CryptoServer CSe

Non-Proprietary
Security Policy

http://hsm.utimaco.com

Imprint

Copyright 2016	**Utimaco IS GmbH** **Germanusstraße 4** **52080 Aachen, Germany** This document may be reproduced only in its original entirety [without revision]. Utimaco IS GmbH accepts no liability for misprints and damage resulting from them.
Phone	+49 (0)241 / 1696-200
Fax	+49 (0)241 / 1696-199
Internet	http://hsm.utimaco.com
e-mail	hsm@utimaco.com
Document Number	2012-0009
Document Version	1.3.3
Date	September 28th, 2016
Status	Released

Table of Contents

1 Introduction

CryptoServer CSe is a hardware security module made by Utimaco IS GmbH. If run in FIPS mode, CryptoServer CSe (**CryptoServer CSe**) meets overall FIPS 140-2 Level 3 requirements, with level 4 requirements in section “Physical Security”.

This document describes the security policy of CryptoServer CSe (Hardware P/N CryptoServer CSe Version 4.00.4.2; Firmware Package Version 4.0.3.0) if run in FIPS mode.

2 Module Overview

The CryptoServer CSe is an encapsulated, protected security module which is realized as a multi-chip embedded cryptographic module as defined in FIPS 140-2 (Hardware P/N CryptoServer CSe Version 4.00.4.2; Firmware Package Version 4.0.3.0). Its realization meets overall FIPS 140-2 Level 3 requirements, with level 4 requirements in section “Physical Security”. The primary purpose of this module is to provide secure cryptographic services such as encryption or decryption (for various cryptographic algorithms like Triple-DES, RSA 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 module.

If not in FIPS mode, the CryptoServer CSe’s flexible firmware architecture enables it to be used 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 CSe offers hardware-based as well as deterministic random number generation in FIPS mode and non-FIPS mode. The hardware based RNG is only used to 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.

The CryptoServer CSe is encased in a hard opaque commercial grade metal case which contains a tamper response envelope around the module: 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) and encapsulated by metal shells, a special tamper detection envelope (which is a special foil bearing a flexible printed circuit with a serpentine geometric pattern of conductors) and potting material (epoxy resin).

The module's cryptographic boundary is defined by the outer metal case on five of the six sides of the module and the epoxy surface on the bottom side of the module. Figure 1 below show views of the cryptographic boundary from the side and top. The dashed lines indicate the cryptographic boundary.

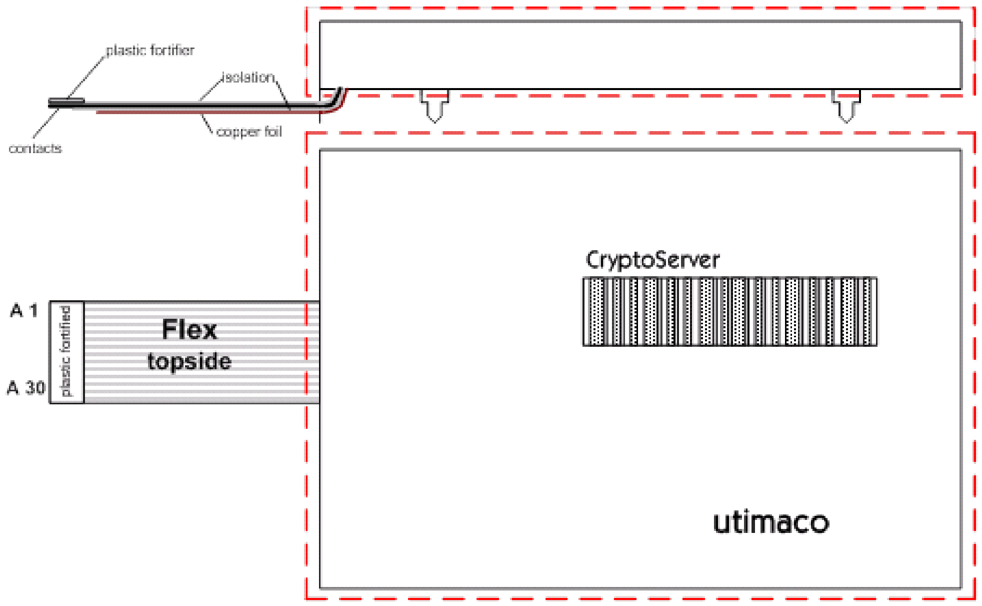

Figure 1: –CryptoServer CSe – side view and top view

To enable communication with a host, this encapsulated cryptographic module is mounted on a carrier card which supports a PCI express interface and two USB interfaces. The connection between the cryptographic module and the carrier card is done by the ribbon cable shown in Figure 1.

The following picture shows the cryptographic module CryptoServer CSe mounted on a PCIe carrier card:

