Oracle

Database 11g & 11g

Release 2 TDE Microsoft Windows Server 2008 (R2)

Integration Guide

CryptoServer

utimaco®

Imprint

Copyright 2020	Utimaco IS GmbH
	Germanusstr. 4
	D-52080 Aachen
	Germany
Phone	AMERICAS +1-844-UTIMACO (+1 844-884-6226)
	EMEA +49 800-627-3081
	APAC +81 800-919-1301
Internet	https://support.hsm.utimaco.com/
e-mail	support@utimaco.com
Document Version	1.0.0
Date	06/10/2025
Status	PUBLISHED
Document No.	IG-2025-0022

Table of Contents

1 Introduction

This paper provides an integration guide explaining how to integrate a Hardware Security Module (HSM) -- *CryptoServer* -- with the *Oracle Database 11g & 11g Release 2* Transparent Data Encryption (TDE).

1.1 Concepts

Database encryption helps address compliance requirements associated with public and private privacy and security mandates such as PCI and California SB1386. *Oracle Database Transparent Data Encryption (TDE)* encrypts data that is stored in an *Oracle* database and decrypts data retrieved from an *Oracle* database.

Oracle Advanced Security TDE column encryption was introduced in *Oracle Database 10g Release 2*, enabling encryption of application table columns such as credit card and social security numbers. Oracle Advanced Security TDE tablespace encryption and support for hardware security modules (HSM) were introduced with *Oracle Database 11g*.

The *SafeGuard CryptoServer* is the HSM developed by *Utimaco Safeware*, i.e. a physically protected specialized computer unit designed to perform sensitive cryptographic tasks and to securely manage cryptographic keys and data. In a *SafeGuard CryptoServer* security system security-relevant actions can be executed and security relevant information can be stored. It can be used as a universal, independent security component for heterogeneous computer systems.

The *Oracle Database* server master key is an encryption key used to encrypt secondary keys used for column encryption and tablespace encryption. This key is stored inside and secured by the *SafeGuard CryptoServer*.

The *SafeGuard CryptoServer* has been certified with *Oracle Advanced Security Transparent Data Encryption* to provide an even higher level of security. The industry-standard API PKCS#11 is used to integrate the *Oracle Database TDE* with a *SafeGuard CryptoServer* solution.

2 Requirements

Ensure that you are familiar with the *Oracle Database 11g (Release 2)* database documentation and setup, and have a copy of the *SafeGuard CryptoServer Manual for System Administrators*[2]. This document also assumes that a supported operating system has already been installed.

Software- and Hardware Requirements

HSM Model	SafeGuard CryptoServer CS-Series/S-Series/Se-Series PCI SafeGuard CryptoServer CS-Series/S-Series/Se-Series LAN SafeGuard CryptoServer Simulator
HSM Firmware	SafeGuard SecurityServer 2.60.0 SafeGuard SecurityServer 3.00.2
Software	Oracle Database 11g (Release 2) SafeGuard SecurityServer 2.60.0

2.1 Supported Operating Systems

The interoperability of the *SafeGuard CryptoServer* solution, an operating system and the *Oracle Database 11g (Release 2) TDE* have been tested successfully for the following combinations:

Operating System	SafeGuard SecurityServer Version	Oracle Database Version	PCI Support	Ethernet Support
Microsoft Windows Server 2008 (R2)	2.10.2 2.60.0 3.00.2	11.1.0.7.0 11.2.0.1.0	Yes	Yes

3 Oracle Database 11g(Release2) with Microsoft Windows Server 2008

To integrate the *SafeGuard CryptoServer* into the *Oracle Database 11g (Release 2) TDE* complete the following steps:

1. Install *SafeGuard CryptoServer* hardware an/d software
2. Install *Oracle Database 11g (Release 2)*

At this point you have to choose between two following approaches:

- Configure the Oracle Software Wallet first and then migrate to TDE with *SafeGuard CryptoServer*. If you choose this, you have to perform the following steps:
  - Configure and test Transparent Data Encryption (TDE) with the Oracle Wallet
  - Migrate from TDE with the Oracle Wallet to TDE with the *SafeGuard CryptoServer PKCS#11 library*
- or you can choose to configure TDE with *SafeGuard CryptoServer* right away. Then you have to perform the next step:
  - Configure Oracle Advanced Security Transparent Data Encryption with the *SafeGuard CryptoServer PKCS#11 library*

The first approach is appropriate for getting familiar with TDE without configuring the HSM first. After migration the master key is stored in the HSM.

