Unisys OS 2200 Cryptographic Library
Version 2R1
Unisys Corporation

Unisys OS 2200 Cryptographic Library FIPS 140-2
Non-Proprietary Security Policy
Version 2.08

Copyright Notice

Acknowledgements

The principal author of this document is:

James R. Heit
Consultant james.heit@Unisys.com
3199 Pilot Knob Road
Eagan, MN 55121

For further information contact:

Robert L. Bergerson
Manager robert.bergerson@Unisys.com
3199 Pilot Knob Road
Eagan, MN 55121

Brian A. Wegleitner
Security Architect brian.wegleitner@Unisys.com
3199 Pilot Knob Road
Eagan, MN 55121

Revision Table

Revision Number	Date	Author	Description
1.0	Dec. 1, 2009	Unisys	Initial Submission
1.01	Jan. 5, 2010	Unisys	Revisions from initial CMVP Lab review.
1.02	Jan. 14, 2010	Unisys	Added table 3.1.1.
1.03	Jan. 21, 2010	Unisys	Revision from third CMVP Lab review.
1.04	Feb. 22, 2010	Unisys	Minor wording revisions.
1.05	Mar. 1, 2010	Unisys	Added CAVP Certificate Numbers.
1.06	Mar. 1, 2010	Unisys	CMVP requested corrections.
1.07	Mar. 8, 2010	Unisys	Add note signature gen/ver only approved with SHA-1.
1.08	Mar. 8, 2010	Unisys	Requested CMVP corrections.
1.09	Oct. 15, 2015	Unisys	Changed the CryptoLib level on the title page and the acknowledgements.
2.0	Dec. 12, 2017	Unisys	Initial submission for CMVP Lab review of CryptoLib 2R1.
2.01	Jan. 9, 2018	Unisys	Revisions from initial CMVP Lab review. Added Keys and CSPs Table.
2.02	Jan. 17, 2018	Unisys	Revisions from second CMVP Lab review. Updated algorithms tables.
2.03	Jan. 23, 2018	Unisys	Revisions from third CMVP Lab review. Added additional services.
2.04	May 4, 2018	Unisys	Added new CAVP certificate numbers.
2.05	May 23, 2018	Unisys	Revisions requested by CMVP Lab prior to NIST submission.
2.06	May 29, 2018	Unisys	Additional changes requested by CMVP Lab.
2.07	May 30, 2018	Unisys	Added Vendor Affirmed Platforms section.
2.08	Aug. 22, 2018	Unisys	Revisions from NIST CMVP review.

Contents

Tables

Section 1
Introduction

This document specifies the non-proprietary Security Policy for the Unisys OS 2200 Cryptographic Library cryptographic module version 2R1. In this document, the Unisys OS 2200 Cryptographic Library is also referred to as Cryptographic Library, Cryptographic Library Module, CryptoLib, and the module. This Security Policy describes the compliance of Cryptographic Library (CryptoLib) with Federal Information Processing Standards Publication 140-2 (FIPS 140-2).

This document also specifies the required actions to use CryptoLib in a FIPS approved mode of operation. This document may be freely distributed in-whole and without modification.

1.1. Audience

This document is required as part of FIPS 140-2 validation. It describes how the Cryptographic Library Module meets the requirements of FIPS 140-2 validation.

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 Unisys website (http://www.unisys.com) contains information on the full line of products from Unisys.
- The CMVP website (http://csrc.nist.gov/groups/STM/cmvp/) contains information on NIST and the cryptographic module validation program.

Please contact Unisys Corporation for access to proprietary product documentation for the Unisys OS 2200 Cryptographic Library

1.3. Documents

FIPS 140-2 requires a submission package containing several documents, they include:

- Security Policy, this document
- Finite State Model
- Design Documents
- Block Diagram
- Source Listings
- Vendor Evidence

Please contact Unisys Corporation for access to proprietary product documentation for the Unisys OS 2200 Cryptographic Library.

