utimaco®
a member of the Sophos Group

SafeGuard® CryptoServer Se

Security Policy

http://hsm.utimaco.com

Imprint

Utimaco Safeware AG
A member of the Sophos Group
Germanusstr. 4
52080 Aachen, Germany

Phone +49 (0)241 / 1696-200

Fax +49 (0)241 / 1696-199

Internet http://hsm.utimaco.com

e-mail support-cs@utimaco.de

Document Number 2008-0010

Document Version 1.0.8

Date May 22nd, 2012

Status Released

Table of Contents

1 Introduction

SafeGuard® CryptoServer Se is a hardware security module made by Utimaco Safeware AG. If run in FIPS mode, SafeGuard® CryptoServer Se (**CryptoServer Se**, **CryptoServer**) meets overall FIPS 140-2 Level 3 requirements.

This document describes the security policy of CryptoServer Se if run in FIPS mode.

2 Module Overview

The CryptoServer Se 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 Se, Version 3.00.3.1; Firmware Package Version 1.0.1.0). Its realization meets 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, RSA and AES), hashing, signing and verification of data (RSA, ECDSA), 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 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's flexible firmware architecture enables it to be used in almost all proprietary environments in which cryptographic services and highest security are required, for instance in 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. In FIPS mode, the hardware based RNG is only used to seed the Approved deterministic RBG.

The 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.

The picture below shows the CryptoServer Se cryptographic module with its PCIe interface:

Figure 1 – SafeGuard® CryptoServer Se

For communication with a host the PCIe board offers a PCIe interface, two serial interfaces (V.24) and a USB interface.

