware of the BCM5880)
SHA	Secure Hash Algorithm
SMAU	Secure Memory Access Unit
SPI	Synchronous Peripheral Interface
SRAM	Static Random Access Memory
STS TESTING	Statistical Testing
SW	Software
TPM	Trusted Platform Module
NDRNG	Non-Deterministic Random Number Generator
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus