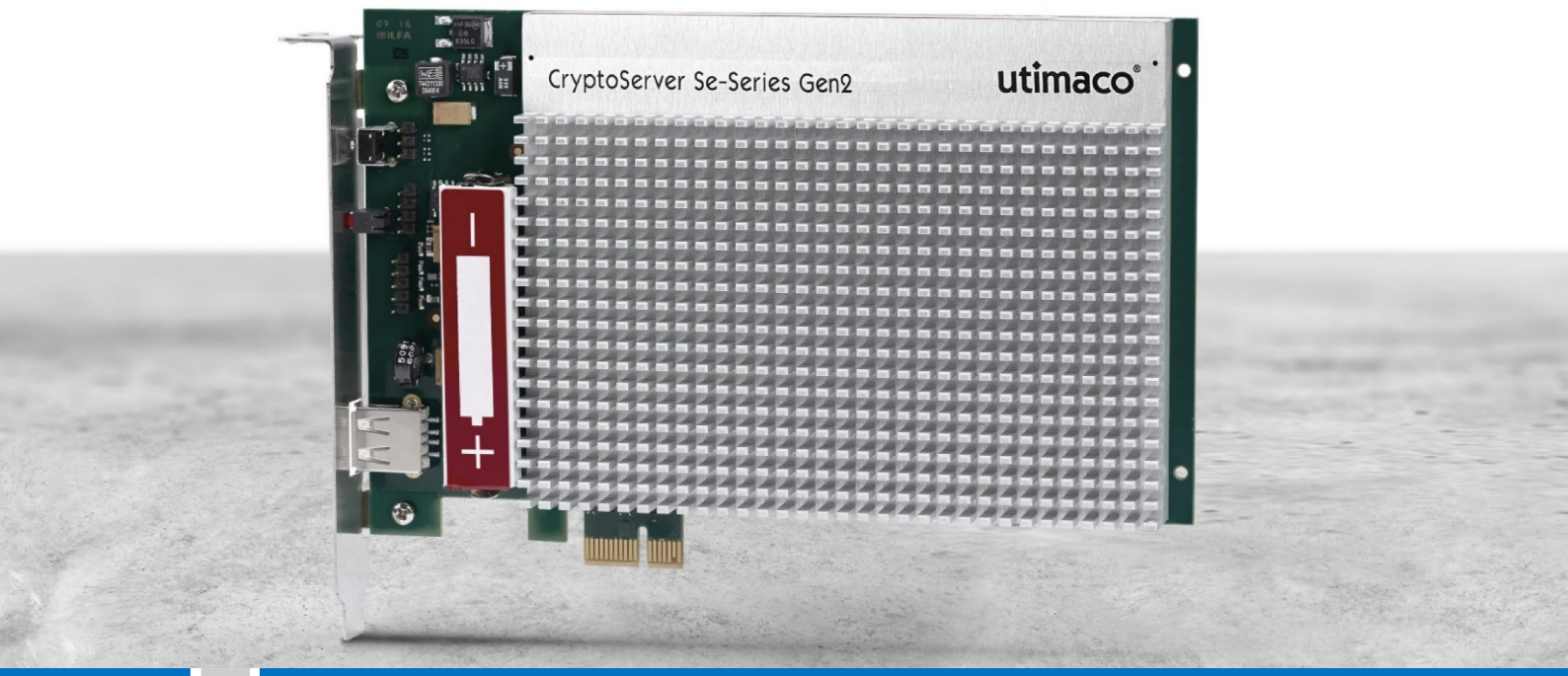

CryptoServer Se-Series Gen2

Non-Proprietary
Security Policy

utimaco®

Imprint

Utimaco IS GmbH
Germanusstr. 4
52080 Aachen
Germany

Phone +49 (0)241 / 1696-200

Fax +49 (0)241 / 1696-199

Internet http://hsm.utimaco.com

E-mail hsm@utimaco.com

Document number 2014-0001

Document version 2.1.3

Date 2021-03-31

Status Released

Table of Contents

1 Introduction 5
1.1 Module Overview 5
1.2 Security Level 8
2 Modes of Operation 9
2.1 Approved Mode of Operation 9
2.1.1 Configuration of Approved Mode 16
2.2 Non-FIPS Modes of Operation 17
2.3 Secure Messaging for Secure Communication with the CryptoServer 19
3 Ports and Interfaces 20
4 Identification and Authentication Policy 21
4.1 Assumption of Roles 21
5 Access Control Policy 23
5.1 Roles and Authenticated Services 23
5.2 Unauthenticated Services 29
5.3 Services in Non-FIPS Modes 31
5.4 Definition of Critical Security Parameters (CSPs) 31
5.5 Definition of Public Keys 32
5.6 Definition of Modes of Access to CSPs 32
6 Security Rules 42
7 Physical Security Policy 46
8 Operational Environment 48
9 Mitigation of Other Attacks Policy 49
10 References 50
11 Definitions and Acronyms 52

1 Introduction

This document defines the security policy for Utimaco's **CryptoServer Se-Series Gen2**, (hereafter denoted the CryptoServer) when run in FIPS mode.

The CryptoServer is a hardware security module made by Utimaco IS GmbH (referred to below also as Utimaco). If run in FIPS mode, the CryptoServer meets FIPS 140-2 overall Level 3 requirements.

Table 1 – CryptoServer Configuration

	Model	HW P/N and Version	FW Version
1	CryptoServer Se12	Hardware P/N CryptoServer Se-Series Gen2 Version 5.01.2.0, 5.01.4.0 and 5.01.4.2 without crypto accelerator	SecurityServer-Se2-Series-4.32.0.3-FIPS
2	CryptoServer Se52		
3	CryptoServer Se500	Hardware P/N CryptoServer Se-Series Gen2 Version 5.01.2.0, 5.01.4.0 and 5.01.4.2 with crypto accelerator (Exar DX8204)	
4	CryptoServer Se1500		

1.1 Module Overview

The CryptoServer is an encapsulated, protected security module realized as a multi-chip embedded cryptographic module as defined in [FIPS140-2]. Its realization meets the overall FIPS 140-2 Level 3 requirements. The primary purpose of this module is to provide secure cryptographic services such as encryption or decryption (for various cryptographic algorithms like Triple-DES and AES), hashing, signing and verification of data (RSA, ECDSA, DSA), random number generation, on-board secure key generation, key storage and further key management functions in a tamper-protected environment.

In FIPS mode, the module offers a general purpose cryptographic API with FIPS Approved algorithms for the above mentioned cryptographic services, as well as an administrative interface. A Secure Messaging concept uses message encryption and MAC authentication to protect communication to and from the cryptographic module.

If not in FIPS mode, CryptoServer's flexible firmware architecture enables its usage in almost all proprietary environments in which cryptographic services and highest security are required, such as archiving systems and payment systems. It can serve as a signature server, time stamp, and generator for PINs, cryptographic keys, or random numbers.

The CryptoServer offers hardware-based as well as deterministic random number generation in FIPS mode and non-FIPS mode. The hardware based RNG is used to seed and re-seed the Approved Deterministic RBG.

Together with Utimaco's appropriate host application software the module also provides cryptographic standard interfaces like PKCS#11, JCE, OpenSSL, CSP/CNG and EKM.

All hardware components of the cryptographic module, including the Central Processing Unit, all memory chips, Real Time Clock, and hardware noise generator for random number generation, are located on a printed circuit board (PCI express board)These hardware components are completely covered with potting material (epoxy resin) and heat sink. This hard, opaque enclosure protects the sensitive CryptoServer hardware components from physical attacks.

For the communication with a host, the PCIe board offers a PCIe interface and two USB interfaces. The picture below shows the CryptoServer with its PCIe interface:

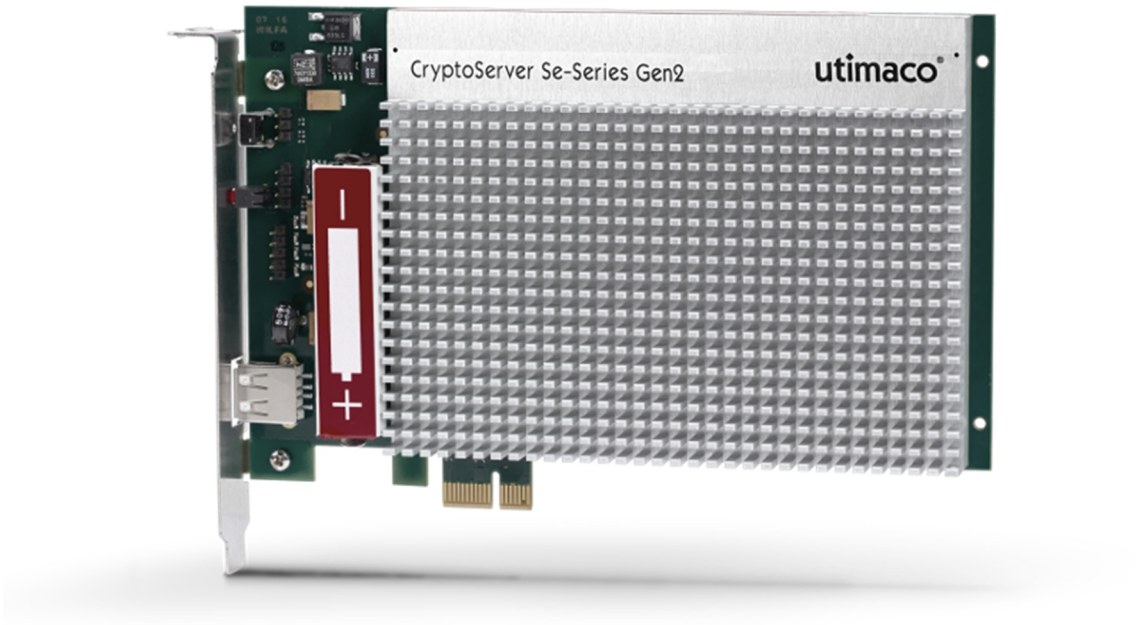

Figure 1 – CryptoServer Se-Series Gen2

The module's cryptographic boundary is defined as the outer perimeter of the heat sink on the top side and the epoxy surface on the bottom side of the module.

Figure 2 and Figure 3 below show views of the cryptographic boundary from the side and the top, and from the bottom. The red dashed line indicates the cryptographic boundary.

