totemo Cryptographic Module (TCM) v3.0

FIPS 140-2 Non-Proprietary Security Policy

FIPS Security Level: 1

Document Version 1.2

Totemo AG
Freihofstrasse 22, CH-8700 Küsnacht, info@totemo.com, www.totemo.com, T +41 44 914 9900, F +41 44 914 9999
CHE-109.335.909 MWST, DUNS-Nr. 48-163-5972, Handelsregisternr. CH-020.3.025.055-6, UID CHE-109.355.909

totemo®
securing data in motion

Table of Contents

Table of Figures

List of Tables

1. Introduction

1.1. Purpose

This is a non-proprietary Cryptographic Module Security Policy for the **totemo** Cryptographic Module from **Totemo AG**. This Security Policy describes how the **totemo** Cryptographic Module meets the security requirements of Federal Information Processing Standards (FIPS) Publication 140-2, which details the U.S. and Canadian Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the National Institute of Standards and Technology (NIST) and the Communications Security Establishment Canada (CSEC) Cryptographic Module Validation Program (CMVP) website at http://csrc.nist.gov/groups/STM/cmvp.

This document also describes how to run the module in a secure FIPS-Approved mode of operation. This policy was prepared as part of the Level 1 FIPS 140-2 validation of the module. The **totemo** Cryptographic Module is referred to in this document as **totemo** TCM, crypto-module, or the module.

1.2. References

This document deals only with operations and capabilities of the module in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on the module from the following sources:

- The **totemo** website (http://www.totemo.com) contains information on the full line of products from **totemo**.
- The CMVP website (http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm) contains contact information for individuals to answer technical or sales-related questions for the module.

1.3. Document Organization

The Security Policy document is one document in a FIPS 140-2 Submission Package. In addition to this document, the Submission Package contains:

- Vendor Evidence document
- Finite State Model document
- Other supporting documentation as additional references

2. totemo Cryptographic Module

2.1. Overview

2.1.1. totemo Product Overview

Totemo AG provides solutions for email encryption, secure managed file transfer and secure mobile communication to enterprise customers and government agencies that need to securely exchange information. **totemo** develops applications that secure data transfers regardless of protocol or application; throughout entire systems and networks. **totemo**'s products are designed to be easy to implement, simple to administer and understand, and require minimal changes in existing network infrastructures.

At the core of each of **totemo**'s products is the **totemo** Security Platform (TSP). The TSP is a dynamic, expandable security architecture, that is based on interoperable standards, and which protects all data in motion, as well as data at rest. The TSP provides security features such as encryption and decryption, authentication and authorization, certificate and key management, and centralized administration. The TSP, a Java-based pluggable application, is platform-independent, highly scalable, transparent to end users, and aligned for future developments. TSP is the cryptographic key and digital certificate management component of **totemo**mail®, **totemo**data® and **totemo**mobile®.

Figure 1 provides a high-level overview of the TSP and the services it provides as well as sample deployment environments.

