CryptoServer LAN

User Manual

Utimaco Safeware AG

Transaction Security

www.utimaco.com

Imprint

Utimaco Safeware AG
Transaction Security
Germanusstr. 4
52080 Aachen

Phone +49 (0)241 / 1696-200

Telefax +49 (0)241 / 1696-199

Internet www.utimaco.de

E-Mail hsm@aachen.utimaco.de

Document Number 2003-0002

Version 2.1.5

Date 11.01.2008

Authors Frank Linneweber
Bernd Doetsch
Dipl-Inf. (FH) Ralf Vennemann
Dipl. Inf. Rainer Herbertz

Table of Contents

1 Introduction

Thank you very much for buying our security system CryptoServer LAN. We hope that our product will meet your expectations. Should you have any complaints or suggestions for improvement please let us know.

Please read this user manual thoroughly before installation and start of operation of CryptoServer LAN.

1.1 About This Manual

Utimaco Safeware AG (Transaction Security) reserves the right to modify or change this user manual without prior notice. Utimaco Safeware AG is not liable for damages resulting from print or translation errors.

In order to facilitate easy usage and quick reference, certain passages of the manual were marked with symbols. The meaning of the symbols will be explained in the following:

Important security advice that should always be followed.

Additional information.

```
This border is used to show messages at
the display of the CryptoServer LAN.
```

2 CryptoServer LAN Security System

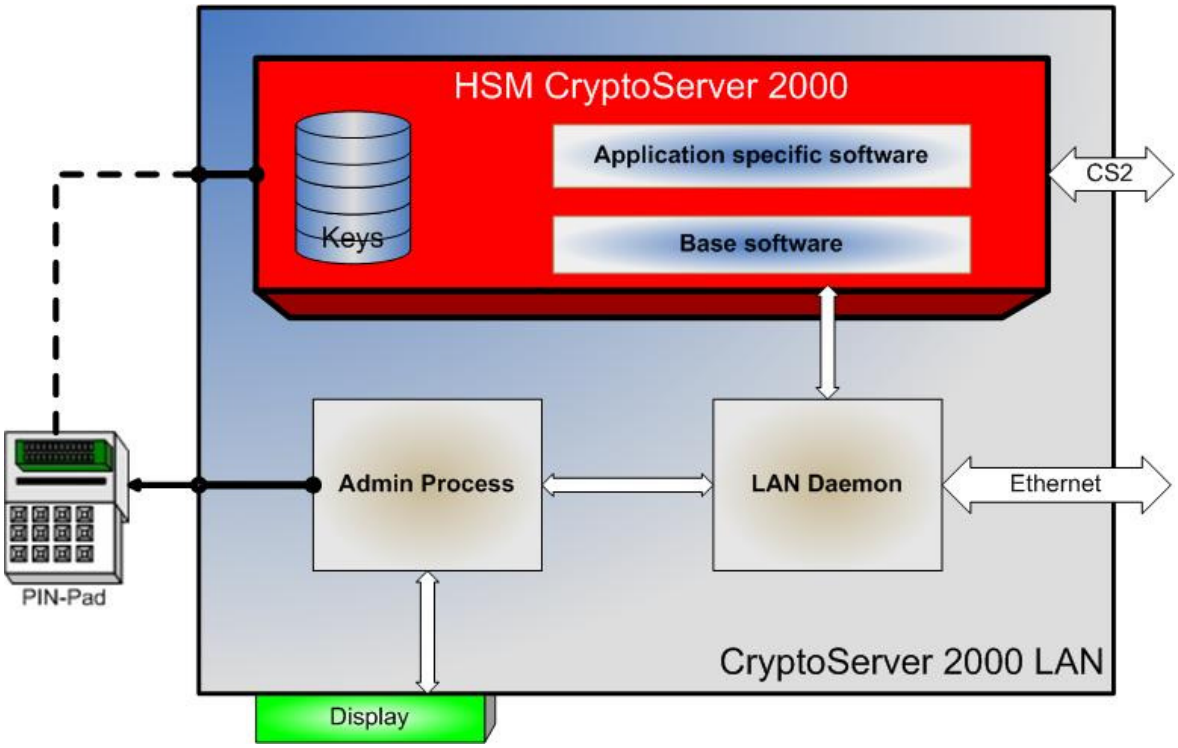
CryptoServer LAN is a universal, independent security component for heterogeneous computer systems. CryptoServer LAN is a security system in which security-relevant actions can be executed and security-relevant information can be stored.

2.1 System Architecture of CryptoServer LAN

CryptoServer LAN consists of the security processor CryptoServer and a communication unit housed in a 19" case.