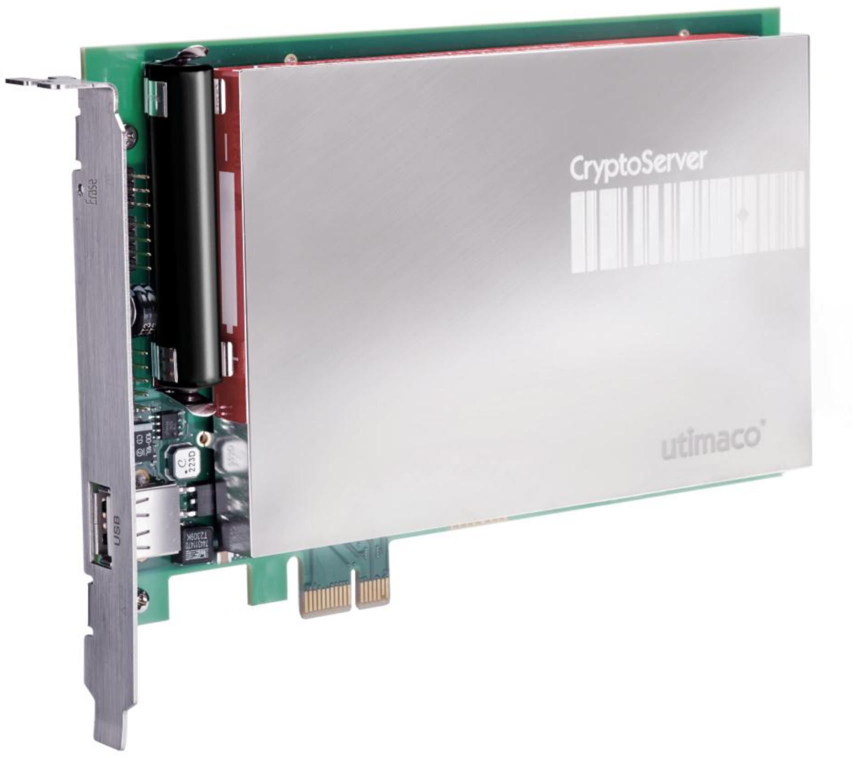

Figure 2: –CryptoServer CSe mounted on the PCIe carrier card

3 Security Level

The cryptographic module meets the overall requirements applicable to Level 3 security in FIPS 140-2. In the section “Physical Security” level 4 (incl. EFP) is reached.

Table 1 – Module Security Level Specification

Security Requirements Section	Level
Cryptographic Module Specification	3
Module Ports and Interfaces	3
Roles, Services and Authentication	3
Finite State Model	3
Physical Security	4
Operational Environment	n/a
Cryptographic Key Management	3
EMI/EMC	3
Self-Tests	3
Design Assurance	3
Mitigation of Other Attacks	3

4 Modes of Operation

4.1 Approved mode of operation

The cryptographic module supports the following FIPS Approved algorithms:

- RSA with variable key sizes :
  - Sign according to [FIPS186-4] with key length 2048 or 3072 bit and hash function SHA-224, SHA-256, SHA-384 or SHA-512
  - Verify according to [FIPS186-2] or [FIPS186-4] with key length 1024, 1536, 2048, 3072 or 4096 bit and hash function SHA-1, SHA-224, SHA-256, SHA-384 or SHA-512
  - Key Generation according to [FIPS186-4] (key lengths 2048 and 3072)

  (see RSA Validation Certificate No. 1845)
- ECDSA according to [FIPS186-4]with EC keys on dedicated elliptic curves (curves P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409 and B-571) and with hash function SHA-224, SHA-256, SHA-384 or SHA-512 for
  - Sign
  - Verify (Verify additionally on curves P-192, K-163, B-163 and additionally with hash function SHA-1)
  - Key Generation according to [FIPS186-4] (curves P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409 and B-571)

  (see ECDSA Validation Certificate No. 730)
- Triple-DES (ECB & CBC) for
  - Data Encryption (24 bytes key length)
  - Data Decryption (16 or 24 bytes key length)

  (see Triple-DES Modes of Operation Validation Certificate No. 1998)
- Triple-DES MAC (Triple-DES MAC generation with 24 bytes key length; Triple-DES MAC verification with 16 or 24 bytes key length)
  (vendor affirmed, based on FIPS Approved Triple-DES core algorithm, see Validation Certificate No. 1998)
- AES (ECB, CBC, CFB8, and OFB) for
  - Data Encryption/Decryption (128, 192, 256 key lengths)

  (see Advanced Encryption Standard Validation Certificate No. 3589)
- AES 128, 192 & 256 bits (CMAC mode)
  (see AES Validation Certificate No. 3589)
  - Data Encryption/Decryption
- AES GCM and CCM is not available in FIPS mode.
- DSA according to [FIPS186-4] for
  - Sign (key size |P|/|Q| = 2048/224, 2048/256 or 3072/256 and hash function SHA-224, SHA-256, SHA-384 or SHA-512)
  - Signature Verification (|P|/|Q| = 1024/160, 2048/224, 2048/256 or 3072/256

  - and hash function SHA-1, SHA-224, SHA-256, SHA-384 or SHA-512)
  - Domain Parameter Generation according to [FIPS186-4] (key size |P|/|Q| = 2048/224, 2048/256, 3072/256)
  - Key Generation (key size |P|/|Q| = 2048/224, 2048/256 or 3072/256) according to [FIPS186-4] (see DSA Validation Certificate No 997)

  (see DSA Validation Certificate No 997)

- SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512 for hashing
  (see Secure Hash Standard Validation Certificates Nos. 2951 (SHA-512 only), 3168 (SHA-512 only) and 2954 (all listed SHA algorithms))
- HMAC (based on SHA-1, SHA-224, SHA-256, SHA-384 or SHA-512)
  - Generation with key size >= 112 bits
  - Verification with key size >= 80 bits

  (see HMAC Validation Certificate No. 2289)
- Hash-based DRBG (based on SHA-512)
  (see DRBG Validation Certificate No. 1089)
- ECC CDH Primitive (ECDH component, curves P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409 and B-571)
  (see CVL Validation Certificate No. 613)
- For random value generation and generation of all cryptographic keys CryptoServer CSe relies on an implemented Deterministic Random Bit Generator (DRBG) that is compliant with NIST Special Publication 800-90A, hash-based, with SHA-512 as transition function (see [NIST 800-90A]). This DRBG is FIPS Approved, see Random Bit Generator DRBG Validation Certificate No. 1089.