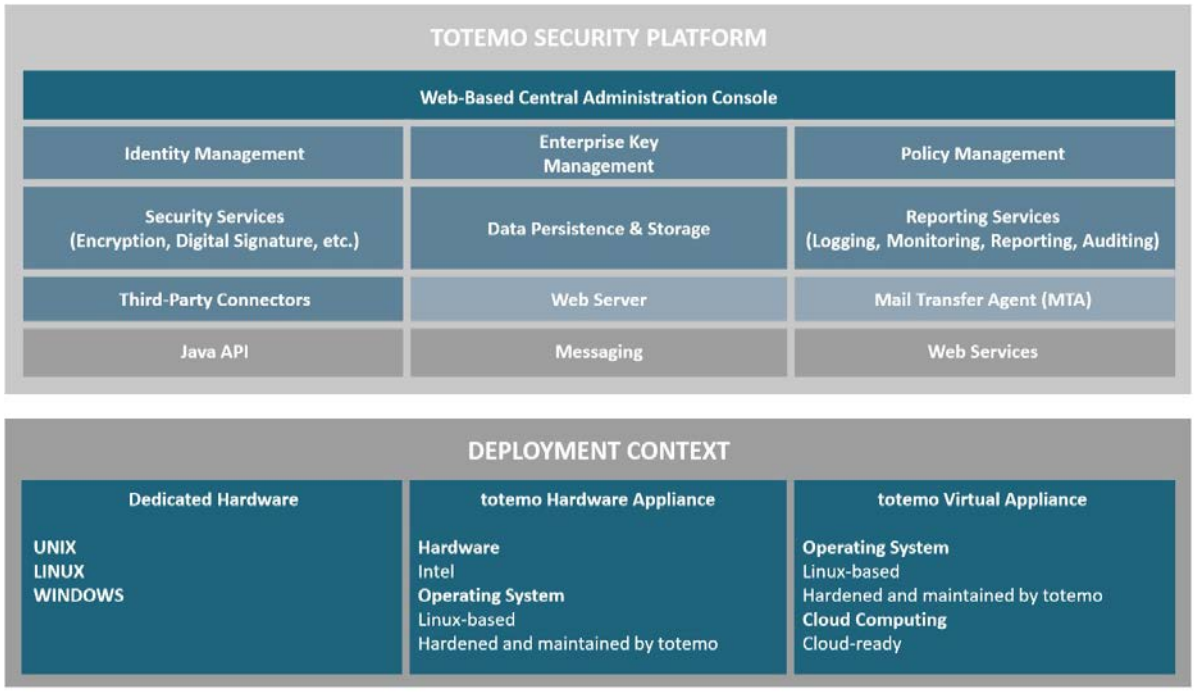

Figure 1 – **totemo** Security Platform

totemomail® is **totemo**'s secure email solution. Its base is a secure email gateway for enterprises and professionals alike. **totemo**mail® leverages the native encryption and digital signature features that exist in all standard mail clients. The solution encrypts emails with internal and external communication partners and even business applications and also enables the sending of large attachments.

totemodata® is a secure file transfer solution which transfers files between people or systems. Like **totemo**mail®, **totemo**data® uses the central security management features of TSP in order to allow members of an organization to exchange large quantities of data without requiring them to become experts on encryption, file transfer protocols, or discretionary access control techniques. **totemo**data® allows users to use virtually any file transfer protocol to upload and download data to the server, while centrally managing access to that data as well as replication and distribution of that data using complex, yet easy-to-configure, business process workflows.

With **totemo**mobile®, business communication remains protected at all times - on all mobile devices and platforms. The solution secures emails and files on mobiles devices and is easily integrated into existing system environments. Like the other products, **totemo**mobile® also leverages the security services and certificate management features of the TSP in order to provide encryption to emails and file transfers when using mobile devices.

totemo Transcoder for BlackBerry provides an additional layer of security by providing protection on not just emails, but data encryption on all traffic between a BlackBerry Enterprise Server and a BlackBerry device. It is easily integrated into existing system environments. As with the other products, the Transcoder leverages the security services and certificate management features of the TSP in order to provide the end-to-end encryption between BlackBerry Devices and BlackBerry Enterprise Server using public-key cryptography.

2.1.2. totemo Cryptographic Module Overview

The **totemo** Cryptographic Module is a Java-based cryptographic library placed at the heart of the TSP, providing encryption, decryption, key production, signature generation and verification, and other cryptographic services to the TSP. **totemo**mail®, **totemo**data®, the server site components of **totemo**mobile® and **totemo** Transcoder for BlackBerry are also written in Java, each incorporating the security features provided by the TSP, which utilizes the Java Cryptography Architecture (JCA)/Java Cryptography Extension (JCE) framework in order to implement cryptographic functionality. By coding their products in Java, **totemo** aims to make them platform-agnostic.

The **totemo** Cryptographic Module is a FIPS module evaluated for overall FIPS Security Level 1, as shown in Table 1.

Table 1 – Security Level Per FIPS 140-2 Section

Section	Section Title	Level
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services, and Authentication	1
4	Finite State Model	1
5	Physical Security	N/A
6	Operational Environment	1
7	Cryptographic Key Management	1
8	EMI/EMC	1
9	Self-tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A

2.2. Module Specification

The **totemo** Cryptographic Module is a software module with a multi-chip standalone embodiment. The overall security level of the module is 1. The TCM is used by calling applications to provide symmetric/asymmetric cipher operation, signature generation/verification, hashing, cryptographic key generation, random number generation, and message authentication functions. The cryptographic boundary of the TCM consists of the shared Java library that is linked with the **totemo** Security Platform. It is designed to execute in a Java Runtime Environment (JRE) installed on **totemo**'s own Operating System (**totemo** Appliance OS 2.3).

The version 3.0 of the module was tested and found compliant on a Pyramid M8110 hardware appliance running **totemo** Appliance OS 2.3 and one Intel Xeon Quad-Core E3-1225v3 processor.

The TCM is defined as a software cryptographic module and therefore has a logical boundary in addition to a physical boundary. The physical and logical boundaries are outlined in section 2.2.1 and 2.2.2 respectively.

2.2.1. Physical Cryptographic Boundary

As a software cryptographic module, there are no physical protection mechanisms implemented. Therefore, the module must rely on the physical characteristics of the host system. The physical boundary of the cryptographic module is defined by the hard enclosure around the host appliance on which it runs. The module supports the physical interfaces of the M8110. These interfaces include the integrated circuits of the system board, the CPU, network adapters, RAM, hard disk, device case, power supply, and fans. Other devices may be attached to the appliance, such as a display monitor, keyboard, mouse, printer, or storage media.

Figure 2 is a block diagram representing the M8110 hardware appliance. The physical cryptographic boundary is defined by the red dotted line. The TCM is stored on the HDD and is loaded into RAM by the OS for execution after the host system powers up. The module will reside in RAM while executing until the host system is powered off or until it is unloaded by the OS.