The CryptoServer is a hardware security module (HSM) based on a tamper proof and tamper-responding microprocessor system which evaluates and defeats physical attacks. Security-relevant information, such as cryptographic keys and secret data, are stored in the tamper-proof buffered on-board memory to ensure maximum integrity and confidentiality. All security relevant software is running inside the CryptoServer.

The communication unit is based on a hardened Linux operating system and contains a LAN daemon (daemon CSXLAN) which establishes the connection between the CryptoServer and the network (e. g. via Ethernet). An integrated display managed by the administration software facilitates the configuration of the system parameters and the administration of the CryptoServer security processor.



CryptoServer LAN system architecture

There may be more than one CryptoServer security processor installed inside the CryptoServer LAN.

For administration tasks, a commercial off-the-shelf VGA screen and a PC keyboard can be connected to the back of the device. However, there is no requirement for a screen or keyboard for normal operation.

CryptoServer LAN has four serial interfaces. Two of them are located on the front side:

- **COM**: This serial port is managed by the administration software of the communication unit. For some administrative tasks like loading or deleting files (see section 4.3.1.2) you must connect a PIN-Pad to this serial port.
- **CS2(1)**: This serial port is connected directly to the CryptoServer and is therefore best suited for loading of secret keys etc. You need special application firmware modules inside the CryptoServer to make use of this port.
- **CS2(2)**: If two CryptoServer HSM's are installed inside the CryptoServer LAN, this serial port is connected directly to the second CryptoServer HSM.

One of the serial interface connectors is located on the back side:

- **Service port**: The CryptoServer writes diagnostic messages to the service port during start-up. You can connect a terminal with 115200 Baud, 8 bits, even parity to this port.

Additional connectors of the CryptoServer LAN:

- VGA connector
- RJ 45 Ethernet connector 10/100/1000 MBit
- Combined keyboard/mouse connector at the back side: To connect the keyboard you must use the included splitter cable.
- PS/2 keyboard connector at the front side

2.2 Connecting a PIN-Pad

Sometimes it is necessary to connect a PIN-Pad to the CryptoServer LAN. Depending on the function to be executed, the PIN-Pad has to be connected either to the connector labeled "COM", "CS2(1)" or "CS2(2)" at the front of the device (see also 4.3.1.2).

Some types of the PIN-Pads get there mains voltage from an AC DC adapter. PIN-Pads that have no AC DC adapter must be additionally connected to the PS/2 connector at the front to set up the power supply.

3 Installation and Operation of the CryptoServer LAN

The only requirements for installation are a 100-240 V power supply and a 10/100/1000 MBit Ethernet link. For service purposes and troubleshooting you need a standard VGA monitor and a PC keyboard with a PS/2 connector.

The CryptoServer LAN can be used as a desktop or a 19´´-rack mount system. Appropriate fastening angles are included.

The CryptoServer LAN is equipped with a 4 x 40 display with 6 keys to control the CryptoServer LAN display administration menu.

Key	Description
OK	Enters the menu system or a submenu; executes the selected item
EXIT	Aborts the current action; goes to the previous menu
1	Selects the previous item in a menu
2	Moves the cursor right in a selection list; selects entries in a multi select menu (e. g. ‘delete files’)
3	Selects the next item in a menu
4	Moves the cursor left in a selection list; acts like the “EXIT” key in a submenu

Note: In this document the keys 1, 2, 3 and 4 are also denoted as arrow keys (↑, →, ↓ and ←, respectively).

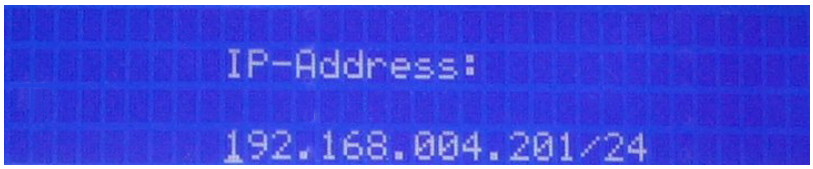
3.1 Installation

Please perform the following steps:

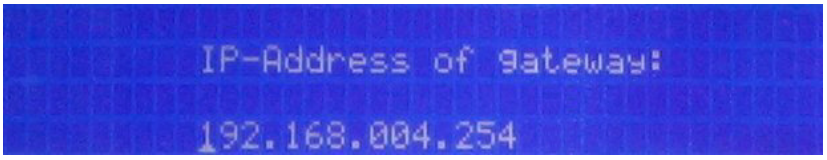
1. Connect the system to a 100-240 V power supply.
2. Connect a twisted pair cable to the RJ 45 connector labeled “Gbit Lan” and your network.
3. Turn the power on. The system is up after approximately 30 seconds. On the display you see the status screen.
4. To setup the IP-address press the “OK” key: the display enters the menu screen. Select the menu item “LAN Box administration > Configuration > IP address“.

