18.12.24

Dell Australia Pty Limited, BSAFE Product Team

Dell BSAFE™ Crypto Module

Module Version 2.0.0.0

FIPS 140-3 Security Policy

Document Version 1.8

BSAFE
FIPS 140 Validated

DELL Technologies

This document is a non-proprietary security policy for the Dell BSAFE™ Crypto Module, version 2.0.0.0 (BSAFE Crypto Module) from Dell Australia Pty Limited, BSAFE Product Team.

Contents:

Preface

With the exception of the non-proprietary *Dell BSAFE™ Crypto Module Security Policy* document, the overall FIPS 140-3 Security Level 1 validation submission documentation is proprietary to Dell Australia Pty Limited and is releasable only under appropriate non-disclosure agreements. For access to the documentation, please contact Dell Support.

This security policy describes how the BSAFE Crypto Module meets the Security Level 1 requirements for all aspects of FIPS 140-3, and how to securely operate it.

Federal Information Processing Standards Publication 140-3 - Security Requirements for Cryptographic Modules (FIPS 140-3) details the U.S. Government requirements for cryptographic modules. More information about the FIPS 140-3 standard and validation program is available on the NIST website.

This document deals only with operations and capabilities of the BSAFE Crypto Module in the technical terms of a FIPS 140-3 cryptographic module security policy. More information about BSAFE Crypto Module and the entire BSAFE product line is available from Dell Support.

Terminology

In this document, the term BSAFE Crypto Module denotes the Dell BSAFE™ Crypto Module, version 2.0.0.0, FIPS 140-3 validated Cryptographic Module for Overall Security Level 1.

The BSAFE Crypto Module is also referred to as:

- The Cryptographic Module
- The module.

1 General

BSAFE Crypto Module is validated with an overall FIPS 140-3 Security Level 1. Security levels for individual areas are shown in the following table:

ISO/IEC 24759 Section 6	FIPS 140-3 Section Title	Security Level
1	General	1
2	Cryptographic Module Specification	1
3	Cryptographic Module Interfaces	1
4	Roles, Services, and Authentication	1
5	Software/Firmware Security	1
6	Operational Environment	1
7	Physical Security[1]	N/A
8	Non-invasive Security	N/A
9	Sensitive Security Parameter Management	1
10	Self-Tests	1
11	Life-cycle Assurance	1
12	Mitigation of Other Attacks	1

Table 1 Security Levels

[1] The module relies on the physical security provided by the host on which it runs.

2 Cryptographic Module Specification

BSAFE Crypto Module is a software module intended to be used as part of a software system, providing cryptographic services to that system.

The module is provided as a static library in Executable and Linkable Format (ELF) format, built for the Intel® x86_64 (64-bit) architecture. It follows the standard x86_64 calling conventions and provides a documented set of functions that can be called from user software. It is intended to be linked directly into the user software system.

The following diagram illustrates the cryptographic module boundary and logical interfaces:

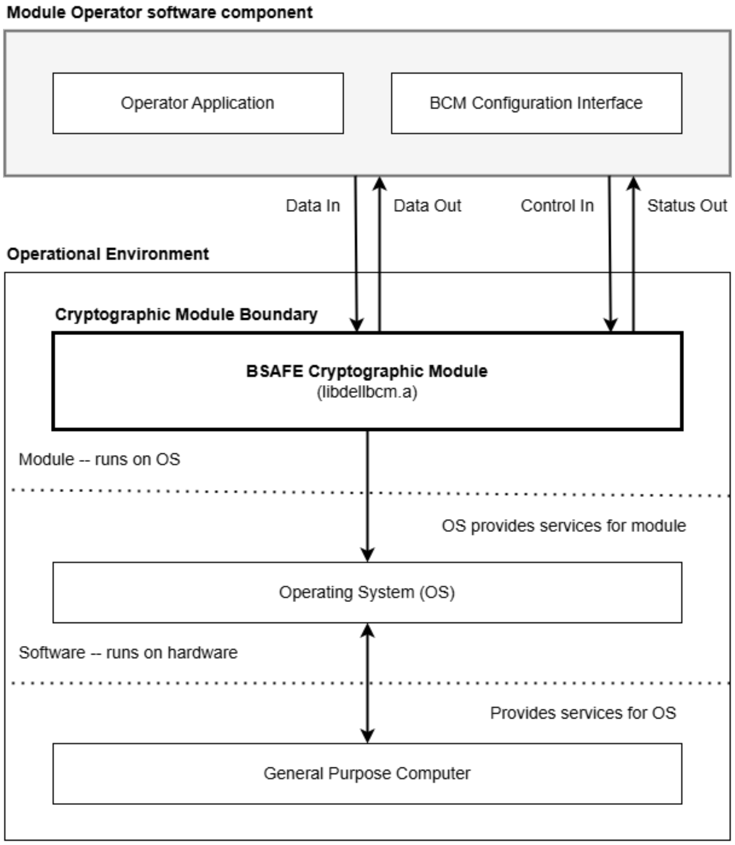

Figure 1 Cryptographic boundary

2.1 Module Description

The module is identified as *Dell BSAFE™ Crypto Module*, version *2.0.0.0,* and consists of a single object file, *fipsobj.o*, in the static library *libdellbcm.a*.

The name and version of the module can be accessed from the APIs `BCM_module_info()` and `BCM_module_version()`.

The FIPS 140-3 validation certificate can be located on the NIST CMVP page using the module name and version reported.

2.2 Software Module Cryptographic Boundaries

Dell BSAFE™ Crypto Module is classified as a multi-chip standalone software cryptographic module for the purposes of FIPS 140-3. As such, it is tested on specific operating systems and computer platforms. The cryptographic boundary includes the module running on selected platforms running selected operating systems. The module is packaged as a library with an object file containing the module's entire executable code. The module relies on the physical security provided by the host computer in which it runs.

The tested operational environment physical perimeter of the module is the case of the general-purpose computer, which encloses the hardware running the module. The physical interfaces for the module are the physical interfaces of the computer running the module, such as the keyboard, monitor and network interface.

The cryptographic module boundary is the linked object file within the final application.

The underlying logical interface to the module is the API, documented in the *Dell BSAFE™ Crypto Module Developers Guide*. The module accepts Control Input through the API calls. Data Input and Output are provided in the variables passed with the API calls. Status Output is provided through the returns and error codes documented for each call. This is illustrated in Figure 1 Cryptographic boundary.

