# SafeGuard Cryptographic Engine

| | |
|---|---|
| *Document Version:* | 1.06.00 |
| *Document Type:* | FIPS 140-2 Level 1 Security Policy |
| *Project Id:* | |
| *File Name:* | SGCE-Fips140-SecurityPolicy-1-06-00.doc |
| *Author(s):* | Roland Reinl |
| *Office / Company:* | Utimaco Safeware AG |

*Abstract:* This document contains the non-proprietary Security Policy for the validation of SafeGuard Cryptographic Engine Version 5.0 according to FIPS 140-2 Level 1.

*Disclaimer:*

# Table Of Contents

# 1 Document Information

## 1.1 Owner / Master Location

Owner of this document is Christian Tobias (CTO).

The location of the master copy is CTO user area, network Utimaco Oberursel at SGCE-Fips140-SecurityPolicy-1-06-00.doc

## 1.2 Change History

| *Version* | *Author* | *Date (finished)* | *Description* |
|---|---|---|---|
| 1.00.00 | RRE | 10.01.2007 | First released version |
| 1.01.00 | RRE | 29.01.2007 | |
| 1.02.00 | CTO | 21.03.2007 | Algorithm Certificate Numbers added |
| 1.03.00 | CTO, GMI | 27.03.2007 | Editorial Changes |
| 1.04.00 | CTO | 03.04.2007 | Editorial Changes |
| 1.05.00 | RRE | 23.08.2007 | Various changes |
| 1.06.00 | CTO | 11.09.2007 | Changes in section 3.5 |

## 1.3 Distribution & Approval History

| *Version* | *Distributed to / approved by* | *Date distributed* | *Date approved* |
|---|---|---|---|
| 1.00.00 | Nuvo / CTO, GMI | 16.01.2007 | 16.01.2007 |
| 1.01.00 | Nuvo / CTO, GMI | 01.02.2007 | 01.02.2007 |
| 1.02.00 | Nuvo / CTO | 21.03.2007 | 21.03.2007 |
| 1.03.00 | Nuvo / CTO | 27.03.2007 | 27.03.2007 |
| 1.04.00 | Nuvo / CTO | 03.04.2007 | 03.04.2007 |
| 1.05.00 | Nuvo / CTO | 23.08.2007 | 23.08.2007 |
| 1.06.00 | Nuvo / CTO | 11.09.2007 | 11.09.2007 |

## 1.4 Assumptions made herein

No assumptions made herein.

# 2 Introduction

## 2.1 Purpose

This document provides the Cryptographic Module Security Policy for a validation according to the standard of FIPS 140-2 for the software product “SafeGuard Cryptographic Engine” (“SGCE”) Version 5.0.

The manufacturer and the vendor of the product is Utimaco Safeware AG.

The SGCE product is claimed to meet the overall requirements applicable to Level 1 security for FIPS 140-2.

This security policy describes the definition and boundaries of the SGCE cryptographic module, its compliance to the security requirements of FIPS 140-2 and how to use SGCE in a secure FIPS 140-2 mode.

## 2.2 References

This document contains only information related to the FIPS 140-2 compliant operation of SGCE. Further information about the SGCE product or information about other products offered by Utimaco Safeware AG is available at the Utimaco website: http://www.utimaco.com.

Information about the FIPS 140-2 standard and the Cryptographic Module Validation Program is available at the following website:

http://csrc.nist.gov/cryptval

## 2.3 Document Organisation

For the complete validation according to FIPS 140-2 the following documents are delivered by the manufacturer:

- Security Policy (this document):
  It contains non-proprietary information about the cryptographic module and its intended method of use.
  This document may be made open to public.

- Vendor Evidence Document:
  It contains additional information, how the cryptographic module meets the security requirements of FIPS 140-2. This information may partly consist of references to other documents.
  This document contains information proprietary to Utimaco Safeware AG and shall not be published.

- Additional documentation:
  Other documents, which contain information required for the validation of the cryptographic module. These documents are referenced by the Security Policy or the Vendor Evidence Document.
  These documents may contain information proprietary to Utimaco Safeware AG and shall not be published.

# 3 SGCE Library Cryptographic Module

## 3.1 Overview

SGCE is a cryptographic toolkit designated to be used for the integration of cryptographic functions into a wide range of applications.