Section 2
Specification

2.1. Overview

CryptoLib is an OS 2200 system software library product that has been validated to the FIPS 140-2 standard. Access to the library is provided through an Application Program Interface (API). The U.S. Government and some commercial customers require FIPS 140 validation of products that use cryptography.

2.2. Specification

This module is classified by the FIPS 140-2 standard as a multi-chip standalone cryptographic module. This module is validated to the following FIPS 140-2 levels.

Table 2–1. Security Level per FIPS 140-2 Sections

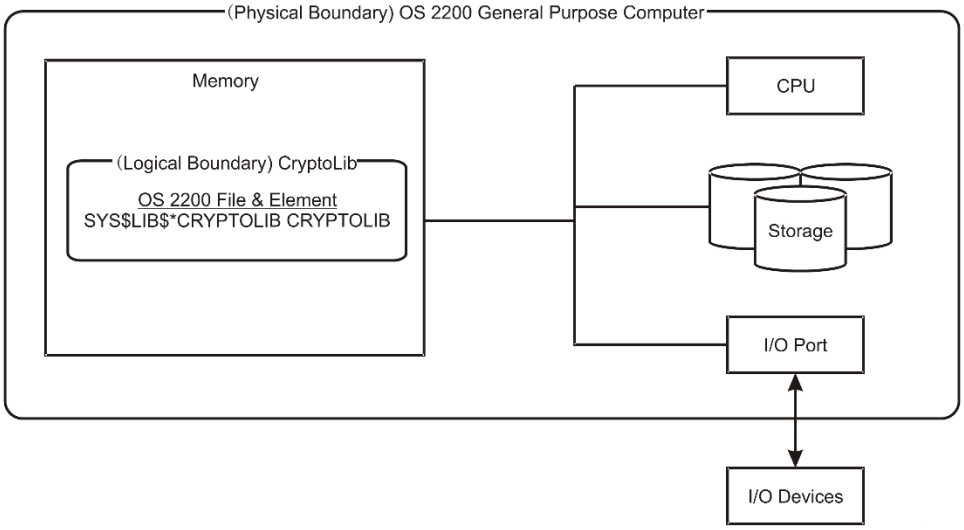
Section	Level
Cryptographic Module Specification	1
Module Ports and Interfaces	1
Roles, Services, and Authentication	1
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A

2.3. Boundary

This module is a software component only, so the logical cryptography boundary contains the software module that makes up the cryptographic module. No source code is provided to the Security Officer or User, only the binary object Module. The physical boundary includes the mainframe containing general

purpose hardware including: the CPU, cache, RAM, disk drives, NICs, and other internal components of the system.

Table 2–2. OS 2200 General Purpose Computer



2.4. Operational Mode

When the module is uninitialized it is considered in non-FIPS mode and prevents any calls to cryptographic functions. Prior to the caller calling CL$init, any calls to cryptographic functions will return with an uninitialized status. The Module must be initialized by calling CL$init. The caller can specify whether they wish to run in FIPS Approved mode or non-FIPS Approved mode by a parameter on CL$init. The CL$init function will perform the Power Up tests, that is, the Known Answer Tests and the module integrity check. The module integrity check is done by using an RSA signature which was computed at build time. If the CL$init computed signature does not match the build signature that is distributed with the Module, the Module will not initialize and no cryptographic functions will be made available. If the Power Up tests fail, the Module will not initialize and no cryptographic functions will be made available and no data output will be output on the Data Output Interface. If the Power Up tests and Module integrity check complete successfully, cryptographic function calls become available to the User or Crypto-officer.

To summarize, to operate CryptoLib in a FIPS Approved mode of operation the following are required:

- The installed version of CryptoLib must be FIPS validated, and have no software corrections applied. Not all versions will be FIPS validated. The installation of CryptoLib is performed via SOLAR. The caller can also check a returned parameter on the CL$init call to see if the CryptoLib version in use is FIPS validated. The installed version can also be checked by executing SYSLIB*CRYPTOLIB.CRYPTOINFO utility.
- The caller must specify that it wishes to run in FIPS Approved mode on the CL$init API call.
- The caller must use only approved cryptographic algorithms. See section 2.8.