2.3 Operational Environments

For FIPS 140-3 validation, the module is tested by an accredited FIPS 140-3 testing laboratory on the following operational environments:

#	Operating System	Hardware Platform	Processor	PAA / Acceleration
1	Dell PowerMaxOS 10	PowerMax storage array compute node	Intel Xeon® Gold 5218	Yes
2	Dell PowerMaxOS 10	PowerMax storage array compute node	Intel Xeon Gold 5218	No
3	Dell PowerMaxOS 10	PowerMax storage array compute node	Intel Xeon Gold 6240L	Yes
4	Dell PowerMaxOS 10	PowerMax storage array compute node	Intel Xeon Gold 6240L	No
5	Dell PowerMaxOS 10	PowerMax storage array compute node	Intel Xeon Gold 6254	Yes
6	Dell PowerMaxOS 10	PowerMax storage array compute node	Intel Xeon Gold 6254	No
7	Dell PowerMaxOS 10	PowerMax storage array compute node	Intel Xeon Platinum 8280L	Yes
8	Dell PowerMaxOS 10	PowerMax storage array compute node	Intel Xeon Platinum 8280L	No

Table 2 Tested Operational Environments

Dell BSAFE affirms compliance for the following operational environment:

#	Operating System	Hardware Platform
1	SUSE® Linux Enterprise Server 15 SP2	Intel x86_64 (64-bit)

Table 3 Vendor Affirmed Operational Environments

> **Note:** When running the module on an affirmed platform, no assurances are made about the minimum strength of generated SSPs, such as keys.

2.4 Cryptographic Algorithms

The following table lists the Dell BSAFE™ Crypto Module Approved algorithms, with the appropriate standards and CAVP validation certificate numbers:

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function
A1204	AES SP 900-38A	ECB, CBC and CTR	128, 192, and 256 bit key sizes	Symmetric encryption
A1204	AES SP 800-38C	CCM	128, 192, and 256 bit key sizes	Symmetric encryption
A1204	AES SP 800-38D	GCM with automatic IV[1] generation	128, 192, and 256 bit key sizes	Symmetric encryption
A1204	AES SP 800-38E	XTS[2]	128 and 256 bit key sizes	Symmetric encryption
A1204	RSA FIPS 186-4	Key generation. Signature generation and signature verification with SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256. Signature verification with SHA-1.	2048 to 4096-bit key size	Key generation, signature generation, and signature verification
A1204	CVL FIPS 186-4	RSASP1[3] component	2048-bit key size	Signature generation
A1204	CVL SP 800-56B Rev. 2	RSADP[4] component	2048-bit key size	Asymmetric encryption

Table 4 Approved Algorithms

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function
A1204	KDF SP 800-132	PBKDF2[5]	112-256 bits strength	Key derivation
A1204	KTS SP 800-38F	AES Key Wrap, and Key Wrap with Padding.	128, 192, and 256 bit key sizes	Key wrapping
A1204	DRBG[6] SP 800-90A	AES-CTR	128, 192, and 256 bit strengths	Random bit generation
A1204	DRBG SP 800-90A	HMAC SHA2-512	256 bits strength	Random bit generation
A1204	SHS FIPS 180-4	SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	112-256 bits strength	Message digesting
A1204	HMAC[7] FIPS 198-1	HMAC with SHA-1, SHA2-256, SHA2-384, SHA2-512	112-256 bits strength	MAC generation
A1204	KAS-IFC-SSC / SP 800-56B Rev.2	**Schemes**: KAS1 **Key Generation methods**: An RSA key pair with a private key in the basic format, with a random public exponent.	2048 to 8192-bit key size	Shared secret generation
VA[8]	CKG of symmetric keys/ SP 800-133 Rev. 2	Direct output of approved DRBG used to generate 128, 192, or 256 bit AES keys. FIPS 140-3 Implementation Guidance, IG D.H.	128-256 bits strength	Key Generation
VA	CKG of asymmetric keys/ SP 800-133 Rev. 2	Direct output of approved DRBG used to generate prime number seeds and private key values. FIPS 140-3 Implementation Guidance, IG D.H.	128-256 bits strength	Key Generation

Table 4 Approved Algorithms (continued)

[1] Initialization Vector (IV).

[2] AES in XTS mode is approved only for hardware storage applications. The two keys concatenated to create the single double-length key must be checked to ensure they are different.

[3] RSA signature primitive 1 (RSASP1).

[4] RSA decryption primitive (RSADP). RSADP shall only be used within the context of an SP800-56B rev 2 Key Transport Scheme (KTS).

[5] Password-based key derivation function 2 (PBKDF2). As defined in the NIST Special Publication 800-132, PBKDF2 can be used in the Approved Mode of Operation when used with Approved symmetric key and message digest algorithms. For more information, see Crypto Officer Guidance.

[6] Deterministic Random Number Generator (DRBG).

[7] Hash-based Message Authentication Code (MAC).

[8] Vendor-affirmed algorithms.

The following table lists the Dell BSAFE™ Crypto Module Non-Approved algorithms, not allowed in the Approved Mode of Operation:

Algorithm / Function	**Use / Function**
MD5	Message Digesting
DES3[1] (three key) in ECB and CBC modes	Symmetric encryption

Table 5 Non-Approved Algorithms Not Allowed in the Approved Mode of Operation

[1] Triple Data Encryption Standard.

Note: BSAFE Crypto Module does not support any non-approved but allowed algorithms, nor non-approved but allowed algorithms with no security claimed.

2.5 Certification Levels

Dell BSAFE™ Crypto Module is validated with an overall Security Level 1 for FIPS 140-3, with Security Level 1 for each individual area. See Table 1, Security Levels for additional details.

2.6 Modes of Operation

The module can operate in Approved mode or Non-Approved mode. The mode selected affects which algorithms are available for use. The following section details the algorithms available in each mode:

- In the Approved mode (`BCM_MODE_FIPS`), the module allows the cryptographic algorithms listed in Table 4, Approved Algorithms. Non-Approved algorithms not allowed in the approved mode of operation are not available.
- Non-Approved mode (`BCM_MODE_NON_FIPS`), the module allows all available cryptographic algorithms.

Approved mode is also referred to as FIPS 140-3 mode in the product documentation.

In each mode of operation, the complete set of services listed in this Security Policy are available to the Crypto Officer.

> **Note:** Critical Security Parameters must not be shared between modes. For example, a key generated in the Approved mode of operation must not be exported and then imported to an application running in Non-Approved mode.

2.6.1 Module Mode Configuration

The module operator must provide a definition of the configuration function `BCM_get_config(BCM_CONFIG *config)` that is called by the module during startup.

The Module mode is set in the operator's function by assigning a value to `config->mode`.

To start the module in the Approved mode of operation, the operator's configuration function should assign the value `BCM_MODE_FIPS` to the mode member of the configuration structure:

```
BCM_STATUS BCM_get_config(BCM_CONFIG *config)
{
  // Start the module in the Approved mode of operation
  config->mode = BCM_MODE_FIPS;
  // ...
  return BCM_OK;
}
```

To start the module in the Non-Approved mode of operation, the operator's configuration function must assign the value `BCM_MODE_NON_FIPS` to the `mode` member of the configuration structure:

```
BCM_STATUS BCM_get_config(BCM_CONFIG *config)
{
  // Start the module in the Non-Approved mode of operation
  config->mode = BCM_MODE_NON_FIPS;
  // ...
  return BCM_OK;
}
```

Note: The default value of the `mode` member is set to `BCM_MODE_FIPS`. Therefore, if the operator's configuration function does not set the mode, the module starts in the Approved mode of operation.