The toolkit contains support for various cryptographic operations like symmetric encryption and decryption, hashing algorithms and a random number generator.

The toolkit runs on a general purpose PC. It is not available as a separate product but is contained in different security software products of the manufacturer, e.g.

- SafeGuard Enterprise (SGN),
- SafeGuard LAN Crypt,
- SafeGuard PrivateDisk,
- SafeGuard PrivateCrypto.

### 3.1.1 Platform Summary

The toolkit is available for different operating system platforms and modes listed below:

- Windows XP application mode (loads DLLs)
- Windows Server 2003 application mode (loads DLLs)
- Windows XP kernel mode (loads SYS drivers)
- FreeBSD 32bit application mode (loads Shared Objects)

The SGCE toolkit has the same structure for all platforms:

It consists of a "Switchboard API" library (SGCE API) and a set of executables each implementing a set of algorithms. SGCE API itself performs no cryptographic task, but acts as an interface to the algorithm executables. There is no way to access the cryptographic executables without calling the SGCE API.

The chapters below contain some more detailed description of the operation of SGCE library on the different platforms.

### 3.1.2 Windows Platform – Application Mode

The algorithm executables are implemented as Windows 32-bit DLLs.

This operation mode is intended for use in Windows 32-bit applications and has been verified for Windows XP SP2 as well as for Windows Server 2003 SP1.

### 3.1.3 Windows Platform – Kernel Mode

The algorithm executables are implemented as Windows kernel drivers.

This operation mode is intended for being used by other Windows kernel drivers and has been verified for Windows XP SP2.

### 3.1.4 FreeBSD Platform

The algorithm executables are implemented as FreeBSD shared objects:

This operation mode is intended for use in FreeBSD applications and has been verified for FreeBSD Version 5.4.

## 3.2 Cryptographic Module Definition

The SGCE cryptographic module is defined as a multi-chip standalone module in the terms of FIPS 140-2.

### 3.2.1 Components

The following components of the delivered product are parts of the cryptographic module:

- The SGCE API (“CRYPTENGN.LIB”),
- The symmetric encryption component for symmetric encryption according to FIPS 197 standard (AES) with 128 bit key length
  (Windows application mode: “CEAESN.DLL”,
  Windows kernel mode: “CEAESM.SYS”,
  FreeBSD: “LIBCEAESF.SO”).
- The symmetric encryption component for symmetric encryption according to FIPS 197 standard (AES) with 256 bit key length
  (Windows application mode: “CEAES2N.DLL”,
  Windows kernel mode: “CEAES2M.SYS”,
  FreeBSD: “LIBCEAES2F.SO”).
- The symmetric encryption component for symmetric encryption according to FIPS 46-3 standard (Triple DES) with 168 bit key length
  (Windows application mode: “CEDES3N.DLL”,
  Windows kernel mode: “CEDES3M.SYS”,
  FreeBSD: “LIBCEDES3F.SO”).
- The cryptographic component for hash calculation according to FIPS 180-2 (SHA-1, SHA-256, SHA-384, SHA-512)
  (Windows application mode: “CESHAN.DLL”,
  Windows kernel mode: “CESHAM.SYS”,
  FreeBSD: “LIBCESHAF.SO”).

- The cryptographic component for calculating HMAC-SHA-256 according to FIPS 198
  (Windows application mode: “CEHMACN.DLL”,
  Windows kernel mode: “CEHMACM.SYS”,
  FreeBSD: “LIBCEHMACF.SO”).
- The cryptographic component for pseudo random number generation according to FIPS 186-2 General Purpose Change Notice dated 5 October 2001 with SHA-1 as G function, a seed-key with length between 20 bytes and 64 bytes and no optional seed
  (Windows application mode: “CERNDN.DLL”,
  Windows kernel mode: “CERNDM.SYS”,
  FreeBSD: “LIBCERNDF.SO”).

  The used seed-key has to be passed from the calling application to SafeGuard Cryptographic Engine. In products of the SafeGuard product family that use SafeGuard Cryptographic Engine, the seed key value is generated by iterated hashing of a buffer of fast changing system variables. The buffer is updated every time a random number is generated to ensure a sufficient level of entropy of the seed key.