The second approach demonstrates TDE with a HSM without initializing the Oracle Wallet first. If the Oracle Wallet is already initialized in *Oracle Database 11.1.0.7*, tablespace encryption will rely on that software wallet even after migration to HSM. It is not possible to migrate the tablespace master key to HSM until *Oracle Database 11g Release 2*.

3.1 Install Safe Guard Crypto Server Hardware and Software

For more information on installing and setting up *SafeGuard CryptoServer PCI(e)* and *SafeGuard CryptoServer LAN* hardware, see the documentation *SafeGuard CryptoServer PCI(e) Operating and Installation Manual* [5] [4] and *SafeGuard CryptoServer LAN Operating and Installation Manual* [3].

To install the *SafeGuard SecurityServer* software perform the following steps:

1. Navigate to the SafeGuard SecurityServer product CD and double click the *CryptoServerSetup<version>.exe* file. The installation GUI appears.
2. Follow the onscreen instructions to complete the *SafeGuard SecurityServer* installation. Make sure that you install the PKCS#11 library (more details in chapters 3.4 and 3.5).
3. Read the *SafeGuard CryptoServer PKCS#11 documentation* [1] and follow the instructions.
4. Be sure to edit the PKCS#11 configuration file:

PKCS#11

cs2_pkcs11.ini

PKCS#11 R2

cs_pkcs11_R2.cfg

- Create a [CryptoServer] section and set the device parameter to point to the *SafeGuard CryptoServer* device to be used.
- Set the **SlotCount** parameter to 1.
- In case of installation on *Microsoft Windows Server 2008* add the following line
- **MultiInitReturnsCKR_OK = true** in the [Global] section.
- You have the possibility to add secure messaging by using an encrypted data layer and strong authentication (e.g. with smartcards) to gain access to the *SafeGuard CryptoServer*.

3.2 Install Oracle Database 11g (Release 2)

The next step is to install the *Oracle Database 11g (Release 2)*:

1. Unzip the installer to a temporary folder.
2. Browse to the temporary folder and start the setup with *setup.exe*. The installation GUI appears.
3. Follow the onscreen instructions to complete the *Oracle Database 11g (Release 2)* installation and reboot *Microsoft Windows Server 2008 (R2)* afterwards.

4. Set the following system variables in order to keep the navigation among significant Oracle paths simple (following system variables are valid for default installation):

`ORACLE_HOME = %ORACLE_BASE%\product\11.1.0\dbhome_1`

`TNS_ADMIN = %ORACLE_HOME%\NETWORK\ADMIN`

For more information on installing the *Oracle Database 11g (Release 2)* on *Microsoft Windows*, see the *Oracle Database 11g (Release 2)* documentation, available at http://www.oracle.com.

3.3 Configureand TestTransparent Data Encryption (TDE) with the Oracle Wallet

To start using Transparent Data Encryption (TDE), create a wallet and set a master key. Oracle recommends that you use a separate encryption wallet to store the master encryption key for your database.

Consider that it is not possible to migrate the master key for TDE tablespace encryption to an HSM after the Oracle Wallet has been initialized in *Oracle Database 11.1.0.7*. In *Oracle Database 11g Release 2*, both master keys for TDE column encryption and TDE tablespace encryption are migrated to a unified master encryption key that can be migrated from the Oracle Wallet to an HSM. To verify that the wallet mechanism is working fine, start a first test with the default software-based wallet:

1. Add the following lines to file `%TNS_ADMIN%\sqlnet.ora`:

```
ENCRYPTION_WALLET_LOCATION =

(SOURCE = (METHOD = FILE) (METHOD_DATA = (DIRECTORY = C:\oracle\WALLETS)))
```

after creating the directory `C:\oracle\WALLETS`.