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

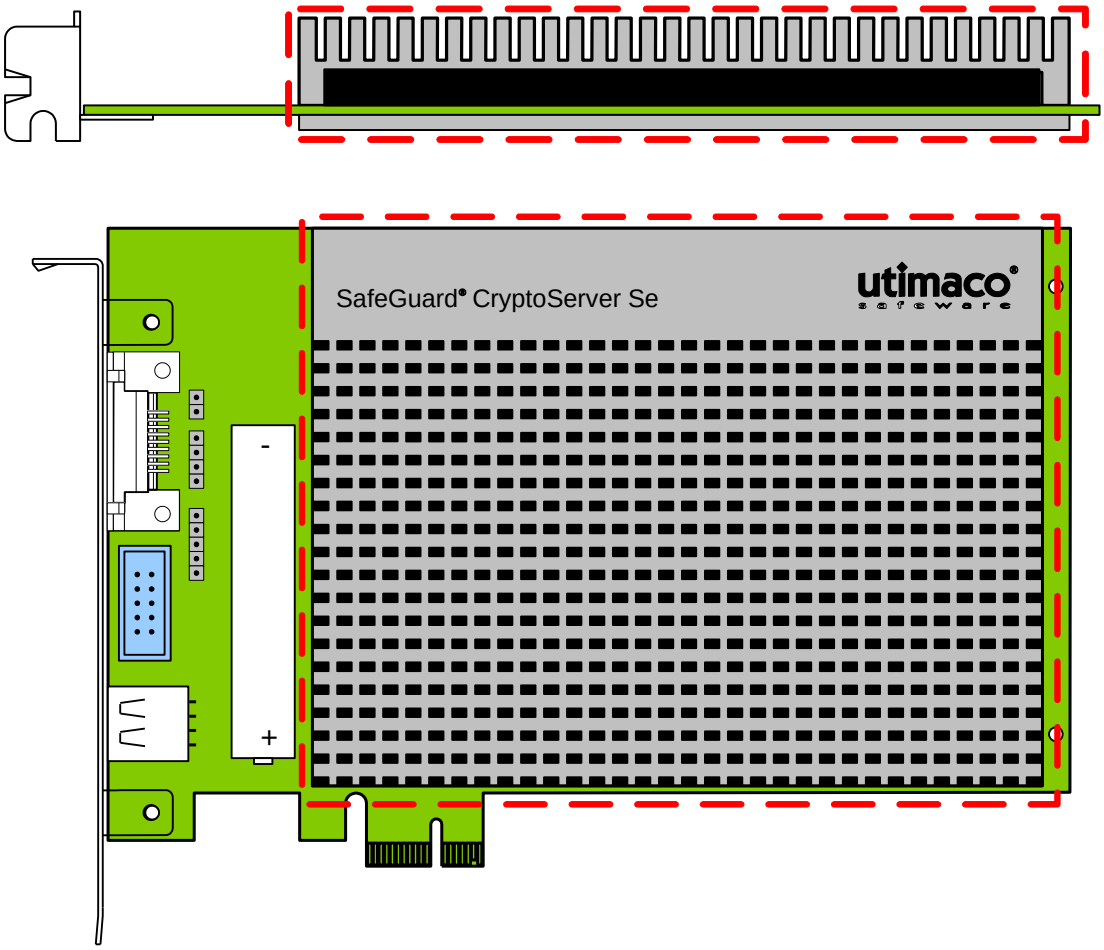

Figure 2 – SafeGuard® CryptoServer Se – side view and top view

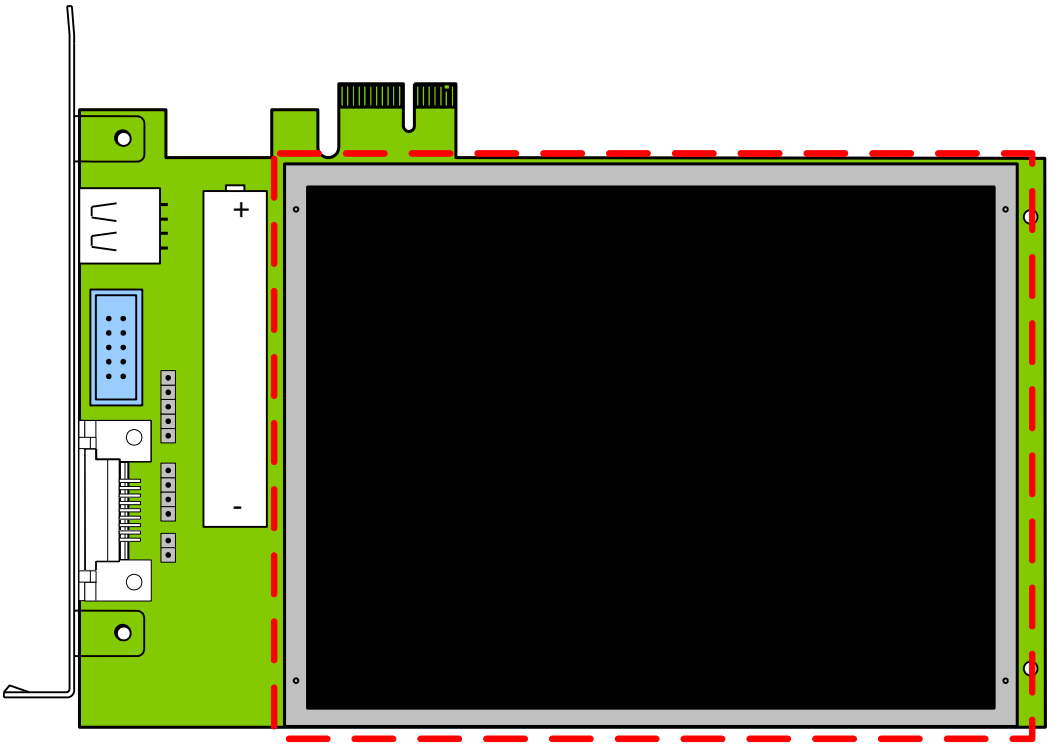

Figure 3 – SafeGuard® CryptoServer Se – bottom view

3 Security Level

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

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	3
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 (minimum 1024 bit key length) for
  - Sign/Verify