### 3.2.2 Operation Scheme

The following three figures depict the cryptographic module and its environment (one figure for each type of operation):

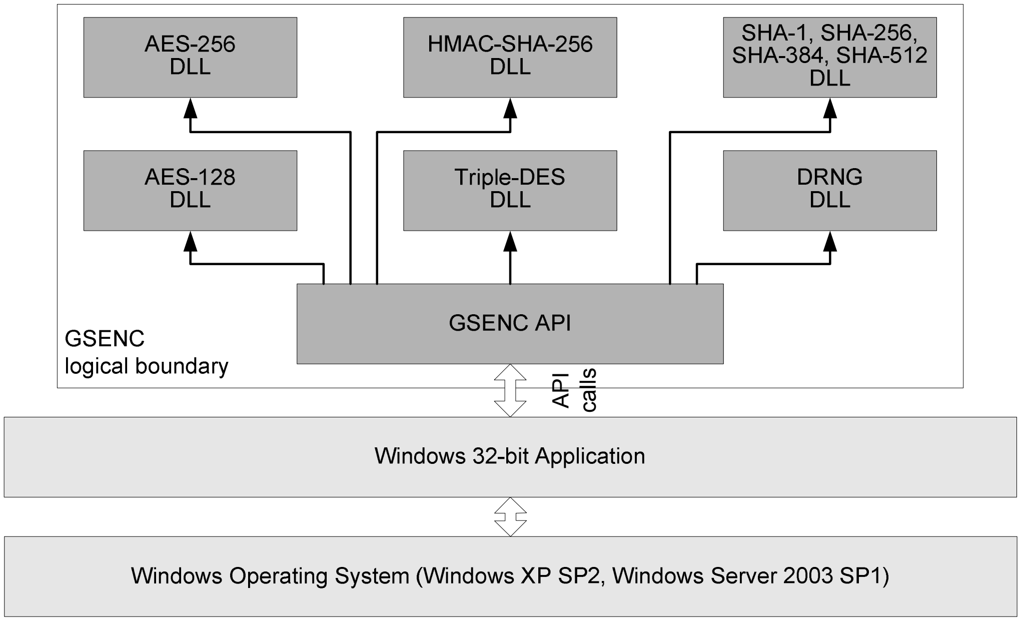

**Figure 1: Cryptographic Module Scheme (Windows application mode)**

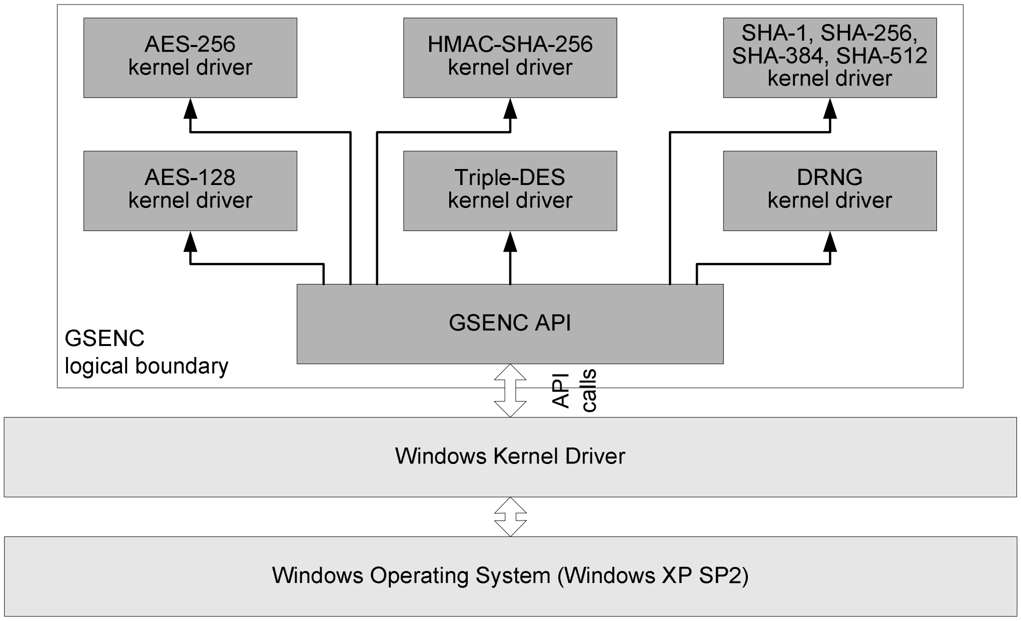

**Figure 2: Cryptographic Module Scheme (Windows kernel mode)**

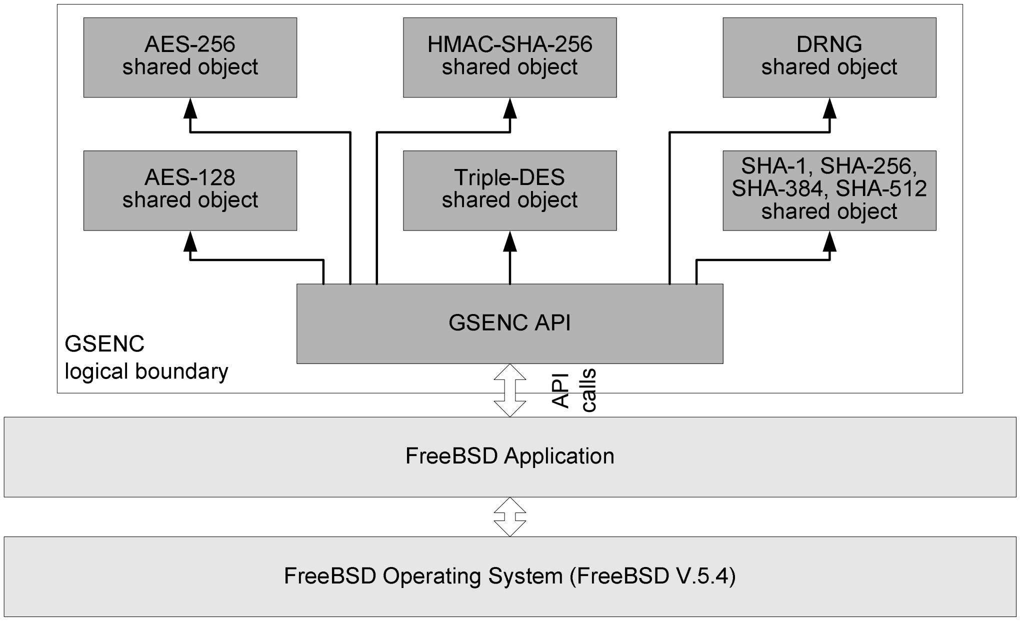

**Figure 3: Cryptographic Module Scheme (FreeBSD application mode)**

### 3.2.3 Hardware Environment

The cryptographic module is a pure software module and is running on a general purpose PC target hardware device.

The following hardware device is expected:

- a general purpose PC equipped with a microprocessor compatible to Intel Pentium 4 (or higher) architecture running one of the mentioned operating system platforms; the code of the cryptographic module is executed on the built-in microprocessor.

### 3.2.4 Cryptographic Algorithms

SGCE cryptographic module provides the following FIPS-Approved algorithms:

| Algorithms | Purpose | FIPS standard | Certificate No. |
|---|---|---|---|
| **AES-128** | Symmetric encryption with 128 bit key length | FIPS PUB 197 | 513 |
| **AES-256** | Symmetric encryption with 256 bit key length | FIPS PUB 197 | 512 |
| **Triple DES (TDEA)** | Symmetric encryption with 168 bits effective key length | FIPS PUB 46-2 | 522 |
| **SHA-1, SHA-256, SHA-384, SHA-512** | Secure hash | FIPS PUB 180-2 | 582 |
| **HMAC-SHA-256** | Message authentication Integrity check | FIPS PUB 198 | 264 |
| **SHA-256 for HMAC** | Secure hash | FIPS PUB 180-2 | 584 |
| **DRNG** | Deterministic Random Number Generator | FIPS PUB 186-2 original Appendix 3.1 | 289 |
| **SHA-1 for DRNG** | Secure hash | FIPS PUB 180-2 / FIPS PUB 186-2 | 583 |

**Table 1: FIPS-Approved Algorithms Provided by SGCE Cryptographic Module**

SGCE cryptographic module does not provide any further cryptographic algorithms either FIPS-Approved nor non-FIPS-Approved.