Once the module is initialized and the pre-operational self-tests (POST) have completed successfully, the overall operating mode of the module can be changed by calling the `BCM_module_configure()` API.

2.6.2 Approved Mode Indicator

The module uses an approved mode indicator combined with a return status code from an approved security service to indicate the use of an approved service.

Approved security services that operate on a `BCM_CTX` use the API `BCM_ctx_is_fips()` with a return code of 1 as an indicator that the context is operating in the approved mode.

Approved security services that operate on a `BCM_KEY` use the API `BCM_key_is_fips()` with a return code of 1 as an indicator that the key is operating in the approved mode.

2.7 Operating the Cryptographic Module

An application using BSAFE Crypto Module is linked to the module file, *libdellbcm.a*, which is then loaded and initialized as part of the application that linked it.

The module initializes itself automatically when loaded, and runs the POST automatically regardless of the mode of operation at startup. If the self-tests complete successfully, the cryptographic services of the module can be used.

The module does not support degraded operation. If any self-test fails the cryptographic services of the module are disabled for both modes of operation.

3 Cryptographic Module Interfaces

BSAFE Crypto Module is a software module that provides APIs only as logical interfaces. Physical ports and interfaces are not provided by the module.

The module conforms to the FIPS 140-3 Security Level 1 requirements for Cryptographic Module Interfaces and does not support a Trusted Channel Interface.

3.1 Ports and Interfaces

The following table lists the ports and interfaces, and the data that passes over each:

Physical Port	Logical Interface	Data that passes over port/interface
N/A	Data input	Service inputs
N/A	Data output	Service outputs
N/A	Control input	Configuration parameters for the API `BCM_module_configure()` which sets the mode of operation.
N/A	Status output	Mode of operation indicator, from either the `BCM_ctx_is_fips()` or `BCM_key_is_fips()` APIs. The state of the module from the API `BCM_module_state()`. For other API status, refer to the Outputs column of Table 7, Roles, Service Commands, Input and Output.

Table 6 Ports and Interfaces

Note: The module does not support a Control Output interface.

4 Roles, Services and Authentication

BSAFE Crypto Module meets all FIPS 140-3 Security Level 1 requirements for Roles, Services and Authentication, implementing only the Crypto Officer role. As allowed by FIPS 140-3, the module does not support identification or authentication of this role. There is no maintenance role, cryptographic bypass capability, or self-initiated cryptographic output. The module does not allow concurrent operators.

4.1 Crypto Officer Role

The Crypto Officer role is responsible for installing and loading the module and has access to all services provided by the module. This role is assumed automatically once the module has been loaded and the POST have run successfully. The POST are automatically run when the module is first loaded. They can be run manually at any time by calling `BCM_module_selftest()`.

4.2 Services

For each service, the Approved Mode indicator is obtained by checking the service status and the Approved mode of the Context or Key. A return status code indicates the service status. For information about individual functions that implement each service, see the *Dell BSAFE™ Crypto Module Developers Guide*.

4.2.1 Roles, Services, Inputs and Outputs

The following is a list of services available to the single Crypto Officer (CO) role.

Role	**Service**	**Input**	**Output**
CO	AES Encryption	Plaintext	Ciphertext, Status
CO	AES Decryption	Ciphertext	Plaintext, Status
CO	Message Digest	Message	Digest, Status
CO	MAC Generation	Secret, Message	MAC, Status
CO	MAC Verification	Secret, Message, MAC	Verify Status, Status
CO	DRBG Initialization	-	-
CO	Random Number Generation	-	Random Bytes, Status
CO	Key Generation	-	Status
CO	Key Import	Key text	Status
CO	Key Export	-	Key text, Status
CO	Key Deletion	-	-

Table 7 Roles, Service Commands, Input and Output

Role	Service	Input	Output
CO	Key Derivation	Secret	Key text, Status
CO	Key Wrap	-	Wrapped key text, Status
CO	Key Unwrap	Wrapped key text	Status
CO	Key Encapsulation (decrypt)	Encapsulated key text	Status
CO	Digital Signature Generation	Message Digest	Signature, Status
CO	Digital Signature Verification	Message Digest, Signature	Verify Status, Status
CO	Show Module Version Information	-	Module Version, Status
CO	Show Status	-	Module Status
CO	Self-test	-	Status

Table 7 Roles, Service Commands, Input and Output

4.2.2 Approved Services

The following is a list of approved services provided by the module:

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator [1]
AES Encryption	Encrypt with symmetric cipher	AES	AES keys	CO	E	K
AES Decryption	Decrypt with symmetric cipher	AES	AES keys	CO	E	K
Message Digest	Digest a message	SHS	-	CO	-	C
MAC Generation	Generate a Message Authentication Code	HMAC	MAC secret	CO	W, E, Z	C
MAC Verification	Verify a Message Authentication Code	HMAC	MAC secret	CO	W, E, Z	C

Table 8 Approved Services

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator [1]
DRBG Initialization	Prepare for random number or key generation	DRBG	Entropy State, DRBG entropy input, DRBG seed, CTR DRBG key value, CTR DRBG V value, HMAC DRBG key value, HMAC DRBG V value	CO	G	C
Random Number Generation	Generate a random number	DRBG	DRBG entropy input, DRBG seed, CTR DRBG key value, CTR DRBG V value, HMAC DRBG key value, HMAC DRBG V value	CO	E	C
Key Generation	Generate a symmetric or asymmetric key	CKG RSA Key generation	AES keys, RSA keys	CO	G	P
Key Import	Import a key into the module	-	AES keys, RSA keys	CO	W	C
Key Export	Export a key from the module	-	AES keys, RSA keys	CO	R	K
Key Deletion	Delete a key from the module	-	AES keys, RSA keys	CO	Z	K
Key Derivation	Derive key text given input secret	KDF (PBKDF2)	KDF secret Derived key text	CO	W, E, Z R, Z	C
Key Wrap	Encrypt an AES key with an AES key encryption key	KTS	AES key (wrapping key) AES key (wrapped key)	CO	E R	K
Key Unwrap	Decrypt an AES key with an AES key encryption key	KTS	AES key (wrapping key) AES key (unwrapped key)	CO	E W	K

Table 8 Approved Services

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator [1]
Key Encapsulation (decrypt)	Decrypt an AES or RSA key with an RSA key encryption key	CVL (RSADP)	RSA key (key encryption key) AES key or RSA key (decrypted key)	CO	E W	K
Digital Signature Generation	Sign a message	RSA (signature generation), CVL (RSASP1)	RSA keys (private key)	CO	E	K
Digital Signature Verification	Verify the signature for a message	RSA (signature verification)	RSA keys (public key)	CO	E	K
Key Agreement	Establish a shared secret	KAS-IFC-SSC	RSA keys (peer public key) RSA keys (peer public key, private key)	CO	W, Z E	K
Show Module Version Information	Provide module version to user	-	-	CO	-	-
Show Status	Provide module status to user	-	-	CO	-	-
Self-test	Run module self-tests on-demand	HMAC	Integrity test key	CO	-	-

Table 8 Approved Services

[1] The indicator for the service is one of the following:

C: The Approved Mode indicator is obtained by checking the service return status code and the mode of the operation's context by calling `BCM_ctx_is_fips()`. For approved services, the FIPS indicator function will return 1.