  (see RSA Validation Certificate No. 841 and 842)
- ECDSA with EC keys on dedicated elliptic curves (curves P-192, P-224, P-256, P-384, P-521, K-163, K-233, K-283, K-409, K-571, B-163, B-233, B-283, B-409 and B-571 as specified and recommended in FIPS 186-2 Appendix 1) for
  - Sign/Verify according ANSI X9.62

  (see ECDSA Validation Certificate No. 221)
- Triple-DES (TDES) (16 or 24 bytes key length) for
  - Data Encryption/Decryption

  (see Triple DES Modes of Operation Validation Certificate No. 1101)

  The operator is responsible for ensuring that no 16 bytes TDES key is used to encrypt more than 2^{20} blocks of data.
- TDES-MAC
  (vendor affirmed, based on FIPS Approved Triple-DES core algorithm, see Validation Certificate No. 1101)
- AES for
  - Data Encryption/Decryption

  (see Advanced Encryption Standard Validation Certificate No. 1711)
- AES (CMAC mode)
  (see AES Validation Certificate No. 1711)
- SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512 for hashing
  (see Secure Hash Standard Validation Certificates No. 1597, 1598 and 1498)
- HMAC (based on SHA-1, SHA-224, SHA-256, SHA-384 or SHA-512)
  (see HMAC Validation Certificate No. 990)
- DRBG hash-based DRBG (based on SHA-512, see DRBG Validation Certificate No. 141)

The Single-DES algorithm is not supported in FIPS mode.

In addition the CryptoServer Se in FIPS mode offers key generation services:

- RSA key pair generation
- EC key pair generation and ECDH key derivation
- Triple-DES key generation
- AES key generation

For random value generation and generation of all cryptographic keys CryptoServer Se relies on an implemented Deterministic Random Bit Generator (DRBG) that is compliant with NIST Special Publication 800-90, hash-based, with SHA-512 as transition function (see *[NIST 800-90]*). This DRBG is FIPS Approved, see Random Bit Generator Validation Certificate No. 141.

CryptoServer Se 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
- RSA Key Wrapping/Unwrapping (key establishment methodology which provides between 80 and 150 bits of encryption strength)
- AES Key Wrapping/Unwrapping (key establishment methodology which provides between 128 and 256 bits of encryption strength depending on the key length)
- TDES Key Wrapping/Unwrapping (key establishment methodology provides 112 bits of encryption strength)
- Diffie-Hellman for key agreement (key establishment methodology which provides 80 bits of encryption strength) – commercially available protocol *[PKCS#3]* for key establishment; see below for the *Secure Messaging* concept

The CryptoServer Se 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.

This is described in the CryptoServer's *Administrator's Guide for CryptoServer Se 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, 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
- DSA (non-compliant)
- MD5, MDC-2 or RIPEMD-160 for hashing
- Single DES
- Retail-TDES MAC

- AES MAC CBC Mode (based on AES Cert. #1711; non-compliant)
- Key generation with True Random Number Generator (based on a physical noise source)
- PIN generation/PIN verification (e. g. VISA/MasterCard)

4.3 Secure Messaging for secure communication with the CryptoServer

The CryptoServer Se 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 encrypted and integrity-protected/signed with an AES or TDES 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 a Session, a 32 bytes AES or 16 bytes TDES session key $\mathbf{K_S}$ will be negotiated between CryptoServer and host, using the Diffie-Hellmann algorithm as the key establishment technique (in accordance with *[PKCS#3]*; for generating its random value, $\mathbf{K_{SM\_MOD\_PRIV}}$. This is needed for the key agreement. The CryptoServer will use its deterministic random bit generator).

The operator is responsible for ensuring that no 16 bytes TDES session key is used to encrypt more than 2^{20} blocks of data. The operator must close a session and open a new session before 2^{20} blocks of data have been exchanged within one session.

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.