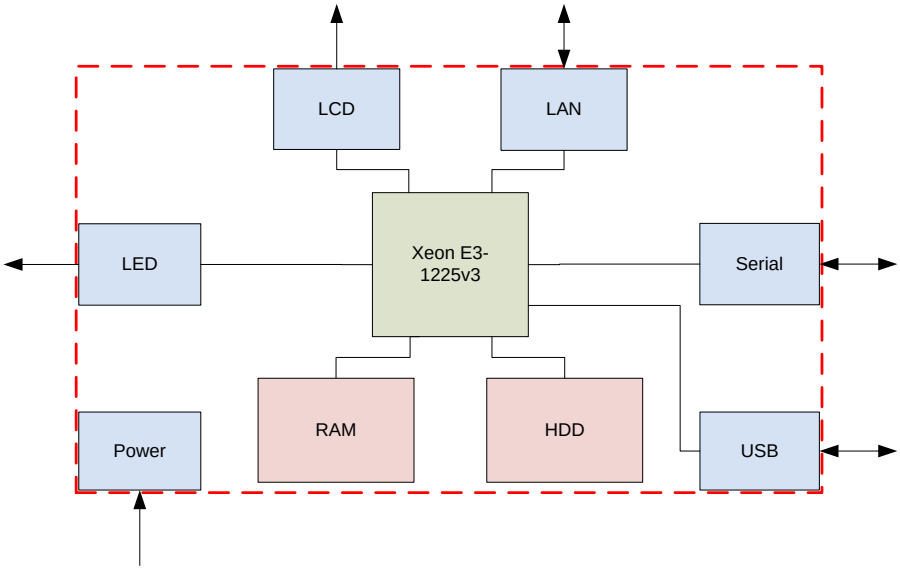

Figure 2 – M8110 Hardware Appliance Block Diagram

2.2.2. Logical Cryptographic Boundary

Figure 3 shows a logical block diagram of the module executing in memory and its interactions with surrounding software components, as well as the module's logical cryptographic boundary. The module's services are designed to be called by the **totemo** Application Software, which define the module's logical interfaces.

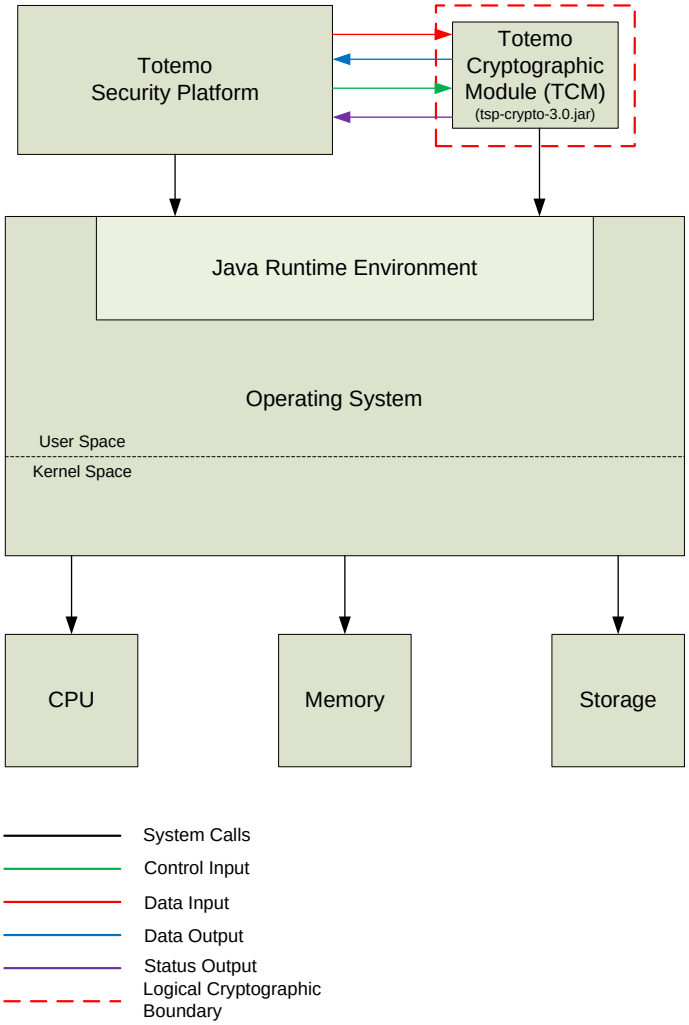

Figure 3 – **totemo** Cryptographic Module Logical Block Diagram and Cryptographic Boundary

2.3. Module Interfaces

The module's physical ports can be categorized into the following logical interfaces defined by FIPS 140-2:

- Data input
- Data output
- Control input
- Status output

As a software module, the module doesn't have any physical characteristics. The module's physical and electrical characteristics, manual controls, and physical indicators are those of the host system. The mapping of the module's logical interfaces in the software to the physical interfaces of the M8110 is described in Table 2 below.

