Rambus Inc.

VaultIP RT-130

FIPS 140-3 Non-Proprietary Security Policy

Prepared by:

atsec information security corporation

4516 Seton Center Parkway, Suite 250

Austin, TX 78759

www.atsec.com

Prepared for:

Rambus Inc.

4453 North First Street, Suite 100

San Jose, CA 95134

www.rambus.com

Table of Contents

1 General 5
1.1 Overview 5
1.2 Security Levels 5
1.3 Additional Information 5
2 Cryptographic Module Specification 7
2.1 Description 7
2.2 Tested and Vendor Affirmed Module Version and Identification 9
2.3 Excluded Components 9
2.4 Modes of Operation 9
2.5 Algorithms 9
2.6 Security Function Implementations 15
2.7 Algorithm Specific Information 19
2.8 RBG and Entropy 20
2.9 Key Generation 20
2.10 Key Establishment 21
2.11 Industry Protocols 21
3 Cryptographic Module Interfaces 22
3.1 Ports and Interfaces 22
4 Roles, Services, and Authentication 23
4.1 Authentication Methods 23
4.2 Roles 23
4.3 Approved Services 23
4.4 Non-Approved Services 36
4.5 External Software/Firmware Loaded 38
5 Software/Firmware Security 40
5.1 Integrity Techniques 40
5.2 Initiate on Demand 40
6 Operational Environment 41
6.1 Operational Environment Type and Requirements 41
7 Physical Security 42
7.1 Mechanisms and Actions Required 42
8 Non-Invasive Security 43
9 Sensitive Security Parameters Management 44
9.1 Storage Areas 44
9.2 SSP Input-Output Methods 44
9.3 SSP Zeroization Methods 45
9.4 SSPs 46
9.5 Transitions 55
10 Self-Tests 56
10.1 Pre-Operational Self-Tests 56
10.2 Conditional Self-Tests 56
10.3 Periodic Self-Test Information 59
10.4 Error States 61
10.5 Operator Initiation of Self-Tests 62
11 Life-Cycle Assurance 63
11.1 Installation, Initialization, and Startup Procedures 63
11.2 Administrator Guidance 63

11.3 Non-Administrator Guidance 63
11.4 Design and Rules 63
11.5 End of Life 63
12 Mitigation of Other Attacks 64
Appendix A. Glossary and Abbreviations 65
Appendix B. References 66

List of Tables

Table 1: Security Levels 5
Table 2: Tested Module Identification – Hardware 9
Table 3: Modes List and Description 9
Table 4: Approved Algorithms 13
Table 5: Vendor-Affirmed Algorithms 13
Table 6: Non-Approved, Allowed Algorithms 13
Table 7: Non-Approved, Allowed Algorithms with No Security Claimed 14
Table 8: Non-Approved, Not Allowed Algorithms 15
Table 9: Security Function Implementations 19
Table 10: Entropy Certificates 20
Table 11: Entropy Sources 20
Table 12: Ports and Interfaces 22
Table 13: Authentication Methods 23
Table 14: Roles 23
Table 15: Approved Services 36
Table 16: Non-Approved Services 38
Table 17: Mechanisms and Actions Required 42
Table 18: Storage Areas 44
Table 19: SSP Input-Output Methods 45
Table 20: SSP Zeroization Methods 45
Table 21: SSP Table 1 50
Table 22: SSP Table 2 55
Table 23: Pre-Operational Self-Tests 56
Table 24: Conditional Self-Tests 59
Table 25: Pre-Operational Periodic Information 60
Table 26: Conditional Periodic Information 61
Table 27: Error States 62

List of Figures

Figure 1 - Xilinx Zynq XC7Z045 FPGA 7
Figure 2: Block Diagram 8

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for the Rambus VaultIP RT-130 cryptographic module (hereafter referred to as “the module” or RT-130 or only VaultIP). It contains a specification of the rules under which the module must operate and describes how this module meets the requirements as specified in FIPS PUB 140-3 (Federal Information Processing Standards Publication 140-3) for a Security Level 2 module.

1.2 Security Levels

Section	Title	Security Level
1	General	2
2	Cryptographic module specification	2
3	Cryptographic module interfaces	2
4	Roles, services, and authentication	2
5	Software/Firmware security	2
6	Operational environment	N/A
7	Physical security	2
8	Non-invasive security	N/A
9	Sensitive security parameter management	2
10	Self-tests	2
11	Life-cycle assurance	2
12	Mitigation of other attacks	N/A
	Overall Level	2

Table 1: Security Levels

1.3 Additional Information

VaultIP is a Silicon IP Security Module which includes a complete set of high-level and low-level cryptographic functions. It offers key management and crypto functions needed for platform and application security such as Content Protection and Mobile Payment, and can be used stand-alone or as a 'Root of Trust' to support a Trusted Execution Environment-based platform.

VaultIP completely shields all key and security sensitive data from all CPUs, interfaces and memory. Security sensitive materials are stored as assets that never leave VaultIP in unencrypted and/or non-authenticated form.

Additionally, VaultIP offers hardware security features that are needed when operating in a Trusted Execution Environment (TEE). These features include One-Time-Programmable memory (OTP) access and management, Random Number Generation / entropy source, timers, (short) monotonic/non-volatile counters and import and export of keys and other assets.

The module provides a slave and a master interface. The slave interface is used to receive commands from one or more host CPUs. The master interface is used for autonomous data reads and writes from and to an external memory, flash or interface.

VaultIP supports many Approved or Allowed cryptographic algorithms.

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The primary application of VaultIP is in mobile communications and consumer electronics appliances, where authentication, encrypted content processing using standard protocols, and protection of keys and other sensitive assets are required. VaultIP is best suited for mobile phones, tablets, wireless handsets, PDA-like devices and set top boxes that have the resources and connectivity to download, store and play back digital media content. These small, battery-powered devices require a low power IP solution with these features available in VaultIP.

VaultIP is primarily aimed to be integrated in the design of Application-Specific Integrated Circuits (ASIC). However, it can also be synthesized in a Field-Programmable Gate Array (FPGA).

Module Type: Hardware

Module Embodiment: SingleChip

Module Characteristics [O]: SubChip

Cryptographic Boundary:

The block diagram in Figure 2 shows the cryptographic module boundary represented with the red line box and the physical boundary shown as the most external thick black line. The orange and grey boxes represent the VaultIP components that comprise the IP core. The VaultIP firmware is stored in Program ROM and Program RAM. Figure 2 shows the details of interfaces that cross the security boundary.

Tested Operational Environment's Physical Perimeter (TOEPP):

For the purpose of this Cryptographic Module Validation, VaultIP is synthesized on the Xilinx Zynq XC7Z045 FPGA chip, which belongs to the Zynq-7000 All Programmable SoC series. The Xilinx ZC706 evaluation board for the XC7Z045 SoC provides the hardware environment for developing and evaluating the hardware design of VaultIP.

Photograph and Block Diagram

The module physical boundary is defined by the Xilinx Zynq XC7Z045 FPGA perimeter. The FPGA is a rectangular enclosure measuring approximately 31 mm x 31 mm x 3 mm.

Figure 1 - Xilinx Zynq XC7Z045 FPGA

The block diagram of the sub-chip module is shown below.

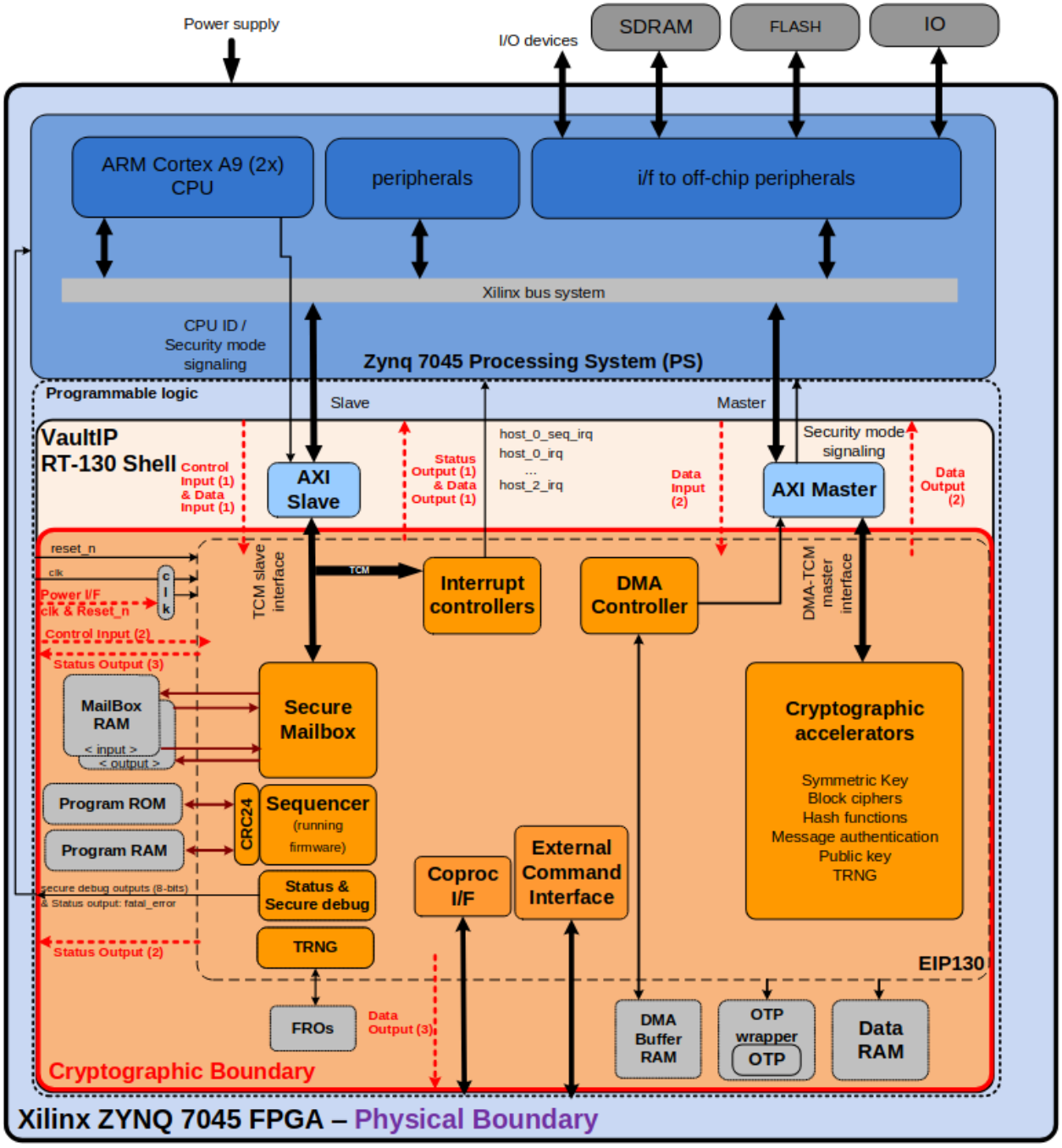

Figure 2: Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
Xilinx Zynq XC7Z045 FPGA	4.3.1	4.6.3	ARM Cortex-A9	

Table 2: Tested Module Identification – Hardware

2.3 Excluded Components

There are no components within the cryptographic boundary excluded from the FIPS 140-3 requirements.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved mode	Automatically entered whenever an approved service is requested	Approved	Equivalent to the indicator of the requested service
Non-approved mode	Automatically entered whenever a non-approved service is requested	Non-Approved	Equivalent to the indicator of the requested service

Table 3: Modes List and Description

Mode Change Instructions and Status:

Once the module is powered on and the self-tests are successful, the module becomes operational and transitions to approved mode automatically.