5 Ports and Interfaces

The physical interface of CryptoServer Se consists of 30 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:

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, RS232 and USB) which are used for data input, data output, control input and status output:

   To enable communication with a host, the module supports a PCIe interface, two RS232 interfaces and two USB interfaces. All requests for services are sent over the PCIe interface. The first RS232 interface is used for status output only. The second RS232 interface and the USB interfaces are 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 an encrypted form: All command and response data (except for status requests) to and from the CryptoServer 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 imported or exported in a wrapped form, i. e. encrypted (via e.g. the *Import Key* and *Export Key* services, see section 7.1 *Roles and services)*.

6 Identification and Authentication Policy

6.1 Assumption of roles

The CryptoServer cryptographic module supports two distinct operator roles: *Cryptographic User* and *Administrator* (called *User Role* and *Crypto Officer Role* in [FIPS140-2]). 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 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 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 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 Se 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
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 256 8-bit-encoded characters)	The probability that a random attempt will succeed or a false acceptance will occur is 1/4.294.967.296 which is less than 1/1,000,000. Due to a correctional delay of 40 milliseconds for every non-successful authentication (there is a maximum limit of 1500 non-successful authentications per minute), the probability of successfully authenticating to the module within one minute is (less than) 1/2.863.311 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 40 milliseconds for every non-successful authentication (there is a maximum limit of 1500 non-successful authentications per minute), the probability of successfully authenticating to the module within one minute is less than 2^{-70} which is less than 1/100,000.

7 Access Control Policy

7.1 Roles and services

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 private, public and secret keys, hashing services and random number generation.	• Change Cryptographic User's Password or Key: This service changes the password or RSA public key which is used for the *Cryptographic User's* authentication. • Get Session Key: This service generates a new Secure Messaging session key for secure communication to the module. • Open Key: This service opens a cryptographic key which is stored inside the cryptographic module and returns a key handle or the key is exported in a wrapped form, encrypted with the Master Backup Key (key for back-up purposes, see 7.3). • Crypt Data: This service encrypts or decrypts data using a TDES or AES key in CBC or ECB mode. The cryptographic user is responsible for ensuring that no 16 bytes TDES user key is used to encrypt more than 2^{20} blocks of data. • Sign Data: This service generates an RSA or ECDSA signature or calculates a TDES MAC or AES MAC or HMAC ((hashed) message authentication code) for given data.. • Verify Signature: This service verifies an RSA or ECDSA signature or a TDES or AES MAC or HMAC. • 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 a given key. • Generate Random Number: This service generates a random number using the DRBG. • Generate Key: This service generates a cryptographic key (TDES: 16 or 24 bytes, AES, RSA or EC) using the DRBG. In FIPS mode this service will not generate Single-DES keys. On request the generated key is not stored but exported in a wrapped form, encrypted with the Master Backup Key. • Agree Secret: This function calculates a shared secret from two EC keys. The shared secret is exported in a wrapped form, encrypted with the Master Backup Key. • Export Key: This service outputs a cryptographic key. Secret or private

Role	Authorized Services
	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. Secret or private keys are only imported in a wrapped form (i. e. encrypted with a Key Encryption Key[1]). Public keys can also be imported in plaintext. On request the imported key can be exported again in a wrapped form, encrypted with the Master Backup Key. • Delete Key: This service deletes a key from the module. • List Keys: This service outputs the key properties (such as the algorithm, key name, key size, etc.) of all keys stored inside the cryptographic module. • Get Key Property: This service returns one or more key properties (key attributes) of a given cryptographic key. It can export the public part of a key but no secret or private key parts. • Set Key Property: This service sets one or more key properties for a given cryptographic key (but no key parts). • Backup Key: This service outputs a cryptographic key for back-up purposes. The key is exported in a wrapped form, encrypted with a Master Backup Key (back-up key, see 7.3). • Restore Key: This service imports the back-up of a cryptographic key into the cryptographic module. The key is imported in a wrapped form, encrypted with a Master Backup Key (back-up key, see 7.3). Optionally the key can also be exported in a wrapped form, encrypted with the Master Backup Key.
Administrator: This role provides all services necessary for firmware and user management.	• Change Administrator's Password or Key: This service changes the password or RSA public key which is used for the *Administrator*'s authentication. • Get Session Key: This service generates a new Secure Messaging session key for secure communication to the module. • Add Operator: This service adds a *Cryptographic User* or *Administrator* to the cryptographic module. • Delete Operator: This service deletes a *Cryptographic User* or *Administrator* 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

[1] *Secret User Key* (**K_{USR_AES}** or **K_{USR_TDES}**) or *Public RSA User Key* **$K_{USR\_RSA\_PRIV}$** with attribute "WRAP"

Role	Authorized Services
	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.) 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. • 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. • 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. (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 Time, SetTimeRel: The *Administrator* can use these services to set the internal clock on the module.