Table 2 – FIPS 140-2 Logical Interface Mappings

FIPS Interface	Physical Interface	Logical Interface
Data Input	USB ports (6), network ports (7), console port (1)	Arguments for library functions that specify plaintext data, ciphertext, digital signatures, cryptographic keys (plaintext or encrypted), initialization vectors, and passwords that are to be input to and processed by the cryptographic module.

Data Output	Network ports (7), console port (1)	Arguments for library functions that receive plaintext data, ciphertext data, digital signatures, cryptographic keys (plaintext or encrypted), and initialization vectors from the cryptographic module.
Control Input	USB ports (6), network ports (7), console port (1)	Arguments for library functions that initiate and control the operation of the module, such as arguments that specify commands and control data (e.g., algorithms, algorithm modes, digest type, or module settings).
Status Output	Network ports (7), console port (1), LED (11)	Function return codes, error codes, or output arguments that receive status information used to indicate the status of the cryptographic module.

2.4. Roles and Services

There are two roles in the module (as required by FIPS 140-2) that operators may assume: a Crypto Officer role and a User role. The module does not allow multiple concurrent operators while in a FIPS-Approved mode of operation.

Please note that the keys and Critical Security Parameters (CSPs) listed in the tables in the following sections indicate the type of access required using the following notation:

- R – Read: The CSP is read.
- W – Write: The CSP is established, generated, modified, or zeroized.
- X – Execute: The CSP is used within an Approved or Allowed security function or authentication mechanism.

2.4.1. Crypto Officer Role

The Crypto Officer role is in charge of installing and initializing the module for first use, checking the status of the module, and running self-tests on demand. Descriptions of the services available to the Crypto Officer role are provided in Table 3 below.

Table 3 – Crypto Officer Services

Service	Description	CSP and Type of Access
Initialize module	Performs integrity check and power-up self-tests	None
Show status	Returns the current mode of the module	None
Run self-tests on demand	Performs power-up self-tests	None
Zeroize keys	Zeroizes and de-allocates memory containing sensitive data	AES key – W AES CMAC Key – W AES GCM IV – W Triple-DES key – W Triple-DES CMAC Key – W HMAC key – W RSA private/public key – W DSA private/public key – W ECDSA private/public key – W DH public/private components

		– W ECDH public/private key – W ECMQV public/private components – W DRBG Seed – W DRBG Entropy – W DRBG C value – W DRBG V value – W

2.4.2. User Role

The User role has the ability to perform symmetric and asymmetric encryption and decryption, signature generation and verification, hashing, cryptographic key generation, random number generation, and message authentication, among other cryptographic services. Descriptions of the services available to the User role are provided in Table 4.

Table 4 – User Services

Service	Description	CSP and Type of Access
Generate random number	Returns the specified number of random bits to calling application	DRBG Seed – R,W,X DRBG 'C' Value – R,W DRBG 'V' Value – R,W DRBG Entropy – R,W,X
Generate message digest	Compute and return a message digest using SHS algorithms	None
Generate keyed hash (HMAC)	Compute and return a message authentication code	HMAC key – RX
Generate Cipher Hash (CMAC)	Compute and return a cipher message authentication code	AES CMAC Key – RX Triple-DES CMAC Key – RX
Generate symmetric key	Generate and return the specified type of symmetric key (Triple-DES or AES)	AES key – W Triple-DES Key – W
Symmetric encryption	Encrypt plaintext using supplied key and algorithm specification (Triple-DES or AES)	AES key – RX Triple-DES key[1] – RX
Symmetric decryption	Decrypt ciphertext using supplied key and algorithm specification (Triple-DES or AES)	AES key – RX Triple-DES key – RX
Generate asymmetric key pair	Generate and return the specified type of asymmetric key pair (RSA, DSA, or ECDSA)	RSA private/public key – W DSA private/public key – W ECDSA private/public key – W
Key Agreement	Perform key agreement using DH, ECDH and ECMQV	DH Public/Private components – WRX ECDH Public/Private components – WRX

[1] Triple-DES encryption must not be used for more than 2^{16} 64-bit blocks of data for any given key.

		ECMQV Public/Private components – RX
Key Wrapping	Perform key wrap with RSA public key, AES key, or Triple-DES Key	RSA public Key – RX AES Key – RX Triple-DES Key – RX
Key Unwrapping	Perform key unwrap with RSA private key, AES key, or Triple-DES Key	RSA private Key – RX AES Key – RX Triple-DES Key – RX
Signature Generation	Generate a signature for the supplied message using the specified key and algorithm (RSA, DSA, or ECDSA)	RSA private key – RX DSA private key – RX ECDSA private key – RX
Signature Verification	Verify the signature on the supplied message using the specified key and algorithm (RSA, DSA, or ECDSA)	RSA public key – RX DSA public key – RX ECDSA public key – RX

2.5. Physical Security

The **totemo** Cryptographic Module (TCM) is a software module, which FIPS defines as a multi-chip standalone cryptographic module. As such, it does not include physical security mechanisms. Thus, the FIPS 140-2 requirements for physical security are not applicable.