2.5. Validated Platform

The FIPS 140-2 Lab tested this module on the following: OS 2200 Operating System 18.0 running on a Unisys ClearPath Dorado 880 in single-user mode.

2.6. Vendor Affirmed Platforms

In addition to the validated platform listed above, the Unisys OS 2200 Cryptographic Library Module has been tested by Unisys on the following platforms:

Note that this list was current at the time this Security Policy was created. The FIPS mode of the Unisys OS 2200 Cryptographic Library Module will be tested by Unisys on future OS 2200 platforms, and this vendor affirmation applies to those systems as well.

Dorado 800 Series
Dorado 4200 Series
Dorado 4300 Series
Dorado 4400 Series
Dorado 6300 Series
Dorado 6400 Series
Dorado 8300 Series
Dorado 8400 Series

Also please note, no claims can be made as to the correct operation of the Unisys OS 2200 Cryptographic Library Module or the security strengths of the generated keys when operating on a platform that is not listed on the validation certificate.

2.7. Ports and Interfaces

The module is a software component and utilizes Application Program Interface (API) as interfaces to the module. The module's API uses the four logical interfaces (Data Input, Data Output, Control Input, Status Output) defined by FIPS 140-2 in the following matter:

Data Input Interface

All data to all functions that is input to and processed by the Module, from the User or Crypto-officer enters via the Data Input Interface.

Data Output Interface

All functions output data (excluding statuses and return codes which are returned via the Status Output Interface) to the User or Crypto-officer via the Data Output Interface. Prior to module Initialization, and during the Power Up Self-Tests, the Data Output interface is prohibited from output. Upon any Self-Test (Conditional or Power Up) failure the module enters an error state and the Data Output Interface is prohibited from output.

Control Input Interface

All input functions that are used to control the operation of the module enter via the Control Input Interface.

Status Output Interface

All functions provide status information back in statuses and return codes from the Module to the User or Crypto-Officer via the Status Output Interface. Some functions, such as CL$init, also provide output parameters that are defined for status output.

Power Interface

This Module is a software only cryptographic module and does not provide power or maintenance access interface beyond the power provided by the computer.

2.8. Approved Cryptographic Algorithms

The Module supports the following FIPS 140-2 algorithms in approved FIPS mode.

Note: *All Module algorithms, both approved and non-approved, are available for use. To run in FIPS approved mode, only FIPS approved algorithms should be used.*

Table 2–3. FIPS 140-2 Approved Algorithms

Algorithm	Type	Standard	Algorithm Mode and Use	CAVP Algorithm Certificate Number
AES (128,192,256)	Symmetric	FIPS 197	CBC,ECB,CFB128,CTR encrypt/decrypt	5416
Triple-DES	Symmetric	FIPS SP800-67 Revision 2	CBC,ECB,CFB64,CTR encrypt/decrypt; 3 different keys	2723
RSA	Asymmetric	FIPS 186-4 FIPS 186-2	Signature Generation PKCS1.5 (2048/3072 with SHA-224, 256, 384, 512) Signature Verification PKCS1.5 (1024/2048/3072 with SHA-1, 224, 256, 384, 512) Note: SHA-1 affirmed for use with protocols only Signature Verification PKCS1.5 (1024/1536/2048/3072/4096 with SHA-1, 224, 256, 384, 512)	2900
SHA (1,224,256,384,512)	Message digest	FIPS 180-4	Message integrity	4347
HMAC-SHA-1, HMAC-SHA-2 (224,256,384,512)	HMAC	FIPS 198-1	Message integrity	3595
DRBG	Deterministic Random Bit Generator	NIST Special Publication 800-90A	Hash_DRBG using SHA-512 supporting predictive resistance	2111

- The vendor affirms symmetric keys are generated per SP 800-133 (unmodified output from a DRBG).