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

- NDRNG – used to generate the seed material for the Approved DRBG; based on a hardware noise source. The NDRNG produces an estimated 402 bits of entropy per each 512 bit block output.
- RSA Key Wrapping (key length between 2048 and 16384 bits) and Unwrapping (key length between 1024 and 16384 bits) (key establishment methodology which provides between 112 and 270 bits of encryption strength depending on the wrapping key's security strength).
- AES (ECB, CBC, or OFB) Key Wrapping/Unwrapping (Cert. #3589) (key establishment methodology which provides between 128 and 256 bits of encryption strength depending on the wrapping key's security strength)
- Triple-DES (ECB or CBC) Key Wrapping/Unwrapping (Cert. #1998) (key establishment methodology provides 112 bits of encryption strength)
- Diffie-Hellman for key agreement (key establishment methodology which provides 112 bits of encryption strength) – commercially available protocol [PKCS#3] for key establishment; see below for the Secure Messaging concept
- EC Diffie-Hellman for key agreement (curves P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409 and B-571 as specified in [FIPS186-4]; Validation Certificate No. 613 for primitive ECC CDH as of SP 800 56A; (key

establishment methodology which provides between 112 and 256 bits of encryption strength depending on the security strength of the original keys and the selected hash algorithm)

- Diffie-Hellman for key agreement (|P| >= 2048, |Q| >= 224 (key establishment methodology which provides 112 or 128 bits of encryption strength depending on the security strength of the original keys and the selected hash algorithm)

The CryptoServer CSe can be configured for FIPS mode as follows:

- Perform a "GetState" command and confirm that the module is in the initialized state, and in operational or maintenance mode and the alarm state "off".
- Load the FIPS firmware module package using the "LoadPkg" command of Utimaco's CryptoServer CSe administration tool csadm.

This is described in the CryptoServer CSe's Administrator's Guide for CryptoServer CSe in FIPS Mode [CSAdmGuide]. If this has been performed successfully, the module's internally stored FIPS mode indicator flag is set. The user can check whether the cryptographic module is running in FIPS or non-FIPS mode by executing the "GetState" service. The system will then display the FIPS mode indicator.

4.2 Non-FIPS mode of operation

Any firmware that has not been validated for FIPS140-2, if loaded into the module while the module is in delivery state, will not set the module's internally stored FIPS mode indicator flag. This missing flag will indicate to the user of the 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:

- RSA for public key cipher of bulk data
- EC Cryptography for public key cipher of bulk data (ECIES)
- MD5, MDC-2 or RIPEMD-160 for hashing
- Single DES
- Retail-Triple-DES MAC
- AES MAC CBC Mode (based on AES Cert. #3589; non-compliant)
- Key generation with True Random Number Generator (based on a physical noise source which provides entropy to both Approved and Non-Approved modes)
- PIN generation/PIN verification (e. g., VISA/MasterCard)
- Several key derivation algorithms as specified in [PKCS#11].
    - KDF_ENC_DATA
    - KDF_HASH
    - KDF_ECDH
    - KDF_XOR_BASE_AND_DATA
    - KDF_CAT_BASE_AND_KEY

- KDF_CAT_BASE_AND_DATA
- KDF_CAT_DATA_AND_BASE
- KDF_EXTRACT_KEY_FROM_KEY

4.3 Secure Messaging for secure communication with the CryptoServer CSe

The CryptoServer CSe implements a *Secure Messaging* concept which enables any operator to secure their communication with the CryptoServer CSe over the PCIe interface, even from a remote host. With Secure Messaging, commands sent to the CryptoServer CSe and response data received from the CryptoServer CSe can be encrypted and integrity-protected/signed with an AES MAC. 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 each Session, a 32 bytes AES session key $\mathbf{K_S}$ will be negotiated between CryptoServer CSe and host, using the Diffie-Hellmann algorithm as the key establishment technique in accordance with [PKCS#3] and with a parameter size of 2048 bits. For generating its random value $\mathbf{K_{SM\_MOD\_PRIV}}$ that is needed for the key agreement, the CryptoServer CSe will use its deterministic random bit generator.

The CryptoServer CSe 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 CSe and the host application.

5 Ports and Interfaces

The physical interface of CryptoServer CSe consists of a ribbon cable consisting of 18 serial lines (see Figure 1). The device provides the following physical ports on these tracks:

1) Power input (including operational power input and backup power input).
2) An External Erase input, which can be used to zeroize all security relevant information inside the module.
3) External communication ports (PCIe and USB) which are used for data input, data output, control input and status output:

   To enable communication with a host, the encapsulated cryptographic module is mounted on a carrier card which supports a PCIe interface and two USB interfaces. All requests for services are sent over the PCIe interface. The USB interfaces are used for status output only.

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 an encrypted form: All command and response data (except for status requests) to and from the CryptoServer CSe are encrypted and given MAC protection by the Secure Messaging layer. For details, see previous subsection *Secure Messaging for secure communication with the CryptoServer*.

Additionally, all secret or private keys can only be exported in a wrapped form, i. e. encrypted with a Key Encryption Key (via e.g., the *Export Key* or Wrap services, see section 7.1 *Roles and authorized* services).

6 Identification and Authentication Policy

6.1 Assumption of roles

The CryptoServer CSe cryptographic module supports three distinct operator roles:

- *Cryptographic User* (performing key management and cryptographic services),
- *Security Officer* (performing key group specific administration functions like key group specific user management or key group specific configuration management) and
- *Administrator* (performing global configuration and user management).

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

- '*User'* (performing cryptographic services like encryption or signing) and
- '*Key Manager'* (performing 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 a user name and their 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 CSe instead of the password. The CryptoServer CSe recalculates and checks the HMAC value using the operator's password that is stored inside the CryptoServer CSe.
- For *RSA signature based authentication* the user sends an RSA signed command containing their user name to authenticate to the cryptographic module.