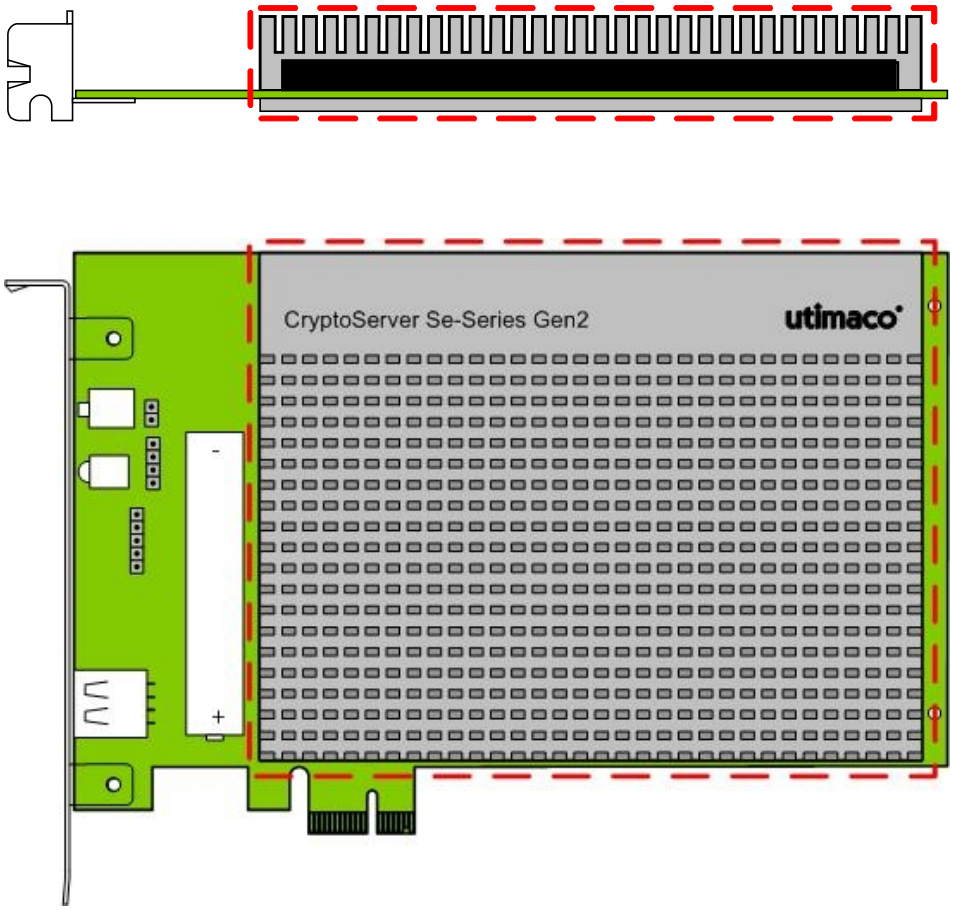

Figure 2 – CryptoServer Se-Series Gen2 – side view and top view

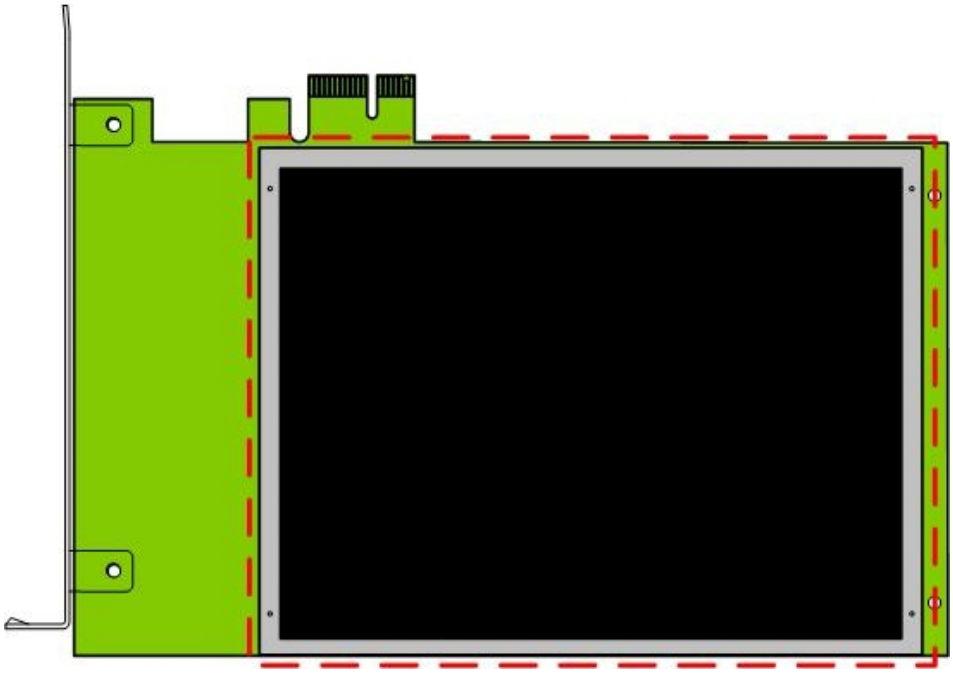

Figure 3 – CryptoServer Se-Series Gen2 – bottom view

1.2 Security Level

The CryptoServer meets the overall requirements applicable to Level 3 security in FIPS 140-2.

Table 2 – Security Level of Security Requirements

Security Requirement	Security Level
Cryptographic Module Specification	3
Cryptographic Module Ports and Interfaces	3
Roles, Services and Authentication	3
Finite State Model	3
Physical Security	3
Operational Environment	N/A
Cryptographic Key Management	3
EMI/EMC	3
Self-Tests	3
Design Assurance	3
Mitigation of Other Attacks	3

2 Modes of Operation

The CryptoServer implements an Approved and a non-Approved mode of operation.

2.1 Approved Mode of Operation

The CryptoServer implements the FIPS Approved and Non-Approved but Allowed cryptographic algorithms listed in the tables below.

Table 3 – Approved and CAVP Validated Cryptographic Algorithms

CAVP Cert #	Algo-rithm	Standard	Mode/ Method	Key Lengths, Curves or Moduli	Use
C1114	AES	FIPS 197; SP 800-38A	CBC, CFB, CTR, ECB, OFB	128, 192, 256	Data Encryption/ Data Decryption
C1130	AES	SP 800-38C	CCM	128, 192, 256	Authenticated Encryption and Decryption, Key Transport Scheme
C1132	AES	SP 800-38F	KW, KWP	128, 192, 256	Key Transport Scheme (Encryption and Decryption)
C1134	AES	SP 800-38B	CMAC	128, 192, 256	Message Authentication (Generation and Verification)
C1245	AES	SP 800-38D	GCM[1], GMAC	128, 192, 256	Authenticated Encryption/ Decryption, Key Transport Scheme (GCM only)
A1019	CVL KDF ANS 9.42	SP 800-135 ANSI X9.42	Concatenation SHA-224, SHA-256, SHA-384, SHA-512 SHA3-224, SHA3-256, SHA3-384, SHA3-512	1-4096	Key Derivation
C1115/C 1116	CVL RSADP	SP 800-56B		2048	Key Transport Scheme and Key Wrapping

[1] The 96 bit IV is randomly generated internally per IG A.5, option 2

CAVP Cert #	Algo-rithm	Standard	Mode/ Method	Key Lengths, Curves or Moduli	Use
C1135	CVL KDF ANS 9.63	SP 800-135; ANSI X9.63	Concatenation SHA-224, SHA-256, SHA-384, SHA-512	128 - 4096	Key Derivation
C1163	CVL KDF TLS	SP 800-135; TLS 1.2	SHA-256, SHA-384, SHA-512		Key Derivation
A1066	DRBG	SP 800-90A	Hash DRBG: SHA-512-based		Random Bit Generation
C1189	DSA	FIPS 186-4		2048/224, 2048/256 or 3072/256	Key Generation
			SHA-224[2], SHA-256, SHA-384, SHA-512	2048/224, 2048/256 or 3072/256	Digital Signature Generation and Domain Parameter Generation
		FIPS 186-4; FIPS 186-2	SHA-1[3], SHA-224[2], SHA-256, SHA-384, SHA-512	1024/160, 2048/224, 2048/256 or 3072/256	Digital Signature Verification and Parameter Verification
C1190 / C1191	ECDSA	FIPS 186-4		NIST Recommended: See below Non-NIST (per IG A.2)[4]: See below, and curve25519	Key Generation

[2] Domain Parameter Generation and Verification with SHA-224 is only possible for key length 2048/224.

[3] Domain Parameter Verification with SHA-1 is only possible for key length 1024/160

[4] Non-NIST-Recommended elliptic curves implemented per IG A.2 are approved per IG A.14, but are not CAVP-testable. Refer to Table 5 for associated security strengths

CAVP Cert #	Algo-rithm	Standard	Mode/ Method	Key Lengths, Curves or Moduli	Use
			SHA-224, SHA-256, SHA-384, SHA-512	NIST Recommended: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, B-571 Non-NIST (per IG A.2)[5]: brainpoolP224r1/ 224t1/ 256r1/ 256t1/ 320r1/ 320t1/ 384r1/ 384t1/ 512r1/ 512t1, secp256k1, FRP256v1	Digital Signature Generation
		FIPS 186-4; FIPS 186-2	SHA-1[6], SHA-224, SHA-256, SHA-384, SHA-512[7]	NIST Recommended: See above, and P-192, K-163, B-163 Non-NIST (per IG A.2): See above	Digital Signature Verification
				NIST Recommended: See above, and P-192, K-163, B-163 Non-NIST (per IG A.2): See above, and curve25519	Public Key Validation
	ENT	SP 800-90B		Generated entropy: 407 Entropy per source output bit: 0.796	Used to generate the seed material for the Approved DRBG
C1136	HMAC	FIPS 198-1	SHA-1, SHA-224, SHA-256,	key size >= 112 bits	Message Authentication Generation

[5] Non-NIST-Recommended elliptic curves implemented per IG A.2 are approved per IG A.14, but are not CAVP-testable. Refer to Table 5 for associated security strengths

[6] Not implemented with K-163

[7] Not implemented with B-Curves

CAVP Cert #	Algo-rithm	Standard	Mode/ Method	Key Lengths, Curves or Moduli	Use
			SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512	key size >= 80 bits	Message Authentication Verification
C1162	KBKDF	SP 800-108	SHA-256; feedback mode	L=256	Key Derivation[8]
C1130	KTS	SP 800-38F	AES CCM	Provides between 128 and 256 bits of encryption strength	Key Transport Scheme
C1132	KTS	SP 800-38F	AES KW, AES KWP	Provides between 128 and 256 bits of encryption strength	Key Transport Scheme
C1245	KTS	SP 800-38F	AES GCM	Provides between 128 and 256 bits of encryption strength	Key Transport Scheme
C1114 and C1134	KTS	SP 800-38F; FIPS 197; SP 800-38A; SP 800-38B	AES CBC and AES CMAC	Provides 256 bits of encryption strength	Secure Messaging with 256-bit session keys K_{SME} and K_{SMM}. Key Transport Scheme (keys derived by SP 800-108, key derivation key established by SP 800-56Ar3 and SP 800-56Cr1)
C1192 / C1193	RSA	FIPS 186-4		2048…16384[9]	Key Generation
			ANSI X9.31, PKCS 1.5, PSS SHA-224, SHA-256, SHA-384, SHA-512	2048…16384[10]	Digital Signature Generation

[8] Used to derive session keys and backup keys.

[9] Even key lengths only. CAVP certification covers all testable RSA modulus sizes: 2048 and 3072 per FIPS 186-4 (IG A.14)

[10] Even key lengths only. CAVP certification covers all testable RSA modulus sizes: 2048 and 3072 per FIPS 186-4 and 4096 per FIPS 186-2, ref. IG A.14

CAVP Cert #	Algo-rithm	Standard	Mode/ Method	Key Lengths, Curves or Moduli	Use
		FIPS 186-4; FIPS 186-2	ANSI X9.31, PKCS 1.5, PSS SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	1024…16384[11]	Digital Signature Verification
C1118	SHA-3	FIPS 202	SHA3-224 SHA3-256 SHA3-384 SHA3-512		Message Digest
C1117	SHS	FIPS 180-4	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512		Message Digest
C1119	SHS	FIPS 180-4	SHA-512		Message Digest (Bootloader)
A1065	SHS	FIPS 180-4	SHA-512		Message Digest (SMOS)
C1121	Triple-DES	SP 800-67; SP 800-38A	CBC, ECB	3-key (24 bytes) and 2-key (16 bytes)	Data Decryption[12]

Table 4 - Approved Cryptographic Algorithms: Vendor Affirmed

Algorithm	Standard	Mode/ Method	Key Lengths, Curves or Moduli	Use
CKG	SP 800-133		512	Unmodified DRBG output used to derive symmetric keys and seeds for asymmetric private keys.
DSA with SHA-3	FIPS 186-4	SHA3-224 SHA3-256 SHA3-384 SHA3-512	(see DSA entry above)	Digital Signature Generation Digital Signature Verification

[11] Even key lengths only. CAVP certification covers all testable RSA modulus sizes: 1024, 2048 and 3072 per FIPS 186-4 and 1024,1536, 2048, 3072, 4096 per FIPS 186-2, ref. IG A.14.