2. Open an SQL Plus session `(Start -> Programs -> Oracle -> OraDB11g_home1 -> Application Development -> SQL Plus)`.
3. Connect to your database as system:

```
SQL> connect system/password
```

4. Create an encryption wallet -- the master key is added into it automatically -- the double quotes are mandatory:

```
SQL> alter system set encryption key identified by

<your wallet password>;
```

5. Encrypt the credit_limit column of the CUSTOMERS table which is owned by the user OE:

```
SQL> alter table oe.customers modify (credit_limit encrypt);
```

6. With the next command, these values listed in the encrypted column are returned in clear text. Transparent Data Encryption decrypts them automatically:

```
SQL> select credit_limit from oe.customers where rownum < 15;
```

7. This command lists the encrypted columns in your database:

```
SQL> select * from dba_encrypted_columns;
```

8. Finally, this view contains information about the wallet itself:

```
SQL> select * from v$encryption_wallet;
```

9. Create an encrypted tablespace:

```
SQL> CREATE TABLESPACE securespace DATAFILE

'C:\app\Administrator\oradata\orcl\secure01.dbf'

SIZE 10M ENCRYPTION DEFAULT STORAGE (ENCRYPT);
```

10. Check if the new tablespace is listed and marked as encrypted:

```
SQL> select tablespace_name, encrypted from dba_tablespaces;
```

11. Close the wallet:

```
SQL> alter system set encryption wallet close;
```

In Oracle Database 11g Release 2, the command to close the wallet has been changed to:

```
SQL> alter system set encryption wallet close identified by

<your wallet password>;
```

12. Exit from your *SQL*Plus* session:

```
SQL> exit
```

13. Start *Oracle Wallet Manager* (`Start -> Programs -> Oracle -> OraDB11g_home1 -> Integrated Management Tools -> Wallet Manager`).

Open the software-based wallet by navigating to the wallet directory *C:\oracle\WALLETS*, check the *Auto-Open* option, click *Save* and the click *Exit*.

14. Verify that an auto-open software wallet has been created in the *C:\oracle\WALLETS* directory. This directory contains two wallets now, the encryption wallet (*ewallet.p12*) and the auto-open wallet (*cwallet.sso*). Check the permission settings of the auto-open wallet: Right click `cwal-`

`let.sso -> Properties -> Security -> Advanced -> Edit -> check "Include inheritable permissions from this object's parent" -> OK`.

Rename the encryption wallet `ewallet.p12` to `ewallet.p24`. This stops *Transparent Data Encryption* from opening the encryption wallet but we have an auto-open wallet now.

3.4 Migrate from TDE with the Oracle Wallet to TDE with SafeGuard CryptoServer

This chapter describes the migration from Oracle Wallet to HSM.

To demonstrate the integration of *Oracle Database 11g (Release 2) TDE* with *SafeGuard CryptoServer* solution initialize a PKCS#11 slot:

1. Navigate with the explorer to the directory of the *SafeGuard SecurityServer* software installation and go into the PKCS#11 library directory (*\Program Files\Utimaco\SafeGuard CryptoServer\Lib*)
2. Open the file PKCS#11 configuration file with double-click.

PKCS#11

cs2_pkcs11.ini

PKCS#11 R2

cs_pkcs11_R2.cfg

3. Double check that the parameter **device** points to your *SafeGuard CryptoServer* hardware, e.g. `PCI:0` for a pci card or `192.168.0.140` for a CryptoServer LAN appliance
4. Save your changes, see also chapter 3.1.
5. Check if the current PKCS#11 environment variable `CS2_PKCS11_INI (CS_PKCS11_R2_CFG PKCS#11 R2)` points to the edited PKCS#11 configuration file `cs2_pkcs11.ini (cs_pkcs_R2.cfg PKCS#11 R2)` file. If the variable does not exist, create it.
6. Open a command line shell. (`Start -> Run -> cmd.exe`)
7. Check if you can connect to the *SafeGuard CryptoServer* using *p11tool* respectively *p11tool2*:

   PKCS#11

   ```
   p11tool ListSlots
   ```

   PKCS#11 R2

   ```
   p11tool2 ListSlots
   ```

8. Initialize the PKCS#11 slot 0 (you can use *p11tool* or the *SafeGuard CryptoServerAdministration Tool (CAT)* for the initialization):

PKCS#11

```
p11tool slot=0 InitToken=<SO_pin> p11tool slot=0 LoginSO=<SO_pin> InitPin=<HSM_passphrase>
```

PKCS#11 R2

```
p11tool2 slot=0 \
    Login=<Administrative user,AuthenticationToken> \
    InitToken=<SO_pin>
p11tool2 slot=0 LoginSO=<SO_pin> InitPin=<HSM_passphrase>
```

Make sure that the HSM_passphrase contains at least 8 characters (letters and numbers or special characters). Otherwise Oracle won't accept it as Oracle Wallet password!