Upon correct authentication the role is selected based on the operator's user name. During authentication a session key $\mathbf{K}_S$ is negotiated which is used to secure subsequent service requests by the operator (see the description of the Secure Messaging concept on page 12). Since the session key (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 CSe 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 2 – 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)	Identity-based operator authentication	User Name and Password or User Name and RSA Signature
Key Manager (sub-role of Cryptographic User)	Identity-based operator authentication	User Name and Password or User Name and RSA Signature
Security Officer	Identity-based operator authentication	User Name and Password or User Name and RSA Signature
Administrator (called *Crypto Officer* in [FIPS140-2])	Identity-based operator authentication	User Name and Password or User Name and RSA Signature

Table 3 – Strengths of Authentication Mechanisms

Authentication Mechanism	Strength of Mechanism
Username and Password (4 characters password chosen from 94 printable ASCII characters)	The probability that a random attempt will succeed or a false acceptance will occur is 1/78,074,896 which is less than 1/1,000,000. Due to a correctional delay of 80 milliseconds for every non-successful authentication there is a maximum limit of 60 * 1000 / 80 = 750 non-successful authentications per minute. This can be stated as allowing only 750 non-successful authentication attempts per minute based on a rate of 80 ms per attempt. Therefore the probability of successfully authenticating to the module within one minute is (less than) 1/104,100 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 2^{-80} (according to SP800-57-Part1 page 67) which is less than 1/1,000,000. Due to a correctional delay of 80 milliseconds for every non-successful authentication there is a maximum limit of 60 * 1000 / 80 = 750 non-successful authentications per minute. This can be stated as allowing only 750 non-successful authentication attempts per minute based on a rate of 80 ms per attempt. Therefore the probability of successfully authenticating to the module within one minute is less than 2^{-70} ($750 * 2^{-80} < 1024 * 2^{-80} = 2^{10} * 2^{-80} = 2^{-70}$) which is less than 1/100,000.

7 Access Control Policy

7.1 Roles and authorized services

General definitions:

An ***Operator*** may be an Administrator, Security Officer or 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 7.3) 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 4 – Services Authorized for Roles

Role	Authorized 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.	• <u>Change Operator's Password or Key: </u>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. • <u>Get Session Key:</u> This service generates a new Secure Messaging session key for secure communication to the module. • <u>List Keys:</u> 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. • <u>Open Key: </u>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. • <u>Get Key Property: </u>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. • <u>Set Key Property: </u>This service sets one or more properties (attributes) for an Assigned key or storage object (but no key parts). • <u>Backup Key: </u>This service outputs a Backup Blob containing an Assigned

Role	Authorized Services
	key or storage object for back-up purposes. • 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. • 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 or SHA-512 hash or HMAC value for given data or for the components of an Assigned key. • Agree Secret: This function calculates a shared secret from two Assigned EC keys. The shared secret is exported in a wrapped form, encrypted with the Master Backup 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. Secret or private keys are only exported in a wrapped form (i.e., encrypted with a Key Encryption Key[1]). Public keys can also be exported in plaintext. • Import Key: This service imports a cryptographic key into the cryptographic module. The key can be imported in a wrapped form (i.e., encrypted with a Key Encryption Key[2]) or in plaintext[3]. On request the imported key can be exported again within a Backup Blob. • 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

[1] *Secret User Key* (**K**$_{\mathbf{USR\_AES}}$ or **K**$_{\mathbf{USR\_TDES}}$) or *Public RSA User Key* **K**$_{\mathbf{USR\_RSA\_PRIV}}$ with attribute “WRAP”

[2] *Secret User Key* (**K**$_{\mathbf{USR\_AES}}$ or **K**$_{\mathbf{USR\_TDES}}$) or *Public RSA User Key* **K**$_{\mathbf{USR\_RSA\_PRIV}}$ with attribute “WRAP”

[3] The key is protected by the Secure Messaging encryption.

Role	Authorized Services
	but exported within two Backup Blobs. • <u>Derive Key:</u> This function derives a Triple-DES or AES key or a Generic Secret from an Assigned base key (DSA or EC). The derived key is stored in the CryptoServer CSe, or exported within a Backup Blob. • <u>Wrap Key:</u> 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 is wrapped with a key encryption key (wrapping key); the wrapping key may be of type Triple-DES, AES or RSA. • <u>Unwrap Key:</u> 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 is wrapped with a key encryption key (wrapping key); the wrapping key may be of type Triple-DES, AES or RSA. • <u>Create Object</u>: This function creates an Assigned cryptographic key or storage object according to the given property list. • <u>Copy Object:</u> 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 CSe 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.	• <u>Change Operator’s Password or Key:</u> 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. • <u>Get Session Key:</u> This service generates a new Secure Messaging session key for secure communication to the module. • <u>List Keys:</u> 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. • <u>Open Key:</u> 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. • <u>Get Key Property:</u> 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. • <u>Generate DSA Parameters:</u> This service generates a DSA Domain Parameter set P, Q and G using the DRBG. • <u>Generate DSA Parameters PQ:</u> This service generates a DSA Domain Parameter set P and Q using the DRBG. • <u>Generate DSA Parameters G:</u> This service generates a DSA Domain

Role	Authorized Services
	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) or in ECB, CBC, OFB mode (AES). • Sign Data: This service generates an RSA, DSA or ECDSA signature or calculates a Triple-DES MAC, AES CMAC, or HMAC ((hashed) message authentication code) for given data with an Assigned signing key. • Verify Signature: This service verifies an RSA, DSA or ECDSA signature or a Triple-DES or AES CMAC, or HMAC using an Assigned verification key. • Compute Hash: This service calculates a SHA-1, SHA-224, SHA-256, SHA-384 or SHA-512 hash or HMAC value for given data or for the components of an Assigned key. • Agree Secret: This function calculates a shared secret from two Assigned EC keys. The shared secret is exported in a wrapped form, encrypted with the Master Backup 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. • Restore User: This service creates a new user in the user database. All information about the user (name, permission, authentication token, etc.)