## 3.3 Interfaces

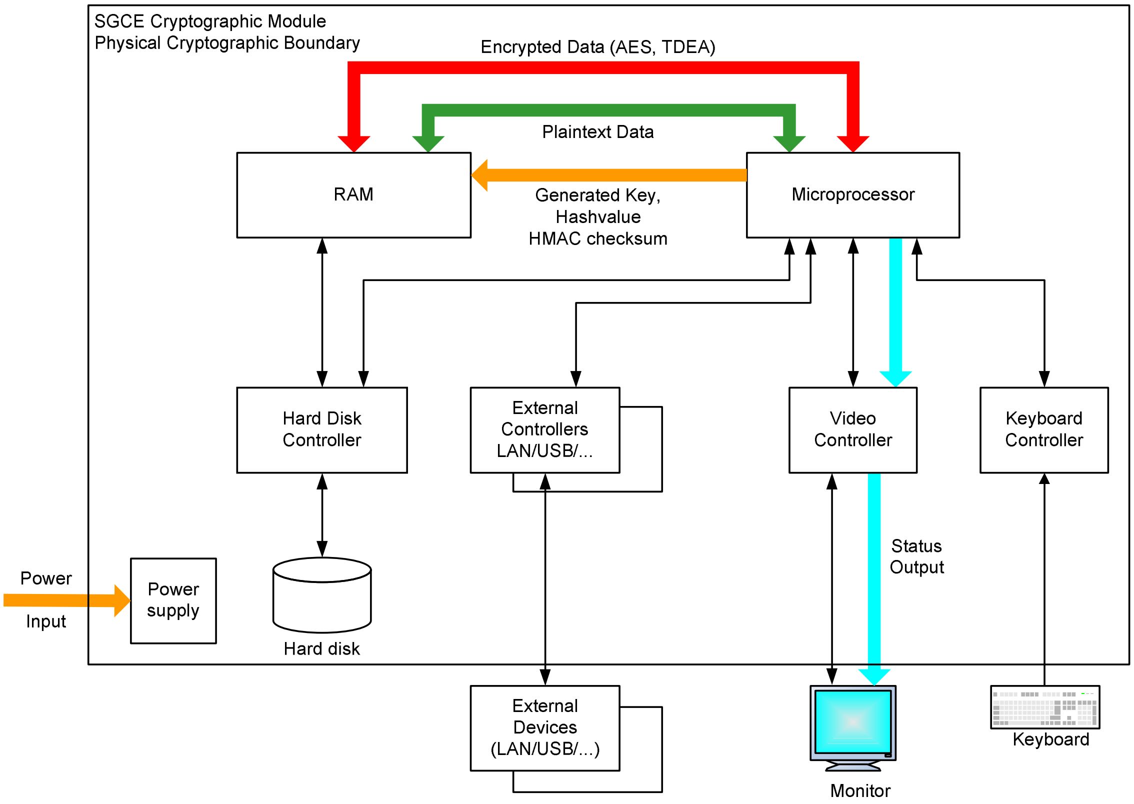

**Figure 4: Hardware Block Diagram for SGCE Cryptographic Module**

The physical boundary of the SGCE cryptographic module is the physical boundary of the device, on which the containing application is running. This is for all mentioned platforms:

The general purpose PC with its case and external interfaces for keyboard, HIDs (e.g. mouse), display, data storage devices (e.g. hard disk, CD-ROM), network ports, USB, serial and parallel interface ports etc.

The logical interfaces of the SGCE cryptographic module are defined as the API calls of the SGCE library functions.

Data input are certain API function calls and their parameters. The parameters may directly contain input data or may be pointer referencing memory data buffers with input data.

Data output are parameters of certain API function calls. The parameters are pointer referencing memory data buffers where output data shall be stored.

Control input are certain API functions calls for initialisation and status check.

Status output are the return values of the included API function calls and the parameters to the special status request function.

The logical interfaces, physical interfaces and cryptographic module interfaces can be mapped like shown in the table below:

| **Logical Interface (FIPS 140-2)** | **Module Interface** | **Physical Port** |
|---|---|---|
| Data Input Interface | API function calls containing parameters with input data or pointers to input data buffers | Keyboard, HIDs, data storage devices, external ports (network, USB, serial etc.) |
| Data Output Interface | Parameters of API function calls pointing to output data buffers | Display, data storage devices, external ports (network, USB, serial etc.) |
| Control Input Interface | API function calls provided for intialisation and control of the module | Keyboard, HIDs, data storage devices, external ports (network, USB, serial etc.) |
| Status Output Interface | Return values of certain API function calls | Display, data storage devices, external ports (network, USB, serial etc.) |
| Power Interface | not applicable | PC/handheld power interface |

**Table 2: Interfaces of SGCE Cryptographic Module**

## 3.4 Roles and Services

The SGCE security policy supports two roles:

- Crypto Officer role and
- User role

The Crypto Officer role is applied to the following individuals:

- A developer, who is building an application, which incorporates the SGCE cryptographic module by linking the SGCE library together with the application and deploying the application with the SGCE DLLs and/or kernel drivers.
- A system administrator, who is installing an application containing the SGCE cryptographic module on a target system.