[12] CAVP certification also covers encryption, which is not used in Approved mode

Algorithm	Standard	Mode/ Method	Key Lengths, Curves or Moduli	Use
ECDSA with SHA-3	FIPS 186-4	SHA3-224 SHA3-256 SHA3-384 SHA3-512	(see ECDSA entry above)	Digital Signature Generation Digital Signature Verification
KAS-SSC	SP 800-56Ar3	FFC DH	|p| = 2048, |q| = 224 or 256 (Provides 112 bits of encryption strength)	Shared Secret computation[13]
KAS-SSC	SP 800-56Ar3	ECC DH	P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, B-571 (Provides between 112 and 256 bits of encryption strength)	Shared Secret Computation[14]
KAS-SSC	SP 800-56Ar3	(Cofactor) Ephemeral Unified Model C(2e, 0s, ECC CDH)	P-521	Secure Messaging: Shared secret computation. The SP 800-56Cr1 KDF (HMAC-SHA-256) is used to derive AES CBC and AES CMAC keys for KTS
KDA: NIST One-Step KDF	SP 800-56C r1	One-step concatena-tion KDF	HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512, HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512	Key Derivation
KDF 135 (X9.63) with SHA-3	SP 800-135; ANSI X9.63	Concatena-tion KDF	SHA3-224, SHA3-256, SHA3-384, SHA3-512 key lengths: 128 - 4096	Key Derivation
KTS	SP 800-56B	KTS-OAEP-basic key transport scheme	2048 - 16384[15] (Provides between 112 and 256 bits of encryption strength)	Key transport scheme (using CVL RSADP C1115/C1116)

[13] Primitive alone or with SP 800-135 ANSI X9.42 KDF

[14] Primitive alone or with the SP 800-56Cr1 One-Step KDF or the SP 800-135 ANSI X9.63 KDF

[15] Even key lengths only

Algorithm	Standard	Mode/ Method	Key Lengths, Curves or Moduli	Use
RSA[16]	FIPS 186-4	SHA3-224 SHA3-256 SHA3-384 SHA3-512	(see RSA entry above)	Digital Signature Generation Digital Signature Verification

The security strength of the Non-NIST Recommended elliptic curves is as follows:

Table 5 – Security Strength of Non-NIST Elliptic Curves

EC Curve	Security Strength	Reference
brainpoolP224r1 / brainpoolP224t1	112	[ECCBP]
curve25519	128	[RFC 7748]
secp256k1	128	[SEC2]
FRP256v1	128	[ANSSI]
brainpoolP256r1 / brainpoolP256t1	128	[ECCBP]
brainpoolP320r1 / brainpoolP320t1	160	[ECCBP]
brainpoolP384r1 / brainpoolP384t1	192	[ECCBP]
brainpoolP512r1 / brainpoolP512t1	256	[ECCBP]

The CryptoServer also implements and uses the following non-FIPS Approved but Allowed algorithms:

[16] Except for X9.31 padding: RSA sign/verify with X9.31 padding does not support SHA3 hashes.

Table 6 – Non-Approved but Allowed Cryptographic Functions

Algorithm	Standards, Methods	Key Lengths, Curves or Moduli	Caveat	Use
Diffie-Hellman (shared secret computation)	Primitive FFC DH	|p| > 2048, |q| >= 224, except 256	Provides between 112 and 256 bits of encryption strength	Shared Secret computation[17]
Diffie-Hellman (key agreement)				Key Agreement[18]
EC Diffie-Hellman (shared secret computation)	IG D.8 scenario X2[19]	Non-NIST (per IG A.2)[20]: brainpoolP224r1/ 224t1/ 256r1/ 256t1/ 320r1/ 320t1/ 384r1/ 384t1/ 512r1/ 512t1, secp256k1, FRP256v1, curve25519	Provides between 112 and 256 bits of encryption strength	Shared Secret Computation[21]
EC Diffie-Hellman (key agreement)				Key Agreement[22]
RSA (CVL C1115/C1116, key wrapping)	SP 800-56B Wrapping with encryption scheme RSAES-PKCS-v1_5 according to [PKCS#1]	2048 - 16384[23]	Provides between 112 and 256 bits of encryption strength	Key establishment methodology (key wrapping and unwrapping) using RSADP CVL

2.1.1 Configuration of Approved Mode

The CryptoServer can be configured for FIPS mode by using Utimaco's command-line tool for administration, csadm, as follows by an authenticated Administrator:

1. Perform the csadm Dev=<device address> GetState command.
2. Verify that the module is in the INITIALIZED state, and in Operational or Maintenance mode and the alarm state is "OFF".

[17] Primitive alone

[18] Shared secret computation with SP 800-135 ANSI X9.42 KDF

[19] As indicated in IG D.8 X2 (b), "the rules of SP 800-56A Rev3 have been followed whenever possible, given that the curves may not be defined in a NIST publication."

[20] Non-NIST-Recommended elliptic curves implemented per IG A.2 are allowed per IG D.8 scenario X2. Refer to Table 5 for associated security strengths

[21] Primitive alone

[22] Shared secret computation with the SP 800-56Cr1 One-Step KDF or the SP 800-135 ANSI X9.63 KDF.

[23] Even key lengths only

3. Load the FIPS validated firmware package by using the csadm LoadPkg command with the flag “ForceClear”.

   The CryptoServer Administrator’s Guide [ANSSI] describes the csadm commands in more detail. If the firmware package has been successfully loaded, the CryptoServer’s internally stored FIPS mode indicator flag is set.

4. To verify that the CryptoServer is in the Approved mode of operation, perform the csadm Dev=<device address> GetState command again and verify that the following line is available in the command output:

   FIPS mode = ON

2.2 Non-FIPS Modes of Operation

The module also includes firmware for non-FIPS modes of operations. Non-FIPS mode firmware will not set the module’s internally stored FIPS mode indicator flag. This missing flag will indicate to the user of the cryptographic module that the module is running in non-FIPS mode when the user requests the “GetState” service.

For example, while the module is in delivery state, non-FIPS firmware that could provide one or more of the following non-FIPS validated algorithms can be loaded into the module:

Table 7 – Non-FIPS Validated Cryptographic Algorithms

Algorithm	Use
RSA public key cipher of bulk data (non-compliant)	Encryption/Decryption (key sizes 512-16384)
EC Cryptography public key cipher of bulk data (ECIES)	Encryption/Decryption
MD5, MDC-2 or RIPEMD-160	Hashing
Single DES	Encryption/Decryption
TDES, TDES MAC (FIPS 113; SP 800-20) (non-compliant)	Encryption, Message Authentication
Triple DES ANSI retail MAC (ANSI X9.19: 1986, Financial Institution Retail Message Authentication)	Message Authentication
AES GCM mode (non-compliant to requirements of IG A.5 scenario 3)	Encryption/Decryption and Message Authentication
AES MAC CBC Mode (non-compliant)	Message Authentication
ECC point multiplication according to **TR-03111** on P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, B-571, brainpoolP224r1/ 224t1/ 256r1/ 256t1/ 320r1/ 320t1/ 384r1/ 384t1/ 512r1/ 512t1, secp256k1, or FRP256v1	Shared Secret Computation
Several key derivation algorithms as specified in [PKCS#11]: • **KDF_ENC_DATA**: Derive key using the result of an encryption (chaining mode ECB or CBC) of a given text with a base key (DES, AES) (CBC: and given IV)	Key derivation (see [PKCS#11] for details)

Algorithm	Use
• **KDF_HASH**: Derive key using the hash value over the key components of a base key (DES or AES or Generic Secret; Hash algorithm must output at least as many bits as are needed for the requested key size within key template). • **KDF_ECDH:** Derive key according to ANSI X9.63 with ECDH primitive and ANSI kdf on P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, B-571, brainpoolP224r1/ 224t1/ 256r1/ 256t1/ 320r1/ 320t1/ 384r1/ 384t1/ 512r1/ 512t1, secp256k1, FRP256v1, or curve25519 • The hash based functions (**KDF_HASH**, **KDF_DH**, **KDF_ECDH_COF** and **KDF_ECDH**) may utilize one of the following hash algorithms: MD5, RipeMD 160, SHA-1. • **KDF_XOR_BASE_AND_DATA**: Derive key by XOR'ing the key components of a base key (DES, AES, Generic Secret) with given data. • **KDF_CAT_BASE_AND_KEY**: Concatenate a base key with a second key (both DES, AES, Generic Secret) to derive new key. • **KDF_CAT_BASE_AND_DATA**: Concatenate a base key (DES, AES, Generic Secret) with given data to derive new key. • **KDF_CAT_DATA_AND_BASE**: Concatenate given data with a base key (DES, AES, Generic Secret) to derive new key. • **KDF_EXTRACT_KEY_FROM_KEY**: Extract part of a base key (DES, AES, Generic Secret) to derive new key.	ECDH (key agreement; key establishment; non-compliant below 112 bits of encryption strength) Diffie-Hellman (key agreement; key establishment; non-compliant below 112 bits of encryption strength)

2.3 Secure Messaging for Secure Communication with the CryptoServer

The CryptoServer implements a Secure Messaging concept, which enables any operator to secure their communication with the CryptoServer over the PCIe interface even from a remote host. With Secure Messaging, commands sent to the CryptoServer and response data received from the CryptoServer can be AES CBC encrypted and integrity-protected/signed with an AES CMAC. In FIPS mode, Secure Messaging must be performed for every sensitive command, i.e., for every command that is only available for authenticated users.

To perform Secure Messaging, the operator must open a Secure Messaging Session. For a Session, two 32-byte AES session keys (Session Encryption key $\mathbf{K_{SME}}$, Session MAC Key $\mathbf{K_{SMM}}$) are negotiated between CryptoServer and host, using (Cofactor) Ephemeral Unified Model EC Diffie-Hellman (P-521) and the SP 800-56Cr1 One-Step KDF as the key establishment technique, with additional key derivation per SP 800-108. For generating its random value $\mathbf{K_{SM\_MOD\_PRIV}}$ that is needed for the key agreement, the CryptoServer uses its deterministic random bit generator. Optionally, a Secure Messaging Session with mutual authentication may be requested. In this case the CryptoServer returns additionally a signature over the answer data which on the host side can be used for authentication of the HSM towards the host.

The CryptoServer can simultaneously manage multiple sessions (with multiple operators): Each session manages its own session key, which is identified by a session ID. All commands using the same session ID and the same session key are said to belong to one session. In this way, a secure channel is established between the CryptoServer and the host application.

3 Ports and Interfaces

The physical interface of the CryptoServer consists of 19 printed circuit board tracks, embedded inside the printed circuit board (PCB) and passing the cryptographic boundary to the outer world (see Figure 1). The device provides the following physical ports on these tracks:

- Power input (including operational power input and backup power input).
- An External Erase button, which acts as a control input and can be used to zeroize all security relevant information inside the module.
- An LED indicating that the External Erase Button is pressed,
- External communication ports (PCIe and USB) that are used for data input, data output, control input and status output:

To enable communication with a host, the CryptoServer supports a PCIe interface and two USB interfaces. All requests for services are sent over the PCIe interface. The first USB interface is used for status output only. The second USB interface is not used in FIPS mode. All Critical Security Parameters (CSPs) are input and output over the services that are offered over the PCIe interface. In particular, CSPs are entered and output only in a wrapped form: All command and response data (except for status requests) to and from the CryptoServer are AES CBC encrypted and AES CMAC authenticated by the Secure Messaging layer. For details, see previous subsection 2.3 “Secure Messaging for Secure Communication with the CryptoServer“.

Additionally, all secret or private keys may optionally be exported encrypted with a Key Encryption Key (via e. g. the *Export Key* or *Wrap* services, see section 5.1 “Roles and Authenticated Services“*)*.

4 Identification and Authentication Policy

4.1 Assumption of Roles

The CryptoServer supports the following operator roles:

- The *Cryptographic User* is allowed to perform key management and cryptographic services.
- The *Security Officer* is allowed to perform key group specific administration functions like key group specific user management or key group specific configuration management.
- The *Administrator* is allowed to perform global configuration and user management.
- The *NTP Manager* is allowed to perform time synchronization functions on the CryptoServer by using an NTP server over a network.