K: The Approved Mode indicator is obtained by checking the service return status code and the mode of the operation's key by calling `BCM_key_is_fips()`. For approved services, the FIPS indicator function will return 1.

4.2.3 Non-Approved Services

The following is a list of non-approved services provided by the module:

Service	Description	Algorithm Accessed	Role	Indicator[1]
Message Digest	Digest a message	MD5	CO	C
Symmetric Encryption	Encrypt with symmetric cipher	DES3 with ECB and CBC modes	CO	K
Symmetric Decryption	Decrypt with symmetric cipher	DES3 with ECB and CBC modes	CO	K

Table 9 Non-Approved Services

[1] The indicator for the service is one of the following:

C: The Approved Mode indicator is obtained by checking the service return status code and the mode of the operation's context by calling `BCM_ctx_is_fips()`. For these non-approved services the FIPS indicator function will return 0.

K: The Approved Mode indicator is obtained by checking the service return status code and the mode of the operation's key by calling `BCM_key_is_fips()`. For these non-approved services the FIPS indicator function will return 0.

4.3 Operation Authentication

The module does not implement authentication. The Crypto Officer role is implicitly assumed once the module is loaded, and cleared on module unload.

5 Software/Firmware Security

This section covers integrity measures to demonstrate protection of the software component of BSAFE Crypto Module, which is the whole of the module.

5.1 Approved Integrity Techniques

The module is an object file. When the module object file is created, a MAC is calculated over the executable code and static data sections, with the resulting integrity block embedded into the object file. The built-in Integrity Test Key is used as the MAC secret.

During the pre-operational software integrity test when the module is loaded, a MAC is again calculated over the executable code and static data. The resulting MAC is compared to the integrity block calculated when the module was created.

If the MAC differs, the pre-operational software integrity test fails, the POST fails, the module enters the `BCM_MODULE_STATE_INTEGRITY_FAILED` state and cannot be used for any cryptographic operation. Any operation attempted will return the `BCM_ERROR_FIPS_INTEGRITY_FAILURE` error status code.

The only way to clear this error condition is to unload the module and load it again.

The pre-operational software integrity test uses HMAC-SHA2-256.The Cryptographic Algorithm Self-Tests (CASTs) are run prior to the pre-operational software integrity test to ensure the MAC implementation used in the integrity test has been self-tested before it is used in the pre-operational software integrity test.

5.2 On Demand Integrity Test Method

The module provides the `BCM_module_selftest()` API for on-demand integrity testing.

5.3 Executable Module Form

The module is built as a single ELF object file with an embedded FIPS 140-3 integrity signature. The ELF object file exports symbols for the operations it supports.

6 Operational Environment

BSAFE Crypto Module is FIPS 140-3 validated to operate in a modifiable environment.

The module is provided for operating systems running on a general purpose computer platform based on an Intel CPU.

6.1 Compliance

Each instance of the module that is loaded within an operating system maintains its own instance of internal SSPs. Any additional SSPs loaded into a given instance of the module are not available to other instances.

The supported operating environments provide process isolation, with resource and memory protection. Each instance of the module is isolated from others such that SSPs can only be accessed or modified in the module to which they belong.

BSAFE Crypto Module does not spawn additional processes.

6.2 Laboratory Validated Operational Environments

For FIPS 140-3 validation, the module is tested by an accredited FIPS 140-3 testing laboratory. For the tested operational environments, refer to Table 2, Tested Operational Environments.

6.3 Affirmation of Compliance for Other Operational Environments

For the vendor affirmed operational environments, refer to Table 3, Vendor Affirmed Operational Environments.

The Cryptographic Module Validation Program (CMVP) makes no statement as to the correct operation of the module or the security strengths of the generated keys when a specific operational environment that is not listed on the validation certificate is used.

6.4 Configuration Restrictions

The module runs on a General Purpose Computer running one of the operational environments listed in **Tested Operational Environments** and **Vendor Affirmed Operational Environments**.

Each supported operational environment manages its own processes and memory in a logically separated manner. The process management setting is not configurable on the supported operational environments.

7 Physical Security

BSAFE Crypto Module is classified as a multi-chip standalone cryptographic module. The module is comprised of software only, validated at FIPS 140-3 Security Level 1, and does not claim any physical security.

8 Non-invasive Security

BSAFE Crypto Module does not implement any non-invasive mitigation techniques.

9 Sensitive Security Parameters Management

The following tables list the SSPs present in the module, the relevant standards and details of how they are used and accessed.

Name	Description	Type
VM	Memory in the Operational Environment of the module.	Volatile

Table 10 Storage Areas

Protection of the SSPs in volatile memory is provided by the operating environment which isolates the memory of separate processes

Name	From	To	Format Type	Distribution Type	Entry Type
App Write	Operator Application in TOEPP[1]	VM	Plaintext	Manual	Electronic
App Read	VM	Operator Application in TOEPP	Plaintext	Manual	Electronic

Table 11 SSP Input-Output Methods

[1] The module's Tested Operational Environment's Physical Perimeter. The TOEPP for the BSAFE Crypto Module is the host computer.

Sensitive security parameters are input and output using the module APIs. No security function or algorithm is used to transport the data.

Method	Description	Rationale	Operator Initiated Capability
Immediate	Temporary SSPs are zeroized immediately after use	Intermediate values are zeroised at the end of a calculation	N/A
Implicit	SSPs are zeroized when the cryptographic object is deleted by the module	SSPs are zeroized when the `BCM_CTX` object is deleted	N/A
Explict	SSPs are zeroized when the associated key or cryptographic object is deleted by the application	SSPs are zeroized when the application no longer needs them	API call to delete object

Table 12 SSP Zeroization Methods

BSAFE Crypto Module encapsulates symmetric and asymmetric keys as `BCM_KEY` objects. For multi-part cryptographic operations, the module defines several object types to encapsulate the intermediate state of the operation.