The Crypto Officer role has the responsibility of correctly developing, deploying and installing applications with SGCE cryptographic module (see also chapter *4 Secure Operation*).

Any individual, who is operating an application containing the SGCE cryptographic module is assumed to hold the User role. Operators performing the User role do not have access to the SGCE product as it is delivered, but to an application and OS platform, where components of the SGCE cryptographic module are included. Assuming this, the User role does not have direct access to the cryptographic operation services included into the SGCE cryptographic module (see table 3 below). This role is not enabled to directly use these services, but these services are hidden behind the respective application. However, the User role must be enabled to retrieve status and version information as well as to execute the self-tests on request.

There is no authentication mechanism provided by the SGCE cryptographic module neither for the User role nor for the Crypto Officer role.

The cryptographic module provides the following services:

- Symmetric data encryption (AES-128, AES-256 and TDEA with 168 bits key size)
- Symmetric data decryption (AES-128, AES-256 and TDEA with 168 bits key size)
- Hash generation (SHA-1, SHA-256, SHA-384 and SHA-512)
- MAC generation (HMAC-SHA-256)
- Pseudo-Random Number Generation (FIPS 186-2)
- Input and zeroize keys
- Show status and version number
- Run self-tests

The services are provided to the User role as well as to the Crypto Officer (CO) role as specified in the table below:

| Role | Service | Affected Keys and CSPs | Access |
|---|---|---|---|
| **CO** | Input Key | Any Key | Execute |
| | Symmetric Encrypt/Decrypt | AES Key, TDEA Key | Execute |
| | Hash calculation | None | Execute |
| | MAC generation | HMAC Key | Execute |
| | Generate Random Number | DRNG Seed-key | Execute |
| | Zeroize Key | Any Key | Execute |
| | Show Status and Version | None | Read |
| | Run Self-Tests | HMAC Key for Integrity<br>Integrity Checksum | Execute |
| **User** | Show Status and Version | None | Read |
| | Run Self-Tests | HMAC Key for Integrity<br>Integrity Checksum | Execute |

**Table 3: Roles and Services of SGCE Cryptographic Module**

## 3.5 Key Management

SGCE cryptographic module uses the following keys:

- Symmetric encryption key for AES and TDEA,
- Key for generating HMAC-SHA-256,
- Seed-key for pseudo-random number generator.

Keys can be encrypted using AES-128, AES-256 or TDEA.

Keys can be zeroized by overwriting the key memory with zeroes by an API function.

The keys are input into the SGCE cryptographic module as parameters of respective API functions. Keys have to be loaded into the RAM before and are then forwarded in the form of a memory pointer as API function parameter to the SGCE cryptographic module.

The input of keys is therefore within the responsibility of the application using the cryptographic modules. If inputting keys electronically from outside the cryptographic boundary, the application shall do this in encrypted form using a FIPS approved encryption algorithm.

Each key is temporarily stored in RAM until the key is zeroized, the operating system is shut down or the PC is powered off. At that point, all encryption keys loaded into memory are destroyed.

However, the built-in Pseudo-Random Number Generator may be used to generate keys, if random keys are required. The correct key generation method has to be implemented by the application using the cryptographic module. In this case, a random number has to be generated by using the respective API function, a key shall be generated by the application using a FIPS approved method of key generation and then the key can then be determined to be used by any cryptographic operation by calling another API function of SGCE cryptographic module.

## 3.6 Physical Security

As the SGCE cryptographic module is a pure software module, there is no physical security requirement to be fulfilled by the module itself.

However, the Crypto Officer shall ensure the physical security of the computer systems, where the application with the SGCE cryptographic module is developed.