The *Cryptographic User* role can optionally be split into two different user roles:

- A *User* who is allowed to perform cryptographic services like encryption or signing,
- A *Key Manager* who is allowed to perform key management services like key generation or key backup/restore.

Additionally, any user is allowed to perform non-sensitive services such as requesting status information without prior authentication.

The cryptographic module uses identity-based operator authentication to enforce the separation of roles. Two authentication methods are supported by the module: Password authentication and RSA signature authentication.

- For *password-based authentication* the operator must enter its user name and its password to log in. The user name is an alphanumeric string. The password is a binary string of a minimum of four (4) characters. To prevent the password from being eavesdropped, an HMAC is calculated including authentication data, command data, and a random challenge. The hash algorithm for the HMAC calculation is SHA-256. This HMAC value is sent to the CryptoServer instead of the password. The CryptoServer recalculates and checks the HMAC value using the operator’s password that is stored inside the CryptoServer.
- For *RSA signature-based authentication* the user sends an RSA signed command containing its user name to authenticate to the cryptographic module.

Upon correct authentication the role is selected based on the operator's user name. During authentication, session keys $\mathbf{K}_{SME}$ and $\mathbf{K}_{SMM}$ are negotiated which is used to secure subsequent service requests by the operator (see the description of the Secure Messaging concept in section 2.3). Since the session keys (and session ID) are stored in volatile memory, all information about the authentication and session is lost if the module is powered down.

The CryptoServer supports multiple simultaneous operators, each using their own session key for message authentication for the service requests. This ensures the separation of the authorized roles and services performed by each operator.

At the end of a session, the operator can log out, or, after 15 minutes of inactivity, the session key is invalidated inside the cryptographic module.

Table 8 – Roles and Required Identification and Authentication

Role	Type of Authentication	Authentication Data
Cryptographic User (called *User* in [FIPS140-2])	Identity-based operator authentication	User Name and Password or User Name and RSA Signature
User (sub-role of Cryptographic User)		
Key Manager (sub-role of Cryptographic User)		
Security Officer		
Administrator (called *Crypto Officer* in [FIPS140-2])		
NTP Manager		

Table 9 – Strength of Authentication Mechanisms

Authentication Mechanism	Strength of Mechanism
Username and Password (minimum 4 characters password chosen from 94 printable ASCII characters)	The probability that a random attempt will succeed or a false acceptance will occur is 1/(94^4), which is less than 1/1,000,000. Due to a correctional delay of 120 milliseconds for every non-successful authentication there is a maximum limit of 60 * 1000 / 120 = 500 non-successful authentications per minute. This can be stated as allowing only 500 non-successful authentication attempts per minute based on a rate of 120 ms per attempt. Therefore the probability of successfully authenticating to the module within one minute is (less than) 500 * 1/(94^4), which is less than 1/100,000.
RSA Signature (minimum 1024 bit key)	The probability that a random attempt will succeed or a false acceptance will occur is less than or equal to approximately [1/(2^{80})] (according to SP 800-57-Part1 Table 2) which is less than 1/1,000,000. Due to a correctional delay of 120 milliseconds for every non-successful authentication, there is a maximum limit of 60 * 1000 / 120 = 500 non-successful authentications per minute. This can be stated as allowing only 500 non-successful authentication attempts per minute based on a rate of 120 ms per attempt. Therefore, the probability of successfully authenticating to the module within one minute is less 500 * [1/(2^{80})], which is less than 1/100,000.

5 Access Control Policy

The CryptoServer offers different administration and cryptographic services, some of them require operator authentication (denoted below as authenticated services) and other can be used by all users without any authentication (denoted below as unauthenticated services).

This chapter describes all services provided by the CryptoServer and identifies the operator roles allowed to access those services (see sections 5.1 and 5.2). Furthermore, it specifies the Critical Security Parameters (CSPs) of the CryptoServer (see section 5.4) and the public keys stored on it (see section 5.5). Section 5.6 specifies for each user role, the services an operator is authorized to perform within that role and for each service within each role, the type(s) of access to the cryptographic keys and CSPs.

5.1 Roles and Authenticated Services

General definitions:

- An ***Operator*** may be an Administrator, an NTP Manager, a Security Officer or a Cryptographic User, User or Key Manager.
- An ***Object*** may be a (cryptographic) key, a storage object or a configuration object.
- A ***Backup Blob*** contains an Object. Secret keys (incl. Generic Secrets) and private key parts are always encrypted with the Master Backup Key (back-up key, see section 5.4) within a Backup Blob.
- Each Object and each Operator may be assigned to a ***Key Group.***
- An Object is ***Local*** if it is assigned to a Key Group; an Object which is assigned to no Key Group is called ***Global***.
- An Object is ***Assigned*** to an Operator if both are assigned to the same Key Group, or if the Object is Global.

Table 10 – Authenticated Services

Role	Authenticated Services
Cryptographic User:	This role provides all cryptographic services, i.e., services for management and use of Assigned private, public and secret keys, hashing services and random number generation. It comprises all services authorized for *Key Managers* and all services authorized for *Users*.
Key Manager: This role provides all key management services.	■ Change Operator's Password or Key: This service changes the password or RSA public key which is used for the *Key Manager's* authentication and resets the user's counter for consecutive failed authentication attempts. ■ Get Session Key: This service generates a new Secure Messaging session key for secure communication to the module. ■ List Keys: This service outputs the key properties (such as the algorithm, key name, key size, etc.) of all Assigned cryptographic keys and storage objects stored inside the cryptographic module.

Role	Authenticated Services
	■ Open Key: This service opens an Assigned Object which is stored inside the cryptographic module and returns a key handle or a Backup Blob containing the Object itself. ■ Get Key Property: This service returns one or more properties (attributes) of an Assigned Object. It can export the public part of a cryptographic key but no secret or private key parts. ■ Set Key Property: This service sets one or more properties (attributes) for an Assigned key or storage object (but no key parts). ■ Backup Key: This service outputs a Backup Blob containing an Assigned key or storage object for back-up purposes. The Backup Blob additionally AES CBC encrypted and authenticated with an AES CMAC by Secure Messaging. ■ Restore Key: This service imports a Backup Blob containing the back-up of an Assigned key or storage object into the cryptographic module. Optionally the key or storage object can also be exported within a Backup Blob. All Backup Blobs are additionally AES CBC encrypted and authenticated with an AES CMAC by Secure Messaging. ■ Delete Key: This service deletes an Assigned key or storage object from the module. ■ Generate DSA Parameters: This service generates a DSA Domain Parameter set P, Q and G using the DRBG. ■ Generate DSA Parameters PQ: This service generates a DSA Domain Parameter set P and Q using the DRBG. ■ Generate DSA Parameters G: This service generates a DSA Domain Parameter G by given P and Q (optionally) using the DRBG. ■ Compute Hash: This service calculates a SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 or SHA-3 hash or HMAC value for given data or for the components of an Assigned key. ■ Generate Key: This service generates a cryptographic key (Triple-DES, AES, RSA, DSA, EC or Generic Secrets) using the DRBG. On request, the generated key is not stored but exported within a Backup Blob. ■ Export Key: This service outputs an Assigned cryptographic key. The exported key is AES CBC encrypted and authenticated with an AES CMAC by Secure Messaging. ■ Import Key: This service imports a cryptographic key into the cryptographic module. The key must be AES CBC encrypted and authenticated with an AES CMAC by Secure Messaging. On request the imported key can be exported again. ■ Generate Key Pair: This service generates a cryptographic key pair (RSA, DSA or EC) using the DRBG and stores the two key parts in different key objects. On request the generated key parts are not stored but exported within two Backup Blobs.

Role	Authenticated Services
	■ Derive Key: This function derives an AES key or a Generic Secret from an Assigned base key (DSA or EC). The derived key or secret is stored in the CryptoServer, or exported within a Backup Blob. ■ Split Key: This function cuts keying material (stored as a Generic Secret) in non-overlapping DES and/or AES keys or Generic Secrets. The original key is deleted from the database, the derived keys are stored in the CryptoServer, or exported within a Backup Blob. ■ Wrap Key: This function exports an Assigned key in form of a key blob, which is formatted as required by PKCS#11 (see [PKCS#11]). The key blob is additionally AES CBC encrypted and authenticated with an AES CMAC by Secure Messaging. ■ Unwrap Key: This function imports an Assigned key from an encrypted key blob. The key is encoded as specified by PKCS#11 (see [PKCS#11]). The key blob is additionally AES CBC encrypted and authenticated with an AES CMAC by the current Secure Messaging session. ■ Create Object: This function creates an Assigned cryptographic key or storage object according to the given property list. The created object is either stored within the CryptoServer or exported within a Backup Blob. ■ Copy Object: This function copies an Assigned key or storage object. A template may be given that contains an additional list of properties which should be added to the original properties or replace existing properties. The copied object is either stored within the CryptoServer or exported within a Backup Blob.
User: This role provides all cryptographic services, i. e., services for use of private, public and secret keys, hashing services and random number generation.	■ Change Operator's Password or Key: This service changes the password or RSA public key which is used for the User's authentication and resets the User's counter for consecutive failed authentication attempts. ■ Get Session Key: This service generates a new Secure Messaging session key for secure communication to the module. ■ List Keys: This service outputs the key properties (such as the algorithm, key name, key size, etc.) of all Assigned keys and storage objects stored inside the cryptographic module. ■ Open Key: This service opens an Assigned Object which is stored inside the cryptographic module and returns a key handle or a Backup Blob containing the Object itself. ■ Get Key Property: This service returns one or more properties (attributes) of an Assigned Object. It can export the public part of a key but no secret or private key parts. ■ Generate DSA Parameters: This service generates a DSA Domain Parameter set P, Q and G using the DRBG. ■ Generate DSA Parameters PQ: This service generates a DSA Domain Parameter set P and Q using the DRBG.

Role	Authenticated Services
	■ Generate DSA Parameters G: This service generates a DSA Domain Parameter G by given P and Q (optionally) using the DRBG. ■ Generate Random Number: This service generates a random number using the DRBG. ■ Crypt Data: This service encrypts or decrypts data using an Assigned Triple-DES or AES key in CBC or ECB mode (Triple-DES, decryption only) or in ECB, CBC, OFB, CTR, GCM, CCM mode (AES). ■ Sign Data: This service generates an RSA, DSA or ECDSA signature or calculates an AES CMAC, AES GMAC or HMAC for given data with an Assigned signing key. ■ Verify Signature: This service verifies an RSA, DSA or ECDSA signature or a Triple-DES MAC, AES CMAC, AES GMAC or HMAC using an Assigned verification key. ■ Compute Hash: This service calculates a SHA-1, SHA-2 or SHA-3 hash or HMAC value for given data or for the components of an Assigned key. ■
Administrator: This role provides all services necessary for firmware and user management.	■ Change Operator's Password or Key: This service changes the password or RSA public key, which is used for an operator's authentication, and resets the operator's counter for consecutive failed authentication attempts. ■ Get Session Key: This service generates a new Secure Messaging session key for secure communication to the module. ■ Add Operator: This service adds an Operator to the cryptographic module. ■ Delete Operator: This service deletes an Operator from the cryptographic module. ■ Add Group User (for Security Officer): This service adds a *Security Officer* to the cryptographic module. ■ Delete Group User (for Security Officer): This service deletes a *Security Officer* from the cryptographic module. ■ Backup User: This service exports all user account data for a given user for backup purposes. All secrets (passwords) are encrypted in the exported data with the Master Backup Key and additionally AES CBC encrypted and authenticated with an AES CMAC by Secure Messaging. ■ Restore User: This service creates a new user in the user database. All information about the user (name, permission, authentication token, etc.) is taken from a backup data block that was output by the *Backup User* service and which is additionally AES CBC encrypted and authenticated with an AES CMAC by Secure Messaging.