Role	Authorized Services
	is taken from a backup data block that was output by the *Backup User* service. • 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 CSe. • 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, encrypted with the session key of the current Secure Messaging session. The generated key is not stored inside the CryptoServer CSe. • Import Master Backup Key: This service imports a Master Backup Key (back-up key). The key is only imported in a wrapped form, encrypted with the session key of 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. • Set Maximum Failure Counter: This service sets the maximum number of allowed consecutive failed authentication attempts before a user is blocked. • Set Time, SetTimeRel: 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 object. • 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.

Role	Authorized Services
	• Backup Key (for the Global configuration object): This service outputs the Global configuration object for back-up purposes. • Restore Key (for the Global configuration object): This service imports the back-up of the Global configuration object into the cryptographic module. • 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. • 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): This service outputs an Assigned Local configuration object for back-up purposes. • Restore Key (for Local configuration objects): This service imports the back-up of a Local configuration object into the cryptographic module. • Delete Key (for Local configuration objects): This service deletes an Assigned Local configuration object by setting all configuration attributes to their default values.

Role	Authorized Services
	• Init Key Group: This service deletes all Local Objects belonging to a given Key Group.

7.2 Unauthenticated services

In addition the CryptoServer CSe supports the following unauthenticated services, i. e. services which are available to any operator:

- Get Boot Log: Retrieve a log file which contains log messages made by the operating system and other firmware modules (or by the boot loader if the command is called in boot loader 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 module.
- Get Maximum Fail Count: This service outputs the maximum number of allowed consecutive failed authentication attempts before a user is blocked.
- List Files: Retrieve a list of all files stored in the flash file system of the module.
- List Active Modules: List all currently active firmware modules.
- List Operators: Read a list of all *Security Officers*, *Cryptographic Users, Key Managers, Users* 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 a log file.
- Get Audit Config: Read 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 (8 bytes random value generated by the CryptoServer CSe’s deterministic random bit generator) for using the challenge/response mechanism in the next authenticated command.
- Get Authentication State: This function returns the current authentication level and an optional list of all operators that are authenticated within the current session.
- Get CXI Info: This function returns some status information about the CXI firmware module like module version number or the fill level of the database.
- Get Personalization Key: This function returns the public part of the Local Personalization Key.
- Verify Genuineness: This function enables any user to verify the genuineness of the CryptoServer CSe by signing a challenge with the Local Personalization Key.

- 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: Zeroizes the cryptographic module including all critical security parameters. In particular, all CSPs that are not wrapped by the Master Key will be zeroized. This service will be executed if an external erase input is given. (Note: after zeroization, CryptoServer CSe is no longer in FIPS mode.)

If the 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.

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

7.3 Definition of Critical Security Parameters (CSPs)

The following CSPs are contained in the module:

- CryptoServer CSe's *Master Key* **K_{CS}** (Encrypts all other secret and private keys stored within the crypto module) (AES 32 bytes)
- *Local Secret DH Key* **$K_{SM\_MOD\_PRIV}$** (generated by the module and used to generate a shared secret via Diffie Hellman for Secure Messaging, see section 4.3)
(DSA 2048 bits, volatile storage only)
- *Final Shared Secret* **S_{SM}** (calculated by Diffie Hellman algorithm and used to derive a Session Key for Secure Messaging, see section 4.3) (volatile storage only)
- *Session Key* **K_S** (derived from the Final Shared Secret **S_{SM}** and used for Secure Messaging, see section 4.3) (32 bytes AES, volatile storage only)
- *DRBG Secrets* **S_{DRBG}** used by the Deterministic Random Bit Generator (DRBG) as specified in [NIST 800-90A] (volatile storage only):
    - Entropy input **S_{DRBG_EI}** generated by the NDRNG
    - Seed **S_{DRBG_SEED}** calculated from Entropy input **S_{DRBG_EI}**
    - Working state constant **S_{DRBG_C}** calculated from the **S_{DRBG_SEED}** Seed
    - Working state value **S_{DRBG_V}** initially calculated from the **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 **K_{CS}**:[4]

- *Local Private Personalization* Key **K_{LP_PRIV}** (Used to authenticate the Module towards the user) (ECDSA)
- *Private User Keys (Exportable via RSA, Triple-DES or AES Wrapping):*
    - **$K_{USR\_RSA\_PRIV}$** (RSA; Signature Generation, Key Decryption)
    - **$K_{USR\_DSA\_PRIV}$** (DSA; Signature Generation, Key Agreement)
    - **$K_{USR\_EC\_PRIV}$** (EC; Signature Generation, Key Agreement)
- *Secret User Keys (Exportable via RSA, Triple-DES or AES Wrapping)*:
    - **K_{USR_AES}** (AES; for Key Encryption, Data Encryption or MAC)
    - **K_{USR_TDES}** (Triple-DES; for Key Encryption, Data Encryption or MAC)
    - *Generic Secret* **K_{USR_GS}** (to be used as HMAC key; at least 112 bits for HMAC generation)
- *Master Backup Key* **MBK** (AES 16, 24 or 32 bytes, key for back-up purposes)
- *Operator Password* **PSW_{AUTH}** (for authentication)