## 3.7 Operational Environment

The cryptographic module has been validated to be FIPS 140-2 compliant on the following hardware device:

- Lenovo NetVista 8307 PC equipped with Intel Pentium 4 processor 2.66 GHz, 512kB L2 cache, 256 MB RAM, 40 GB ATA-100 hard disk drive, 48x CD-ROM drive, 16x DVD-ROM drive, ATI Radeon 7000 AGP video controller, integrated Intel 10/100 MHz Ethernet network controller with RJ-45 ethernet port, IBM Standard PS/2 104 key keyboard, IBM Standard PS/2 two button mouse, six USB V2.0 ports.

running the following operating systems (one PC for each OS):

- Windows XP SP2 for application mode (DLLs) and kernel mode (SYS drivers),
- Windows Server 2003 SP1 for application mode (DLLs),
- FreeBSD Version 5.4 for application mode (SO – shared objects)

## 3.8 Self Tests

The SGCE performs the following tests at initialization before running any cryptographic operations:

- Software integrity test (HMAC-SHA-256) of all cryptographic components (DLL or kernel module or shared object):
  A HMAC-SHA-256 checksum is calculated for every component file and the checksums are compared to given checksums.
- Known Answer Tests for all cryptographic algorithms (AES-128, AES-256, TripleDES, SHA-1, SHA-256, SHA-284, SHA-512, HMAC, DRNG)
- Continuous RNG test for DRNG

The cryptographic module also has the ability to run self-tests on demand.

If the software integrity test or any known-answer test fails, the respective API function returns with an error code and the module enters the error state. In this state the cryptographic module refuses all cryptographic operations. The Show Status function returns an error code in this case, indicating, that the cryptographic module is not ready for operation.

If the software integrity test and all known-answer tests pass successfully, the cryptographic module is ready for operation and the Show Status function returns, that the cryptographic module is in FIPS Approved mode of operation.

## 3.9 Mitigation of Other Attacks

The module does not contain security mechanisms to mitigate other attacks outside the security requirements of FIPS 140-2.

# 4 Secure Operation

## 4.1 Overview

The following chapter describes how to use SGCE in a way, that it meets the security requirements of FIPS 140-2 Level 1.

## 4.2 Application Development and Installation

All individuals developing applications, which use FIPS approved components from SGCE cryptographic module are assumed holding the Crypto Officer role.

Each individals installing applications, which use FIPS approved components from SGCE cryptographic module are also assumed holding the Crypto Officer role.

They shall follow the instructions for building secure applications and installing applications using SGCE as described herein and within in the SGCE API Reference document.

Especially the following rules shall be observed by the Crypto Officer:

- The Crypto Officer is responsible for the secure installation of the SGCE cryptographic module together with the developed application on the target PC.
- The operating system on the target PC with SGCE cryptographic module installed shall be configured to single user mode.
- All keys entered from the outside into the cryptographic boundary shall be imported encrypted.
- The operator must be enabled to view the status and the version of the SGCE cryptographic module.
- The operator must be enabled to perform the self-test of the cryptographic module.

# 5 Terms and Definitions

## 5.1 Abbreviations

| | |
|---|---|
| DLL | Dynamically linkable library |
| FIPS | Federal Information Processing Standards |
| HID | Human Interface Device |
| OS | Operating System |
| PDA | Personal Digital Assistant |
| SGCE | SafeGuard Cryptographic Engine |
| SO | Shared Object |
| USB | Universal Serial Bus |

# 6 References

[FIPS 46-3] “FIPS PUB 46-3, Data Encryption Standard (DES)”, National Institute of Standards and Technology, 25 October 1999

[FIPS 140-2] “FIPS PUB 140-2, Security Requirements for Cryptographic Modules”, National Institute of Standards and Technology, May 25, 2001

[FIPS 180-2] “FIPS PUB 180-2, Secure Hash Standard with Change Notice 1”, National Institute of Standards and Technology, February 25, 2004

[FIPS 186-2] “FIPS Pub 186-2, Digital Signature Standard (DSS)”, National Institute of Standards and Technology, 27 January 2000

[FIPS 197] “FIPS PUB 197, Advanced Encryption Standard (AES)”, National Institute of Standards and Technology, November 26, 2001

[FIPS 198] “FIPS PUB 198, The Keyed-Hash Message Authentication Code (HMAC)”, National Institute of Standards and Technology, March 06, 2002

[SGCE-VED] “SGCE Cryptographic Library, FIPS 140-2 Level 1 Vendor Evidence Documentation”, Version 1.00, Utimaco Safeware AG, January 2007