The mode of operation is assumed based on the service invoked i.e., the module switches back and forth between approved and non-approved modes based on the service called. By default, the module is in approved mode. The non-approved mode of operation is entered when non-approved services are requested (Section 4.4). The module switches back to approved mode of operation when an approved service in Section 4.3 is called.

The module implements the approved service indicator as described in Section 4.3.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A5264	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
AES-CCM	A5264	Key Length - 128, 192, 256	SP 800-38C
AES-CMAC	A5264	Direction - Generation, Verification Key Length - 128, 192, 256	SP 800-38B
AES-CTR	A5264	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A5264	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A5264	Direction - Decrypt, Encrypt IV Generation - External, Internal IV Generation Mode - 8.2.2 Key Length - 128, 192, 256	SP 800-38D
AES-GMAC	A5264	Direction - Decrypt, Encrypt IV Generation - External, Internal IV Generation Mode - 8.2.2 Key Length - 128, 192, 256	SP 800-38D
AES-KWP	A5264	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-XTS Testing Revision 2.0	A5264	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E
Counter DRBG	A5264	Prediction Resistance - No Mode - AES-256 Derivation Function Enabled - No	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A5264	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - extra bits	FIPS 186-5
ECDSA KeyVer (FIPS186-4)	A5264	Curve - P-192	FIPS 186-4
ECDSA KeyVer (FIPS186-5)	A5264	Curve - P-224, P-256, P-384, P-521	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A5264	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512 Component - No	FIPS 186-5
ECDSA SigVer (FIPS186-4)	A5264	Component - No Curve - P-192, P-224, P-256, P-384, P-521 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-4
ECDSA SigVer (FIPS186-5)	A5263	Curve - P-256 Hash Algorithm - SHA2-256	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A5264	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-5

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA-1	A5264	Key Length - Key Length: 112-512 Increment 8	FIPS 198-1
HMAC-SHA2-224	A5264	Key Length - Key Length: 112-512 Increment 8	FIPS 198-1
HMAC-SHA2-256	A5264	Key Length - Key Length: 128-512 Increment 8	FIPS 198-1
HMAC-SHA2-384	A5264	Key Length - Key Length: 192-1024 Increment 8	FIPS 198-1
HMAC-SHA2-512	A5264	Key Length - Key Length: 256-1024 Increment 8	FIPS 198-1
HMAC-SHA3-224	A5264	Key Length - Key Length: 112-1152 Increment 8	FIPS 198-1
HMAC-SHA3-256	A5264	Key Length - Key Length: 128-1088 Increment 8	FIPS 198-1
HMAC-SHA3-384	A5264	Key Length - Key Length: 192-832 Increment 8	FIPS 198-1
HMAC-SHA3-512	A5264	Key Length - Key Length: 256-576 Increment 8	FIPS 198-1
KAS-ECC Sp800-56Ar3	A5264	Domain Parameter Generation Methods - P-224, P-256, P-384, P-521 Function - Key Pair Generation Scheme - fullUnified - KAS Role - Initiator, Responder KDF Methods - oneStepKdf - Key Length - 512 ephemeralUnified - KAS Role - Initiator, Responder KDF Methods - oneStepKdf - Key Length - 512 onePassUnified - KAS Role - Initiator, Responder KDF Methods - oneStepKdf - Key Length - 512 onePassDh - KAS Role - Initiator, Responder KDF Methods - oneStepKdf - Key Length - 512 staticUnified - KAS Role - Initiator, Responder KDF Methods -	SP 800-56A Rev. 3

Algorithm	CAVP Cert	Properties	Reference
		oneStepKdf - Key Length - 512	
KDF SP800-108	A5264	KDF Mode - Counter, Feedback Supported Lengths - Supported Lengths: 112-1152 Increment 8	SP 800-108 Rev. 1
KTS-IFC	A5264	Modulo - 2048, 3072 Key Generation Methods - rsakpg1-basic Scheme - KTS-OAEP-basic - KAS Role - responder Key Transport Method - Key Length - 1024	SP 800-56B Rev. 2
RSA SigGen (FIPS186-5)	A5264	Modulo - 2048, 3072 Signature Type - pkcs1v1.5, pss	FIPS 186-5
RSA SigVer (FIPS186-2)	A5264	Signature Type - PKCS 1.5, PKCSPSS Modulo - 1536	FIPS 186-4
RSA SigVer (FIPS186-4)	A5264	Signature Type - PKCS 1.5, PKCSPSS Modulo - 1024, 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-5)	A5264	Modulo - 2048, 3072 Signature Type - pkcs1v1.5, pss	FIPS 186-5
SHA-1	A5264	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-224	A5264	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A5255	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A5263	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A5264	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A5264	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A5264	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA3-224	A5264	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
SHA3-256	A5264	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
SHA3-384	A5264	Message Length - Message Length: 0-65536 Increment 8	FIPS 202

Algorithm	CAVP Cert	Properties	Reference
SHA3-512	A5264	Message Length - Message Length: 0-65536 Increment 8	FIPS 202

Table 4: Approved Algorithms

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG (symmetric)	Key Type:Symmetric	N/A	SP 800-133r2, Section 4, example 1
CKG (asymmetric)	Key Type:Asymmetric	N/A	SP 800-133r2, Section 4, example 1

Table 5: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

Name	Properties	Implementation	Reference
ECDSA key pair generation	Curves:brainpoolP224r1, brainpoolP256r1, brainpoolP384r1, brainpoolP512r1 (112, 128, 192, 256 bits of security)	Rambus Root of Trust RT-130 (RAM)	FIPS 140-3 IG C.A; RFC 5639
ECDSA signature generation	Curves:brainpoolP224r1, brainpoolP256r1, brainpoolP384r1, brainpoolP512r1 (112, 128, 192, 256 bits of security)	Rambus Root of Trust RT-130 (RAM)	FIPS 140-3 IG C.A; RFC 5639
ECDSA signature verification	Curves:brainpoolP192r1, brainpoolP224r1, brainpoolP256r1, brainpoolP384r1, brainpoolP512r1 (96, 112, 128, 192, 256 bits of security)	Rambus Root of Trust RT-130 (RAM)	FIPS 140-3 IG C.A; RFC 5639
KAS-ECC	Curves:brainpoolP224r1, brainpoolP256r1, brainpoolP384r1, brainpoolP512r1 (112, 128, 192, 256 bits of security)	Rambus Root of Trust RT-130 (RAM)	FIPS 140-3 IG C.A; RFC 5639

Table 6: Non-Approved, Allowed Algorithms

Non-Approved, Allowed Algorithms with No Security Claimed:

Name	Caveat	Use and Function
Image de-obfuscation	Firmware images obfuscated using a non-approved AES key are considered plaintext and unprotected (IG 2.4.A)	De-obfuscation of the RAM firmware image

Name	Caveat	Use and Function
AES SIV	SSPs obfuscated using this algorithm are considered plaintext and unprotected (IG 2.4.A)	De-obfuscation of Asset Key Blobs or OTP Key Blobs

Table 7: Non-Approved, Allowed Algorithms with No Security Claimed

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
AES CTR using external IV	Encryption
AES ICM	Encryption, Decryption
AES GCM using external IV	Authenticated encryption
AES GCM using IV generated with non-approved entropy source configuration	Authenticated encryption
SHA-1 standalone	Message digest
AES CBC-MAC	MAC
AES GMAC using IV generated with non-approved entropy source configuration	MAC
HMAC with key sizes less than 112 bits	MAC
Non-approved entropy source configuration	Random number generation
TwoStep KDF	Key derivation
CKG with key sizes less than 112 bits	Symmetric key generation
CKG with non-approved entropy source configuration	Symmetric key generation
ECDSA key pair generation with non-approved entropy source configuration	Key pair generation
ECDSA with P-192	Key pair generation, Signature generation
ECDSA with SHA-1	Signature generation
ECDSA with non-approved entropy source configuration	Signature generation
ECDSA (pre-hashed message)	Signature generation, Signature verification
RSA with modulus size not 2048 or 3072 bits	Signature generation
RSA with modulus size not 1024, 1536, 2048, or 3072 bits	Signature verification
RSA with SHA-1	Signature generation
RSA-PSS with non-approved entropy source configuration	Signature generation

Name	Use and Function
RSA-PSS with invalid salt length	Signature generation, Signature verification
Diffie-Hellman	Key pair generation, Key pair verification, Shared secret computation
EC Diffie-Hellman	Shared secret computation
Ed25519	Key pair generation, Signature generation, Signature verification
X25519	Key pair generation, Shared secret computation
RSA-OAEP	Key encapsulation
RSA-PKCS#1v1.5	Key encapsulation, Key un-encapsulation
ECIES	Key encapsulation, Key un-encapsulation