Examples are `BCM_CIPHER` objects for symmetric ciphers and `BCM_MAC` objects for message authentication codes. These objects are created explicitly by the user with function calls to the module.

The module defines a `BCM_CTX` object which can contain SSPs in the form of internal random number generator (RNG) state and associated entropy state. A single `BCM_CTX` is created automatically when the module starts and is retained by the module as the default context. `BCM_CTX` objects can be created explicitly by the user with function calls to the module.

To zeroize all unprotected SSPs and key components, perform the following procedure:

1. For each object, delete all:
    1. `BCM_CIPHER` cryptographic objects with a call to `BCM_cipher_delete()`.
    2. `BCM_MAC` cryptographic objects with a call to `BCM_mac_delete()`.
    3. `BCM_DIGEST` cryptographic objects with a call to `BCM_digest_delete()`.
    4. `BCM_KEY` objects with a call to `BCM_key_delete()`.
    5. `BCM_CTX` objects with a call to `BCM_ctx_delete()`.
2. Delete the default context created at startup. To do this, unload the module or call `BCM_module_unload()`.

Key / SSP Name / Type	**Strength**	**Security Function and Cert. Number**	**Generation**	**Import / Export**	**Establishment**	**Storage**	**Zeroisation**	**Use & related keys**
RSA keys (public key / PSP; private key / CSP)	112 - 150 bits	RSA KAS-IFC-SSC (A1204)	CKG SP 800-133 Rev. 2. FIPS 186-4 method	App Write, App Read	N/A	VM	Explicit	Signature generation and verification. Key agreement.
AES keys (CSP)	128, 192, and 256 bits	AES, KTS (A1204)	CKG, SP 800-133 Rev. 2. Direct output of approved DRBG.	App Write, App Read	N/A	VM	Explicit	Symmetric encryption. Key wrapping.
MAC secret (CSP)	128 - 256 bits	HMAC (A1204)	N/A (Application input)	App Write	N/A	VM	Immediate	MAC generation and verification
KDF secret (CSP)	112 - 256 bits	KDF (PBKDF2) (A1204)	N/A (Application input)	App Write	N/A	VM	Immediate	Key derivation
Derived key text (CSP)	112 - 256 bits	KDF (PBKDF2) (A1204)	PBKDF2	App Read	N/A	VM	Immediate	Key derivation
CTR DRBG key value (CSP)	128, 192, and 256 bits	DRBG, CKG (A1204)	Obtained from SP 800-90B compliant entropy source	N/A	N/A	VM	Implicit	Random bit generation

Table 13 SSPs

Key / SSP Name (continued)/ Type	Strength	Security Function and Cert. Number	Generation	Import / Export	Establis hment	Storage	Zeroisation	Use & related keys
CTR DRBG V value (CSP)	128 bits	DRBG, CKG (A1204)	Obtained from SP 800-90B compliant entropy source	N/A	N/A	VM	Implicit	Random bit generation
HMAC DRBG key value (CSP)	256 bits	DRBG, CKG (A1204)	Obtained from SP 800-90B compliant entropy source	N/A	N/A	VM	Implicit	Random bit generation
HMAC DRBG V value (CSP)	256 bits	DRBG, CKG (A1204)	Obtained from SP 800-90B compliant entropy source	N/A	N/A	VM	Implicit	Random bit generation
DRBG entropy input (CSP)	128, 192, and 256 bits	DRBG (A1204)	Obtained from SP 800-90B compliant entropy source	N/A	N/A	VM	Implicit	Random bit generation
DRBG seed (CSP)	128, 192, and 256 bits	DRBG (A1204)	Obtained from SP 800-90B compliant entropy source	N/A	N/A	VM	Implicit	Random bit generation
Entropy state (CSP)	256 bits	DRBG (A1204)	Internal state of SP 800-90B compliant entropy source. Obtained from noise source.	N/A	N/A	VM	Implicit	Random bit generation
Integrity test key (not considered an SSP)	144 bits	HMAC (A1204)	Built-in constant value.	N/A	N/A	Built into module image	N/A	Self-test

Table 13 SSPs

9.1 Deterministic Random Bit Generator

BSAFE Crypto Module provides the following approved DRBGs for use in both Approved and Non-Approved modes:

DRBG	Entropy Obtained (bits)
CTR DRBG with AES-128	128
CTR DRBG with AES-192	192
CTR DRBG with AES-256[1]	256
HMAC DRBG with SHA2-512	256

Table 14 Approved Random Bit Generators

[1] CTR DRBG with AES-256 is the default DRBG.

Use	Details
RSA key-pair generation	Prime generation, and Miller-Rabin prime number testing[1] Blinding of random values
RSA key validation	Prime recovery testing[2]
Symmetric key generation	Direct generation of symmetric keys
Initialization vector generation	Direct generation of IVs for symmetric encryption
RSA PKCS #1 PSS signing	Generation of random value for message encoding

Table 15 DRBG Output Uses

[1] All seeds for asymmetric key generation are generated using the direct output of the approved DRBG.

[2] For details refer to NIST SP 900-56B Rev. 2, Appendix C.

9.2 Entropy Sources

BSAFE Crypto Module provides an entropy source that is internal to the module. This entropy source generates entropy that is used to seed the Approved RBGs.

The entropy source is provided as an ENT (NP) that is compliant with SP 800-90B, and is estimated to provide a min entropy of 7.96875 bits per byte.

Entropy Sources	**Minimum Number of Bits of Entropy**	**Details**
Execution time jitter	256	Instantiation of HMAC DRBG
Execution time jitter	128	Instantiation of CTR DRBG with AES-128
Execution time jitter	192	Instantiation of CTR DRBG with AES-192
Execution time jitter	256	Instantiation of CTR DRBG with AES-256
Execution time jitter	DRBG instantiation bits	Application calls `BCM_random_seed()`
Execution time jitter	64	Application calls `BCM_secure_random_bytes()`

Table 16 Non-Deterministic Random Bit Generation Specification

The `BCM_CTX` object manages an approved DRBG and an Entropy NDRBG. The first call to a random number generation service using the `BCM_CTX` object instantiates the Entropy NDRBG and the DRBG. At instantiation, the DRBG issues a GET call to the Entropy NDRBG for the number of bits of entropy equivalent to the security strength of the DRBG.

A call to the `BCM_random_seed()` API with the `BCM_CTX` object resets the DRBG seed. The DRBG issues a GET call to the Entropy NDRBG for the number of bits of entropy equivalent to the security strength of the DRBG.

A call to the `BCM_secure_random_bytes()` API with the `BCM_CTX` object adds a fixed number of bits of entropy to the DRBG state before the DRBG generates output. The DRBG issues a GET call to the Entropy NDRBG for 64 bits of entropy.

The entropy is collected from the jitter in the CPU execution time for performing an HMAC over the state and previous noise sample. By collecting multiple jitter samples, a bit stream that meets the statistical measurements which indicate a bit stream is random, is produced and whitened.