- Per SP800-67 revision 2, the user is responsible for ensuring the module's limit to 2^16 encryptions with the same Triple-DES key.

2.9. Non-Approved But Allowed Cryptographic Algorithms

Table 2–4. FIPS 140-2 Non-Approved But Allowed Algorithms

Algorithm	Type	Standard	Algorithm Mode and Use
NDRNG	Non-Deterministic Random Number Generator		Not user callable; Used to generate Non-deterministic random numbers used to seed the approved DRBG. The minimum number of bits of entropy requested per each GET function call is 1536 bytes. This is enough to ensure that determining the value of the entropy for the DRBG requires at least as many operations as determining the value of any randomly generated key.
Diffie-Hellman	Key agreement		Key agreement [Sizes 2048/4096]

2.10. Non-Approved Cryptographic Algorithms

The Module supports the following FIPS 140-2 algorithms in non-Approved FIPS mode.

Table 2–5. FIPS 140-2 Non-Approved Algorithms

Algorithm	Type	Standard	Algorithm Mode and Use
AES (128,192,256)	Symmetric	SP800-38D	GCM encrypt/decrypt
DES (56)	Symmetric	FIPS 46-3	CBC, ECB, CFB64, CTR encrypt/decrypt
Triple-DES	Symmetric	FIPS SP800-67 Revision 2	2 different keys
Diffie-Hellman	Key agreement		Key agreement [Sizes 1024/1536]
DSA	Asymmetric	FIPS 186-2; PQG(gen) MOD(1024, 2048, 3072); PQG(ver) MOD(1024, 2048, 3072); KEYGEN(Y) MOD(1024, 2048, 3072); SIG(gen) MOD(1024, 2048, 3072); SIG(ver) MOD(1024, 2048, 3072);	Parameter generation and verification; key generation using a FIPS Approved RNG; signature generation and verification using SHA-1 and SHA-2 only.
MD2	Message digest	RFC1115	Message integrity
MD5	Message digest	RFC1321	Message integrity
HMAC-MD5	HMAC	RFC2104	Message integrity
RC4	Symmetric		Encrypt/decrypt
RSA	Symmetric	FIPS 186-2	Key generation; Signature Generation PKCS1.5 (1024 with SHA-1, 224, 256, 384, 512);
Non-approved RNGs	Random Number Generators	FIPS 186-2 General Purpose x-Change Notice w/SHA-1; C programming rand() function interface.	Random number Generator

Section 3
Roles, Services and Authentication

3.1. Roles and Services

There are two roles supported by Cryptographic Library, Crypto Officer and User, as defined in the FIPS 140-2 standard. The Crypto Officer and User are defined as any entity that can access the services provided by the module and each role is implicitly assumed based on the service being executed. There are no restrictions on this access. The Crypto Officer role may perform the install, uninstall, initialization, and self-tests services of the module on the host system. The User role can call any API functions provided by the module.

Table 3–1. Roles and Services

Service	Role	Approved/Non-Approved
General Services		
Installation	Crypto Officer	Approved
Initialization	Crypto Officer	Approved
Self-Tests	Crypto Officer	Approved
Show status	User	Approved
Uninstall	Crypto Officer	Approved
PEM encode/decode	User	Non-Approved
BIGNUM Mathematical	User	Non-Approved
RFC 2898 KDF	User	Non-Approved
ASN.1 encode/decode	User	Approved
Diffie-Hellman (DH)		
Generation of key pair	User	Allowed
Generation of shared secret	User	Allowed
Digital Signature (RSA & DSA)		
DSA Key generation	User	Non-Approved
DSA Signature	User	Non-Approved
DSA Verification	User	Non-Approved
RSA Key generation	User	Non-Approved