Table 8: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Signature verification (ROM)	DigSig-SigVer	Verify a digital signature (ROM)		ECDSA SigVer (FIPS186-5): (A5263) SHA2-256: (A5263)
Encryption	BC-UnAuth	Encrypt a plaintext		AES-CBC: (A5264) AES-CTR: (A5264) AES-ECB: (A5264) AES-XTS Testing Revision 2.0: (A5264)
Decryption	BC-UnAuth	Decrypt a ciphertext		AES-CBC: (A5264) AES-CTR: (A5264) AES-ECB: (A5264) AES-XTS Testing Revision 2.0: (A5264)
Authenticated encryption	BC-Auth	Encrypt a plaintext		AES-CCM: (A5264) AES-GCM: (A5264)

Name	Type	Description	Properties	Algorithms
Authenticated decryption	BC-Auth	Decrypt a plaintext		AES-CCM: (A5264) AES-GCM: (A5264)
Message digest	SHA	Compute a message digest		SHA2-224: (A5264) SHA2-256: (A5264) SHA2-384: (A5264) SHA2-512: (A5264) SHA3-224: (A5264) SHA3-256: (A5264) SHA3-384: (A5264) SHA3-512: (A5264)
MAC	MAC	Compute a MAC tag		AES-CMAC: (A5264) AES-GMAC: (A5264) SHA-1: (A5264) HMAC-SHA-1: (A5264) HMAC-SHA2-224: (A5264) HMAC-SHA2-256: (A5264) HMAC-SHA2-384: (A5264) HMAC-SHA2-512: (A5264) HMAC-SHA3-224: (A5264) HMAC-SHA3-256: (A5264) HMAC-SHA3-384: (A5264) HMAC-SHA3-512: (A5264)
Random number generation	DRBG	Generate random bytes		SHA2-256: (A5255) Counter DRBG: (A5264)
Key wrapping (KTS)	KTS-Wrap	Wrap a key	Key size:128, 192, 256 bits Standard:SP	AES-KWP: (A5264)

Name	Type	Description	Properties	Algorithms
			800-38F IG D.G:approved Key confirmation:no Caveat:Key establishment methodology provides between 128 and 256 bits of security strength	
Key unwrapping (KTS)	KTS-Unwrap	Unwrap a wrapped key	Key size:128, 192, 256 bits Standard:SP 800-56Brev2 IG D.G:approved Key confirmation:no Caveat:Key establishment methodology provides between 128 and 256 bits of security strength	AES-KWP: (A5264)
Key derivation	KBKDF	Derive a key from a key derivation key		KDF SP800-108: (A5264)
Symmetric key generation	CKG	Generate a symmetric key	Standard:SP 800-133r2, Section 4, example 1	CKG (symmetric): () Counter DRBG: (A5264)
Key pair generation	AsymKeyPair-KeyGen	Generate an EC key pair		ECDSA KeyGen (FIPS186-5): (A5264)
Key pair verification	AsymKeyPair-KeyVer	Verify an EC key pair		ECDSA KeyVer (FIPS186-4): (A5264) ECDSA KeyVer (FIPS186-5): (A5264)
Signature generation	DigSig-SigGen	Generate a digital signature		ECDSA SigGen (FIPS186-5): (A5264) RSA SigGen

Name	Type	Description	Properties	Algorithms
				(FIPS186-5): (A5264)
Signature verification	DigSig-SigVer	Verify a digital signature		SHA-1: (A5264) ECDSA SigVer (FIPS186-4): (A5264) ECDSA SigVer (FIPS186-5): (A5264) RSA SigVer (FIPS186-2): (A5264) RSA SigVer (FIPS186-4): (A5264) RSA SigVer (FIPS186-5): (A5264)
KAS	KAS-Full	Establish a shared key among two parties	Curve:P-224, P-256, P-384, P-521 Security strength:112, 128, 192, 256 bits IG:IG D.F Scenario 2, path (2), end-to-end Key confirmation:no Key derivation:KDA (tested as part KAS certificate) Caveat:Key establishment methodology provides between 112 and 256 bits of security strength	KAS-ECC Sp800-56Ar3: (A5264)
KTS-Decapsulation	KTS-Decap	Un-encapsulate an encapsulated key	Modulus size:2048, 3072 bits RSA key generation method:N/A Standard:SP 800-56Brev2 IG	KTS-IFC: (A5264)

Name	Type	Description	Properties	Algorithms
			D.G:approved Key confirmation:no Caveat:Key establishment methodology provides between 112 and 128 bits of security strength	

Table 9: Security Function Implementations

2.7 Algorithm Specific Information

AES-GCM IV (IG C.H): VaultIP is compliant with scenario 2 of FIPS 140-3 IG C.H in [FIPS140-3_IG]. The internal IV is generated in the encryption operation using the RBG-based construction method as defined in section 8.2.2 of [SP800-38D]. VaultIP generates an IV with a length of 96 bits, initialized with random data obtained from the SP800-90Ar1 DRBG implemented in the module.

AES-XTS (IG C.I): The AES algorithm in XTS mode can be only used for the cryptographic protection of data on storage devices, as specified in [SP800-38E]. VaultIP implements a check to ensure that the two AES keys used in XTS-AES algorithm are not identical, meeting the requirement of FIPS 140-3 IG C.I in [FIPS140-3_IG].

SP800-56Ar3 assurances (IG D.F): To comply with the assurances found in Section 5.6.2 of SP 800-56Ar3, the keys for KAS-ECC must be generated using the approved key generation services specified in Section 2.9. The module performs full public key validation on the generated public keys. Additionally, the module performs full public key validation on the received public keys. If the module is used to perform key agreement with the “One-Pass Diffie-Hellman”, “Static Unified Model”, or “One-Pass Unified Model” schemes, a trusted third party is used to obtain the assurance of private key possession for the static peer public key.

RSA modulus size (IG C.F): In compliance with FIPS 186-5, the RSA Signature Generation uses module sizes greater or equal to 2048 bits. The 1536 bits RSA is used in approved mode for FIPS 186-2 signature verification, the 1024-bit modulus is used in approved mode for FIPS 186-4 signature verification and the modulus size for FIPS 186-5 signature verification are 2048 and 3072 bits. All supported modulus sizes have been CAVP tested.

SP800-56Br2 assurances (IG D.G): The entity using the IUT must obtain required assurances listed in section 6.4 of SP 800-56Br2 as follows:

- The entity requesting the RSA key unwrapping (un-encapsulation) service from the module, shall only use an RSA private key that was generated by an active FIPS validated module that implements FIPS 186-5 compliant RSA key generation service and performs the key pair validity and the pairwise consistency as stated in section 6.4.1.1 of the SP 800-56Br2. Additionally, the entity shall renew these assurances over time by using any method described in section 6.4.1.5 of the SP 800-56Br2.

Legacy use (IG C.M): Per SP800-131r2, the SHA-1 with FIPS 186-4 RSA and ECDSA Digital Signature Verification is used in approved mode (for legacy use), the FIPS 186-4 ECDSA Signature Verification with P-192 is used in approved mode (for legacy use), RSA Digital Signature Verification is used in approved mode (for legacy use) with 1024-bit or 1536-bit modulus.

2.8 RBG and Entropy

Cert Number	Vendor Name
E167	Rambus Inc.

Table 10: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
EIP130 TRNG Entropy Source	Physical	Xilinx Zynq XC7Z045 FPGA	256 bits	Full Entropy	SHA2-256 (A5255)

Table 11: Entropy Sources

The module provides an SP800-90Ar1-compliant Deterministic Random Bit Generator (DRBG) using CTR_DRBG mechanism with AES-256 for generation of key components of asymmetric keys, and random number generation. The DRBG does not employ a derivation function. The DRBG is seeded and reseeded with 384 bits of entropy input (corresponding to 384 bits of entropy) provided from the entropy source inside the module. This corresponds to scenario 1 of IG 9.3.A.

The module complies with the Public Use Document (URL provided in section 11.2) for ESV certificate E167 by reading entropy data from the SHA2-256 conditioning function, which corresponds to the conditioned GetEntropy() function. Outputs of multiple GetEntropy() calls are concatenated to receive the entropy input length greater than 256 bits. The output is truncated to get the entropy input string which is not a multiple of 256. The 384 bits of entropy source output is obtained by calling the GetEntropy() twice, with each call providing 256 bits of output. The second call output is truncated to 128 bits and concatenated to the 256-bit output from the first call. The operational environment on the ESV certificate is identical to the Xilinx Zynq XC7Z045 FPGA, in which the sub-chip components are contained. There are no maintenance requirements for the entropy source.

2.9 Key Generation

VaultIP provides services for generating symmetric and asymmetric keys compliant with [SP800-133r2] section 4 example 1 (vendor affirmed).

VaultIP implements symmetric key generation for AES and HMAC keys (”Asset Load (random)” service), using random data obtained from a Deterministic Random Bit Generator (DRBG) compliant with [SP800-90Ar1].

VaultIP implements asymmetric key generation for ECDSA and EC Diffie-Hellman key pairs ("Key pair generation" service) with the following methods:

- ECDSA key pairs are generated in accordance with [SP800-133r2] section 5.1 which maps to [FIPS186-5] appendix A.3.1. A seed (i.e. the random value) used in asymmetric key generation is directly obtained from the [SP800-90Ar1] DRBG.
- EC Diffie-Hellman key pairs are generated in accordance with [SP800-133r2] section 5.2 i.e. key generation method specified in [SP800-56Ar3] section 5.6.1.2.1 used by approved key-establishment schemes which maps to [FIPS186-5].

Intermediate key generation values are not output from the cryptographic module during or after processing the service.

VaultIP implements a key-based key derivation function (KBKDF) in Counter or Feedback modes ("Asset Load (derive)" service) using HMAC-SHA-256 or AES-CMAC [SP800-108r1-upd1].

2.10 Key Establishment

Vault IP also provides EC Diffie-Hellman key agreement compliant with [SP800-56Ar3] and using SHA-256 as a one-step key derivation function compliant with section 4.1 of [SP800-56Cr2] according to scenario 2 path (2) of IG D.F. The key agreement scheme provides between 112 and 256 bits of security strength.

VaultIP provides SSP transport to the dynamic assets entered in encrypted form:

- Key wrapping with AES-KWP, 128, 192 or 256-bit keys, compliant with [SP800-38F] and IG D.G. Security strength ranges from 128 to 256 bits.
- Key un-encapsulation with KTS-IFC and 2048, 3072-bit keys, compliant with [SP800-56Br2] and IG D.G. Security strength is 112 or 128 bits.