Role	Authenticated Services
	■ List Master Backup Keys: This service outputs information (key type, key size, key check value, etc.) about all Master Backup Keys (back-up keys) that are stored inside the CryptoServer. ■ Generate Master Backup Key: This service generates and outputs a Master Backup Key (back-up key). The key is only exported in a wrapped form, AES CBC encrypted and authenticated with an AES CMAC by the current Secure Messaging session. The generated key is not stored inside the CryptoServer. ■ Import Master Backup Key: This service imports a Master Backup Key (back-up key). The key is only imported if AES CBC encrypted and authenticated with an AES CMAC by the current Secure Messaging session. ■ Load File: This service loads files. If a file with the same file name is currently loaded, that current file will be replaced. This command is usually used to load and replace firmware modules. If the file is a firmware module, the old file will only be replaced if the RSA signature for the firmware module is verified successfully. (Note: loading non-FIPS-validated firmware onto the cryptographic module will cause the module to cease being FIPS-validated.) ■ Delete File: This service is used to delete files. (Note: deleting FIPS-validated firmware from the cryptographic module will cause the module to cease being FIPS-validated.) ■ Clear Audit Log: This service deletes the audit log file except for the first ‘k’ parts. ■ Clear Audit Log Files: This service deletes audit log files up to the given file number ‘n’. Optionally it can be checked before, if the youngest file to be deleted has not changed compared to the latest audit log file that was read out. ■ Generate Audit Log Key: This service generates and stores an (RSA or ECDSA) Audit Log Signature Key which may be used for signing audit log files with function ‘Get Signed Audit Log’. ■ Get Signed Audit Log: This service returns the requested audit log file, signed with the Audit Log Signature Key. ■ List DB Search Keys: This service returns all search keys of a given database. ■ Export DB Entry: This service exports a given database entry encrypted by the CryptoServer’s Master Backup Key. ■ Import DB Entry: This service imports an encrypted database entry created by the function Export DB Entry. ■ Set Maximum Failure Counter: This service sets the maximum number of allowed consecutive failed authentication attempts before a user is blocked. ■ Set Administration-Only Mode: This service switches the CryptoServer into Administration-Only Mode (all cryptographic services are blocked, only administrative services are available) or back to the Operational Mode.

Role	Authenticated Services
	■ Set Startup Mode: This service configures the startup mode of the CryptoServer. If the startup mode is set to 1, the CryptoServer will always boot into Administration-Only Mode after a restart. ■ Set Time, Set Time Rel: These services are used to set the internal clock on the module. ■ List Keys (for the Global configuration object): This service lists the Global configuration objects. ■ Open Key (for configuration objects): This service opens a configuration object and returns a reference, or the configuration object itself is exported. ■ Get Key Property (for configuration objects): This service returns one or more configuration properties. ■ Set Key Property (for the Global configuration object): This service sets one or more Global configuration properties. ■ Backup Key (for the Global configuration object): This service outputs the Global configuration object for back-up purposes. The backup blob is AES CBC encrypted and authenticated with an AES CMAC by Secure Messaging. ■ Restore Key (for the Global configuration object): This service imports the back-up of the Global configuration object into the cryptographic module. The backup blob is AES CBC encrypted and authenticated with an AES CMAC by Secure Messaging. ■ Delete Key (for the Global configuration object): This service deletes all Global configuration values by setting them to their default values.
Security Officer: This role provides all services necessary for Key Group specific user and configuration management.	■ Change Operator's Password or Key: This service changes the password or RSA public key which is used for the *Security Officer's* authentication and resets his counter for consecutive failed authentication attempts. ■ Get Session Key: This service generates a new Secure Messaging session key for secure communication to the module. ■ Add Group User (for a *Cryptographic User, Key Manager* or *User)*: This service adds a *Cryptographic User, Key Manager* or *User* to the cryptographic module. The added operator and the authorizing *Security Officer* must be assigned to the same Key Group. ■ Delete Group User (for a *Cryptographic User, Key Manager* or *User)*: This service deletes a *Cryptographic User, Key Manager* or *User* from the cryptographic module. The deleted operator and the authorizing *Security Officer* must be assigned to the same Key Group. ■ List Keys (for Local configuration objects): This service lists all Assigned Local configuration objects. ■ Open Key: This service opens an Assigned Object and returns a reference or a Backup Blob containing the Object itself.

Role	Authenticated Services
	■ Get Key Property: This service returns one or more properties (attributes) of an Assigned Object. It can export the public part of a key but no secret or private key parts. ■ Set Key Property: This service allows the Security Officer to set a Local configuration value, or to set the TRUSTED attribute of an Assigned key encryption key. ■ Backup Key (for Local configuration objects): Output an Assigned Local configuration object for backup purposes. The backup blob is AES CBC encrypted and authenticated with an AES CMAC by Secure Messaging. ■ Restore Key (for Local configuration objects): Import the backup copy of a Local configuration object into the cryptographic module. The backup blob is AES CBC encrypted and authenticated with an AES CMAC by Secure Messaging. ■ Delete Key (for Local configuration objects): Delete an Assigned Local configuration object by setting all configuration attributes to their default values. ■ Init Key Group: Delete all Local Objects belonging to a given Key Group.
NTP Manager: This role provides all services necessary for NTP time synchronization on the CryptoServer by using an NTP server over a network	■ Change Operator's Password or Key: This service changes the password or RSA public key which is used for the *NTP Manager's* authentication and resets his counter for consecutive failed authentication attempts. ■ Get Session Key: This service generates a new Secure Messaging session key for secure communication to the module. ■ Change Activation State: Change the state of the NTP firmware module from deactivated to activated and vice versa. ■ Set NTP Settings: Allow setting the NTP attributes `MaxAdjustPerOperation` and `MaxAdjustPerDay` for the maximum time adjustment that can be performed with the 'Set Time Delay' function. ■ Set Time NTP: This service sets the time of the CryptoServer.

5.2 Unauthenticated Services

In addition to the services requiring operator authentication, the CryptoServer supports the following unauthenticated services available to any operator without any authentication required.

- Get Boot Log: Retrieve a log file containing log messages written by the operating system and other firmware modules (or by the boot loader if the command is called in bootloader mode) during the boot process.
- Show Status (or "GetState"): View the current status of the cryptographic module, including the FIPS mode indicator.
- Get Time: Read out the current time of the internal Real Time Clock of the CryptoServer.

- Get Maximum Fail Count: Output the maximum number of allowed consecutive failed authentication attempts before a user is blocked.
- Get Startup Mode: Show the startup mode of the CryptoServer.
- Get HSM Auth Key: Retrieve the public part of the device-individual HSM Authentication Key for mutual authentication. On first execution of the service, the HSM Authentication Key is generated.
- Get Audit Log Key: Retrieve the public part of the Audit Log Signature Key.
- List Files: Retrieve a list of all files stored in the CryptoServer.
- List Active Modules: List all currently active firmware modules.
- List Operators: Read a list of all Security Officers, Cryptographic Users, Key Managers, Users, NTP Managers and Administrators.
- Get Operator Info: Retrieve all non-sensitive information about the specified operator.
- End Session: Terminate a Secure Messaging session by invalidating the relevant session key.
- Get Audit Log: Read an audit log file.
- Get Audit Config: Read the configuration for auditing.
- Get Memory Info: Return statistical information about the file system usage.
- Echo: Communication test (echo input data).
- Get Challenge: Generate and output a challenge (16 bytes random value generated by the CryptoServer's deterministic random bit generator) for using the challenge/response mechanism in the next authenticated command.
- Get Authentication State: Return the current authentication state and an optional list of all operators that are authenticated within the current session.
- Get CXI Info: Return some status information about the CXI firmware module, for example, module version number or the fill level of the database.
- Set Time Delay: Adjust the CryptoServer time (RTC) by a given number of seconds and milliseconds. The relative time change cannot exceed the limits given by the `MaxAdjustPerOperation` and `MaxAdjustPerDay` NTP attributes.
- Get NTP Settings: Return the current settings of the `MaxAdjustPerOperation` and `MaxAdjustPerDay` NTP attributes.
- P11 Permissions: Return information about the roles regarding users who are currently logged in to the CryptoServer (defined according to [PKCS#11]: Cryptographic User, Security Officer and Key Manager), restricted to users matching the specified Key Group.
- Initiate Self Tests: At any time, the execution of the self-tests required by FIPS 140-2 can be forced by performing a reset or power-cycle of the module. During self-test execution, no further command processing is possible.
- Zeroize: Zeroize the cryptographic module including all critical security parameters. All CSPs that are not wrapped by the Master Key are zeroized. This service is executed only after an external erase. (Note: After zeroization, the CryptoServer is no longer in FIPS mode.)

If the cryptographic module is in FIPS error state, the only services that are available are a small subset of these unauthenticated services. These services only output status information and do not perform any cryptographic function.

5.3 Services in Non-FIPS Modes

The services that the cryptographic module provides are the same between the Approved and non-Approved (Non-FIPS) modes. Non-Approved algorithms can be used in lieu of the Approved algorithms in the non-Approved modes.

5.4 Definition of Critical Security Parameters (CSPs)

The following CSPs are contained in the module:

- CryptoServer's *Master Key* $\mathbf{K_{CS}}$ (AES CBC 32 bytes)
- *Local Secret ECDH Key* $\mathbf{K_{SM\_MOD\_PRIV}}$ (generated by the module and used to establish a shared session key derivation key via EC Diffie Hellman for Secure Messaging, see section 2.3)
  (ECDSA for curve NIST-P521, volatile storage only)
- *Session Key Derivation Key* $\mathbf{K_{KD}}$ (established according to [NIST SP 800-56A r3] using the EC Diffie Hellman algorithm and used to derive Session Keys for Secure Messaging, see section 2.3) (volatile storage only)
- *Session Keys* $\mathbf{K_{SME}}$ *and* $\mathbf{K_{SMM}}$ (derived from the *Session Key Derivation Key* $\mathbf{K_{KD}}$ and used for Secure Messaging, see section 2.3) (32 bytes AES, volatile storage only)
- *DRBG Secrets* $\mathbf{S_{DRBG}}$ used by the Deterministic Random Bit Generator (DRBG) as specified in [NIST 800-90A] (volatile storage only):
  - Entropy input $\mathbf{S_{DRBG\_EI}}$ generated by the NDRNG
  - Seed $\mathbf{S_{DRBG\_SEED}}$ calculated from Entropy input $\mathbf{S_{DRBG\_EI}}$
  - Working state constant $\mathbf{S_{DRBG\_C}}$ calculated from the $\mathbf{S_{DRBG\_SEED}}$ Seed
  - Working state value $\mathbf{S_{DRBG\_V}}$ initially calculated from the $\mathbf{S_{DRBG\_SEED}}$ Seed and updated each time the DRBG is called

The following CSPs are stored within the cryptographic module encrypted with the Master Key $\mathbf{K_{CS}}$:[24]

- Private device-individual *HSM Authentication Key* $\mathbf{K_{HA\_PRIV}}$ (3072-bit RSA key)
- Private *Audit Log Signature Key* $\mathbf{K_{AL\_PRIV}}$ (NIST-P256 based ECDSA key or 3072-bit RSA key)
- Private User Keys:
  - $\mathbf{K_{USR\_RSA\_PRIV}}$ (RSA; Signature Generation, Key Decryption)
  - $\mathbf{K_{USR\_DSA\_PRIV}}$ (DSA; Signature Generation, Key Agreement)
  - $\mathbf{K_{USR\_EC\_PRIV}}$ (EC; Signature Generation, Key Agreement)
- Secret User Keys:
  - $\mathbf{K_{USR\_AES}}$ (AES; for Key Encryption, Data Encryption or MAC)
  - $\mathbf{K_{USR\_TDES}}$ (Triple-DES; for Key Decryption, Data Decryption)

[24] Note: These non-volatile CSPs are not subject to the zeroization requirement since they are stored in encrypted form (using the AES algorithm).