2.6. Operational Environment

The **totemo** Cryptographic Module was tested and found to be compliant with FIPS 140-2 requirements on an Intel Xeon E3-1225v3 processor. The processor executes **totemo** Appliance OS 2.3 on which JRE 8.0 is executing. The TCM and **totemo** software applications using the TCM will be loaded into and executed by a Java Virtual Machine (JVM) provided by the JRE.

2.6.1. Use of External RNG

The module makes use of the JVM's configured SecureRandom entropy source to provide entropy when required. The module will request entropy as appropriate to the security strength and seeding configuration for the DRBG that is using it. In approved mode the minimum amount of entropy that would be requested is 112 bits with a larger minimum, such as 256 bits, being set if the security strength of the operation requires it. The module will wait until the SecureRandom.generateSeed() returns the requested amount of entropy, blocking if necessary.

2.7. Cryptographic Key Management

The module implements the FIPS-Approved algorithms listed in Table 5 below.

Table 5 – FIPS-Approved Algorithm Implementations

Algorithm	Certificate Number
AES in ECB, CBC, CFB-128, OFB, CCM, CMAC, GCM modes encrypt/decrypt with 128-, 192- and 256-bit keys	5901
AES in KW mode for key wrapping with 128-, 192- and 256-bit keys	5901
Triple-DES in ECB, CBC, CFB-8, CFB-64, CMAC modes en-	2870

crypt/decrypt; KO 1, 2[2]	
RSA (FIPS 186-4) Key Generation for 2048- and 3072-bit keys	3089
RSA (PKCS #1 v1.5 and PSS) Signature Generation on 2048-, 3072-, and 4096-bit keys with support for SHA-3[3]	3089
RSA (PKCS #1 v1.5, PSS, and X9.31) Signature Verification on 2048-, 3072-, and 4096-bit[4] keys with support for SHA-3[3]	3089
RSA decryption (CVL)	2125
DSA (FIPS 186-4) Key Generation for 2048- and 3072-bit keys	1496
DSA Signature Generation and Verification for 2048- and 3072-bit keys with support for SHA-3[3]	1496
ECDSA Key Generation with NIST Recommended Curves: P-224, P-256, P-384, and P-521	1572
ECDSA Signature Generation and Verification for P-224, P-256, P-384, and P-521 with support for SHA-3[3]	1572
SHA-1[5], SHA-224, SHA-256, SHA-384, SHA-512, SHA512/224, SHA512/256 hash	4657
SHA3-224, SHA3-256, SHA3-384, SHA3-512	61
HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512, HMAC-SHA-512/224, HMAC-SHA-512/256 keyed hash	3881
HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512	3881
ECDH and ECMQV key agreement	202
Diffie-Hellman key agreement	202
SP 800-90Ar1 HASH_DRBG	2463
CKG (NIST SP 800-133)	Vendor Affirmed

In accordance with FIPS 140-2 IG D.12, the cryptographic module performs Cryptographic Key Generation (CKG) as per SP800-133 (vendor affirmed). The resulting generated symmetric key and the seed used in the asymmetric key generation are the unmodified output from SP800-90Ar1 DRBG.

The cryptographic module implements the following non-Approved key-establishment algorithms, which are allowed for use in a FIPS-Approved mode of operation:

- RSA (CVL Cert. #2125, key wrapping; key establishment methodology provides 112 to 256[6] bits of encryption strength)
- Diffie-Hellman (key agreement; key establishment methodology provides between 112 and 256 bits of encryption strength)
- ECDH (key agreement; key establishment methodology provides between 112 and 256 bits of encryption strength)
- ECMQV (key agreement; key establishment methodology provides between 112 and 256 bits of encryption strength)

[2] The use of TDES Keying Option 2 is restricted to legacy use, for decryption only.

[3] Vendor-affirmed when used with SHA-3 (CAVP testing does not offer testing with SHA-3 variants).

[4] Vendor affirmed when used with k=4096 for signature verification; the module supports k=4096 for all operations, but CAVP testing of k = 4096 is only available for signature generation.

[5] The use of SHA-1 in the FIPS-Approved mode of operation shall be limited to random number generation and signature verification.