To test *Oracle Database 11g (Release 2) TDE* with *SafeGuard CryptoServer*:

1. Copy the PKCS#11 library *cs2_pkcs11.dll* respectively PKCS#11 R2 library *cs_pkcs11_R2.dll* (`\Program Files\Utimaco\SafeGuard CryproServer\Lib) to %SYSTEM_DRIVE%\oracle\extapi\ [32,64]\hsm\Utimaco\<version>\`).

For example:

```
C:\oracle\extapi\32\hsm\Utimaco\1.00.00\cs2_pkcs11.dll
```

You will have to create the directory manually first.

1. Change the parameter `METHOD of ENCRYPTION_WALLET_LOCATION` in the file `%TNS_ADMIN%\sqlnet.ora` to HSM:

```
ENCRYPTION_WALLET_LOCATION =

(SOURCE = (METHOD = HSM)(METHOD_DATA =

(DIRECTORY = C:\oracle\WALLETS)))
```

3. Before migration rename *ewallet.p24* back to *ewallet.p12* at *C:\oracle\WALLETS* and check the permission settings of the file: `right-click ewallet.p12 > Properties > Security > Advanced > Edit > check "Include inheritable permissions from this object's parent" and click OK`.
4. Log back into the database:

```
SQL> connect system/password
```

5. Migrate master encryption key from Oracle Wallet to HSM:

```
SQL> alter system set encryption key identified by <HSM_passphrase> migrate using
<wallet_password>;
```

where:

- `HSM_passphrase`

  `HSM_passphrase` is the passphrase of the PKCS#11 user which has been given at the initialization of the PKCS#11 slot. The master key in the *SafeGuard CryptoServer* is not used for tablespace encryption in *Oracle Database 11g R1*. It still relies on the software wallet created in section 3.3, step 4 (since it is not possible to migrate the tablespace master key in *Oracle Database 11g R1*). In *Oracle Database 11 Release 2*, the two master encryption keys in the wallet are migrated to a unified master encryption key in the HSM.

- `wallet_password`

`wallet_password` is the password for the software wallet created in section 3.3, step 4.

You can check all operations performed by HSM with the PKCS#11 log file *cs2_pkcs11.log* repectively *cs_pkcs11_R2.log* (PKCS#11 R2). For this purpose set the log level to highest level in the configuration file. (don't keep it this way because of performance loss; delete the log file afterwards) and set the log path to `c:\temp`.

6. The next query returns the values listed in the encrypted column in plain text:

```
SQL> select credit_limit from oe.customers where rownum < 15;
```

Transparent Data Encryption decrypts them automatically, now using the HSM master key.

7. Close the wallet:

```
SQL> alter system set encryption wallet close;
```

In Oracle Database 11g Release 2, the command to close the wallet has been changed to:

```
SQL> alter system set encryption wallet close identified by
<your_wallet_password>;
```

8. Exit from your SQL*Plus session:

```
SQL> exit
```

9. Start Oracle Wallet Manager (`Start -> Programs -> Oracle -> OraDB11g_home1 -> Integrated Management Tools -> Wallet Manager`).
10. Open the software-based wallet by navigating to the wallet directory *C:\oracle\WALLETS* and click Change Password. Use the same string you used for the HSM (`HSM_passphrase`) as the new password for the software based wallet. Click Save and then click Exit.
11. Log back into the database:

```
SQL> connect system/password

SQL> alter system set wallet open identified by <HSM_passphrase>;
```

Now that the password for the Oracle Wallet and HSM are the same, both are opened with one command. If a password change is not feasible, use an auto-open wallet and rename or remove the wallet from the `ENCRYPTION_WALLET_LOCATION` specified in *sqlnet.ora*.

NEVER delete the encryption wallet, and NEVER forget the password of the encryption wallet.

12. Verify that HSM is now used for master key storage as well (migration is a re-key operation for the master key for column encryption in *Oracle Database 11g R1*, or for both master encryption keys in *Oracle Database 11g Release 2*):

```
SQL> select * from v$encryption_wallet;
```

13. Create an encrypted tablespace, using the master encryption key from the software wallet in *Oracle Database 11g R1*, or the unified master key from the HSM with *Oracle Database 11g Release 2*:

```
SQL> CREATE TABLESPACE securespace2

DATAFILE 'C:\app\Administrator\oradata\orcl\secure02.dbf'

SIZE 10M ENCRYPTION DEFAULT STORAGE(ENCRYPT);
```

14. Check if the new tablespace is listed and marked as encrypted:

```
SQL> select tablespace_name, encrypted from dba_tablespaces;
```

15. Now test re-keying of the master encryption key in HSM and check column encryption one more time:

```
SQL> alter system set encryption key identified by

<HSM_passphrase>;
```

Verify that a new master encryption key is created in HSM by watching the logfile *cs2_pkcs11.log*.