2.11 Industry Protocols

The module does not implement any industry protocol.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
TCM slave	Data Input Data Output Control Input Status Output	Service requests, service input data, service output data, service result codes
DMA-TCM master	Data Input Data Output	Bulk service input and output data
Coprocessor interface	Data Output	Asset (SSP) data
MODULE_STATUS register	Status Output	Module status
soft_reset pin	Control Input	Soft reset
abort_req pin	Control Input	Soft reset
reset_n pin	Control Input	Hard reset
clk pin	Control Input	Clock signal
fatal_error pin	Status Output	Fatal error
power pin	Power	Power

Table 12: Ports and Interfaces

The slave and master interfaces connect the VaultIP RT-130 to the AXI bus system. The slave interface is used to receive commands from one or more host CPUs and send the appropriate response. The master interface is used for autonomous data reads and writes from and to an external memory, flash or interface.

4 Roles, Services, and Authentication

4.1 Authentication Methods

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
Login service	32-bit PIN value must be provided via the Login service	Module compares 32-bit PIN value provided by operator with Login PIN provisioned during module installation	32 bits (guess probability = 1/2^32, approx. 10^-9.63)	22.74 bits (guess probability = 614/2^32, approx. 10^-6.84) at 614 attempts per minute

Table 13: Authentication Methods

The Crypto Officer role is initialized via the “Provision Random HUK” service, which accepts the 32-bit Login PIN and instructs the module to generate the Hardware Unique Key (HUK) and install the 32-bit Login PIN. The Login PIN is stored in One Time Programmable (OTP) memory and is protected against disclosure, modification, and substitution like any CSP. It cannot be altered except by zeroization of the whole OTP memory.

After power-up, the module will require the authentication of the Crypto Officer role before allowing execution of most services (exceptions listed in the table below). Authentication is performed through the use of a 32-bit Login PIN provided in the input parameters of the Login service. The module compares this PIN with the 32-bit Login PIN stored in the OTP during installation. If the comparison succeeds, then the Login service succeeds and the module is unlocked. Otherwise, the module enters the firmware error state and must be hard reset to allow a new authentication attempt.

The Crypto Officer role is always authenticated, both in approved mode and non-approved modes of operation. No authentication data can be output by any of the available services.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	Login service

Table 14: Roles

No support is provided for multiple concurrent operators.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
System Boot	Continue initialization of the module by loading	FW accepted bit is set in the MODULE_ST	Firmware signature, de-obfuscatio	N/A	Signature verification (ROM)	Unauthenticated

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	the RAM firmware image	ATUS register	n key, ECDSA public key, de-obfuscation IV			
Encryption	Encrypt a plaintext	FAsvc bit is set in the service output	Plaintext, IV (if applicable), asset store reference to key	Ciphertext	Encryption	Crypto Officer - Static AES key: E - Dynamic AES key: E
Decryption	Decrypt a ciphertext	FAsvc bit is set in the service output	Ciphertext, IV (if applicable), asset store reference to key	Plaintext	Decryption	Crypto Officer - Static AES key: E - Dynamic AES key: E
Authenticated Encryption	Encrypt a plaintext	FAsvc bit is set in the service output	Plaintext, IV, asset store reference to key	Ciphertext, MAC tag	Authenticated encryption	Crypto Officer - Static AES key: E - Dynamic AES key: E
Authenticated Decryption	Decrypt a ciphertext	FAsvc bit is set in the service output	Ciphertext, IV, MAC tag, asset store reference to key	Plaintext or fail	Authenticated decryption	Crypto Officer - Static AES key: E - Dynamic AES key: E
Hash	Compute a message digest	FAsvc bit is set in the service output	Message	Digest value	Message digest	Crypto Officer
MAC Tag Generation	Generate a MAC tag	FAsvc bit is set in the service output	Message, asset store reference to key	MAC tag	MAC	Crypto Officer - Static AES key: E - Dynamic AES key: E - Static HMAC key: E - Dynamic

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						HMAC key: E
MAC Tag Verification	Verify a MAC tag for a message	FAsvc bit is set in the service output	Message, MAC tag, asset store reference to key	Pass/fail	MAC	Crypto Officer - Static AES key: E - Dynamic AES key: E - Static HMAC key: E - Dynamic HMAC key: E
RNG Configuration (reseed)	Reseed the DRBG	FAsvc bit is set in the service output	N/A	N/A	Random number generation	Crypto Officer - Entropy input: G,E,Z - DRBG seed: G,E,Z - Internal state (V, Key): G,E,Z
RNG Get Random Number	Generate random bytes using the DRBG or entropy source	FAsvc bit is set in the service output	Output size	Random bytes	Random number generation	Crypto Officer - Internal state (V, Key): E
RNG Post-Processing Verification	Verify the conditioning component and DRBG self-tests using known inputs	N/A	Known noise input or DRBG state values	Test random bits	None	Crypto Officer
RNG Hardware Self-Test Verification	Verify the entropy source health tests using known inputs	N/A	Health test parameters, known noise input	Result	None	Crypto Officer
Symmetric Wrap	Wrap key material using AES KWP	FAsvc bit is set in the service output	Key wrapping key or asset store	Wrapped key material	Key wrapping (KTS)	Crypto Officer - Static AES key: E - Dynamic

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			reference to key wrapping key, key material to be wrapped			AES key: E - AES key wrapping key: W,E
Symmetric Unwrap	Unwrap key material using AES KWP	FAsvc bit is set in the service output	Key wrapping key, wrapped key material	Unwrapped key material	Key unwrapping (KTS)	Crypto Officer - AES key wrapping key: W,E
Asset Create	Allocate space for an asset in the Dynamic Asset Store	N/A	Asset size	Asset store reference to SSP	None	Crypto Officer
Static Asset Search	Search for an asset in the Static Asset Store	N/A	Asset number	Asset store reference to SSP	None	Unauthenticated
Asset Load (derive)	Derive a key from a key derivation key and store the result in the Dynamic Asset Store	FAsvc bit is set in the service output	Asset store reference to key derivation key, asset store reference to derived key	N/A	Key derivation	Crypto Officer - HUK: E - Dynamic AES key: G - Dynamic HMAC key: G - Static key derivation key: E - Dynamic key derivation key: G,E
Asset Load (import)	De-obfuscate obfuscated key material using AES-SIV and store the result in the Dynamic Asset Store (AES key,	N/A	Obfuscated key material, asset store reference to key	N/A	None	Crypto Officer - Dynamic AES key: W - Dynamic HMAC key: W - Dynamic key

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	HMAC key, key derivation key, EC public/private key, or RSA public/private key)					derivation key: W - Dynamic EC public key: W - Dynamic EC private key: W - Dynamic RSA public key: W - Dynamic RSA private key: W
Asset Load (random)	Generate symmetric key material using the DRBG, store the result in the Dynamic Asset Store, and optionally output the obfuscated result	FAsvc bit is set in the service output	Key size, asset store reference to key material	Obfuscated key material (optional)	Symmetric key generation	Crypto Officer - Internal state (V, Key): E - Dynamic AES key: G,R - Dynamic HMAC key: G,R - Dynamic key derivation key: G,R
Asset Load (plaintext)	Store plaintext key material in the Dynamic Asset Store and optionally output the obfuscated result	N/A	Plaintext key material, asset store reference to key	Obfuscated key material (optional)	None	Crypto Officer - Dynamic AES key: R,W - Dynamic HMAC key: R,W - Dynamic key derivation key: R,W - Dynamic EC public key: R,W - Dynamic EC private key: R,W - Dynamic RSA public

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						key: R,W - Dynamic RSA private key: R,W
Asset Load (unwrap)	Unwrap wrapped key material using AES KWP and store the result in the Dynamic Asset Store	FAsvc bit is set in the service output	Wrapped key material, asset store reference to key	N/A	Key unwrapping (KTS)	Crypto Officer - Dynamic AES key: W - Dynamic HMAC key: W - Dynamic key derivation key: W - Dynamic EC public key: W - Dynamic EC private key: W - Dynamic RSA public key: W - Dynamic RSA private key: W
Asset Delete	Delete an asset from the Dynamic Asset Store	N/A	Asset store reference to SSP	N/A	None	Crypto Officer - Dynamic AES key: Z - Dynamic HMAC key: Z - Dynamic key derivation key: Z - Dynamic EC public key: Z - Dynamic EC private key: Z - Dynamic RSA public key: Z - Dynamic

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						RSA private key: Z
Asset Export	Export an asset via the coprocessor interface	N/A	Asset store reference to SSP, coprocessor selection	Asset data	None	Crypto Officer - Dynamic AES key: R - Dynamic HMAC key: R - Dynamic key derivation key: R - Dynamic EC public key: R - Dynamic EC private key: R - Dynamic RSA public key: R - Dynamic RSA private key: R
Public Data Read	Read a Public Data asset	N/A	Asset store reference	Public Data asset	None	Unauthenticated
Monotonic Counter Read	Read a Monotonic Counter asset	N/A	Asset store reference	Monotonic Counter value	None	Unauthenticated
Monotonic Counter Increment	Increment a Monotonic Counter asset	N/A	Asset store reference	N/A	None	Crypto Officer
OTP Data Write	De-obfuscate obfuscated key material using AES-SIV and store the result in the Static Asset Store (HUK, AES key, HMAC key, key derivation	N/A	Obfuscated key material, asset store reference to key	N/A	None	Crypto Officer - HUK: W - Static AES key: W - Static HMAC key: W - Static key derivation key: W - Static EC

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	key, EC public/private key, or RSA public/private key)					public key: W - Static EC private key: W - Static RSA public key: W - Static RSA private key: W
Provision Random HUK	Generate a random HUK using the DRBG and store the result (together with the provided Login PIN) in the Static Asset Store	HUK has FIPSApproved bit set	Login PIN, HUK size	AES-SIV obfuscated HUK	Symmetric key generation	Crypto Officer - Internal state (V, Key): E - Login PIN: W - HUK: G,R
Secure Timer	Start, stop, or read a timer	N/A	Asset store reference	Timer value	None	Crypto Officer
Dynamic Asset Store Reset	Zeroize the Dynamic Asset Store	N/A	N/A	N/A	None	Crypto Officer - Dynamic AES key: Z - Dynamic HMAC key: Z - Dynamic key derivation key: Z - Dynamic EC public key: Z - Dynamic EC private key: Z - Dynamic RSA public key: Z - Dynamic