[6] Calculated using Equation 1 of FIPS 140-2 IG 7.5 for 2048- to 15360-bit keys

- AES (Cert. #5901, key unwrapping; key establishment methodology provides 128 to 256 bits of encryption strength)
- Triple-DES (Cert. #2870, key unwrapping; key establishment methodology provides 80 to 112 bits of encryption strength)

The cryptographic module implements the following non-approved cryptographic functions, which can be used only in non-FIPS mode of operation. Non-approved algorithms make use of the existing approved services and the following KDF services (KBKDF and PBKDF) in the non-approved mode of operation:

- AES (non-compliant[7])
- ARC4 (RC4)
- Blowfish
- Camellia
- CAST5
- DES
- Diffie-Hellman KAS (non-compliant[8])
- DSA (non-compliant[9])
- DSTU4145
- ECDSA (non-compliant[10])
- ElGamal
- GOST28147
- GOST3410-1994
- GOST3410-2001
- GOST3411
- HMAC-GOST3411
- HMAC-MD5
- HMAC-RIPEMD128
- HMAC-RIPEMD160
- HMAC-RIPEMD256
- HMAC-RIPEMD320
- HMAC-TIGER
- HMAC-WHIRLPOOL
- IDEA
- KBKDF using SHA-512/224 or SHA-512/256 (non-compliant)
- MD5
- OpenSSL PBKDF (non-compliant)
- PKCS#12 PBKDF (non-compliant)
- PKCS#5 Scheme 1 PBKDF (non-compliant)
- PRNG X9.31
- RC2
- RIPEMD128

[7] Support for additional modes of operation.

[8] Support for additional key sizes and the key establishment of keys of less than 112 bits of security strength.

[9] Deterministic signature calculation, support for addition digests and key sizes.

[10] Deterministic signature calculation, support for addition digests and key sizes.

- RIPEMD160
- RIPEMD256
- RIPEMD320
- RSA (non-compliant[11])
- RSA KTS (non-compliant[12])
- SCrypt
- SEED
- Serpent
- SipHash
- SHACAL-2
- TIGER
- Triple-DES (non-compliant[13])
- Twofish
- WHIRLPOOL

[11] Support for additional digests and signature formats, PKCS#1 1.5 key wrapping, support for additional key sizes.

[12] Support for additional key sizes and the establishment of keys of less than 112 bits of security strength.

[13] Support for additional modes of operation.

The module supports the critical security parameters (CSPs) listed below in Table 6. Internally generated keys and CSPs listed in Table 6 are generated with an Approved Random Number Generator.

Table 6 – List of Cryptographic Keys, Cryptographic Key Components, and CSPs

Key	Key Type	Generation / Input	Output	Storage	Zeroization	Use
AES key	AES128-, 192-, 256-bit key	API call parameter or internally generated	Output via GPC internal path	Plaintext in volatile memory	Unload module, power cycle	Encryption, Decryption
AES CMAC Key	AES CMAC 128-, 192-, 256-bit key	API call parameter	Output via GPC internal path	Plaintext in volatile memory	Unload module, power cycle	Message Authentication with AES
AES GCM IV[14]	Random data[15]	Internally generated	Never	Keys are not persistently stored by the module	Unload module, API call, Remove Power	IV input to AES GCM function
Triple-DES key[16]	Triple-DES 192-bit key	API call parameter or internally generated	Output via GPC internal path	Plaintext in volatile memory	Unload module, power cycle	Encryption, decryption
Triple-DES CMAC Key	Triple-DES CMAC 192-bit key	API call parameter	Output via GPC internal path	Plaintext in volatile memory	Unload module, power cycle	Message Authentication with Triple-DES
HMAC key	128- to 512-bit HMAC Key	API call parameter	Output via GPC internal path	Plaintext in volatile memory	Unload module, power cycle	Message Authentication with SHA-2 and SHA-3 family
RSA private key	RSA 2048-, 3072-, 4096-, 7680-, 15360-bit key	API call parameter or internally generated	Output via GPC internal path	Plaintext in volatile memory	Unload module, power cycle	Signature generation, decryption
RSA public key	RSA 2048-, 3072-, 4096-, 7680-, 15360-bit key	API call parameter or internally generated	Output via GPC internal path	Plaintext in volatile memory	Unload module, power cycle	Signature verification, encryption
DSA private key	Private key at least of size 224-, 256-, 384-,	API call parameter or	Output via GPC inter-	Plaintext in volatile	Unload module, pow-	Signature generation,

[14] The module was tested and validated with IVs in the range of 8 to 1024 bits. In the FIPS Mode of Operation, the IV length is restricted to 96 bits or greater.

[15] IV generation is per IG A.5 scenario 2 and has a minimum of 112 bits of entropy strenght.

[16] The module was tested and validated with TDES Keying Option 1 (3-Key). Option 2 (2-Key) was tested and validated only for decryption; encryption is not permitted in FIPS mode.