The encryption key for the individual table `OE.CUSTOMERS` is now encrypted with the new master key:

```
SQL> select credit_limit from oe.customers where rownum < 15;
```

16. Remove the master key from the database memory:

```
SQL> alter system set encryption wallet close

[identified by <HSM_password>];
```

17. And try listing encrypted data again which shouldn't work this time:

```
SQL> select credit_limit from oe.customers where rownum < 15;
```

fails because no access to master key.

18. Exit from your SQL*Plus session:

```
SQL> exit
```

3.5 Configure Oracle Advanced Security TDE with SafeGuard CryptoServer

This chapter demonstrates the integration of *Oracle Database 11g (Release 2)* TDE with *SafeGuard CryptoServer* solution. It provides examples for column and tablespace encryption with HSM. If you already initialized the Oracle Wallet, follow the instructions of the last chapter to migrate from Oracle Wallet to HSM. Consider that it won't be possible (in *Oracle Database 11.1.0.7*) to migrate the tablespace master key to HSM after the Oracle Wallet has already been initialized.

First initialize a PKCS#11 slot.

1. Navigate with the explorer to the directory of the *SafeGuard SecurityServer* software installation and go into the PKCS#11 library directory (*\Program Files\Utimaco\SafeGuard CryptoServer\Lib*)
2. Open the file *cs2_pkcs11.ini* with double-click.
3. Double check that the parameter $_{\text{device}}$ points to your *SafeGuard CryptoServer* hardware, e.g. PCI:0 for a pci card or 192.168.0.140 for a CryptoServer LAN appliance
4. Save your changes, see also chapter 3.1.

5. Check if the current PKCS#11 environment variable CS2_PKCS11_INI (CS_PKCS11_R2_CFG PKCS#11 R2) points to the edited PKCS#11 configuration file *cs2_pkcs11.ini* (*cs_pkcs_R2.cfg* PKCS#11 R2) file. If the variable does not exist, create it.
6. Open a command line shell. (`Start -> Run -> cmd.exe`)
7. Check if you can connect to the *SafeGuard CryptoServer* using *p11tool* respectively *p11tool2*:

PKCS#11

```
p11tool ListSlots
```

PKCS#11 R2

```
p11tool2 ListSlots
```

8. Initialize the PKCS#11 slot 0 (you can use *p11tool* or the *SafeGuard CryptoServerAdministration Tool (CAT)* for the initialization):

PKCS#11

```
p11tool slot=0 InitToken=<SO_pin>
```

```
p11tool slot=0 LoginSO=<SO_pin> InitPin=<HSM_passphrase>
```

PKCS#11 R2

```
p11tool2 slot=0 \
    Login=<Administrative user,AuthenticationToken> \
    InitToken=<SO_pin>
p11tool2 slot=0 LoginSO=<SO_pin> InitPin=<HSM_passphrase>
```

To test the Oracle Database11g (Release 2) TDE with the *SafeGuard CryptoServer*:

1. Copy the PKCS#11 library *cs2_pkcs11.dll* (`\Program Files\Utimaco\SafeGuard CryproServer\Lib) to %SYSTEM_DRIVE%\oracle\extapi\[32,64]\hsm\Utimaco\<version>\`).

For example:

`C:\oracle\extapi\32\hsm\Utimaco\1.00.00\cs2_pkcs11.dll`

You will have to create the directory manually first.

2. Add the following line to the file *%TNS_ADMIN%\sqlnet.ora*:

```
ENCRYPTION_WALLET_LOCATION = (SOURCE = (METHOD = HSM))
```

3. Open an SQL Plus session (`Start > Programs > Oracle > OraDB11g_home1 > Application Development > SQL Plus`).
4. Connect to your database as system:

```
SQL> connect system/password
```

5. Create the master key. It is added into HSM automatically. The double quotes are mandatory:

```
SQL> alter system set encryption key identified by <HSM_passphrase>;
```

where `HSM_passphrase` is the passphrase of the PKCS#11 user which has been given at the initialization of the PKCS#11 slot. You can check all operations performed by HSM by watching the logs located at *c:\tmp\cs2_pkcs11.log*. For this purpose set the log level to `15` in *cs2_pkcs11.ini* (don't keep it this way because of performance loss; delete the log file afterwards) and set the log path to *c:\temp*.