Table 3–1. Roles and Services

Service	Role	Approved/Non-Approved
RSA Signature	User	Approved
RSA Verification	User	Approved
Digest Algorithms and Message Authentication (SHA, HMAC)		
MD2	User	Non-Approved
MD5	User	Non-Approved
MD5 HMAC	User	Non-Approved
SHA 1 Digest	User	Approved
SHA 224 Digest	User	Approved
SHA 256 Digest	User	Approved
SHA 384 Digest	User	Approved
SHA 512 Digest	User	Approved
SHA 1 HMAC	User	Approved
SHA 2 HMAC	User	Approved
Non-Approved Random Number Generation		
Non-Approved RNG Seeding	User	Non-Approved
Non-Approved RNG Random number request	User	Non-Approved
Random Number Generation - RNG FIPS 186-2 General Purpose x-Change Notice w/ SHA-1		
RNG Seeding	User	Non-Approved
RNG Random number request	User	Non-Approved
Random Number Generation - DRBG NIST Special Publication 800-90A SHA-512		
DRBG Instantiate	User	Approved
DRBG Reseed	User	Approved
DRBG Random Number Generation	User	Approved
DRBG Uninstantiate	User	Approved
Symmetric Encryption		
AES Decryption	User	Approved
AES Encryption	User	Approved
DES Decryption	User	Non-Approved
DES Encryption	User	Non-Approved
RC4 Decryption	User	Non-Approved
RC4 Encryption	User	Non-Approved

Table 3–1. Roles and Services

Service	Role	Approved/Non-Approved
Triple-DES Decryption	User	Approved
Triple-DES Encryption	User	Approved
Zeroization		
AES Zeroization	User	Approved
DES Zeroization	User	Non-Approved
Triple-DES Zeroization	User	Approved
DSA Zeroization	User	Non-Approved
MD5 HMAC Zeroization	User	Non-Approved
RC4 Zeroization	User	Non-Approved
RSA Zeroization	User	Approved
SHA 1 HMAC Zeroization	User	Approved
SHA 2 HMAC Zeroization	User	Approved
DRBG Zeroization	User	Approved

3.2. Authentication

This Module does not support any authentication or identification services to determine the user.

Section 4
Physical Security

4.1 Module Physical Security

This module is a software library solution, and thus claims no physical security.

Section 5
Cryptographic Key Management

5.1. Keys and CSPs

The table below provided a complete list of Critical Security Parameters (CSPs) used within the module:

Table 5–1. Keys and CSPs Table

Key/ CSP Name	Key/ CSP Type	Generation/ Input	Output	Storage	Zeroization	Use
AES Key	AES 128, 192, or 256-bit key CBC,ECB,CFB128,CTR modes	Internally using the DRBG; Input in plaintext via API	Available in plaintext in user provided storage.	Plaintext, RAM	CL$AES_clearkey(); Reboot Operating System; Cycle host power	Encryption/ Decryption Crypto Officer: Read, Write, Delete User: Read, Write, Delete
Triple-DES Key	3 Different 64-bit Keys The user is responsible for using [each] key for up to 2^16 encryptions. CBC,ECB,CFB64,CTR modes	Internally using the DRBG; Input in plaintext via API	Available in plaintext in user provided storage.	Plaintext, RAM	CL$3DES_clearkey(); Reboot Operating System; Cycle host power	Encryption/ Decryption Crypto Officer: Read, Write, Delete User: Read, Write, Delete
HMAC with SHA-1 and SHA-2 Keys	HMAC-SHA-1, HMAC-SHA-2 (224,256,384,512)	Internally using the DRBG; Input in plaintext via API	None	Plaintext, RAM	Cleared on API call exit; Reboot Operating System; Cycle host power	Keyed Hash Key Crypto Officer: Read, Write, Delete User: Read, Write, Delete
RSA Private Key	Private component of an RSA key pair Key sizes: 1024, 1536 (Sig Ver only); 2048, 3072,4096 (Sig Gen and Sig Ver)	Input in plaintext via API	Available in plaintext in user provided storage.	Plaintext, RAM	CL$RSA_free(); Reboot Operating System; Cycle host power	Signature generation and verification Crypto Officer: Read, Write, Delete User: Read, Write, Delete
RSA Public Key	Private component of an RSA key pair	Internally computed based on the private key; Input in plaintext via API	Available in plaintext in user provided storage.	Plaintext, RAM	CL$RSA_free(); Reboot Operating System; Cycle host power	Signature generation and verification Crypto Officer: Read, Write, Delete User: Read, Write, Delete