7.2 Unauthenticated Services

In addition the CryptoServer Se 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.
- 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 *Cryptographic 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 Memory Info: Return statistical information about the file system usage.
- Echo: Communication test (echo input data).
- Get Challenge: Generate and output a challenge (8 byte 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: This function returns the current permission mask and an optional list of all users that are authenticated within the current session.
- Get CXI Info: This function returns some status information about the CXI module.
- 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 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: Zeroize 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 is no longer in FIPS mode.)

If the module is in FIPS error state, the only services that are available are unauthenticated services that only output status information and do *not* perform any cryptographic function.

7.3 Definition of Critical Security Parameters (CSPs)

The following CSPs are contained in the module:

- CryptoServer's Master Key $\mathbf{K_{CS}}$ (AES 32 bytes)
- Local Secret DH Key $\mathbf{K_{SM\_MOD\_PRIV}}$ (generated by the module and used to generate a shared secret via Diffie Hellman for Secure Messaging, see section 4.3) (DSA 1024 or 2048 bits, volatile storage only)
- Final shared secret $\mathbf{S_{SM}}$ (calculated by Diffie Hellman algorithm and used to derive a Session Key for Secure Messaging, see section 4.3) (volatile storage only)
- Session Key $\mathbf{K_S}$ (derived from the final shared secret $\mathbf{S_{SM}}$ and used for Secure Messaging, see section 4.3) (32 bytes AES or 16 bytes TDES, volatile storage only)
- DRBG secrets $\mathbf{S_{DRBG}}$ used by the Deterministic Random Bit Generator (DRBG) as specified in [NIST 800-90] (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 $\mathbf{K_{CS}}$:[2]

- Private part of Local Personalization Key $\mathbf{K_{LP\_PRIV}}$ (ECDSA)
- Private RSA User Key $\mathbf{K_{USR\_RSA\_PRIV}}$ (Signature Generation, Key Decryption)
- Private EC User Key $\mathbf{K_{USR\_EC\_PRIV}}$ (Signature Generation, Key Derivation)
- Secret User Key $\mathbf{K_{USR\_AES}}$ (AES) or $\mathbf{K_{USR\_TDES}}$ (TDES); (for Key Encryption, Data Encryption or MAC)
- Master Backup Key **MBK** (AES 16, 24 or 32 bytes, key for back-up purposes)
- *Administrator's* Password $\mathbf{PSW_{ADM}}$ (for authentication)
- *Cryptographic User's* Password $\mathbf{PSW_{CRU}}$ (for authentication)

The following CSP is generated but not stored within the cryptographic module and exported after being encrypted with the Master Backup Key:

- Shared secret $\mathbf{S_{USR}}$ as generated by the *Agree Secret* function

7.4 Definition of Public Keys

The following public keys are contained in the cryptographic module:

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

- 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 Local Personalization Key (ECDSA) $\mathbf{K_{LP\_PUB}}$
- EC Public User Key $\mathbf{K_{USR\_EC\_PUB}}$ (Signature Verification, Key Derivation)
- RSA Public User Key $\mathbf{K_{USR\_RSA\_PUB}}$ (Signature Verification, Key Encryption)
- *Administrator's* Public Authentication Key $\mathbf{K_{AUTH\_ADM\_PUB}}$ (RSA)
- *Cryptographic User's* Public Authentication Key $\mathbf{K_{AUTH\_CRU\_PUB}}$ (RSA)