If the bits of entropy are not available on a GET call then the Entropy NDRBG generates an error status code that is returned to the application by the random number generation service.

9.3 Transition periods

The module addresses the requirements of FIPS 140-3. *Transitioning the use of cryptographic algorithms and key lengths* (NIST SP 800-131A Rev. 2) provides more specific guidance in regards to transition periods or time frames where an algorithm or key length transitions from Approved to Non-Approved.

None of the Approved algorithms provided by the module are affected by transition periods or time frames.

Recommendation for Key Management Part 1 (NIST SP 800-57 Part 1 Rev. 5) specifies security strengths that are acceptable for protecting data going forward. Application writers should consider these acceptable use dates with respect to the expected deployment lifetime of the application and the life of the data being protected. This life depends on the type of key and use, but are from 1-3 years. Refer to NIST SP 800-57, Part 1 for more explanation..

Strength	**Last Date Acceptable**
< 112	Already disallowed
112	31 Dec 2030
>= 128	Acceptable to 2031 and beyond

Table 17 Security Strength Time Frames

The correspondence between security strength, algorithms and key size is specified in the following:

- *Recommendation for Pair-wise Key Establishment Using Integer Factorization Cryptography* (NIST SP 800-56B Rev. 2)
- *Recommendation for Key Management Part 1* (NIST SP 800-57 Part 1 Rev. 5)
- *Recommendation for Applications Using Approved Hash Algorithms* (NIST SP 800-107 Rev. 1).

Strength	**Symmetric**	**RSA**	**Hash**	**MAC and KDF**
< 80		1024	SHA-1	
112	3DES	2048	SHA2-224	
128	AES-128	3072	SHA2-256	SHA-1
192	AES-192	7680	SHA2-384	SHA2-224, SHA2-512/224
256	AES-256	15360	SHA2-512	SHA2-256, SHA2-512/256, SHA2-384, SHA2-512

Table 18 Correspondence between Security Strength, Algorithms and Key Size

10 Self-tests

BSAFE Crypto Module performs a number of pre-operational and conditional self-tests to ensure proper operation.

The cryptographic services of the module are disabled when the self-tests are running.

- When self-tests are running all cryptographic operations fail and return the `BCM_ERROR_FIPS_MODULE_NOT_READY` return status code.
- The `BCM_module_selftest()` status interface returns a state of `BCM_MODULE_STATE_NOT_READY`.

The `BCM_module_selftest()` API runs self-tests on demand after the module has loaded. The on-demand self-tests are the software integrity test and the cryptographic algorithm self-tests. The module can also be reloaded to execute the self-tests on-demand.

If a self-test that is run by `BCM_module_selftest()` fails, the module enters the self-test error state.

For all self-test failures, the library notifies the user through the return status codes for the API.

10.1 Pre-operational Self-tests

The following table lists the pre-operational self-tests:

Algorithm	Test Properties	Type	Details
HMAC	HMAC-SHA2-256 signature:32 bytes HMAC secret:18 bytes	KAT	Pre-operational software integrity test executes automatically when the module is loaded into memory
Entropy source	1024 noise samples	RCT and APT	Pre-operational critical functions test runs when a `BCM_CTX` objects creates an entropy source

Table 19 Pre-operational Self-tests

10.1.1 Pre-operational Self-test Notes

If all POST pass, the cryptographic services of the module are enabled and the module can be used. The `BCM_module_state()` status interface returns a state of `BCM_MODULE_STATE_READY`.

If the pre-operational software integrity test fails, the module enters the self-test error state.

The repetition count test (RCT) and adaptive proportion test (APT) are defined in SP 800-90B.

The pre-operational software integrity tests used are described in detail in Approved Integrity Techniques.

10.2 Conditional Self-tests

The following table lists the conditional self-tests:

Algorithm	Test	Type	Details	Condition
AES	128, 192 and 256-bit AES keys	KAT	Encrypt and decrypt self-tests	Module startup
KTS	128, 192 and 256-bit AES keys	KAT	Wrap and unwrap self-tests	Module startup
DRBG (AES-CTR)	128, 192 and 256-bit strength	KAT	Random bit generation self-tests	Module startup
DRBG (AES-CTR)	128, 192 and 256-bit strength	Fault-Detection Test	SP 800-90A Rev. 1 health test. Instantiate, generate, reseed, uninstantiate tests	Module startup
DRBG (HMAC SHA2-512)	256-bit strength	KAT	Random bit generation self-test	Module startup
DRBG (HMAC SHA2-512)	256-bit strength	Fault-Detection Test	SP 800-90A Rev. 1 health test. Instantiate, generate, reseed, uninstantiate tests	Module startup
HMAC (SHA-1)	HMAC with SHA-1	KAT	MAC generation self-test	Module startup
HMAC (SHA2)	HMAC with SHA2-256, SHA2-384, SHA2-512	KAT	MAC generation self-test	Module startup
KDF	PBKDF2 with SHA2-256, 1000 iterations, 32 byte output	KAT	Key derivation self-test	Module startup
RSA	2048-bit RSA key	KAT	RSA signature and verification self-tests	Module startup

Table 20 Conditional Self-tests

Algorithm	Test	Type	Details	Condition
RSA	2048-bit RSA key	KAT	RSA encryption and decryption self-tests	Module startup
SHS (SHA-1)	SHA-1	KAT	Message digest generation self-test	Module startup
SHS (SHA2)	SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	KAT	Message digest generation self-test	Module startup
RSA (key generation)	2048-bit to 4096-bit signing keys	PCT	RSA sign and verify	Generation of RSA signature key pair
RSA (key generation)	2048-bit to 4096-bit encryption keys	PCT	RSA encrypt and decrypt	Generation of RSA encryption key pair
Entropy source	1 bit of entropy per byte	Fault-Detection Test	Entropy continuous RCT and APT testing, as defined in SP 800-90B	Creation of new DRBG instance Reseed of DRBG instance Generation of secure random bytes

Table 20 Conditional Self-tests

10.2.1 Conditional Self-test Notes

For the KATs, the module includes a set of fixed inputs for each algorithm along with corresponding pre-calculated expected outputs. The cryptographic algorithm is run with the fixed inputs and the algorithm outputs are compared with the expected outputs. If there is any difference the algorithm self-test fails, the CAST fail and the module enters the self-test error state.

If a pair-wise consistency test fails, the key-generation operation fails and returns an error indicator through a return status code. The error is cleared by reattempting the key-generation operation.

If the Entropy source self-tests fail then the entropy collection operation returns an error indicator through a return status code. The error cannot be cleared from the NDRBG object. A new NDRBG object must be created to collect entropy.