DRBG entropy input	Entropy used in the seeding of the DRBG Key size: 256-bits	Internally computed using system obtained unpredictable information that is SHA-512 digested, giving a security strength of 256 bits. Upon each call to the entropy source, 256 bits are returned. Testing estimates that this source is assumed to have full entropy.	None	Plaintext, RAM	Reboot Operating System; Cycle host power	Instantiation and Reseeding of the Hash_DRBG Crypto Officer: Read, Write, Delete User: Read, Write, Delete
DRBG seed	Seed for Hash_DRBG Key size: 888-bits	Internally computed including user supplied personalization bytes	None	Plaintext, RAM	Reboot Operating System; Cycle host power	Instantiation and Reseeding of the Hash_DRBG Crypto Officer: Read, Write, Delete User: Read, Write, Delete
DRBG state	Hash_DRBG secret internal state values V and C Key size: 888-bits	Internally computed using the DRBG seed	None	Plaintext, RAM	CL$random2_uninstantiate(); Reboot Operating System; Cycle host power	Reseeding and random byte generation Crypto Officer: Read, Write, Delete User: Read, Write, Delete
Integrity Test Key	2048-bit RSA Public Key	Hard-coded in Plaintext in the module	None	Hard-coded in Plaintext	Zeroization is not required for Public Keys.	Verify signature of module Crypto Officer: Read User: Read

It is the user's responsibility to ensure that Keys/CSPs generated using non-Approved services are not used for the Approved services, and vice versa.

5.2. Key Generation

This module provides cryptographic functions for key generation. These APIs are called by applications that reside outside the cryptographic boundary. All symmetric and HMAC keys can be created for encryption/decryption and hashing. All keys are generated by using the approved NIST Special Publication 800-90A Deterministic Random Bit Generator. The module does not implement a FIPS-approved RSA key generation method.

5.3. Key Agreement

This module provides non-Approved RSA encrypt/decrypt and Allowed Diffie-Hellman primitives, which calling applications can use to implement approved/allowed key establishment methods.

5.4. Key Storage, Entry, and Output

This module does not store any private or secret critical security parameters (CSPs) in persistent state media. All CSPs generated or passed to the Module remain in the User's (calling application) memory. The User must utilize the API's correctly to guarantee FIPS 140-2 compliance.

5.5. Key Zeroization

This module does not store any private or secret CSPs and it is the User's responsibility to ensure all CSPs are deleted in a way that will make them unavailable. This Module provides functions to overwrite memory that contains keying material with zeroes. Once overwritten, the keying material will become unavailable. It is the User's responsibility to ensure the correct API is called to overwrite the keying material. Rebooting the Operating System or cycling the system's power will also zeroize CSPs.

Section 6
Electromagnetic Interference/ Electromagnetic Compatibility (EMI/EMC)

6.1 Module EMI/EMC

This module runs on hardware that meets the applicable EMI/EMC requirements for FIPS 140-2 specified by 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class A (i.e., for business use).

Section 7
Self-Tests

7.1. Power Up Tests

Power Up tests, also known as Known Answer Tests (KATs), are tests where a cryptographic value is calculated and compared against a result that was previously calculated and stored. Before any User may call into the Module the module must be initialized as described in section 2.4. Specifying FIPS_ON on the call will cause CryptoLib to perform the KATs for approved functions. If any KAT fails, the Module will prevent any cryptographic calls from being performed.