  - *Generic Secret* $\mathbf{K_{USR\_GS}}$ (to be used as keying material or as a HMAC key; at least 112 bits for HMAC generation)
- *Master Backup Key* **MBK** (AES CBC 16, 24 or 32 bytes, key for back-up purposes)
- *Operator Password* $\mathbf{PSW_{AUTH}}$ (for authentication)

The functionality of keys is dependent on their attributes, as indicated by the vendor-imposed security rules in Section 6. Keys with “CRYPT” or “DECRYPT” attribute can be used for encryption, keys with the “SIGN” or “VERIFY” attribute can used for digital signatures or MACs or HMACs, keys with the "WRAP" or "UNWRAP" attribute can be used for key wrapping, and keys with the “DERIVE” attribute can be used for key establishment.

5.5 Definition of Public Keys

The following public keys are contained in the cryptographic module:

- *Production Key* (RSA 2048 bit) $\mathbf{K_{PROD\_PUB}}$
- *Module Signature Key* (RSA 4096 bit) $\mathbf{K_{MDL\text{-}SIG\_PUB}}$
- *Default Administrator Key* (RSA 1024 bit) $\mathbf{K_{ADMIN\text{-}DEF\_PUB}}$
- Public part of the device-individual *HSM Authentication Key* $\mathbf{K_{HA\_PUB}}$ (exportable 3072-bit RSA key)
- Public *Audit Log Signature Key* $\mathbf{K_{AL\_PUB}}$ (NIST-P256 based ECDSA key or 3072-bit RSA key)
- Public User Keys:
  - $\mathbf{K_{USR\_EC\_PUB}}$ (EC; Signature Verification, Key Agreement)
  - $\mathbf{K_{USR\_DSA\_PUB}}$ (DSA; Signature Verification, Key Agreement)
  - $\mathbf{K_{USR\_RSA\_PUB}}$ (RSA; Signature Verification, Key Encryption)
- Operator’s Public Authentication Key $\mathbf{K_{AUTH\_PUB}}$ (RSA)

The following public keys are used temporarily within the cryptographic module:

- *Remote Public ECDH Key* $\mathbf{K_{SM\_HOST\_PUB}}$ (generated by the host and used to establish a Session Key Derivation Key via EC Diffie Hellman for Secure Messaging)
  (ECDSA for curve NIST P-521, volatile storage only)
- *Local Public ECDH Key* $\mathbf{K_{SM\_MOD\_PUB}}$ (generated by the module and used to establish a Session Key Derivation Key via EC Diffie Hellman for Secure Messaging)
  (ECDSA for curve NIST P-521, volatile storage only)

5.6 Definition of Modes of Access to CSPs

Table 11, Table 12, Table 13, and Table 14 define the relationship between the different services provided by the cryptographic module and access to CSPs. The types of access (for example, Use/Write/Update) are given in the right-hand column.

The following types of access are possible:

- *Write:* the CSP is created (newly written) and stored.
- *Update:* replaces the current value of the CSP with a new value.

- *Use:* the value of the CSP is used for some cryptographic calculation
- *Wrapped Export:* the CSP is wrapped by some wrapping key and exported from the cryptographic module.
- *Export:* the CSP is exported from the cryptographic module (only possible for public RSA, DSA or EC keys $\mathbf{K_{USR\_RSA\_PUB}}$, $\mathbf{K_{USR\_DSA\_PUB}}$ and $\mathbf{K_{USR\_EC\_PUB}}$).
- *Delete:* invalidates the CSP
- *(xxx):* The access type (one of the access types listed above) is set in brackets if this access type is conditional.

The following definitions are used in all tables mentioned above:

- Any *User Key* can be a *Secret User Key* (K_{USR_AES}, K_{USR_TDES} or K_{USR_GS}) or a *Private* and/or *Public User Key* ($K_{USR\_RSA\_PRIV}$, $K_{USR\_RSA\_PUB}$, $K_{USR\_DSA\_PRIV}$, $K_{USR\_DSA\_PUB}$, $K_{USR\_EC\_PRIV}$, $K_{USR\_EC\_PUB}$) [25]
- A *Secret Data Encryption Key* is a *Secret AES or DES User Key* (K_{USR_AES} or K_{USR_TDES}) with attribute[26] "CRYPT"/"DECRYPT". [27]
- A *Secret Key Encryption Key* can be a *Secret AES or Triple-DES User Key* (K_{USR_AES} or K_{USR_TDES}) with attribute[28] "WRAP"/"UNWRAP".[27]
- A *Secret MAC Key* can be a *Secret User Key* (K_{USR_AES} or K_{USR_GS}) with attribute[29] "SIGN"/"VERIFY".
- A *Key Derivation Key* can be a *Private or Public EC or DSA User Key* ($K_{USR\_EC\_PRIV}$, $K_{USR\_EC\_PUB}$, $K_{USR\_DSA\_PRIV}$, $K_{USR\_DSA\_PUB}$) with attribute[30] "DERIVE".
- A *Private Sign Key* can be a *Private RSA, DSA or EC User Key* ($K_{USR\_RSA\_PRIV}$, $K_{USR\_DSA\_PRIV}$ or $K_{USR\_EC\_PRIV}$) with attribute[29] "SIGN".
- A *Public Verify Key* can be a *Public RSA, DSA or EC User Key* ($\mathbf{K_{USR\_RSA\_PUB}}$, $\mathbf{K_{USR\_DSA\_PUB}}$ or $\mathbf{K_{USR\_EC\_PUB}}$) with attribute[29] "VERIFY".

* General remark concerning the access to internal or external keys: If a key is marked with an asterisk, the key may be an internal[31] or an external[32] key. In case that such a key is accessed, the following CSPs must additionally be used:

  - When an internal *Secret or Private User Key* is to be accessed, the *Master Key* $\mathbf{K_{CS}}$ must be used to decrypt or encrypt the internal key.
  - When an external key is to be accessed, the **MBK** must be used to verify or update the MAC and/or to decrypt or encrypt the secret or private key part.

[25] In validated FIPS mode, TDES keys cannot be generated.

[26] See chapter 6, vendor imposed security rule 9.

[27] In validated FIPS mode, TDES keys can only be used for decryption and unwrapping.

[28] See chapter 6, vendor imposed security rule 12.

[29] See chapter 6, vendor imposed security rule 10.

[30] See chapter 6, vendor imposed security rule 11.

[31] An "internal key" is any User Key that is stored inside the cryptographic module.

[32] An "external key" is any User Key that is stored outside the cryptographic module in the form of a secured *Backup Blob* (e. g. as result of the *Backup Key* service). A *Backup Blob is integrity protected with a MAC;* secret and private key parts are always encrypted with the Master Backup Key **MBK**.

** General remark concerning *DRBG Secrets* $S_{DRBG:}$

- If a new block of random values must be generated but no reseeding is required, the *DRBG Secrets* S_{DRBG_C} and S_{DRBG_V} are used and S_{DRBG_V} is updated.
- If a new block of random values must be generated and reseeding is required, all *DRBG Secrets* S_{DRBG_EI}, S_{DRBG_SEED}, S_{DRBG_C} and S_{DRBG_V} are updated and used.

Below, the four left-hand columns indicate the *Roles* for which each service is available.

An asterisk in brackets (*) indicates that the service can be executed by the user but no keys or CSPs are accessed by the service.

Table 11 – CSP and Key Access Rights within Roles & Services – General Services

Role				Service	Cryptographic Keys and CSPs Access Operation	Type of Access
Ad-minis trator	Security Officer	CU, KM, U[33]	NTP Manager			
X	X	X	X	any command authentication	*Public Authentication Key* K_{AUTH_PUB} or *Password* PSW_{AUTH} of respective operator	*Use*
X	X	X	X	any command using *Secure Messaging*	*Session Keys* K_{SME} *and* K_{SMM}	*Use*[34]
X	X	X	X	Get Session Key	*DRBG Secrets* S_{DRBG}**	*Use, Update*
					Remote Public ECDH Key $K_{SM\_HOST\_PUB}$	*Use*
					Local Private ECDH Key $K_{SM\_MOD\_PRIV}$	*Use*
					Local Public ECDH Key $K_{SM\_MOD\_PUB}$	*Export*
					Session Key derivation key K_{KD}	*Use*
					Session Keys K_{SME} *and* K_{SMM}	*Write*
					Device-individual private HSM Authentication Key K_{HA_PRIV}	*Use* *Write*[35]
					Device-individual public HSM Authentication Key K_{HA_PUB}	*Write*[35]
(all without authentication)				End Session	*Session Keys* K_{SME} *and* K_{SMM}	*Delete*[36]

[33] Cryptographic User, Key Manager, User

[34] KTS with AES CBC + CMAC

[35] If the key pair is not present

[36] Invalidated within Key Cache; Key Cache is zeroized on power cycle and in case of an alarm.

Role				Service	Cryptographic Keys and CSPs Access Operation	Type of Access
Ad-minis trator	Security Officer	CU, KM, U[33]	NTP Manager			
(all without authentication)				Get HSM Auth Key	*Device-individual public HSM Authentication Key* K_{HA_PUB}	*Export, Write*[35]
					Device-individual private HSM Authentication Key K_{HA_PRIV}	*Use, Write*[35]
(all without authentication)				Get Audit Log Key	*Public Audit Log Signature Key* K_{AL_PUB}	*Export*
X	X	X	X	Change Operator's Key or Password	*Public Authentication Key* K_{AUTH_PUB} *or Password* PSW_{AUTH} *of Operator*	*Update*
					If operator uses a password: *CryptoServer's Master Key* K_{CS}	*(Use)*
(without authentication; only executed when an external erase is triggered by pushing the 'Erase' push-button on the PCIe card)				Zeroize	*CryptoServer's Master Key* K_{CS}	*Delete*[37]
					All CSPs that are stored temporarily in the Key Cache (volatile storage)	*Delete*[38]
					All CSPs that are stored wrapped with the Master Key	*Delete*[39]

Table 12 – CSP and Key Access Rights within Roles & Services – Administration

Role				Service	Cryptographic Keys and CSPs Access Operation	Type of Access
Ad-minis-trator	Security Officer	CU, KM, U[40]	NTP Manager			
X				Add Operator	*Public Authentication Key* K_{AUTH_PUB} or *Password* PSW_{AUTH} *of Operator*	*Write*
					If operator uses password: *CryptoServer's Master Key* K_{CS}	*(Use)*
X				Delete Operator	Public Authentication Key K_{AUTH_PUB} or Password PSW_{AUTH} of Operator	Delete[41]
X	X			Add Group User	Public Authentication Key K_{AUTH_PUB} or Password PSW_{AUTH} of Operator	*Write*
					If operator uses password: *CryptoServer's Master Key* K_{CS}	*(Use)*
X	X			Delete Group User	*Public Authentication Key* K_{AUTH_PUB} or *Password* PSW_{AUTH} *of Operator*	*Delete*[41]

[37] Zeroized by overwriting the Key-RAM five times, alternately with 00_h and FF_h patterns.

[38] Key Cache is zeroized by overwriting each memory cell of the Key Cache five times, alternately with 00_h and FF_h patterns.

[39] CSPs are invalidated by zeroizing the Master Key K_{CS} because they are encrypted with the Master Key K_{CS}.

[40] Cryptographic User, Key Manager, User

[41] Invalidated within database; no zeroization needed because it is stored encrypted with the Master Key K_{CS}.