10.3 Error States

The following table lists the error states:

State Name	**Description**	**Indicator**
Self-test Error	Module pre-operational integrity test failure	Cryptographic services return `BCM_ERROR_FIPS_INTEGRITY_FAILURE` error code. The `BCM_module_state()` status interface returns a state of `BCM_MODULE_STATE_INTEGRITY_FAILED`.
Self-test Error	CAST failure	Cryptographic services return `BCM_ERROR_FIPS_SELFTEST_FAILURE` error code. The `BCM_module_state()` status interface returns a state of `BCM_MODULE_STATE_SELFTEST_FAILED`.
Self-test Error	On-demand integrity test failure	Cryptographic services return `BCM_ERROR_FIPS_INTEGRITY_FAILURE` error code. The `BCM_module_state()` status interface returns a state of `BCM_MODULE_STATE_INTEGRITY_FAILED`.
Self-test Error	On-demand CAST failure	Cryptographic services return `BCM_ERROR_FIPS_SELFTEST_FAILURE` error code. The `BCM_module_state()` status interface returns a state of `BCM_MODULE_STATE_SELFTEST_FAILED`.

Table 21 Error States

When the module enters the self-test error state then cryptographic services for the module can be re-enabled only by reloading the module.

11 Life-cycle Assurance

11.1 Installation, Initialization, Startup, Operation and Maintenance

11.1.1 Installation

The module is linked into the application at compile time, and installed as part of the target application. There is no physical installation, operation or maintenance of the module.

11.1.2 Initialization

There are no specific initialization steps required for the module.

11.1.3 Startup

The module is started by starting the application that includes it. The module uses operating system services to perform the module startup when the application is started. This module startup includes running the pre-operational integrity test.

Before cryptographic services are made available by the module, the pre-operational integrity tests must complete successfully. These ensure that the application has made no modification to the module as part of its development or installation. For more information about the pre-operational integrity test, see Software/Firmware Security.

11.1.4 Maintenance

Maintenance applies only to the application maintainers. If modifications are made to the application, such as a new version or patch to the application, the module's pre-operational integrity test ensures that the module contained within is unaltered. Application writers should not attempt to modify the module ELF object file as the module will refuse to load or perform cryptographic operations.

11.2 Crypto Officer Guidance

For details of the administrative functions, security parameters, and logical interfaces available to the Crypto Officer, refer to Crypto Officer Role.

The following sections detail the requirements for algorithm use in the Approved Mode of Operation.

11.2.1 Module Management

General Crypto Officer Guidance

Users should take care to zeroize SSPs when they are no longer needed. BSAFE Crypto Module objects should be deleted when no longer needed, which will zeroize sensitive data managed by the module. User variables containing SSP data on the stack or heap should be zeroized after use or before going out of scope.

11.2.2 Random Number Generation Operations

Using the CTR DRBG with AES-256 is recommended as it is fast and provides 256 bits of cryptographic strength, so it is suitable for all purposes. This is the module default DRBG.

11.2.3 Key Generation

When using an approved DRBG to generate keys, the security strength of the DRBG must be at least as great as the security strength of the key being generated. For details about the comparable security strengths of symmetric block ciphers and asymmetric key algorithms refer to Table 2 of SP 800-57 Part 1 Rev. 5.

The module's default DRBG provides a security strength as great as that of any supported keys.

11.2.4 Symmetric Key Operations

GCM Mode Ciphers

When using GCM feedback mode for symmetric encryption, the authentication tag length and authenticated data length may be specified as input parameters, but the IV must not be specified. For compliance with IG C.H scenario 2 it must be generated internally. The generated IV is fully random, generated by the module's approved DRBG, with a default length of 96 bits. No special considerations are required provided the system has sufficient entropy.

XTS Mode Ciphers

AES in XTS mode is approved only for hardware storage applications.

The data encryption key and tweak key components of the double-length XTS key must be checked to ensure they are different. This check is performed automatically by the module.

Triple DES

Triple DES is not available as an approved algorithm in the Approved Mode of Operation. When Triple DES is used in Non-Approved mode, the amount of data that can be encrypted is restricted to 2^{16} *64-bit* blocks.

11.2.5 Asymmetric Key Operations

In the following, *Protect* refers to cryptographically protecting data for later use, e.g. signing, encrypting or wrapping. *Process* refers to processing previously protected data, e.g. verifying, decrypting or unwrapping.

Purpose	**RSA modulus length**	**Note**
Protect and Process	2048, 3072, 4096	Sizes approved in FIPS 186-4 and FIPS 140-3 IG. (CAVP validated)

Table 22 Approved RSA modulus length for digital signatures

Purpose	RSA modulus length	Note
Process only	1024	May be used for verification only.

Table 22 Approved RSA modulus length for digital signatures

11.2.6 Digital Signatures

Keys used for digital signature generation and verification shall not be used for any other purpose. The module generates or loads keys with a particular purpose that is either signing or encryption. The same purpose must always be used for a given key when exported and loaded into the module again.

The length of an RSA key pair for digital signature generation must be greater than or equal to 2048 bits. For digital signature verification, the length must be greater than or equal to 2048 bits, however 1024 bits is allowed for legacy-use only.

RSA keys must pass validation before use. Keys generated by the module will pass validation. For RSA PKCS #1 PSS, the size relationship between the hash function output block length (`hLen`) and the length of the salt (`sLen`) shall be `0 <= slen <= hLen`.

The SHA-1 digest is disallowed for the generation of digital signatures. The digest must be an approved algorithm. Verification of signatures with a SHA-1 digest is allowed for legacy use.

The security strength of both the key and the digest functions shall be chosen to meet or exceed the required security strength for the digital signature. The security strength of the digest function should be stronger or the same as that of the key.

11.2.7 Message Authentication Code Operations

HMACs

The key length for an HMAC generation or verification must be between 112 and 256 bits, inclusive.

For HMAC verification, a length of the secret key greater than or equal to 80 and less than 112 is allowed for legacy-use.

11.2.8 Key Derivation Function Operations

Password-based Key Derivation

Keys generated using PBKDF2 shall only be used in data storage applications.

The minimum password length is 14 characters, which has a strength of approximately 112 bits, assuming a randomly selected password using the extended ASCII printable character set is used.

For random passwords, that is, a string of characters from a given set of characters in which each character is equally likely to be selected, the strength of the password is given by

`S = L*(log N / log 2)`

where:

- `N` is the number of possible characters. For example:
  for the ASCII printable character set `N` = 95
  for the extended ASCII printable character set `N` = 218.
- `L` is the number of characters.

A password of strength `S` can be guessed at random with the probability of 1 in 2^S.

The minimum length of the randomly-generated portion of the salt is 16 bytes.

The iteration count is as large as possible, with a minimum of 10,000 iterations recommended.

The derived key size can range from 1 byte to a maximum of $(2^{32} - 1) * b$, where `b` is the digest size of the message digest function in bytes.