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						RSA private key: Z
Show status	Return information about the module state	N/A	N/A	Module status	None	Unauthenticated
Show version	Return information about the module hardware and firmware	N/A	N/A	Module version	None	Unauthenticated
Self-Test	Perform all CASTs	FAsvc bit is set in the service output	N/A	N/A	None	Crypto Officer
Reset	Reset the module to its initial state	N/A	N/A	N/A	None	Crypto Officer
Login	Authenticate as the Crypto Officer	N/A	Login PIN	N/A	None	Unauthenticated - Login PIN: E
Authenticated Unlock Start	Step 1 in the two-step protocol to enable Secure Debug for peripherals	FAsvc bit is set in the service output	Asset store reference to authentication key	Nonce	Random number generation	Crypto Officer - Internal state (V, Key): E
Authenticated Unlock Verify	Step 2 in the two-step protocol to enable Secure Debug for peripherals	FAsvc bit is set in the service output	Nonce, signature	N/A	Signature verification	Crypto Officer - Static EC public key: E - Dynamic EC public key: E
Set Secure Debug	Activate a Secure Debug port for a peripheral	N/A	Port number	N/A	None	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Key pair generation	Generate a key pair	FAsvc bit is set in the service output	Curve or modulus size, asset store references to key pair	N/A	Key pair generation	Crypto Officer - Internal state (V, Key): E - Static EC public key: G - Dynamic EC public key: G - Static EC private key: G - Dynamic EC private key: G - Static RSA public key: G - Dynamic RSA public key: G - Static RSA private key: G - Dynamic RSA private key: G - Intermediate key generation value: G,E,Z
Key pair verification	Verify a key pair	FAsvc bit is set in the service output	Asset store references to key pair	Pass/fail	Key pair verification	Crypto Officer - Static EC public key: G - Dynamic EC public key: G - Static EC private key: G - Dynamic EC private key: G

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- NIST SP 800-56Arev3 domain parameters: E
Signature generation	Generate a signature for a message	FAsvc bit is set in the service output	Message, asset store reference to private key	Signature	Signature generation	Crypto Officer - Static EC private key: E - Dynamic EC private key: E - Static RSA private key: E - Dynamic RSA private key: E
Signature verification	Verify a signature for a message	FAsvc bit is set in the service output	Message, signature, asset store reference to public key	Pass/fail	Signature verification	Crypto Officer - Static EC public key: E - Dynamic EC public key: E - Static RSA public key: E - Dynamic RSA public key: E
KAS	Establish a shared key among two parties	FAsvc bit is set in the service output	Asset store reference(s) to owner private key(s), asset store reference(s) to peer public key(s), asset store	N/A	KAS	Crypto Officer - Static AES key: G - Dynamic AES key: G - Static HMAC key: G - Dynamic HMAC key: G - Static key derivation key: G

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			reference to shared key			- Dynamic key derivation key: G - Static EC public key: E - Dynamic EC public key: E - Static EC private key: E - Dynamic EC private key: E - Shared secret: G,E,Z - NIST SP 800-56Arev3 domain parameters: E
Key un-encapsulation	Un-encapsulate key material using KTS-IFC	FAsvc bit is set in the service output	Encapsulated key material, asset store reference to owner private key, asset store reference to un-encapsulated key	N/A	KTS-Decapsulation	Crypto Officer - Dynamic AES key: W - Dynamic HMAC key: W - Dynamic key derivation key: W - Dynamic EC public key: W - Dynamic EC private key: W - Dynamic RSA public key: W - Static RSA public key: E - Dynamic

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						RSA private key: W,E
Register Read	Read from a specified address	N/A	Address	Read data	None	Unauthenticated
Register Write	Write to a specified address	N/A	Address, write data	N/A	None	Crypto Officer
Clock Switch	Activate/deactivate clocks	N/A	Clock configuration	N/A	None	Crypto Officer
Zeroize Output Mailbox	Zeroize the output mailbox	N/A	N/A	N/A	None	Crypto Officer
Select OTP Zeroize	Step 1 in the OTP zeroization process	N/A	N/A	N/A	None	Crypto Officer
Zeroize OTP	Step 2 in the OTP zeroization process	N/A	N/A	N/A	None	Crypto Officer - Login PIN: Z - HUK: Z - Static AES key: Z - Static HMAC key: Z - Static key derivation key: Z - Static EC public key: Z - Static EC private key: Z - Static RSA public key: Z - Static RSA private key: Z - NIST SP 800-56Arev3 domain

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						parameters: Z
Sleep Mode	Move the module to the Sleep Mode	N/A	N/A	N/A	None	Crypto Officer
Resume From Sleep	Restore the module to the operational state	N/A	N/A	N/A	None	Unauthenticated
Firmware Check	Verify a firmware image using ECDSA signature verification	FAsvc bit is set in the service output	Firmware image, firmware signature verification key	Pass/fail	Signature verification	Crypto Officer - Firmware signature verification key: W,E
Update RollbackID	Update the RollbackID	N/A	New RollbackID	N/A	None	Crypto Officer
Hard reset	Forcefully reset the module using a hardware pin	N/A	N/A	N/A	None	Unauthenticated

Table 15: Approved Services

The following convention is used to specify access rights to SSPs:

- **Generate (G)**: The module generates or derives the SSP.
- **Read (R)**: The SSP is read from the module (e.g. the SSP is output).
- **Write (W)**: The SSP is updated, imported, or written to the module.
- **Execute (E)**: The module uses the SSP in performing a cryptographic operation.
- **Zeroize (Z)**: The module zeroizes the SSP.

Each service utilizing approved security functions provides an indicator implemented as follows:

- Each service will return output parameters as part of an “output token”.
- Security-relevant services will include a field in the output token called the “FIPS-Approved Service indication” (FAsvc). If the FAsvc bit is not present, the service is not a security-relevant service.
- If this bit is present and set, the service is considered approved. If the bit is present but not set, the service is considered non-approved.
- The Provision Random HUK service does not set the “FAsvc” bit in the output token; instead, it sets the “FIPSApproved” bit on the generated HUK as the service indicator.

4.4 Non-Approved Services

Name	Description	Algorithms	Role
Encryption	Encrypt a plaintext	AES CTR using external IV AES ICM	Crypto Officer
Decryption	Decrypt a ciphertext	AES CTR using external IV AES ICM	Crypto Officer
Authenticated Encryption	Encrypt a plaintext	AES GCM using external IV AES GCM using IV generated with non-approved entropy source configuration	Crypto Officer
Hash	Compute a message digest	SHA-1 standalone	Crypto Officer
MAC Tag Generation	Generate a MAC tag	AES CBC-MAC AES GMAC using IV generated with non-approved entropy source configuration HMAC with key sizes less than 112 bits	Crypto Officer
MAC Tag Verification	Verify a MAC tag for a message	AES CBC-MAC HMAC with key sizes less than 112 bits	Crypto Officer
RNG Configuration (reconfiguration)	Use non-approved entropy source configuration	Non-approved entropy source configuration	Crypto Officer
RNG Get Random Number	Generate random bytes using a non-approved entropy source configuration	Non-approved entropy source configuration	Crypto Officer
Asset Load (derive)	Derive a key from a key derivation key and store the result in the Dynamic Asset Store	TwoStep KDF	Crypto Officer
Asset Load (random)	Generate symmetric key material using the DRBG, store the result in the Dynamic Asset Store, and optionally output the obfuscated result	CKG with key sizes less than 112 bits CKG with non-approved entropy source configuration	Crypto Officer
Authenticated Unlock Start	Step 1 in the two-step protocol to enable Secure Debug for peripherals	Non-approved entropy source configuration	Crypto Officer
Key pair generation	Generate a key pair	ECDSA key pair generation with non-approved entropy source configuration ECDSA with P-192	Crypto Officer

Name	Description	Algorithms	Role
		Diffie-Hellman Ed25519 X25519	
Key pair verification	Verify a key pair	Diffie-Hellman	Crypto Officer
Signature generation	Generate a signature for a message	ECDSA with P-192 ECDSA with SHA-1 ECDSA with non-approved entropy source configuration ECDSA (pre-hashed message) RSA with modulus size not 2048 or 3072 bits RSA with SHA-1 RSA-PSS with non-approved entropy source configuration RSA-PSS with invalid salt length Ed25519	Crypto Officer
Signature verification	Verify a signature for a message	ECDSA (pre-hashed message) RSA with modulus size not 1024, 1536, 2048, or 3072 bits RSA-PSS with invalid salt length Ed25519	Crypto Officer
Key agreement	Establish a shared key among two parties	Diffie-Hellman X25519	Crypto Officer
Shared secret computation	Establish a shared secret among two parties	Diffie-Hellman EC Diffie-Hellman X25519	Crypto Officer
Key encapsulation	Encapsulate key material	RSA-OAEP RSA-PKCS#1v1.5 ECIES	Crypto Officer
Key un-encapsulation	Un-encapsulate key material	RSA-PKCS#1v1.5 ECIES	Crypto Officer

Table 16: Non-Approved Services

4.5 External Software/Firmware Loaded

Upon startup, the ROM firmware component loads the RAM firmware from external storage into the sub-chip cryptographic subsystem. The integrity of the RAM firmware is determined by verifying an ECDSA P-256 with SHA2-256 signature stored in the firmware that was computed at build time. If the signature verification fails, the firmware load test fails. The

public key used to verify this signature is provided with the RAM firmware, the private key associated with this public key is controlled by the vendor. The hash of the public key is compared with a hash value stored in ROM to ensure the authenticity of the provided public key.

All data output is inhibited during the execution of the firmware load test and the firmware loading process.

5 Software/Firmware Security

5.1 Integrity Techniques

The module employs a CRC24 as integrity technique to integrity verify the ROM code during startup.