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

- Remote Public DH Key $\mathbf{K_{SM\_HOST\_PUB}}$ (generated by the host and used to generate a shared secret via Diffie Hellman for Secure Messaging)
  (DSA 1024 or 2048 bits, volatile storage only)
- Local Public DH Key $\mathbf{K_{SM\_MOD\_PUB}}$ (generated by the module and used to generate a shared secret via Diffie Hellman for Secure Messaging)
  (DSA 1024 or 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.
- *Use*:* use of an internal[3] or external[4] User Key is possible. The following CSPs must also be used:
  - When an internal key is to be used, the **Master Key** $\mathbf{K_{CS}}$ must be used to decrypt the internal key.
  - When an external key is to be used, the **MBK** must be used to verify the MAC and to decrypt the secret or private key.
- *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 or EC keys $\mathbf{K_{USR\_RSA\_PUB}}$ and $\mathbf{K_{USR\_EC\_PUB}}$).
- *Delete:* invalidates the CSP

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

[4] 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 MAC protected; secret and private key parts are always encrypted with the Master Backup Key **MBK**.

- *(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 ($\mathbf{K_{USR\_AES}}$ or $\mathbf{K_{USR\_TDES}}$) or a Private and/or Public RSA or EC User Key **(**$\mathbf{K_{USR\_RSA\_PRIV}}$, $\mathbf{K_{USR\_RSA\_PUB}}$, $\mathbf{K_{USR\_EC\_PRIV}}$, $\mathbf{K_{USR\_EC\_PUB}}$**)**
- A *Secret Data Encryption Key* is a Secret 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 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 ($\mathbf{K_{USR\_AES}}$ or $\mathbf{K_{USR\_TDES}}$) with attribute[7] "SIGN"/"VERIFY".
- A *Private Sign Key* can be a Private RSA or EC User Key ($\mathbf{K_{USR\_RSA\_PRIV}}$ or $\mathbf{K_{USR\_EC\_PRIV}}$).
- A *Public Verify Key* can be a Public RSA or EC User Key ($\mathbf{K_{USR\_RSA\_PUB}}$ or $\mathbf{K_{USR\_EC\_PUB}}$).

** General remark concerning *DRBG secrets* $\boldsymbol{S_{DRBG:}}$

- 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 secret DRBG values $\mathbf{S_{DRBG\_EI}}$, $\mathbf{S_{DRBG\_SEED}}$, $\mathbf{S_{DRBG\_C}}$ and $\mathbf{S_{DRBG\_V}}$ are updated and used.

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

Table 5 – CSP and Key Access Rights within Roles & Services

Role		**Service**	**Cryptographic Keys and CSPs Access Operation**	**Type of Access**
Admini-strator	**Crypto-graphic User**			
	X	any command authentication performed by *Cryptographic User*	*Public Authentication Key* $\boldsymbol{K_{AUTH\_CRU\_PUB}}$ *or Password* $\boldsymbol{PSW_{CRU}}$ *of Cryptographic User*	*Use*

[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.

Role		Service	Cryptographic Keys and CSPs Access Operation	Type of Access
Admini-strator	**Crypto-graphic User**			
X		any command authentication performed by *Administrator*	*Public Authentication Key* $K_{AUTH\_ADM\_PUB}$ *or Password* PSW_{ADM} *of Administrator*	*Use*
X	X	any command using *Secure Messaging*	*Session Key* K_S	*Use*
X	X	Get Session Key	*DRBG secrets* S_{DRBG}**	*Use* and *Update*
			Remote Public DH Key $K_{SM\_HOST\_PUB}$	*Use*
			Local Secret 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*[8]
(all without authentication)		Verify Genuineness	*Local Personalization Key* K_{LP_PRIV} *(Private Part)*	*Use*
(all without authentication)		Get Personal. Key	*Local Personalization Key* K_{LP_PUB} *(Public Part)*	*Export*
X	X	Change Operator's Key or Password	*Public Authentication Key* $K_{AUTH\_XXX\_PUB}$ *or Password* PSW_{XXX} *of Operator*	*Update*
			If operator uses password: *CryptoServer's Master Key* K_{CS}	*(Use)*
X		Add Operator	*Public Authentication Key* $K_{AUTH\_XXX\_PUB}$ *or Password* PSW_{XXX} *of Operator*	*Write*
			If operator uses password: *CryptoServer's Master Key* K_{CS}	*(Use)*
X		Delete Operator	*Public Authentication Key* $K_{AUTH\_XXX\_PUB}$ *or Password* PSW_{XXX} *of Operator*	*Delete*[9]