6. Encrypt the credit_limit column of the CUSTOMERS table which is owned by the user OE:

```
SQL> alter table oe.customers modify (credit_limit encrypt);
```

7. With the next command, these values listed in the encrypted column are returned in clear text:

```
SQL> select credit_limit from oe.customers where rownum < 15;
```

Transparent Data Encryption decrypts them automatically using keys which are encrypted by the master key stored in HSM.

8. This command lists the encrypted columns in your database:

```
SQL> select * from dba_encrypted_columns;
```

9. This view contains information about the wallet:

```
SQL> select * from v$encryption_wallet;
```

10. Now test re-keying of the master encryption key in HSM and check column encryption one more time:

```
SQL> alter system set encryption key identified by <HSM_passphrase>;
```

Verify that a new master encryption key is created in HSM by watching the PKCS#11 logfile.

11. The encryption key for the individual table `OE.CUSTOMERS` is now encrypted with the new master key:

```
SQL> select credit_limit from oe.customers where rownum < 15;
```

12. Finally create an encrypted tablespace using HSM:

```
SQL> CREATE TABLESPACE securespace1

DATAFILE 'C:\app\Administrator\oradata\orcl\secure01.dbf'

SIZE 10M ENCRYPTION DEFAULT STORAGE(ENCRYPT);
```

13. Check if the new tablespace is listed and marked as encrypted:

```
SQL> select tablespace_name, encrypted from dba_tablespaces;
```

14. Remove the master key from the database memory:

```
SQL> alter system set encryption wallet close;
```

In *Oracle Database 11g Release 2*, the command to close the wallet has been changed to:

```
SQL> alter system set encryption wallet close identified by <your wallet
password>;
```

15. And try listing encrypted data again which shouldn't work this time:

```
SQL> select credit_limit from oe.customers where rownum < 15; fails because no
access to master key
```

16. Exit from your SQL*Plus session:

```
SQL> exit
```

For more information, see the *Oracle Database 11g Release 2 TDE* documentation available at http://www.oracle.com.

4 Further Information

This document forms a part of the information and support which is provided by the Utimaco Safeware. Additional documentation can be found on the product CD in the documentation directory.

All SafeGuard CryptoServer product documentation is also available at the Utimaco Safeware website: http://hsm.utimaco.com

5 Troubleshooting

The following table describes problems you might encounter when you configure a *SafeGuard CryptoServer* solution with *Oracle Database 11g (Release2) TDE*:

Error message	Solution
ORA-28376 Cannot find PKCS11 library	Check the PKCS#11 library path and confirm that the library path is correct. For example: `C:\oracle\extapi\32\hsm\Utimaco\1.00.00\cs2_pkcs11.dll`
ORA-2877 No need to migrate from wallet to HSM	No Oracle Wallet was created to migrate from or the migration or the migration already happened.
ORA-28353 Failed to open wallet	Ensure that the HSM passphrase or software wallet password is correct (use quotation marks). Also ensure that the content of *sqlnet.ora* is correct.
ORA-28374 Typed master key not found	Check the permission of *ewallet.p12* file; ensure that the checkbox "`Include inheritable permissions from this object's parent`" is selected
ORA-00600 internal error code, arguments: [kzthsminit: C_initialize]	Reboot Microsoft Windows Server after the installation in order to restart Oracle services.

6 References

[1] UTIMACO SAFEWARE AG. *CryptoServer PKCS#11 Interface*, 2011. 2006-0003.

[2] UTIMACO SAFEWARE AG. *SafeGuard CryptoServer - Manual for System Administrators*, 2011. M0100001-en.

[3] UTIMACO SAFEWARE AG. *SafeGuard CryptoServer LAN - Operating Manual*, 2011. M010-0006-en.

[4] UTIMACO SAFEWARE AG. *SafeGuard CryptoServer PCI - Operating and Installation Manual*, 2011. M010-0003-en.

[5] UTIMACO SAFEWARE AG. *SafeGuard CryptoServer PCIe - Operating and Installation Manual*, 2011. M010-0004-en.