The following CSP is generated but not stored within the cryptographic module and

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

exported after being encrypted with the Master Backup Key:

- *Shared Secret* **S_{USR}** as generated by the *Agree Secret* function (part of the ECDH agreement scheme)

7.4 Definition of Public Keys

The following public keys are contained in the cryptographic module:

- *Production Key* (RSA 2048 bit) **K_{PROD_PUB}**
- *Module Signature Key* (RSA 4096 bit) **$K_{MDL\text{-}SIG\_PUB}$**
- *Default Administrator Key* (RSA 1024 bit) **$K_{ADMIN\text{-}DEF\_PUB}$**
- *Local Public Personalization Key* (ECDSA) **K_{LP_PUB}**
- *Public User Keys:*
  - **$K_{USR\_EC\_PUB}$** (EC; Signature Verification, Key Agreement)
  - **$K_{USR\_DSA\_PUB}$** (DSA; Signature Verification, Key Agreement)
  - **$K_{USR\_RSA\_PUB}$** (RSA; Signature Verification, Key Encryption)
- Operator's *Public Authentication Key* **K_{AUTH_PUB}** (RSA)

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

- *Remote Public DH Key* **$K_{SM\_HOST\_PUB}$** (generated by the host and used to generate a shared secret via Diffie Hellman for Secure Messaging)
  (DSA 2048 bits, volatile storage only)
- *Local Public DH Key* **$K_{SM\_MOD\_PUB}$** (generated by the module and used to generate a shared secret via Diffie Hellman for Secure Messaging)
  (DSA 2048 bits, volatile storage only)

7.5 Definition of modes of access to CSPs

Table 5 defines the relationship between the different module services and access to CSPs. The types of access (e.g., Use/Write/Update) are given in the right-hand column.

The following types of access are possible:

- *Write:* the CSP is created (newly written).
- *Update:* replaces the 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 key encryption key and exported from the cryptographic module.
- *Export:* the key (plaintext) is exported from the cryptographic module (only possible for public RSA, DSA or EC keys **$K_{USR\_RSA\_PUB}$, $K_{USR\_DSA\_PUB}$** and **$K_{USR\_EC\_PUB}$**).
- *Delete:* invalidates the CSP
- *(xxx):* access type is set in brackets if this access type is conditional.

The following definitions are used in Table 5:

- *Any User Key* can be a *Secret User Key* ($\mathbf{K}_{USR\_AES}$, $\mathbf{K}_{USR\_TDES}$ or $\mathbf{K}_{USR\_GS}$) or a *Private* and/or *Public User Key* **(**$\mathbf{K}_{USR\_RSA\_PRIV}$, $\mathbf{K}_{USR\_RSA\_PUB}$, $\mathbf{K}_{USR\_DSA\_PRIV}$, $\mathbf{K}_{USR\_DSA\_PUB}$, $\mathbf{K}_{USR\_EC\_PRIV}$, $\mathbf{K}_{USR\_EC\_PUB}$**)**
- A *Secret Data Encryption Key* is a *Secret AES or Triple-DES User Key* ($\mathbf{K}_{USR\_AES}$ or $\mathbf{K}_{USR\_TDES}$) with attribute[5] "CRYPT"/"DECRYPT".
- A *Secret Key Encryption Key* can be a *Secret AES or Triple-DES User Key* ($\mathbf{K}_{USR\_AES}$ or $\mathbf{K}_{USR\_TDES}$) with attribute[6] "WRAP"/"UNWRAP".
- A *Secret MAC Key* can be a *Secret User Key* ($\mathbf{K}_{USR\_AES}$, $\mathbf{K}_{USR\_TDES}$ or $\mathbf{K}_{USR\_GS}$) with attribute[7] "SIGN"/"VERIFY".
- A *Key Derivation Key* can be a *Private or Public EC or DSA User Key* ($\mathbf{K}_{USR\_EC\_PRIV}$, $\mathbf{K}_{USR\_EC\_PUB}$, $\mathbf{K}_{USR\_DSA\_PRIV}$, $\mathbf{K}_{USR\_DSA\_PUB}$**)** with attribute[8] "DERIVE".
- A *Private Sign Key* can be a *Private RSA, DSA or EC User Key* ($\mathbf{K}_{USR\_RSA\_PRIV}$, $\mathbf{K}_{USR\_DSA\_PRIV}$ or $\mathbf{K}_{USR\_EC\_PRIV}$) with attribute[7] "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[7] "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[9] or an external[10] 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.

** General remark concerning *DRBG Secrets* $\mathbf{S}_{DRBG}$:

- If a new block of random values must be generated but no reseeding is required, the *DRBG Secrets* $\mathbf{S}_{DRBG\_C}$ and $\mathbf{S}_{DRBG\_V}$ are used and $\mathbf{S}_{DRBG\_V}$ is updated.
- If a new block of random values must be generated and reseeding is required, all *DRBG Secrets* $\mathbf{S}_{DRBG\_EI}$, $\mathbf{S}_{DRBG\_SEED}$, $\mathbf{S}_{DRBG\_C}$ and $\mathbf{S}_{DRBG\_V}$ are updated and used.

Below, the two 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.

[5] See chapter 9, vendor imposed security rule 9.

[6] See chapter 9, vendor imposed security rule 12.

[7] See chapter 9, vendor imposed security rule 10.

[8] See chapter 9, vendor imposed security rule 11.

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

[10] 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**.