Table 7–1. Approved Cryptographic Power Up Tests

Algorithm	Power Up Self Test
AES	Encrypt/Decrypt KAT
Triple-DES	Encrypt/Decrypt KAT
RSA	Sign/Verify Test
HMAC	HMAC-SHA-1 KAT HMAC-SHA-256 KAT HMAC-SHA-512 KAT
DRBG	NIST Special Publication 800-90A SHA-512 Hash_DRBG KAT and Health Tests.
Software Integrity	RSA signature verification

7.2. Conditional Self-Tests

Conditional self-tests are executed implicitly when they are necessary. This module implements continuous random number generator (CRNG) conditional self-tests as required by FIPS 140-2 Level 1 requirements.

Continuous Random Number Generator (CRNG)

The module implements four random number generators, one of which is FIPS Approved and one of which is Allowed.

Table 7–2. CRNG test FIPS 140-2

Algorithm	Conditional Self-Test
Non-approved RNG	CRNG test FIPS 140-2 Section 4.9.2; Test failure results in an error status and the generated number is discarded. The module will be unusable after a test failure.
Non-approved RNG FIPS 186-2, Appendix 3.1, Change Notice 1	CRNG test FIPS 140-2 Section 4.9.2; Test failure results in an error status and the generated number is discarded. The module will be unusable after a test failure.
NDRNG OS random data digested with SHA-512.	FIPS 140-2 Section 9.8. A CRNG test is applied to the seed material generated by this NDRNG. Test failure results in an error status being returned to the caller. The module will be unusable after a test failure. This NDRNG is used to seed the approved NIST Special Publication 800-90A Hash_DRBG.

Note: The CRNGT test is not required for the SP800-90A-compliant DRBG, as per FIPS IG 9.8.

The following additional Conditional Self-Tests are not applicable to this module:

Bypass Conditional Self-Test (Not Applicable)

This module does not support a bypass capability.

Firmware Load Conditional Self-Test (Not Applicable)

This module does not reference any externally cryptographic modules or devices.

Manual Key Entry Conditional Self-Test (Not Applicable)

This module does not allow keys to be manually entered.

7.3. Critical Function Tests

This module does not implement any critical function tests for FIPS 140-2 Level 1.

Section 8
Design Assurance

8.1 Module Design Assurance

Unisys manages and maintains source code and associated User documentation using the PRIMUS source control system. PRIMUS is also used for product build management, and tracking which versions of the files are used in each release. For information on the secure installation, initialization, and startup of the module please refer to section 2.4.

Section 9
Mitigation of Other Attacks

9.1 Module Attack Mitigation

This module has no prevention against specific attacks made on the module.

Appendix A
Glossary

Abbreviation	Expanded Form
3DES	Triple Digital Encryption Standard
AES	Advanced Encryption Standard
API	Application Programming Interface
CBC	Cipher Block Chaining Mode
CFB	Cipher Feed Back Mode
CMVP	Cryptographic Module Validation Program
CryptoLib	Unisys OS 2200 Cryptographic Library
CSP	Critical Security Parameter
CTR	Counter Mode
DES	Digital Encryption Standard
DH	Diffie-Hellman
DSA	Digital Signature Algorithm
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book Mode
EMC	Electro Magnetic Compatibility
EMI	Electro Magnetic Interference
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standard
GCM	Galois/Counter Mode
HMAC	Hashed MAC
KAT	Known Answer Test
KDF	Key Derivation Function
MAC	Message Authentication Code
MD2	Message Digest Algorithm 2
MD5	Message Digest Algorithm 5
NIST	National Institute of Standards and Technology
RSA	Rivest-Shamir-Adleman

Abbreviation	Expanded Form
RNG	Random Number Generator
SHA	Secure Hash Algorithm