Role				Service	Cryptographic Keys and CSPs Access Operation	Type of Access
Ad-minis-trator	Security Officer	CU, KM, U[40]	NTP Manager			
X				Backup User	*Public Authentication Key* K_{AUTH_PUB} or *Password* PSW_{AUTH} *of Operator*	*Wrapped Export*
					Master Backup Key ***MBK***	*Use*
					CryptoServer's Master Key K_{CS}	*Use*
X				Restore User	*Public Authentication Key* K_{AUTH_PUB} or *Password* PSW_{AUTH} *of Operator*	*Write* or *Update*
					Master Backup Key ***MBK***	*Use*
					CryptoServer's Master Key K_{CS}	*Use*
X				Load File	If file to be loaded is a firmware module: *Public Module Signature Key* $K_{MDL\text{-}SIG\_PUB}$	(Use)
X				Delete File	---	---
X				Clear Audit Log	---	---
X				Set Max Fail Cnt	---	---
X				Set Time	---	---
X				Set Time Rel	---	---
			X	Set Time NTP	---	---
			X	Change Activation State	---	---
			X	Set NTP Settings	---	---
X				List Master Backup Keys	---	---
X				Clear Audit Log Files	---	---
X				Generate Audit Log Key	*Public Audit Log Signature Key* K_{AL_PUB}	*Write, Export*
					Private Audit Log Signature Key K_{AL_PRIV}	*Write, Use*
X				Get Signed Audit Log	*Private Audit Log Signature Key* K_{AL_PRIV}	*Use*
X				List DB Search Key	---	---
X				Export DB Entry	*Master Backup Key* ***MBK***	*Use*
					If database entry whose back-up copy will be exported contains a *User Key* or the *Audit Log Signature Key*: *Any User Key* or *Private* and *Public Audit Log Signature Key* K_{AL_PRIV} and K_{AL_PUB}	*(Wrapped Export)*

Role				Service	Cryptographic Keys and CSPs Access Operation	Type of Access
Ad-minis-trator	Security Officer	CU, KM, U[40]	NTP Manager			
					If database entry whose back-up copy will be exported is a user database entry: *Public Authentication Key* K_{AUTH_PUB} or *Password* PSW_{AUTH} *of Operator*	*(Wrapped Export)*
					If database entry whose back-up copy will be exported contains a secret part (private/secret key or password): *CryptoServer's Master Key* K_{CS}	*(Use)*
X				Import DB Entry	*Master Backup Key MBK*	*Use*
					If database entry whose back-up copy will be imported contains a *User Key* or the *Audit Log Signature Key*: *Any User Key* or *Private* and *Public Audit Log Signature Key* K_{AL_PRIV} and K_{AL_PUB}	*(Write* or *Update)*
					If database entry whose back-up copy will be imported is a user database entry: *Public Authentication Key* K_{AUTH_PUB} or *Password* PSW_{AUTH} *of Operator*	*(Write* or *Update)*
					If database entry whose back-up copy will be imported contains a secret part (private/secret key or password): *CryptoServer's Master Key* K_{CS}	*(Use)*
X				Set Administration -Only Mode	---	---
X				Set Startup Mode	---	---
X				Generate Master Backup Key	Master Backup Key **MBK**	*Wrapped Export*
					Session Keys K_{SME} and K_{SMM}	*Use*
					DRBG Secrets S_{DRBG}**	*Use* and *Update*
X				Import Master Backup Key	Master Backup Key **MBK**	*Write* or *Update*
					Session Keys K_{SME} and K_{SMM}	*Use*
					CryptoServer's Master Key K_{CS}	*Use*

Table 13 – CSP and Key Access Rights within Roles & Services – Key Management

Role					Service	Cryptographic Keys and CSPs Access Operation	Type of Access
Ad-minis-trator	Secu-rity Officer	Cryptographic User		NTP Manager			
		User	Key Mgr				
	X				Init Key Group	*Any User Key*	*Delete*[42]
(*)	X	X	X		Open Key	If requested key is to be exported: *Any User Key**	(*Wrapped Export)*
(*)	(*)	(*)	(*)		List Keys	---	---
(*)	(*)		X		Delete Key	*Any User Key*	*Delete*[42]
X	X	X	X		Get Key Property*	If Public User Key is requested: *Any Public User Key** (***K**USR_RSA_PUB,* ***K**USR_DSA_PUB* or ***K**USR_EC_PUB*)*	*(Export)*
(*)	(*)		(*)		Set Key Property*	--- (if an external key is addressed, the ***MBK*** is used to verify and update the MAC)	---
(*)	(*)		X		Backup Key	*Any User Key*	*Wrapped Export*
						Master Backup Key ***MBK***	*Use*
						If key whose back-up copy will be exported is *Private* or *Secret User Key*: *CryptoServer's Master Key* $\boldsymbol{K_{CS}}$	*(Use)*
(*)	(*)		X		Restore Key	*Any User Key*	*Write* or *Update* or *Wrapped export*
						Master Backup Key ***MBK***	*Use*
						If key which will be restored is *Private* or *Secret User Key* and shall be stored internally: *CryptoServer's Master Key* $\boldsymbol{K_{CS}}$	*(Use)*

[42] Invalidated within database; no zeroization needed because it is only stored encrypted with the Master Key $\boldsymbol{K_{CS}}$.

Role					Service	Cryptographic Keys and CSPs Access Operation	Type of Access
Ad-minis-trator	Secu-rity Officer	Cryptographic User		NTP Mana ger			
		User	Key Mgr				
			X		Generate Key, Generate Key Pair	*DRBG Secrets* S_{DRBG}**	*Use* and *Update*
						*Any User Key**	*Write* or *Update* (if the generated key shall be stored in the CryptoServer) or *Wrapped Export* (if the generated key shall be exported outside the CryptoServer)
			X		Export Key	*Any User Key**	*Wrapped Export*
						Optional: *Secret Key Encryption Key** or *Public RSA User Key* $K_{USR\_RSA\_PUB}$*	*(Use)* [43]
						Only if random padding is required: *DRBG Secrets* S_{DRBG}**	*(Use* and *Update)*
			X		Import Key	*Any User Key**	*Write* or *Update* or *Wrapped Export*
						Optional: *Secret Key Encryption Key** or *Private RSA User Key* $K_{USR\_RSA\_PRIV}$*	*(Use)*[43]
			X		Derive Key (option ECDH_COF or DH or TLS12_PRF)	*Key Derivation Key(s)**	*Use*
						*Secret User Key**	*Write* or *Update or Wrapped Export*
			X		Split Key	*Generic Secret* K_{USR_GS}*	*Use* and *Delete*[44]
						*Secret User Key**	*Write* or *Update* or *Wrapped Export*

[43] Key (un)wrapping: AES KW(P), AES CCM, AES GCM or RSADP

[44] Invalidated within database; no zeroization needed because it is only stored encrypted with the Master Key K_{CS}.

Role					Service	Cryptographic Keys and CSPs Access Operation	Type of Access
Ad-minis-trator	Secu-rity Officer	Cryptographic User		NTP Mana ger			
		User	Key Mgr				
			X		Wrap	*Any User Key**	*Wrapped Export*
						*Secret Key Encryption Key** or *Public RSA User Key* $K_{USR\_RSA\_PUB}$*	*Use*[43]
						Only if random padding is required: *DRBG Secrets* S_{DRBG}**	*(Use* and *Update)*
			X		Unwrap	*Any User Key**	*Write* or *Update* or *Wrapped Export*
						*Secret Key Encryption Key** or *Private RSA User Key* $K_{USR\_RSA\_PRIV}$*	*Use*[43]
			X		Create Object	*Any User Key**	*Write* or *Update* or *Wrapped Export*
			X		Copy Object	*Any User Key**	*Write* or *Wrapped Export*

Table 14 – CSP and Key Access Rights within Roles & Services – Cryptographic Services

Role					Service	Cryptographic Keys and CSPs Access Operation	Type of Access
Ad-minis-trator	Secu-rity Officer	Crypto-graphic User		NTP Mana ger			
		User	Key Mgr				
		X			Crypt Data	*Secret Data Encryption Key**	*Use* [43]
						If random padding is required: *DRBG Secrets* S_{DRBG}**	*(Use* and *Update)*
		X			Sign Data	*Private Sign Key** or *Secret MAC Key**	*Use*
						If random padding is required: *DRBG Secrets* S_{DRBG}**	*(Use* and *Update)*
		X			Verify Signature	*Public Verify Key** or *Secret MAC Key**	*Use*
		X			Generate Random Number	*DRBG Secrets* S_{DRBG}**	*Use* and *Update*
		X	X		Compute Hash	optional*: Generic Secret* K_{USR_GS}*	*(Use)*
		X	X		Generate DSA Parameters (_PQ/G)	*DRBG Secrets* S_{DRBG}**	*Use* and *Update*

6 Security Rules

The cryptographic module's design complies with the cryptographic module's security rules.

This section documents the security rules enforced by the cryptographic module to implement the security requirements of a FIPS 140-2 Level 3 module.

1. The cryptographic module provides at least two distinct operator roles. These are the *User* role and the *Crypto Officer* role.
2. The cryptographic module provides identity-based authentication.
3. No access to any cryptographic services is permitted until the operator has been authenticated into the "Cryptographic User", "User", "Key Manager", "Security Officer" or "Administrator" role by the module.
4. The cryptographic module performs the following tests:
   a) Power up Self-Tests:
      i) Cryptographic Algorithm Tests:
         (1) AES Known Answer Tests (encrypt and decrypt: ECB, CBC, OFB) (Cert. #C1114)
         (2) AES-CMAC Known Answer Test (Cert. #C1134)
         (3) AES GMAC, GCM encrypt and GCM decrypt Known Answer Tests (Cert #C1245)
         (4) DRBG Known Answer Tests according to [NIST 800-90A] (testing the Instantiate Function, the Generate Function and the Reseed Function) (Cert. #A1066)
         (5) DSA Pair-wise Consistency Test (sign/verify) (Cert. #C1189)
         (6) ECDSA Pair-wise Consistency Test (sign/verify) (Cert. #C1190 / C1191)
         (7) ECC DH and FFC DH Known Answer tests (meeting IG D.8)
         (8) HMAC Known Answer Tests[45] (Cert. #C1136)
         (9) KBKDF SP 800-108 Known Answer Test (Cert. #C1162)
         (10) KDF Known Answer Tests for:
            (a) ANSI X9.42 KDF (Cert. #A1019)
            (b) ANSI X9.63 KDF (Cert. #C1135)
            (c) NIST 56C KDF
            (d) TLS 1.2 KDF (Cert. #C1163)
         (11) RSA Known Answer Tests (sign and verify) (Cert. #C1192 / C1193)
         (12) RSA decryption primitive Known Answer Tests (wrap and unwrap) (Cert. #C1115/C1116)
         (13) SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 Known Answer Tests (Cert. #C1117)

[45] kat_hmac_sha1, kat_hmac_sha224, kat_hmac_sha256, kat_hmac_sha384, kat_hmac_sha512, kat_hmac_sha3_224, kat_hmac_sha3_256, kat_hmac_sha3_384, kat_hmac_sha3_512

(14) SHA3-224, SHA3-256, SHA3-384, and SHA3-512 Known Answer Tests (Cert. #C1118)

(15) BL SHA: SHA-512 Known Answer Test (Cert. #C1119)

(16) SMOS SHA: SHA-512 Known Answer Test (Cert. #A1065)

(17) Triple-DES ECB and CBC encrypt and decrypt Known Answer Tests (Cert. #C1121)

ii) Firmware Integrity Test (CRC (32 bit) verification for boot loader program code, SHA-512 hash value verification for the module program code for every firmware module)

iii) Entropy source Power-Up tests:

(1) According to SP 800-90B:

(a) Repetition Count Test according to SP 800-90B §4.4.1

(b) Adaptive Proportion Test according to SP 800-90B §4.4.2

(2) According to [AIS 20/31] (RNG class PTG.2):

(a) Continuous Chi-Squared Test according to AIS 20/31 §5.5.3

(b) Start-up Chi-Squared Test according to AIS 20/31 §5.5.2

iv) Critical Functions Tests

(1) SDRAM Test

(2) Master Key Consistency Test

(3) Temperature Test

b) Conditional Self-Tests:

i) *Continuous Random Number Generator (RNG) Test* performed on DRBG: Prior to each use, the DRBG is tested using the conditional test specified in FIPS 140-2 §4.9.2.

ii) Entropy source Continuous tests:

(1) According to SP 800-90B:

(a) Repetition Count Test according to SP 800-90B §4.4.1

(b) Adaptive Proportion Test according to SP 800-90B §4.4.2

(2) According to [AIS 20/31] (RNG class PTG.2):

(a) Continuous Chi-Squared Test according to AIS 20/31 §5.5.3

iii) DSA Key *Pair-wise Consistency Test* (sign/verify) for DSA key generation

iv) ECDSA Key *Pair-wise Consistency Test* (sign/verify) for EC key generation

v) RSA Key *Pair-wise Consistency Test* (sign/verify and encrypt/decrypt) for RSA key generation

vi) *Firmware Load Test* (via RSA 4096 signature verification, Cert. #C1192 / C1193)

vii) *Public Key Validation* as required by SP 800-56Ar3 (Cofactor) Ephemeral Unified Model (full public key validation according to SP 800-56Ar3 section 5.6.2.3.3)

5. At any time, the operator can force the module to perform the power-up self-test.
6. Data output is inhibited during key generation, self-tests, zeroization, and error states.