Table 5 – 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[11]			
X	X	X	any command authentication	*Public Authentication Key **K_{AUTH_PUB}*** or *Password **PSW_{AUTH}*** of respective operator	*Use*
X	X	X	any command using *Secure Messaging*	*Session Key **K_S***	*Use*
X	X	X	Get Session Key	*DRBG Secrets **S_{DRBG}*****	*Use* and *Update*
				*Remote Public DH Key **$K_{SM\_HOST\_PUB}$***	*Use*
				*Local Private DH Key **$K_{SM\_MOD\_PRIV}$***	*Use*
				*Local Public DH Key **$K_{SM\_MOD\_PUB}$***	*Export*
				*Final Shared Secret **S_{SM}***	*Use*
				*Session Key **K_S***	*Write*
(all without authentication)			End Session	*Session Key **K_S***	*Delete*[12]
(all without authentication)			Verify Genuineness	*Local Private Personalization Key **K_{LP_PRIV}***	*Use*
(all without authentication)			Get Personal. Key	*Local Public Personalization Key* **K_{LP_PUB}**	*Export*
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 password: *CryptoServer CSe's Master Key **K_{CS}***	*(Use)*
(without authentication; only executed when an external erase is triggered by a short-circuit of the 'External Erase' pins on the PCIe card)			Zeroize	*CryptoServer CSe's Master Key **K_{CS}***	*Delete*[13]
				All CSPs that are stored temporarily in the Key Cache (volatile storage)	*Delete*[14]
				All CSPs that are stored wrapped with the Master Key	*Delete*[15]

[11] Cryptographic User, Key Manager, User

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

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

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

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

Table 6 – CSP and Key Access Rights within Roles & Services – General Administration

Role			Service	Cryptographic Keys and CSPs Access Operation	Type of Access
Ad-ministrator	Security Officer	CU, KM, U[16]			
X			Add Operator	*Public Authentication Key* ***K_{AUTH_PUB}*** or *Password* ***PSW_{AUTH}*** of Operator	*Write*
				If operator uses password: *CryptoServer CSe's Master Key* ***K_{CS}***	*(Use)*
X			Delete Operator	*Public Authentication Key* ***K_{AUTH_PUB}*** or *Password* ***PSW_{AUTH}*** *of Operator*	*Delete*[17]
X	X		Add Group User	*Public Authentication Key* ***K_{AUTH_PUB}*** or *Password* ***PSW_{AUTH}*** *of Operator*	*Write*
				If operator uses password: CryptoServer CSe's *Master Key* ***K_{CS}***	*(Use)*
X	X		Delete Group User	*Public Authentication Key* ***K_{AUTH_PUB}*** or *Password* ***PSW_{AUTH}*** of Operator	*Delete*[17]
X			Backup User	*Public Authentication Key* ***K_{AUTH_PUB}*** or *Password* ***PSW_{AUTH}*** of Operator	*Wrapped Export*
				Master Backup Key ***MBK***	*Use*
				CryptoServer CSe'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 CSe'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			List Master Backup Keys	---	---
X			Generate Master Backup Key	*Master Backup Key* ***MBK***	*Wrapped Export*
				Session Key ***K_S***	*Use*
				DRBG Secrets ***S_{DRBG}******	*Use* and *Update*
X			Import Master Backup Key	*Master Backup Key* ***MBK***	*Write* or *Update*
				Session Key ***K_S***	*Use*
				CryptoServer CSe's *Master Key* ***K_{CS}***	*Use*

[16] Cryptographic User, Key Manager, User

[17] Invalidated within database; password cannot be accessed because it is encrypted with the Master Key **K_{CS}**.

Table 7 – 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				
		User	Key Mgr			
	X			Init Key Group	*Any User Key*	*Delete*[18]
(*)	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*[18]
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 CSe's Master Key **K**$_{CS}$*	*(Use)*
(*)	(*)		X	Restore Key	*Any User Key*	*Write, 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 CSe's Master Key **K**$_{CS}$*	*(Use)*
			X	Generate Key, Generate Key Pair	*DRBG Secrets **S**$_{DRBG}$***	*Use* and *Update*
					*Any User Key**	*Write* or *Update* (if generated key is to be stored in CryptoServer CSe), or *Wrapped Export* (if generated key is to be exported from CryptoServer CSe)

[18] Invalidated within database; password cannot be accessed because it is 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				
		User	Key Mgr			
			X	Export Key	*Any User Key**	*Export* (only possible if key to be exported is a public key), or *Wrapped Export* (mandatory if *Private* or *Secret User Key* is exported)
					Optional if public key is exported; otherwise mandatory: *Secret Key Encryption Key* or Public RSA User Key* $\boldsymbol{K}_{USR\_RSA\_PUB}$*****	*(Use)*
					Only if random padding is required: *DRBG Secrets* $\boldsymbol{S}_{DRBG}$**	*(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* $\boldsymbol{K}_{USR\_RSA\_PRIV}$*****	*(Use)*
			X	Derive Key	*Key Derivation Key**	*Use*
					*Secret User Key**	*Write or Update* or *Wrapped Export*
			X	Wrap	*Any User Key**	*Wrapped Export*
					Secret Key Encryption Key or Public RSA User Key* $\boldsymbol{K}_{USR\_RSA\_PUB}$*****	*Use*
					Only if random padding is required: *DRBG Secrets* $\boldsymbol{S}_{DRBG}$**	*(Use* and *Update)*
			X	Unwrap	*Any User Key**	*Write* or *Update* or *Wrapped Export*
					Secret Key Encryption Key or Private RSA User Key* $\boldsymbol{K}_{USR\_RSA\_PRIV}$*****	*Use*
			X	Create Object	*Any User Key**	*Write* or *Update* or *Wrapped Export*
			X	Copy Object	*Any User Key**	*Write* or *Wrapped Export*