5.2 Initiate on Demand

Integrity tests are performed when the module is powered on. The integrity test can be performed on demand by powering off and powering on the module.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Non-Modifiable

How Requirements are Satisfied: N/A

7 Physical Security

7.1 Mechanisms and Actions Required

Mechanism	Inspection Frequency	Inspection Guidance
Tamper-evident coating covering the FPGA components: integrated heat spreader, substrate with solder ball grid array, silicon chip with TMI	N/A	N/A

Table 17: Mechanisms and Actions Required

The integrated heat spreader (IHS) serves as a protective shell for the processing silicon chip. The IHS lid, the substrate with solder ball grid array, and the silicon chip covered with Thermal Interface material (TMI) are production-grade components. They provide opacity in the visible spectrum.

8 Non-Invasive Security

The module does not implement any non-invasive security mechanisms.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
Static Asset Store	SSPs are stored in One Time Programmable (OTP) memory	Static
Dynamic Asset Store	SSPs are maintained in Data RAM	Dynamic

Table 18: Storage Areas

The Static Asset Store and Dynamic Asset Store maintain internal separation of the SSPs (including CSPs) in approved and non-approved modes of operation. Each asset internally maintains a "Fips Approved" bit which indicates if the asset can be used in an approved service or not.

SSPs that are not stored in an Asset Store are only transiently used for a specific service. They are by definition exclusive between approved and non-approved services.

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
OTP coprocessor export	Static Asset Store	Coprocessor interface	Plaintext	Manual	Electronic	
OTP obfuscated import	Operator calling application (TOEPP)	Static Asset Store	Plaintext	Manual	Electronic	
OTP obfuscated export	Static Asset Store	Operator calling application (TOEPP)	Plaintext	Manual	Electronic	
RAM coprocessor export	Dynamic Asset Store	Coprocessor interface	Plaintext	Manual	Electronic	
RAM plaintext import	Operator calling application (TOEPP)	Dynamic Asset Store	Plaintext	Manual	Electronic	
RAM obfuscated import	Operator calling application (TOEPP)	Dynamic Asset Store	Plaintext	Manual	Electronic	

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
RAM obfuscated export	Dynamic Asset Store	Operator calling application (TOEPP)	Plaintext	Manual	Electronic	
RAM encrypted import	Operator calling application (TOEPP)	Dynamic Asset Store	Encrypted	Manual	Electronic	Key unwrapping (KTS)
RAM encrypted export	Dynamic Asset Store	Operator calling application (TOEPP)	Encrypted	Manual	Electronic	Key wrapping (KTS)

Table 19: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Asset Delete service	Zeroize and delete an SSP from the Dynamic Asset Store	Memory occupied by SSPs is overwritten with zeroes, which renders the SSP values irretrievable	By invoking the Asset Delete service
Zeroize OTP service	Zeroize all OTP memory, including all SSPs contained in the Static Asset Store	OTP memory is overwritten by ones, which renders the SSP values for all SSPs in the Static Asset Store irretrievable	By invoking the Select OTP Zeroize and Zeroize OTP services
Dynamic Asset Store Reset service	Zeroize all SSPs contained in the Dynamic Asset Store	Dynamic Asset Store memory is overwritten by zeroes, which renders the SSP values for all SSPs in the Dynamic Asset Store irretrievable	By invoking the Dynamic Asset Store Reset service
Automatic	SSP is automatically zeroized by the module when no longer needed	Memory occupied by SSPs is overwritten with zeroes, which renders the SSP values irretrievable	N/A
Module reset	De-allocates the volatile memory used to store SSPs in the Dynamic Asset Store	Volatile memory used by the module is overwritten within nanoseconds when the module is reset	Via the soft_reset, abort_req, or reset_n pins, or by invoking the Reset service

Table 20: SSP Zeroization Methods

SSP zeroization when overwriting RAM or OTP memory is performed without delay. Additionally, control is not returned to the operator until the zeroization is completed, preventing any potential compromise of the zeroized SSP. All data output is inhibited during zeroization.

The Asset Delete, Zeroize OTP, and Dynamic Asset Store Reset services provide an explicit indicator when the service (i.e., zeroization) completes: Result output parameter. Automatic SSP zeroization is indicated to the operator by successful completion of the relevant service.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Login PIN	PIN value used to authenticate the Crypto Officer	32 bits - 32 bits	Authentication data - CSP			
HUK	Hardware Unique Key used to derive trusted keys	128, 256 bits - 128, 256 bits	Root key - CSP	Symmetric key generation		Key derivation
Static AES key	AES key used for encryption, decryption, and computing MAC tags	XTS: 256, 512 bits; other modes: 128, 192, 256 bits - XTS: 128, 256 bits; other modes: 128, 192, 256 bits	Symmetric key - CSP	Symmetric key generation Key derivation	Key unwrapping (KTS) KAS KTS-Decapsulation	Encryption Decryption Authenticated encryption Authenticated decryption MAC Key wrapping (KTS)
Dynamic AES key	AES key used for encryption, decryption, and computing MAC tags	XTS: 256, 512 bits; other modes: 128, 192, 256 bits - XTS: 128, 256 bits; other modes: 128, 192, 256 bits	Symmetric key - CSP	Symmetric key generation Key derivation	Key unwrapping (KTS) KAS KTS-Decapsulation	Encryption Decryption Authenticated encryption Authenticated decryption MAC
AES key wrapping key	AES key used for wrapping	128, 192, 256 bits -	Symmetric key - CSP		Key wrapping (KTS)	Key wrapping (KTS)

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	and unwrapping	128, 192, 256 bits			Key unwrapping (KTS)	Key unwrapping (KTS)
Static HMAC key	HMAC key used for computing MAC tags	112-1152 bits - 112-256 bits	Symmetric key - CSP	Symmetric key generation Key derivation	Key unwrapping (KTS) KAS KTS-Decapsulation	MAC
Dynamic HMAC key	HMAC key used for computing MAC tags	112-1152 bits - 112-256 bits	Symmetric key - CSP	Symmetric key generation Key derivation	Key unwrapping (KTS) KAS KTS-Decapsulation	MAC
Entropy input	Entropy input used to seed the DRBG	384 bits - 384 bits	Entropy input - CSP	Random number generation		Random number generation
DRBG seed	DRBG seed derived from the entropy input	384 bits - 384 bits	Seed - CSP	Random number generation		Random number generation
Internal state (V, Key)	Internal state of CTR_DRBG instance	384 bits - 256 bits	Internal state - CSP	Random number generation		Random number generation
Static key derivation key	Symmetric key used to derive symmetric keys	112-1152 bits - 112-256 bits	Symmetric key - CSP	Symmetric key generation Key derivation	Key unwrapping (KTS) KTS-Decapsulation	Key derivation
Dynamic key derivation key	Symmetric key used to derive symmetric keys	112-1152 bits - 112-256 bits	Symmetric key - CSP	Symmetric key generation Key derivation	Key unwrapping (KTS) KTS-Decapsulation	Key derivation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Static EC public key	Public key used for ECDSA and KAS-ECC	P-192, P-224, P-256, P-384, P-521, brainpoolP192r1, brainpoolP224r1, brainpoolP256r1, brainpoolP384r1, brainpoolP512r1 - 96-256 bits	Public key - PSP	Key pair generation	KTS-Decapsulation	Key pair verification Signature verification KAS
Dynamic EC public key	Public key used for ECDSA and KAS-ECC	P-192, P-224, P-256, P-384, P-521, brainpoolP192r1, brainpoolP224r1, brainpoolP256r1, brainpoolP384r1, brainpoolP512r1 - 96-256 bits	Public key - PSP	Key pair generation	KTS-Decapsulation	Key pair verification Signature verification KAS
Static EC private key	Private key used for ECDSA and KAS-ECC	P-224, P-256, P-384, P-521, brainpoolP224r1, brainpoolP256r1, brainpoolP384r1, brainpoolP512r1 - 112-256 bits	Private key - CSP	Key pair generation	KTS-Decapsulation	Key pair verification Signature generation KAS
Dynamic EC private key	Private key used for ECDSA and KAS-ECC	P-224, P-256, P-384, P-521, brainpoolP224r1, brainpoolP256r1, brainpoolP384r1, brainpoolP51	Private key - CSP	Key pair generation	KTS-Decapsulation	Key pair verification Signature generation KAS

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		2r1 - 112-256 bits				
Static RSA public key	Public key used for RSA	1024, 1536, 2048, 3072 bits - 80-132 bits	Public key - PSP		KTS-Decapsulation	Signature verification
Dynamic RSA public key	Public key used for RSA	1024, 1536, 2048, 3072 bits - 80-132 bits	Public key - PSP		KTS-Decapsulation	Signature verification
Static RSA private key	Private key used for RSA	2048, 3072 bits - 110-132 bits	Private key - CSP		KTS-Decapsulation	Signature generation KTS-Decapsulation
Dynamic RSA private key	Private key used for RSA	2048, 3072 bits - 110-132 bits	Private key - CSP		KTS-Decapsulation	Signature generation KTS-Decapsulation
Firmware signature verification key	Public key used for firmware signature verification	P-256 - 128 bits	Public key - PSP			Signature verification
Shared secret	Shared secret established as part of KAS-ECC	224-512 bits - 112-256 bits	Shared secret - CSP		KAS	KAS
Intermediate key generation value	Temporary value generated during key pair generation services	224-4096 bits - 112-256 bits	Intermediate value - CSP			
NIST SP 800-56Arev3 domain parameters	Domain parameters used as part of KAS-ECC	P-192, P-224, P-256, P-384, P-521, brainpoolP192r1, brainpoolP224r1, brainpoolP25	Domain parameter - PSP			KAS

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		6r1, brainpoolP384r1, brainpoolP512r1 - 96-256 bits				