7. Status information does not contain CSPs or sensitive data that if misused could lead to the compromising of the module.
8. The module supports concurrent operators.
9. The successful completion of the power-up self-tests is indicated by executing the csadm “GetState” command which returns state = INITIALIZED and FIPS mode = ON.

The following security rules are imposed by the vendor:

1. The module zeroizes all plaintext CSPs within a maximum of 4 ms after any attack or alarm (see chapter 7 below).
2. If the cryptographic module remains inactive in any valid role for a maximum period of 15 minutes, the module automatically logs off the operator.
3. The module provides functionality for protecting command and response data on their way to and from the module via a *Secure Messaging* mechanism. This mechanism encrypts and integrity protects the data with the AES encrypting algorithm and CMAC. In FIPS mode, the use of Secure Messaging is mandatory for every command that has to be authenticated.
4. The module implements a Challenge-Response mechanism to prevent the replay of older authenticated messages.
5. The module prohibits the export of plaintext secret or private cryptographic keys or other CSPs.
6. The module supports an “Exportable” attribute for every stored private or secret cryptographic key. The module only permits the (wrapped) export of a key if this attribute is set.
7. The module supports a “Deny_backup” attribute for every stored private or secret cryptographic key. The module only permits the MBK encrypted export (export for backup purposes) of a key if this attribute is NOT set.
8. The module supports an (optional) “Key Group” attribute for every stored key and for every registered operator. Access to a key can be restricted by assigning this key to a specific key group. Operators who are not assigned to the same key group are forbidden to access or even ‘see’ the key.
   A key is assigned to a key group by setting its key group attribute value to the desired key group name. An operator is assigned to a key group by setting their operator key group attribute value to the desired key group name.
9. The module supports the “CRYPT” (”DECRYPT”) attribute for every stored secret cryptographic AES or Triple-DES key. The module only permits encryption (decryption) with a secret user key if this attribute is set. In FIPS mode this attribute cannot be set for private or public user keys. In particular, RSA and EC keys cannot be used for bulk data encryption or decryption. In FIPS mode, Triple-DES keys cannot be used for encryption and cannot be generated.
10. The module supports the “SIGN” (”VERIFY”) attribute for every private, public or secret cryptographic key. The module only permits the generation (verification) of a signature with a private (public) user key only if this attribute is set. The module allows the generation (verification) of a MAC or HMAC with a secret user key only if this attribute is set. In FIPS mode, Triple-DES keys cannot be used for TDES MAC calculation and verification. This attribute can only be set if attributes DERIVE and WRAP/UNWRAP are not set.

11. The module supports a “DERIVE” attribute for private and public cryptographic EC or DSA keys. The module only permits key derivation with a private or public user key if this attribute is set.
    This attribute cannot be set for RSA keys or secret user keys. This attribute can only be set if attributes SIGN and VERIFY are not set.
12. The module supports the “WRAP” (”UNWRAP”) attribute for every stored secret AES, Triple-DES or public (private) RSA key. The module only permits the key to be used to encrypt (decrypt) other keys for export (import) if, and only if, this attribute is set.
    This attribute cannot be set for EC or DSA keys. In FIPS mode, Triple-DES keys cannot be used for key wrapping. This attribute can only be set if attributes SIGN and VERIFY are not set.
13. The module supports the attribute “TRUSTED” (default: false) for every stored wrapping key (attribute “WRAP” = TRUE), which can only be set to TRUE by a *Security Officer*. It also supports the “WRAP WITH TRUSTED” attribute (default: false) for any key. If set to TRUE, the key can only be wrapped with a wrapping key that has the attribute “TRUSTED” set to TRUE.

7 Physical Security Policy

The CryptoServer is a multi-chip embedded cryptographic module encapsulated in a hard, opaque, tamper-evident coating.

On the top side of the module a (hollow) metal heat sink is directly mounted on the printed circuit board, on three edges, and the space between the PCB and the heat sink is completely filled with potting material (epoxy resin) (see Figure 2). On the bottom side of the PCB, a metal frame is stuck directly onto the printed circuit board, and the space inside the metal frame is completely filled with potting material (see Figure 3). Epoxy hardness testing was performed over the module's operating temperature range from -10 ˚C to +60˚C.

The heat sink and potting material together define the top and bottom sides of the module and deliver a hard, opaque coating. All the cryptographic module's hardware components (which are all mounted on the PCB) are entirely covered by this coating.

Each CryptoServer Se-Series Gen2 with Hardware P/N Version 5.01.4.2 additionally has a sensor patch wire. If this is disrupted (i.e. in the case of physical attack), it activates the tamper response. This feature falls under FIPS 140-2 Area 11 (Mitigation of Other Attacks).

The CryptoServer module with its tamper-evident enclosure (the heat sink and the potting material) implements the following physical security mechanisms:

- Active tamper response and zeroization circuitry.
- The cryptographic module's hardware components are covered by hard, opaque potting material or the heat sink which show evidence of tampering on the enclosure when a physical attack is attempted.
- The potting material is hard and opaque enough to prevent direct observation and easy penetration to the depth of the underlying hardware components. It is highly probable that anyone attempting to penetrate to the depth of the circuitry will break off large pieces of potting material and tear important hardware components off the module, causing serious damage to the module.
- Temperature sensors that activate a tamper response if the module is outside of the defined temperature range of –16°C to 66°C (10.4°F to 150.8°F).
- Voltage sensors that monitor the power supply of the module and activate a tamper response if the power input is outside of the defined range (including low or removed battery).
- Tamper response and zeroization circuitry is active while module is in standby mode (powered down).
- Zeroization is performed within less than 4 milliseconds after tamper detection (temperature or voltage outside of defined range).
- Module stops operation if its internal temperature is outside of its operational temperature range of -5°C to 62°C (23°F to 143.6°F).
- The module regularly inverts all bits of the plaintext CSPs to avoid "burn in" of information into SRAM cells.

To ensure security of the cryptographic module, the module must be periodically inspected for evidence of tampering. The recommended inspection schedule depends on the

customer’s application area. This may vary between inspecting the module once a week and once a year.

The physical security mechanisms listed above function autonomously and under all circumstances.

8 Operational Environment

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

9 Mitigation of Other Attacks Policy

The cryptographic module has been designed to mitigate several physical attacks, Simple and Differential Power Analysis (SPA/DPA) and timing analysis.

Table 15 - Mitigation of Other Attacks

Other Attacks	Mitigation Mechanism
SPA/DPA	SPA/DPA attacks are mitigated by use of hardware components assembled into a special design of the power management circuit, such that it is not feasible to monitor power consumption to determine the value of an algorithm's key. Power consumption of the module does not depend on the value of cryptographic keys.
Timing Analysis	It is not feasible to determine the value of an algorithm's keys by measuring the execution time of a cryptographic operation. Triple-DES and AES operations are executed in fixed time. If blinding is switched on for RSA and ECDSA, the input data for a single RSA and ECDSA signature generation is randomized by use of a blinding technique so that the input parameters of the algorithm are not known by the operator. In this case it is not possible to gain knowledge about the private key by the amount of time required by the signature operation. Blinding is not supported for bulk signing.
Mechanical attack	Each CryptoServer Se-Series Gen2 with Hardware P/N Version 5.01.4.2 additionally has a sensor patch wire installed in the epoxy. If this is disrupted (i.e. in the case of physical attack), it activates the tamper response (zeroization).
Temperature and voltage	The module provides physical EFP temperature and voltage protections that are outside the scope of FIPS 140-2 physical security Level 3. A tamper response (zeroization) is activated if the module is outside the defined temperature range (–16°C to 66°C) or voltage range

10 References

Reference	Title/Company
[ANSSI]	ANSSI: "Avis relatif aux paramètres de courbes elliptiques définis par l'Etat français" in: Journal Officiel de la République Française (JORF), n° 0241 du 16 octobre 2011 page 17533 text n° 30 (Announcement about elliptic curve parameters set by the French government). NOR: PRMD1123151V. Available: https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000024668816
[CSAdmGuide]	CryptoServer - Administrator's Guide for CryptoServer Se/CSe/Se2 in FIPS Mode, Doc. no 2011-0002 / Utimaco IS GmbH
[ECCBP]	RFC 5639: Elliptic Curve Cryptography ECC Brainpool Standard - Curves and Curve Generation, March 2010, including Errata, http://tools.ietf.org/html/rfc5639
[FIPS140-2]	FIPS PUB 140-2, Security Requirements for Cryptographic Modules / National Institute of Standards and Technology (NIST), May 2001
[FIPS186-2]	FIPS PUB 186-2: Digital Signature Standard (DSS) / National Institute of Standards and Technology (NIST), January 2000
[FIPS186-4]	FIPS PUB 186-4: Digital Signature Standard (DSS) / National Institute of Standards and Technology (NIST), July 2013
[NIST 800-90A]	NIST Special Publication 800-90A, Recommendation for Random Number Generation Using Deterministic Random Bit Generators / National Institute of Standards and Technology (NIST), January 2012
[NIST SP 800-56A r3]	NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography
[PKCS#1]	PKCS#1: RSA Encryption Standard v2.1, 14th June 2002 / RSA Laboratories, http://www.rsa.com/rsalabs/node.asp?id=2125
[PKCS#3]	PKCS#3: Diffie-Hellman Key Agreement Standard v1.4, 1st November 1993 / RSA Laboratories, http://www.rsa.com/rsalabs/node.asp?id=2126
[PKCS#11]	PKCS#11: Cryptographic Token Interface Standard v2.20, 28th June 2004 / RSA Laboratories, http://www.rsa.com/rsalabs/node.asp?id=2133
[PCIHSM]	Payment Card Industry (PCI) PIN Transaction Security (PTS) Hardware Security Module (HSM) Security Requirements, PCI Security Standards Council, Version 2.0, May 2012
[RFC 7748]	RFC 7748: Elliptic Curves for Security / Internet Research Task Force (IRTF), January 2016, ISSN 2070-1721, including Errata ID 4730 reported and verified on 2016-07-05
[SEC2]	SEC2: Recommended Elliptic Curve Domain Parameters – Certicom Research – September 20, 2000, Version 1.0

Reference	Title/Company
[AIS 20/31]	Application Notes and Interpretation of the Scheme (AIS): AIS 20/AIS 31: A proposal for: Functionality classes for random number generators, Version 2.0 / Wolfgang Killmann (T-Systems GEI GmbH, Bonn), Werner Schindler (Bundesamt für Sicherheit in der Informationstechnik/BSI, Bonn), 18. September 2011

11 Definitions and Acronyms

AES	Advanced Encryption Standard
CSP	Critical Security Parameter
DES	Data Encryption Standard
DH	Diffie-Hellman
DPA	Differential Power Analysis
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
EC	Elliptic Curve
ECDH	Elliptic Curve Diffie-Hellman Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
KDF	Key Derivation Function
MAC	Message Authentication Code
MBK	Master Backup Key
NDRNG	Non-deterministic Random Number Generator
PCB	Printed Circuit Board
PCI	Payment Card Industry
PTS	PIN Transaction Security
RNG	Random Number Generator
SHA	Secure Hash Algorithm
SPA	Simple Power Analysis
Triple-DES	Triple-DES with key size 16 or 24 bytes