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

[9] 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
Admini-strator	**Crypto-graphic User**			
X		Backup User	*Public Authentication Key* $K_{AUTH\_XXX\_PUB}$ *or Password* PSW_{XXX} *of Operator*	*Wrapped Export*
			Master Backup Key ***MBK***	*Use*
			CryptoServer's Master Key K_{CS}	*Use*
X		Restore User	*Public Authentication Key* $K_{AUTH\_XXX\_PUB}$ *or Password* PSW_{XXX} *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 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's Master Key K_{CS}	*Use*
	X	Open Key	If requested key is to be exported: *any user key*	(*Wrapped Export)*
			If requested key is to be exported: *CryptoServer's Master Key* K_{CS}	*(Use)*
			If requested key is to be exported: *Master Backup Key* ***MBK***	*(Use)*
	X	Crypt Data	*Secret Data Encryption Key*	*Use**

Role		Service	Cryptographic Keys and CSPs Access Operation	Type of Access
Admini-strator	**Crypto-graphic User**			
			If random padding is required: *DRBG secrets* $\boldsymbol{S_{DRBG}}$**	*(Use* and *Update)*
	X	Sign Data	*Private Sign Key or Secret MAC Key*	*Use**
			If random padding is required: *DRBG secrets* $\boldsymbol{S_{DRBG}}$**	*(Use* and *Update)*
	X	Verify Signature	*Public Verify Key or Secret MAC Key*	*Use**
	X	Compute Hash	optionally*: any user key*	(Use*)
	X	Generate Random Number	*DRBG secrets* $\boldsymbol{S_{DRBG}}$**	*Use* and *Update*
	X	Generate Key	*DRBG secrets* $\boldsymbol{S_{DRBG}}$**	*Use* and *Update*
			Any User Key	*Write* (if generated key is to be stored in CryptoServer), or *Wrapped Export* (if generated key is to be exported from CryptoServer)
			If generated key is to be stored in CryptoServer: *CryptoServer's Master Key* $\boldsymbol{K_{CS}}$	*(Use)*
			If generated key is to be exported from CryptoServer: *Master Backup Key* ***MBK***	*(Use)*
	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 key is exported)
			If key to be exported is private or secret key: *CryptoServer's Master Key* $\boldsymbol{K_{CS}}$	*(Use)*

Role		Service	Cryptographic Keys and CSPs Access Operation	Type of Access
Admini-strator	**Crypto-graphic User**			
			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 if public key is imported; otherwise mandatory: *Secret Key Encryption Key or Private RSA User Key* $\boldsymbol{K_{USR\_RSA\_PRIV}}$	*(Use*)*
			If key to be imported is private or secret key: *CryptoServer's Master Key* $\boldsymbol{K_{CS}}$	*(Use)*
			If imported key is to be exported from CryptoServer in the form of an MBK encrypted Backup Blob: *Master Backup Key MBK*	*(Use)*
	X	List Keys	---	---
	X	Delete Key	*Any User Key*	*Delete*[10]
	X	Get Key Property	If Public User Key is requested: *Any Public User Key (*$\boldsymbol{K_{USR\_RSA\_PUB}}$ *or* $\boldsymbol{K_{USR\_EC\_PUB}}$*)*	*(Export)*
			When executed on an external key: ***MBK***	(Use)
	X	Set Key Property	When executed on an external key: ***MBK***	(Use)
	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 key: *CryptoServer's Master Key* $\boldsymbol{K_{CS}}$	*(Use)*