Table 21: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Login PIN	OTP obfuscated import	Static Asset Store:Obfuscated	For the lifetime of the module	Zeroize OTP service	
HUK	OTP obfuscated import OTP obfuscated export	Static Asset Store:Obfuscated	For the lifetime of the module	Zeroize OTP service	
Static AES key	OTP coprocessor export OTP obfuscated import	Static Asset Store:Obfuscated	Until explicitly zeroized	Zeroize OTP service	HUK:Derived From Static key derivation key:Derived From Dynamic key derivation key:Derived From Shared secret:Derived From
Dynamic AES key	RAM coprocessor export RAM plaintext import RAM obfuscated import RAM obfuscated export RAM encrypted import RAM	Dynamic Asset Store:Obfuscated	Until explicitly zeroized or module reset	Asset Delete service Module reset Dynamic Asset Store Reset service	HUK:Derived From Static key derivation key:Derived From Dynamic key derivation key:Derived From Shared secret:Derived From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	encrypted export				
AES key wrapping key	RAM plaintext import		For the duration of the service	Automatic	
Static HMAC key	OTP coprocessor export OTP obfuscated import	Static Asset Store:Obfuscated	Until explicitly zeroized	Zeroize OTP service	HUK:Derived From Static key derivation key:Derived From Dynamic key derivation key:Derived From Shared secret:Derived From
Dynamic HMAC key	RAM coprocessor export RAM plaintext import RAM obfuscated import RAM obfuscated export RAM encrypted import RAM encrypted export	Dynamic Asset Store:Obfuscated	Until explicitly zeroized or module reset	Asset Delete service Dynamic Asset Store Reset service Module reset	HUK:Derived From Static key derivation key:Derived From Dynamic key derivation key:Derived From Shared secret:Derived From
Entropy input			From generation until DRBG seed is created	Automatic	
DRBG seed			While the DRBG is being instantiated	Automatic	Entropy input:Derived From
Internal state (V, Key)			From DRBG instantiation until	Automatic Module reset	DRBG seed:Derived From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
			DRBG termination		
Static key derivation key	OTP coprocessor export OTP obfuscated import	Static Asset Store:Obfuscated	Until explicitly zeroized	Zeroize OTP service	HUK:Derived From Shared secret:Derived From
Dynamic key derivation key	RAM coprocessor export RAM plaintext import RAM obfuscated import RAM obfuscated export RAM encrypted import RAM encrypted export	Dynamic Asset Store:Obfuscated	Until explicitly zeroized or module reset	Asset Delete service Dynamic Asset Store Reset service Module reset	HUK:Derived From Shared secret:Derived From
Static EC public key	OTP coprocessor export OTP obfuscated import	Static Asset Store:Obfuscated	Until explicitly zeroized	Zeroize OTP service	Static EC private key:Paired With
Dynamic EC public key	RAM coprocessor export RAM plaintext import RAM obfuscated import RAM obfuscated export RAM encrypted import RAM	Dynamic Asset Store:Obfuscated	Until explicitly zeroized or module reset	Asset Delete service Dynamic Asset Store Reset service Module reset	Dynamic EC private key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	encrypted export				
Static EC private key	OTP coprocessor export OTP obfuscated import	Static Asset Store:Obfuscated	Until explicitly zeroized	Zeroize OTP service	Static EC public key:Paired With
Dynamic EC private key	RAM coprocessor export RAM plaintext import RAM obfuscated import RAM obfuscated export RAM encrypted import RAM encrypted export	Dynamic Asset Store:Obfuscated	Until explicitly zeroized or module reset	Asset Delete service Dynamic Asset Store Reset service Module reset	Dynamic EC public key:Paired With
Static RSA public key	OTP coprocessor export OTP obfuscated import	Static Asset Store:Obfuscated	Until explicitly zeroized	Zeroize OTP service	Static RSA private key:Paired With
Dynamic RSA public key	RAM coprocessor export RAM plaintext import RAM obfuscated import RAM obfuscated export RAM encrypted import RAM	Dynamic Asset Store:Obfuscated	Until explicitly zeroized or module reset	Asset Delete service Dynamic Asset Store Reset service Module reset	Dynamic RSA private key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	encrypted export				
Static RSA private key	OTP coprocessor export OTP obfuscated import	Static Asset Store:Obfuscated	Until explicitly zeroized	Zeroize OTP service	Static RSA public key:Paired With
Dynamic RSA private key	RAM coprocessor export RAM plaintext import RAM obfuscated import RAM obfuscated export RAM encrypted import RAM encrypted export	Dynamic Asset Store:Obfuscated	Until explicitly zeroized or module reset	Asset Delete service Dynamic Asset Store Reset service Module reset	Dynamic RSA public key:Paired With
Firmware signature verification key	RAM plaintext import		For the duration of the service	Automatic	
Shared secret			For the duration of the service	Automatic	
Intermediate key generation value			For the duration of the service	Automatic	
NIST SP 800-56Arev3 domain parameters		Dynamic Asset Store:Obfuscated Static Asset Store:Obfuscated	Until explicitly zeroized or module reset	Asset Delete service Zeroize OTP service Dynamic Asset Store Reset service Module reset	Static EC public key:Used With Static EC private key:Used With Dynamic EC public key:Used

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
					With Dynamic EC private key:Used With

Table 22: SSP Table 2

9.5 Transitions

The SHA-1 algorithm as implemented by the module will be non-approved for all purposes, starting January 1, 2030.

10 Self-Tests

While the module is executing the self-tests, services are not available, and data output (via the data output interface) is inhibited until the tests are successfully completed. The module does not process service requests from the operator until the tests are completed.

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
CRC24	N/A	Error Detection Code	SW/FW Integrity	CRC24 ok bit is set in the MODULE_STATUS register	CRC24 check is performed on the entire ROM firmware image

Table 23: Pre-Operational Self-Tests

The pre-operational firmware integrity test on the ROM firmware component is performed automatically when the module is initialized. If this test fails, the module transitions to the Hardware Error state.

10.2 Conditional Self-Tests

As part of the initialization, the ROM firmware component performs the SHA2-256 and ECDSA Signature Verification CASTs. Then, the ROM firmware loads the RAM firmware component and performs the firmware load test on the RAM firmware. Only if these tests succeed, will the RAM firmware be executed. Finally, the RAM firmware automatically performs the rest of the CASTs listed in the table below, before transitioning to the operational state. If any of the tests performed by the firmware components fail, the module transitions to the Firmware Error state.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
ECDSA SigVer (FIPS186-5) (A5263) KAT	P-256 with SHA2-256	KAT	CAST	ROM firmware is ready to accept RAM firmware image	KAT signature verification	ROM firmware integrity test passed
SHA2-256 (A5263)	320-bit message	KAT	CAST	ROM firmware is ready to accept RAM firmware image	KAT message digest	ROM firmware integrity test passed
ECDSA SigVer (FIPS186-	P-256 with SHA2-256	Signature verification	SW/FW Load	FW accepted bit is set in the MODULE_STATUS register	Signature verification is performed on the	RAM firmware image is loaded

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
5) (A5263)					entire RAM firmware image	
AES-CBC (A5264)	128-bit key	KAT	CAST	SelftestActive field in System Information output is set to 0	KAT encryption and decryption	RAM firmware load test passed
AES-CCM (A5264)	192-bit key, 88-bit nonce	KAT	CAST	SelftestActive field in System Information output is set to 0	KAT encryption and decryption	RAM firmware load test passed
AES-XTS Testing Revision 2.0 (A5264)	256-bit key	KAT	CAST	SelftestActive field in System Information output is set to 0	KAT encryption and decryption	RAM firmware load test passed
AES-GCM (A5264)	256-bit key, 128-bit IV	KAT	CAST	SelftestActive field in System Information output is set to 0	KAT encryption and decryption	RAM firmware load test passed
AES-CMAC (A5264)	256-bit key	KAT	CAST	SelftestActive field in System Information output is set to 0	KAT MAC tag generation	RAM firmware load test passed
Counter DRBG (A5264)	AES-256	KAT	CAST	SelftestActive field in System Information output is set to 0	KAT instantiate, reseed, generate, generate (compliant to SP 800-90A Section 11.3)	RAM firmware load test passed
SHA-1 (A5264)	256-bit message	KAT	CAST	SelftestActive field in System Information output is set to 0	KAT message digest	RAM firmware load test passed

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA2-224 (A5264)	320-bit message	KAT	CAST	SelftestActive field in System Information output is set to 0	KAT message digest	RAM firmware load test passed
HMAC-SHA2-256 (A5264)	256-bit key	KAT	CAST	SelftestActive field in System Information output is set to 0	KAT MAC tag generation	RAM firmware load test passed
KDF SP800-108 (A5264)	HMAC SHA2-256 in feedback mode	KAT	CAST	SelftestActive field in System Information output is set to 0	KAT key-based key derivation	RAM firmware load test passed
SHA2-512 (A5264)	640-bit message	KAT	CAST	SelftestActive field in System Information output is set to 0	KAT message digest	RAM firmware load test passed
SHA3-512 (A5264)	320-bit message	KAT	CAST	SelftestActive field in System Information output is set to 0	KAT message digest	RAM firmware load test passed
KDA (A5264)	OneStep KDA with SHA2-256	KAT	CAST	SelftestActive field in System Information output is set to 0	KAT key derivation from shared secret	RAM firmware load test passed
ECDSA SigGen (FIPS186-5) (A5264)	P-224 with SHA2-224	KAT	CAST	SelftestActive field in System Information output is set to 0	KAT signature generation	RAM firmware load test passed
ECDSA SigVer (FIPS186-5) (A5264)	P-224 with SHA2-224	KAT	CAST	SelftestActive field in System Information output is set to 0	KAT signature verification	RAM firmware load test passed
KAS-ECC Sp800-56Ar3 (A5264)	KAS-ECC-SSC with P-224	KAT	CAST	SelftestActive field in System Information output is set to 0	KAT shared secret computation	RAM firmware load test passed

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
RSA SigGen (FIPS186-5) (A5264)	2048-bit modulus with PKCS#1 v1.5 padding and SHA2-256	KAT	CAST	SelftestActive field in System Information output is set to 0	KAT signature generation	RAM firmware load test passed
RSA SigVer (FIPS186-5) (A5264)	2048-bit modulus with PKCS#1 v1.5 padding and SHA2-256	KAT	CAST	SelftestActive field in System Information output is set to 0	KAT signature verification	RAM firmware load test passed
ECDSA KeyGen (FIPS186-5) (A5264)	SHA-224, SHA-256, SHA-384, SHA-512	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
ECDSA KeyGen (FIPS186-5) (A5264) Sp800-56Ar3	N/A	PCT	PCT	Successful key pair generation	SP 800-56Ar3 Section 5.6.2.1.4	Key pair generation
Entropy Source RCT-ST	Cutoff: 31 samples	Startup health test	CAST	Entropy source produces entropy	Repetition Count Test	Entropy source startup
Entropy Source APT-ST	Cutoff: 325 samples	Startup health test	CAST	Entropy source produces entropy	Adaptive Proportion Test	Entropy source startup
Entropy Source RCT-C	Cutoff: 31 samples	Continuous health test	CAST	Entropy source produces entropy	Repetition Count Test	DRBG seeding
Entropy Source APT-C	Cutoff: 325 samples	Continuous health test	CAST	Entropy source produces entropy	Adaptive Proportion Test	DRBG seeding