Derived keys can be used as specified in SP 800-132, Section 5.4, option 1a.

11.2.9 Key Transport Schemes

Key Wrapping using AES

The key establishment methodology provides between 128 and 256 bits (inclusive) of encryption strength.

The security strength of the key encryption key must be greater than or equal to the security strength of the key being wrapped.

11.2.10 Key Validation

Asymmetric keys are validated as they enter the module for use in processing existing data.

Before the keys are used to protect data, they must be validated. The module provides services for the validation of RSA keys.

12 Mitigation of Other Attacks

RSA key operations implement blinding, a reversible way of modifying the input data, so as to make the operation immune to timing attacks. Blinding has no effect on the algorithm other than to mitigate attacks on the algorithm.

This mitigation is enabled by default. For optimum security, it should not be disabled. If necessary it can be disabled with `BCM_FLAG_KEY_DISABLE_BLINDING`. For more information, see Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems.

RSA signing operations implement a verification step after private key operations. This verification step is in place to prevent potential faults in optimized Chinese Remainder Theorem (CRT) implementations. It has no effect on the signature algorithm.
This mitigation is enabled by default. For optimum security, it should not be disabled. If necessary it can be disabled with `BCM_FLAG_KEY_DISABLE_SIGNATURE_CHECK`.

For more information, see Breaking public key cryptosystems on tamper resistant devices in the presence of transient faults: Bao, Deng, Han, Jeng and On the Importance of Eliminating Errors in Cryptographic Computations.

RSA PKCS #1 v1.5 encryption padding operations are implemented in constant time in order to make the operation immune to timing attacks.
For this mitigation, constant time padding is built-in and cannot be disabled.
For more information, see. Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1.

13 Acronyms

The following table lists the acronyms used with the module and their definitions:

Term	Definition
AES	Advanced Encryption Standard. A fast symmetric key algorithm with a 128-bit block, and keys of lengths 128, 192, and 256 bits. AES replaces DES as the US symmetric encryption standard.
API	Application Programming Interface.
Attack	An attempt, either a successful or unsuccessful, to break part or all of a cryptosystem. Attack types include an algebraic attack, birthday attack, brute force attack, chosen ciphertext attack, chosen plaintext attack, differential cryptanalysis, known plaintext attack, linear cryptanalysis, and middle person attack.
CAST	Cryptographic Algorithm Self-Tests. A test of all cryptographic functions of an approved cryptographic algorithm that must be performed prior to the first operational use of the cryptographic algorithm. A CAST may be a KAT, a comparison test or a fault-detection test.
CBC	Cipher Block Chaining. A mode of encryption in which each ciphertext depends upon all previous ciphertexts. Changing the IV alters the ciphertext produced by successive encryptions of an identical plaintext.
CCM	Counter with Cipher block chaining Message authentication code. A mode of encryption combining the Counter mode of encryption with Cipher Block Chaining Message Authentication Code (CBC-MAC) for authentication.
CMVP	Cryptographic Module Validation Program.
CSP	Critical Security Parameters are security related information, such as keys or passwords, whose disclosure or modification can compromise security.
CTR	Counter Mode. A mode of encryption which turns a block cipher into a stream cipher. It generates the next keystream block by encrypting successive values of a counter.
CTR DRBG	Counter mode Deterministic Random Bit Generator.
Decryption	The conversion of encrypted data, ciphertext, into its original form. Generally, the reverse of encryption.
DES	Data Encryption Standard. A symmetric encryption algorithm with a 56-bit key with eight parity bits.
DES3	Triple Data Encryption Standard. A symmetric encryption algorithm with three 56-bit key with eight parity bits. Also known as Triple-DES and TDEA.
DRBG	Deterministic Random Bit Generator.
ECB	Electronic Codebook. A mode of encryption which divides a message into blocks and encrypts each block separately.
Encryption	The transformation of plaintext into an apparently less readable form, called ciphertext, using a mathematical process. The ciphertext can be read by anyone who has the key and decrypts (undoes the encryption) the ciphertext.
FIPS	Federal Information Processing Standards.

Table 23 Acronyms and Definitions

Term	Definition
GCM	Galois/Counter Mode. A mode of encryption combining the Counter mode of encryption with Galois field multiplication for authentication.
HMAC	Keyed-Hashing for Message Authentication Code.
HMAC DRBG	HMAC Deterministic Random Bit Generator.
IV	Initialization Vector. Used as a seed value for an encryption operation.
KAT	Known Answer Test.
Key	A string of bits used in cryptography, allowing people to encrypt and decrypt data. Can be used to perform other mathematical operations as well. Given a cipher, a key determines the mapping of the plaintext to the ciphertext. The types of keys include distributed key, private key, public key, secret key, session key, shared key, subkey, symmetric key, and weak key.
Key wrapping	A method of encrypting key data for protection on untrusted storage devices or during transmission over an insecure channel.
MAC	Message Authentication Code.
MD5	A message digest algorithm, which hashes an arbitrary-length input into a 16-byte digest. Designed as a replacement for MD4.
NDRBG	Non-Deterministic Random Bit Generator.
NIST	National Institute of Standards and Technology. A division of the US Department of Commerce (formerly known as the NBS) which produces security and cryptography-related standards.
OS	Operating System.
PBKDF2	Password-based Key Derivation Function 2. A method of password-based key derivation, which applies a MAC algorithm to derive the key.
POST	Pre-Operational Self-Tests.
privacy	The state or quality of being secluded from the view or presence of others.
private key	The secret key in public key cryptography. Primarily used for decryption but also used for encryption with digital signatures.
PRNG	Pseudo-Random Number Generator.
PSP	Public Security Parameters are security related public information (e.g. public keys) whose modification can compromise the security of the cryptographic module.
RNG	Random Number Generator.
RSA	Public key (asymmetric) algorithm providing the ability to encrypt data and create and verify digital signatures. RSA stands for Rivest, Shamir, and Adleman, the developers of the RSA public key cryptosystem.
SHA	Secure Hash Algorithm. An algorithm, which creates a unique hash value for each possible input. SHA takes an arbitrary input, which is hashed into a 160-bit digest.
SHA-1	A revision to SHA to correct a weakness. It produces 160-bit digests. SHA-1 takes an arbitrary input, which is hashed into a 20-byte digest.

Table 23 Acronyms and Definitions

Term	Definition
SHA-2	The NIST-mandated successor to SHA-1, to complement the Advanced Encryption Standard. It is a family of message digest algorithms (SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, and SHA2-512/256), which produce digests of 224, 256, 384, 512, 224, and 256 bits respectively.
SSP	Sensitive Security Parameters include both Critical Security Parameters (CSP) and Public Security Parameters (PSP).
Triple-DES	See DES3.
XTS	XEX-based Tweaked Codebook mode with ciphertext stealing. A mode of encryption used with AES.

Table 23 Acronyms and Definitions