[10] Invalidated within database; secret or private user keys are not accessible because encrypted with the Master Key $\mathbf{K_{CS}}$

Role		Service	Cryptographic Keys and CSPs Access Operation	Type of Access
Administrator	**Cryptographic User**			
	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 key: *CryptoServer's Master Key* $\boldsymbol{K_{CS}}$	*(Use)*
	X	Agree Secret	*Private EC User Key* $\boldsymbol{K_{USR\_EC\_PRIV}}$	*Use**
			Public EC User Key $\boldsymbol{K_{USR\_EC\_PUB}}$	*Use**
			Shared Secret $\boldsymbol{S_{USR}}$	*Wrapped Export*
(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's Master Key* $\boldsymbol{K_{CS}}$	*Delete*[11]
			All CSPs that are stored temporarily in the Key Cache (volatile storage)	*Delete*[12]
			All CSPs that are stored wrapped with the Master Key	*Delete*[13]

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

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

[13] CSPs are invalidated by zeroizing the Master Key $\mathbf{K_{CS}}$ because they are encrypted with the Master Key $\mathbf{K_{CS}}$.

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 two distinct operator roles. These are the *Cryptographic User* role and the *Administrator* 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" or "Administrator" role by the module.
4. The cryptographic module performs the following tests:
   a) Power up Self-Tests:
      i) Cryptographic Algorithm Tests:
         (1) TDES Known Answer Test
         (2) TDES-MAC Known Answer Test
         (3) AES Known Answer Test
         (4) AES-MAC Known Answer Test (CMAC Mode)
         (5) SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512 Known Answer Tests
         (6) RSA Known Answer Tests (sign/verify)
         (7) RSA Pair-wise Consistency Test (encrypt/decrypt)
         (8) EC Pair-wise Consistency Test (sign/verify)
         (9) HMAC Known Answer Test
         (10) DRBG Known Answer Tests according to [NIST 800-90] (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) Conditional Self-Tests:
      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) *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.

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 or TDES 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 not belonging 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 group name. An operator is assigned to a key group by setting their operator key group attribute value to the desired group name.
9. The module supports the "CRYPT" ("DECRYPT") attribute for every stored secret cryptographic 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 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 EC cryptographic 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, TDES 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 keys.

10 Physical Security Policy

10.1 Physical security mechanisms

The CryptoServer Se multi-chip embedded cryptographic module is 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 again completely filled by potting material (see Figure 3).

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.

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 –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 (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 security, the module must be periodically inspected for evidence of tampering. 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 timing analysis.

Table 6 – Mitigation of Other Attacks

Other Attacks	Mitigation Mechanism
Timing Analysis	It is not feasible to determine the value of an algorithm's keys by measuring the execution time of a cryptographic operation. TDES and AES operations are executed in fixed time. 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. For that reason it is not possible to calculate the bits of a private key by the amount of time required by the private RSA operation.

12 References

	Title/Company
[CSAdmGuide]	SafeGuard® CryptoServer Se - Administrator's Guide for CryptoServer Se in FIPS Mode, Doc. no 2011-0002 / Utimaco Safeware AG
[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
[NIST 800-90]	NIST Special Publication 800-90, Recommendation for Random Number Generation Using Deterministic Random Bit Generators (Revised) / National Institute of Standards and Technology (NIST), March 2007
[PKCS#1]	PKCS#1: RSA Encryption Standard v2.1, 14th June 2002 / RSA Laboratories, http://www.rsasecurity.com/rsalabs/pkcs
[PKCS#3]	PKCS#3: Diffie-Hellman Key Agreement Standard v1.4, 1st November 1993 / RSA Laboratories, http://www.rsasecurity.com/rsalabs/pkcs

13 Definitions and Acronyms

AES	Advanced Encryption Standard
CSP	Critical Security Parameter
DES	Data Encryption Standard
DH	Diffie Hellman
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
RNG	Random Number Generator
SHA	Secure Hash Algorithm
TDES	Triple-DES (with key size 16 or 24 bytes)