Table 24: Conditional Self-Tests

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
CRC24	Error Detection Code	SW/FW Integrity	On demand	Manually

Table 25: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
ECDSA SigVer (FIPS186-5) (A5263) KAT	KAT	CAST	On demand	Manually
SHA2-256 (A5263)	KAT	CAST	On demand	Manually
ECDSA SigVer (FIPS186-5) (A5263)	Signature verification	SW/FW Load	On demand	Manually
AES-CBC (A5264)	KAT	CAST	On demand	Manually
AES-CCM (A5264)	KAT	CAST	On demand	Manually
AES-XTS Testing Revision 2.0 (A5264)	KAT	CAST	On demand	Manually
AES-GCM (A5264)	KAT	CAST	On demand	Manually
AES-CMAC (A5264)	KAT	CAST	On demand	Manually
Counter DRBG (A5264)	KAT	CAST	On demand	Manually
SHA-1 (A5264)	KAT	CAST	On demand	Manually
SHA2-224 (A5264)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A5264)	KAT	CAST	On demand	Manually
KDF SP800-108 (A5264)	KAT	CAST	On demand	Manually
SHA2-512 (A5264)	KAT	CAST	On demand	Manually
SHA3-512 (A5264)	KAT	CAST	On demand	Manually
KDA (A5264)	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
ECDSA SigGen (FIPS186-5) (A5264)	KAT	CAST	On demand	Manually
ECDSA SigVer (FIPS186-5) (A5264)	KAT	CAST	On demand	Manually
KAS-ECC Sp800-56Ar3 (A5264)	KAT	CAST	On demand	Manually
RSA SigGen (FIPS186-5) (A5264)	KAT	CAST	On demand	Manually
RSA SigVer (FIPS186-5) (A5264)	KAT	CAST	On demand	Manually
ECDSA KeyGen (FIPS186-5) (A5264)	PCT	PCT	On demand	Manually
ECDSA KeyGen (FIPS186-5) (A5264) Sp800-56Ar3	PCT	PCT	On demand	Manually
Entropy Source RCT-ST	Startup health test	CAST	On demand	Manually
Entropy Source APT-ST	Startup health test	CAST	On demand	Manually
Entropy Source RCT-C	Continuous health test	CAST	On demand	Manually
Entropy Source APT-C	Continuous health test	CAST	On demand	Manually

Table 26: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Hardware error	Hardware failed to verify the integrity of the ROM firmware	ROM firmware integrity test failure	Power off	CRC24 error bit is set in the MODULE_STATUS register
Firmware error	ROM or RAM firmware	Unsuccessful login RAM firmware	Hard reset (reset_n pin) or power off	Fatal error bit is set or FW accepted bit is not set in the

Name	Description	Conditions	Recovery Method	Indicator
	encountered an error	load test failure CAST failure PCT failure DMA error		MODULE_STATUS register

Table 27: Error States

In the Hardware Error state, no firmware input or output is possible at all, only the hardware registers such as the MODULE_STATUS register. In the Firmware Error state, only the *Show status, Show version* and *Hard Reset* services are available. Cryptographic functions and data output are inhibited.

10.5 Operator Initiation of Self-Tests

To perform the on-demand self-tests that includes the pre-operational self-tests and CASTs, the Crypto Officer shall power-off and power-on or perform a hard reset of the module.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

VaultIP synthesized in the Xillinx Zynq XC7Z045 FPGA is a single chip hardware module. The chip is delivered from the vendor via a trusted delivery courier. Upon reception of VaultIP, the customer should verify that the package does not have any irregular tears or openings. The chip comes preloaded with the following code packages:

- 914-130939-460_VaultIP-130-939_Firmware-cfgA_FW4.6.3.zip.
- 915-130939-430_VaultIP-130-939_HW4.3.1.zip

11.2 Administrator Guidance

The Public Use Document related to the ESV certificate is posted here: https://csrc.nist.gov/CSRC/media/projects/cryptographic-module-validation-program/documents/entropy/E167_PublicUse.pdf

The module is configured as a FIPS140-3 module at factory for the Xilinx Zynq XC7Z045 FPGA tested implementation. In this FPGA configuration the Crypto Officer should execute the “Show version" service and verify the following outputs:

- *00040301* is the hardware version
- *01040603* is the firmware version
- *0000d82c* is the EIP-130 component name

The combination of these outputs maps to a unique module with name VaultIP RT-130 with versions 4.6.3, 4.3.1.

The module implicitly transitions between the approved mode and the non-approved mode contingent on the service that is invoked. Therefore, there are no special procedures to administer the approved or non-approved modes.

11.3 Non-Administrator Guidance

N/A

11.4 Design and Rules

The Crypto Officer shall consider the following requirements and restrictions when using the module.

- AES GCM IV see Section 2.7
- AES XTS see Section 2.7
- SP800-56Ar3 assurances see Section 2.7
- RSA modulus size see Section 2.7
- SP800-56Br2 assurances see Section 2.7
- Legacy use see Section 2.7

11.5 End of Life

Secure sanitization of the module consists of performing the Zeroize OTP service then powering off the module. This will zeroize all SSPs in non-volatile and volatile memory.

12 Mitigation of Other Attacks

The module does not implement security mechanisms to mitigate other attacks.

Appendix A. Glossary and Abbreviations

AES	Advanced Encryption Standard
ASIC	Application-Specific Integrated Circuit
CAST	Cryptographic Algorithm Self-Test
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CBC-MAC	Cipher Block Chaining Message Authentication Code
CCM	Counter with Cipher Block Chaining Message Authentication Code
CKG	Cryptographic Key Generation
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CTR	Counter Mode
DMA	Direct Memory Access
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
ECIES	Elliptic Curve Integrated Encryption Scheme
ESV	Entropy Source Validation
FIPS	Federal Information Processing Standards Publication
FPGA	Field Programmable Gate Array
GCM	Galois Counter Mode
GMAC	Galois Message Authentication Code
HMAC	Keyed-Hash Message Authentication Code
HUK	Hardware Unique Key
ICM	Integer Counter Mode
IFC	Integer Factorization Cryptography
IV	Initialization Vector
KAS	Key Agreement Scheme
KAT	Known Answer Test
KDF	Key Derivation Function
KTS	Key Transport Scheme
KWP	AES Key Wrap with Padding
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OAEP	Optimal Asymmetric Encryption Padding
OTP	One-Time Programmable
PCT	Pair-wise Consistency Test
PDA	Personal Digital Assistant
PIN	Personal Identification Number
PSS	Probabilistic Signature Scheme
RAM	Random-Access Memory
ROM	Read-Only Memory
RSA	Rivest, Shamir, Adleman
SHA	Secure Hash Algorithm
SIV	Synthetic Initialization Vector
SoC	System on Chip
SSP	Sensitive Security Parameter
TCM	Tightly-Coupled Memory
TEE	Trusted Execution Environment
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

Appendix B. References

FIPS140-3	**FIPS PUB 140-3 - Security Requirements For Cryptographic Modules** March 2019 https://doi.org/10.6028/NIST.FIPS.140-3
FIPS140-3_IG	**Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program** https://csrc.nist.gov/Projects/cryptographic-module-validation-program/fips-140-3-ig-announcements
FIPS180-4	**Secure Hash Standard (SHS)** August 2015 http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
FIPS186-2	**Digital Signature Standard (DSS)** Jan 2000 https://csrc.nist.gov/files/pubs/fips/186-2/final/docs/fips186-2.pdf
FIPS186-4	**Digital Signature Standard (DSS)** July 2013 http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
FIPS186-5	**Digital Signature Standard (DSS)** February 2023 https://doi.org/10.6028/NIST.FIPS.186-5
FIPS197	**Advanced Encryption Standard** November 2001 http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
FIPS198-1	**The Keyed Hash Message Authentication Code (HMAC)** July 2008 http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf
FIPS202	**SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions** August 2015 http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf
PKCS#1	**Public Key Cryptography Standards (PKCS) #1: RSA Cryptography** Specifications Version 2.1 February 2003 http://www.ietf.org/rfc/rfc3447.txt
RFC 5639	**Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation** March 2010 https://doi.org/10.17487/RFC5639
SP800-38A	**NIST Special Publication 800-38A - Recommendation for Block Cipher Modes of Operation Methods and Techniques** December 2001 http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

SP800-38B	**NIST Special Publication 800-38B - Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication** May 2005 http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf
SP800-38C	**NIST Special Publication 800-38C - Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality** May 2004 http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf
SP800-38D	**NIST Special Publication 800-38D - Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC** November 2007 http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf
SP800-38E	**NIST Special Publication 800-38E - Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices** January 2010 http://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf
SP800-38F	**NIST Special Publication 800-38F - Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping** December 2012 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf
SP800-56Ar3	**NIST Special Publication 800-56A Revision 3 - Recommendation for Pair Wise Key Establishment Schemes Using Discrete Logarithm Cryptography** April 2018 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf
SP800-56Br2	**Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography** March 2019 https://doi.org/10.6028/NIST.SP.800-56Br2
SP800-56Cr2	**Recommendation for Key Derivation through Extraction-then-Expansion** August 2020 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Cr2.pdf
SP800-90Ar1	**NIST Special Publication 800-90A - Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators** June 2015 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf
SP800-90B	**NIST Special Publication 800-90B - Recommendation for the Entropy Sources Used for Random Bit Generation** January 2018 https://doi.org/10.6028/NIST.SP.800-90B

SP800-108r1-upd1	**NIST Special Publication 800-108 - Recommendation for Key Derivation Using Pseudorandom Functions (Revised)** August 2022 https://doi.org/10.6028/NIST.SP.800-108r1-upd1
SP800-133r2	**NIST Special Publication 800-133 Revision 2 - Recommendation for Cryptographic Key Generation** June 2020 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-133r2.pdf
SP800-140Br1	**NIST Special Publication 800-140Br1 - CMVP Security Policy Requirements** November 2023 https://doi.org/10.6028/NIST.SP.800-140Br1