Table 8 – 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	Cryptographic User				
		User	Key Mgr			
		X		Crypt Data	*Secret Data Encryption Key**	*Use*
					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	optionally*: Any User Key**	(Use)
		X	X	Agree Secret	*Private EC User Key **$K_{USR\_EC\_PRIV}$****	*Use*
					*Public EC User Key **$K_{USR\_EC\_PUB}$****	*Use*
					*Shared Secret **S_{USR}***	*Wrapped Export*
		X	X	Generate DSA Parameters (_PQ/G)	*DRBG Secrets **S_{DRBG}** ***	*Use* and *Update*

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 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 three distinct operator roles. These are the *Cryptographic User* role, the *Security Officer* role and the *Administrator* role. The *Cryptographic User* role may be split into two sub-roles *User* and *Key Manager*.
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) <u>Power up Self-Tests:</u>
      i) Cryptographic Algorithm Tests:
         (1) Triple-DES (Encrypt / Decrypt) Known Answer Tests
         (2) Triple-DES MAC Known Answer Test
         (3) AES (Encrypt / Decrypt) Known Answer Tests
         (4) AES-CMAC Known Answer Test
         (5) SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512 Known Answer Tests (*Cert. #2954*)
         (6) SMOS: SHA-512 Known Answer Tests (*Cert. #3168*)
         (7) BL SHA: SHA-512 Known Answer Tests (*Cert. #2951*)
         (8) RSA Known Answer Tests (sign/verify) (*Cert. #1845)*
         (9) RSA Pair-wise Consistency Test (encrypt/decrypt) (*Cert. #1845)*
         (10) EC (ECDSA) Pair-wise Consistency Test (sign/verify)
         (11) DSA Pair-wise Consistency Test (sign/verify)
         (12) HMAC (SHA-1, 224, 256, 384, 512) Known Answer Tests
         (13) DRBG Known Answer Tests according to [NIST 800-90A] (testing the Instantiate Function, the Generate Function and the Reseed Function)
      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) Critical Functions Tests
         (1) SDRAM Test
         (2) Master Key Consistency Test
         (3) Temperature Test
   b) <u>Conditional Self-Tests:</u>
      i) *Continuous Random Number Generator (RNG) Test* performed on DRBG and

Hardware RNG

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

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

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

v) *Firmware Load Test* (via RSA signature verification)

5. At any time the operator is capable of commanding the module to perform the power-up self-test.
6. Prior to each use, the DRBG is tested using the conditional test specified in FIPS 140-2 §4.9.2.
7. Data output is inhibited during key generation, self-tests, zeroization, and error states.
8. Status information does not contain CSPs or sensitive data that if misused could lead to the compromising of the module.
9. The module supports concurrent operators.
10. The successful completion of the power-up self-tests is indicated by executing the “GetState” command which returns state = INITIALIZED and FIPS mode = ON.

This section documents the security rules imposed by the vendor:

1. The module zeroizes all plaintext CSPs within a maximum of 4 milliseconds after any attack or alarm (see section 10 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 MAC. 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.
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.
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.
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 wrap (unwrap) other keys for export (import) if, and only if, this attribute is set.
    This attribute cannot be set for EC or DSA keys.
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.

10 Physical Security Policy

10.1 Physical security mechanisms

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

This coating consists of an inner metal housing surrounded by a special tamper-detection foil and potting material, and all this encased in an outer metal housing.

The CryptoServer CSe module with its tamper-evident enclosure (the coating) implements the following physical security mechanisms:

- Active tamper response and zeroization circuitry.
- Module is entirely encapsulated by a security foil (tamper sensor) that detects all physical and chemical attacks.
- Temperature sensors that activate a tamper response if the module is outside of the defined temperature range of –10°C to 60°C.
- 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 (foil destruction or temperature or voltage outside of defined range).
- Module stops operation if its internal temperature exceeds the upper limit of its operational temperature range (62°C).
- The module regularly inverts all bits of the plaintext CSPs to avoid “burn in” of information into SRAM cells.

To ensure physical security of the cryptographic module, no extra action has to be performed. For a module in FIPS mode, the physical security mechanisms listed above function autonomously and under all circumstances.

11 Mitigation of Other Attacks Policy

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

Table 9 – 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 possible the input data for a private RSA operation is randomized by use of a blinding technique so that the input parameters of the RSA algorithm are not known by the operator. In this case it is not possible to calculate the bits of a private key by the amount of time required by the private RSA operation. This blinding technique is only possible if the public key is available. If the private key was generated within the CryptoServer the public key is always stored with the private key and blinding will be applied. In the case that the user imports a private key without the public key, blinding is not possible and will not be applied.

The CryptoServer's ability to mitigate effectively SPA/DPA and timing attacks on Triple-DES or AES operations has been verified in the context of the CryptoServer's validation process done by the German Credit Association "Deutsche Kreditwirtschaft"[19].

The CryptoServer's ability to mitigate effectively SPA/DPA and timing attacks on RSA operations has been verified as part of the validation process according to the Payment Card Industry (PCI) PIN Transaction Security (PTS) Hardware Security Module (HSM) Security Requirements (see [PCIHSM]).

[19] Also denoted as GBIC (German Banking Industry Committee), a consortium of several financial organizations (e.g. Bundesverband der Deutschen Volksbanken und Raiffeisenbanken, Bundesverband deutscher Banken etc.). The DK arose in August 2011 from the German ZKA (Zentraler Kreditausschuss) and continues its tasks.

12 References

Ref.	Title/Company
[CSAdmGuide]	CryptoServer - Administrator's Guide for CryptoServer Se/CSe in FIPS Mode, Doc. no 2011-0002 / Utimaco IS GmbH
[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
[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 Concil, Version 2.0, May 2012

13 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 Hellmann Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
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