Key	Key Type	Generation / Input	Output	Storage	Zeroization	Use
	512-bit	internally generated	nal path	memory	er cycle	decryption
DSA public key	DSA 2048-, 3072-, 4096-, 7680-, 15360-bit key	API call parameter or internally generated	Output via GPC internal path	Plaintext in volatile memory	Unload module, power cycle	Signature verification, encryption
ECDSA private key	ECDSA Key from NIST Recommended Curves: P-224, P-256, P-384, and P-521	API call parameter or internally generated	Output via GPC internal path	Plaintext in volatile memory	Unload module, power cycle	Signature generation, decryption
ECDSA public key	ECDSA Key from NIST Recommended Curves: P-224, P-256, P-384, and P-521	API call parameter or internally generated	Output via GPC internal path	Plaintext in volatile memory	Unload module, power cycle	Signature verification, encryption
DH public key	Public key of size 2048-, 3072-, 4096-, 7680-, 15360-bit	API call parameter or Internally generated	Output via GPC internal path	Plaintext in volatile memory	Unload module, power cycle	Key Agreement Key Establishment
DH private key	Private key at least of size 224-, 256-, 384-, 512-bit	API call parameter or Internally generated	Output via GPC internal path	Plaintext in volatile memory	Unload module, power cycle	Key Agreement Key Establishment
ECDH public key	ECDH public key for NIST Recommended Curves: P-224, P-256, P-384, and P-521	API call parameter or Internally generated	Output via GPC internal path	Plaintext in volatile memory	Unload module, power cycle	Key Agreement Key Establishment
ECDH private key	ECDH private key for NIST Recommended Curves: P-224, P-256, P-384, and P-521	API call parameter or Internally generated	Output via GPC internal path	Plaintext in volatile memory	Unload module, power cycle	Key Agreement Key Establishment
ECMQV public key	ECDH public key for NIST Recommended Curves: P-224, P-256, P-384, and P-521	API call parameter or Internally generated	Output via GPC internal path	Plaintext in volatile memory	Unload module, power cycle	Key Agreement Key Establishment
ECMQV private key	ECDH private key for NIST Recommended	API call parameter or Internally generated	Output via GPC internal path	Plaintext in volatile memory	Unload module, power cycle	Key Agreement

Key	Key Type	Generation / Input	Output	Storage	Zeroization	Use
	Curves: P-224, P-256, P-384, and P-521					Key Establishment
DRBG Seed	880-bit random value	API call parameter or Internally generated	Never	Plaintext in volatile memory	Unload module, power cycle	Seed input to SP 800-90Ar1 Hash_DRBG
DRBG Entropy	440-bit random value	API call parameter or Internally generated	Never	Plaintext in volatile memory	Unload module, power cycle	Entropy input to SP 800-90Ar1 Hash_DRBG
Hash DRBG V value	Internal hash DRBG state value	Internally generated	Never	Plaintext in volatile memory	Unload module, power cycle	Used for SP 800-90Ar1 Hash_DRBG
Hash DRBG C value	Internal hash DRBG state value	Internally generated	Never	Plaintext in volatile memory	Unload module, power cycle	Used for SP 800-90Ar1 Hash_DRBG

2.8. EMC/EMI

The **totemo** Cryptographic Module is a software module. Therefore, the only electromagnetic interference produced is that of the host platform on which the module resides and executes. FIPS 140-2 requires that the host systems on which FIPS 140-2 testing is performed meet the Federal Communications Commission (FCC) EMI and EMC requirements for business use as defined in Subpart B, Class A of FCC 47 Code of Federal Regulations Part 15. However, all systems sold in the United States must meet these applicable FCC requirements.

2.9. Self-Tests

The **totemo** Cryptographic Module performs self-tests automatically each time the host appliance is powered on and a host application loads it into memory. Conditional self-tests are performed each time the module needs to generate a new random number or a new asymmetric key pair. The module's random bit generator will perform critical function tests as needed to assure its security.

Should any self-test fail, the module's data output interfaces will be inhibited. Only control input and status output commands will be allowed to execute. For errors encountered during conditional pairwise consistency checking, the module will enter a soft error state, which will clear any random bit or keying information and return to normal operation. For all other errors, the module will halt and must be reloaded into memory by either restarting the host application or rebooting the appliance.

2.9.1. Power-Up Self-Tests

The **totemo** Cryptographic Module performs the following self-tests at power-up:

- Software integrity check using RSA 4096 digital signature verification with SHA-384 hash
- Known Answer Tests (KATs)
  - AES KAT
  - AES-CMAC KAT
  - Triple-DES KAT
  - Triple-DES-CMAC KAT
  - RSA Encryption/Decryption KAT
  - RSA Signature Generation KAT
  - RSA Signature Verification KAT
  - DSA Pairwise Consistency Test
  - ECDSA Pairwise Consistency Test
  - SHA-1 KAT
  - SHA-224 KAT
  - SHA-256 KAT
  - SHA-384 KAT
  - SHA-512 KAT
  - SHA-512/224 KAT
  - SHA-512/256 KAT
  - SHA-3-224 KAT
  - SHA-3-256 KAT
  - SHA-3-384 KAT
  - SHA-3-512 KAT
  - HMAC SHA-224 KAT
  - HMAC SHA-256 KAT
  - HMAC SHA-384 KAT
  - HMAC SHA-512 KAT

- HMAC SHA-512/224 KAT
- HMAC SHA-512/256 KAT
- HMAC SHA-3-224 KAT
- HMAC SHA-3-256 KAT
- HMAC SHA-3-384 KAT
- HMAC SHA-3-512 KAT
- DH Key Agreement KAT
- ECDH Key Agreement KAT
- ECMQV Key Agreement KAT
- SP 800-90Ar1 HASH_DRBG KAT

2.9.2. Conditional Self-Tests

The **totemo** Cryptographic Module performs the following conditional self-tests:

- Continuous Random Number Generator Test for the SP 800-90Ar1 HASH_DRBG
- Continuous Random Number Generator Test for the NDRNG entropy source
- RSA Pairwise Consistency Test for signature generation and verification
- RSA Pairwise Consistency test for wrapping and unwrapping
- DSA Pairwise Consistency test for signature generation and verification
- ECDSA Pairwise Consistency test for signature generation and verification

2.9.3. Critical Functions Self-Tests

The **totemo** Cryptographic Module implements the SP 800-90Ar1 HASH_DRBG as its random number generator. This DRBG employs two critical functions which must also be tested on a regular basis to ensure the security of the SP 800-90Ar1 DRBG. Therefore, the following critical function tests are also implemented by the crypto module:

- DRBG Instantiate Critical Function Test
- DRBG Reseed Critical Function Test
- DRBG Generate Critical Function Test
- DRBG Uninstantiate Critical Function Test

2.10. Mitigation of Other Attacks

This section is not applicable. The modules do not claim to mitigate any attacks beyond the FIPS 140-2 Level 1 requirements for this validation.

3. Secure Operation

The **totemo** Cryptographic Module meets Level 1 requirements for FIPS 140-2. The sections below describe how to place and keep the module in FIPS-Approved mode of operation.

3.1. Crypto Officer Guidance

FIPS 140-2 mandates that a software cryptographic module at Security Level 1 be restricted to a single operator mode of operation. The operating system segregates user processes into separate process spaces. Each process space is an independent virtual memory area that is logically separated from all other processes by the operating system software and hardware. The module functions entirely within the process space of the process that invokes it, and thus satisfies the FIPS 140-2 requirement for a single user mode of operation.

3.1.1. Initial Setup

The **totemo** Cryptographic Module is one of many components of the **totemo** Security Platform, which is delivered as part of a **totemo** host application. The Crypto Officer shall follow the installation procedures of the host software application to ensure proper installation and operation of the TCM. Detailed documentation on installing, uninstalling, configuring, managing and upgrading the host application is provided as part of the product documentation set.

To place the **totemo** Cryptographic Module into a FIPS-Approved mode of operation the JRE system property "`tsp.tcm.fipsMode`" shall be set to "FIPS". This can be done via the CLI (use Java startup tion `-Dtsp.tcm.fipsMode=FIPS`), by modifying the JRE system property file directly, or programmatically using the Java API for setting system properties.

3.2. User Guidance

The **totemo** Cryptographic Module is designed for use by the **totemo** Security Platform. The TCM does not have the ability to input or output CSPs beyond its physical boundary, nor does it persistently store CSPs within its logical boundary.

4. Acronyms

This section describes the acronyms.

Table 7 – Acronyms

Acronym	Definition
AES	Advanced Encryption System
CBC	Cipher Block Chaining
CCM	Counter with CBC-MAC
CFB	Cipher Feedback
CKG	Cryptographic Key Generation
CLI	Command Line Interface
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CPU	Central Processing Unit
CSEC	Communications Security Establishment Canada
CSP	Critical Security Parameter
DES	Data Encryption Standard
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECC CDH	ECC Cofactor Diffie Hellman
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
ECMQV	Elliptic Curve Menezes–Qu–Vanstone
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FIPS	Federal Information Processing Standard
GCM	Galois/Counter Mode
GPC	General Purpose Computer
GUI	Graphical User Interface
HDD	Hard Disk Drive
HMAC	(Keyed) Hash Message Authentication Code
JCA	Java Cryptography Architecture
JCE	Java Cryptography Environment
JRE	Java Runtime Environment
JVM	Java Virtual Machine
KAS	Key Agreement Scheme
KAT	Known Answer Test
KBKDF	Key based Key Derivation Function
KDF	Key Derivation Function

KO	Keying Option
LAN	Local Area Network
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MAC	Message Authentication Code
MQV	Menezes–Qu–Vanstone
NIST	National Institute of Standards and Technology
NVLAP	National Voluntary Laboratory Accreditation Program
OAEP	Optimal Asymmetric Encryption Padding
OFB	Output Feedback
OS	Operating System
PBKDF	Password based Key Derivation Function
PKCS	Public Key Cryptography Standard
PSS	Probabilistic Signature Scheme
RAM	Random Access Memory
RSA	Rivest Shamir and Adleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SP	Special Publication
TCM	totemo Cryptography Module
TSP	totemo Security Platform
USB	Universal Serial Bus