   You can change the IP-address with the arrow keys: The keys ← respectively → change the cursor position; the keys ↑ and ↓ change the displayed digit.

   The 2 digits after the slash represent the number of consecutive ´1´ bits in the desired netmask. The number “24” in the image below corresponds to the netmask “255.255.255.0”.



5. For accepting the selected IP-address and netmask press the “OK” key again.
6. If you want to connect to the CryptoServer LAN from outside the current network you must provide a default gateway. You can do this with the menu item ”LAN Box administration > Configuration > Default Gateway“



7. To check if the network is correctly configured select the item ”LAN Box administration > Diagnostic > ping“. Adjust the pre-set IP-address to the one of a host in your network and press the “OK” key.

3.2 Starting Up and Shutting Down

Once the device is switched on, the system automatically powers up. After approx. 30 seconds the CryptoServer LAN is ready for operation.

Before the device is switched off, the CryptoServer LAN should be shut down. For doing this the following alternatives exist:

- Shut down the system by using the menu item “LAN Box administration > Shutdown”.
- If a keyboard is connected to CryptoServer LAN, then the shutdown can be initiated by the key combination <Ctrl> + <Alt> + <Del>.
- Alternatively, the shutdown can be initiated by a software command over TCP (see section 5.3.1).

After approx. 15 seconds the system will power off.

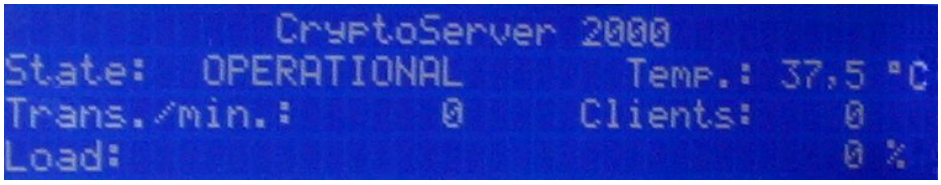
Every device with a CryptoServer security module must steadily be kept under voltage. A steadily inoperative device may lead to an insufficient power supply of the security module which activates the CryptoServer secure deletion procedure.

Consequential repairs are not covered by the guarantee!

A temporary closedown (e. g. during transport) is no danger for the security module.

3.3 Operation of the CryptoServer LAN

Once the CryptoServer LAN is booted the following status screen is displayed:



In this screen the following information is given:

1. Status of the CryptoServer (see below)
2. Temperature in degree Celsius
3. Transactions per minute
4. Number of currently connected clients
5. Load - Average utilization of the last minutes.
   The average utilization is the ratio of the time that requests spent in the CryptoServer to the total time.
6. Battery state.
   If one of the two internal batteries of the CryptoServer LAN is low, a battery symbol is displayed in the center of the screen (between the “State” and the “Temperature” output).

The status field “State” can show the following values:

- **Operational:** The CryptoServer is up and running.
- **Busy:** The CryptoServer performs longer calculations. This can also indicate a configuration problem or a hang-up of the CryptoServer.
- **Initialized:** The CryptoServer is in boot loader mode and cannot process normal requests.
- **Offline:** The CSXLAN daemon was not correctly started.
- **High temperature**: The maximum temperature has been exceeded and the operation is stopped.
- **Alarm**: The sensors have triggered, all sensitive data inside the CryptoServer were deleted. The alarm must be reset and all firmware modules of the CryptoServer must be re-loaded (see [CS2ADMIN]).
- **Defect:** The CryptoServer has detected problems with the memory during boot loading.

If there are more than one CryptoServer security modules installed inside the CryptoServer LAN, the display changes every 5 seconds to show alternately the states of the different CryptoServers.

3.4 Administration

There are two administration tasks:

1. Administration of the Hardware Security Module CryptoServer

a) This can be done via the administration menu on the display (see section 4.2).

b) This can be done with the tool *CSADM* (e. g. for administration per remote access, see [CS2ADMIN])

2. Configuration of the communication unit (LAN box)

This can be done in different ways:

a) The basic configuration like setting the IP-address and the default gateway address can be done with the help of the display and menu system.

b) For more complex configuration changes you have to edit the configuration file located in */etc/csxlan.conf* (see 5.2.6 for the format of the configuration file). To change this file you can:

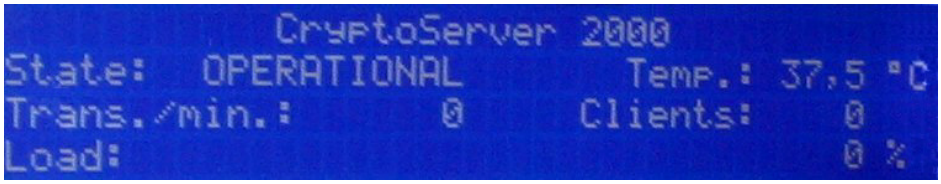
- Export this configuration to a floppy disk or USB memory stick using the administration menu on the display (see 4.2.1.6), change the file and later import the changed file back to the system (see 4.2.1.7).
- Remote copy the configuration file to your local computer using the CSADM tool, change the file and later copy the changed file back to the system (see [CS2ADMIN]).
- Connect a screen and keyboard to the system, log on to the operating system as "root" user and change the file */etc/csxlan.conf* using the vi editor.
- Log into the system using SSH and change the file */etc/csxlan.conf* using the vi editor.
  For security reasons, access via SSH is normally prohibited. You may enable one of the services to get remote access to the system for testing purposes (see 4.2.1.3 and 5.2.4).

To activate the new settings it is necessary to reboot the CryptoServer!

4 Display and Menu System

The CryptoServer LAN is equipped with a 4 x 40 display with 6 keys to control the CryptoServer LAN display administration menu[1]. This chapter describes the handling of the display and the functions of the menu.

Normally the display shows some information about the current status of the CryptoServer like temperature and the current workload:



After pressing the **OK** key, the menu system is shown. In the following sections the menu functions will be explained in detail. The status screen is displayed again after 60 seconds of inactivity. You can adjust this interval by changing the 'Timeout' configuration variable (see section 5.2.6)

[1] For meaning and usage of the six keys 'EXIT', 'OK', ↑, →, ↓ and ← see chapter 3.

4.1 Menu Overview

- LAN Box administration
  - Configuration
    - IP address
    - Default Gateway
    - DHCP
    - SSH Daemon
      - Configuration
      - Generate new keys
    - Card Reader
    - Export csxlan.conf
    - Import csxlan.conf
    - Keyboard layout
    - SNMP
  - State
    - Show Version
    - List clients
  - Diagnostic
    - Trace Level
    - Export Trace File
    - ifconfig
    - ping
  - Update
  - Selftest
  - Reboot
  - Shutdown
- CryptoServer administration
  - Admin – Module Functions
    - Get status of CryptoServer
    - List Files
    - Load File(s)
    - Delete File(s)
    - List currently active modules
    - Get time
    - Set time to system time
    - Set time manually
    - List users
    - Show bootlog

■	= First Level
▣	= Second Level
□	= Third Level

- □ Show memory information
- □ View alarm log
- □ View temperature log
- □ View time log

▣ Bootloader Functions

- □ Get status of CryptoServer
- □ Reset CryptoServer to bootloader
- □ Start OS (normal mode)
- □ Start OS (recovery mode)
- □ Clear CryptoServer
- □ Load base firmware modules
- □ Set RTC to system time
- □ Set RTC manually
- □ Reset Alarm
- □ Change Init Key
- □ View alarm log
- □ View temperature log
- □ View time log

▣ Generic Commands

- □ Restart CryptoServer
- □ Reset CryptoServer to bootloader
- □ Reset CryptoServer
- □ Show driver info

■ PIN-Pad applications

■	= First Level
▣	= Second Level
□	= Third Level

4.2 LAN Box Administration

With this group of menu functions the LAN box (communication unit) itself can be administrated. It contains no functionality for the administration of the security module CryptoServer.

4.2.1 Configuration

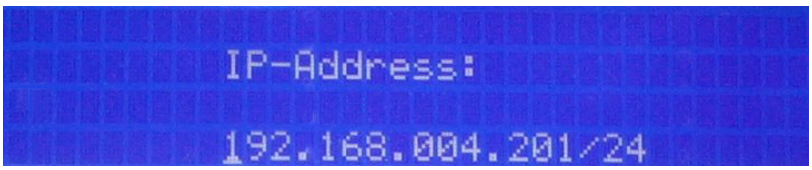
4.2.1.1 IP Address

Here you can configure the static IP address of the CryptoServer LAN.

It is also possible to perform an automatic network configuration via DHCP (see 4.2.1.3.).

If there are more than one Ethernet device installed inside the CryptoServer LAN, you have to choose the one to be configured. The different Ethernet devices are labeled ‘eth0’, ‘eth1’ etc. and are shown in a selection menu. ‘eth0’ is the 1000/100/10 MBit interface (labeled “Gbit Lan”) and ‘eth1’ is the 100/10 MBit interface (labeled “Lan”).

After choosing a device (or if there is only one Ethernet device), you will see in the display the following screen:



The 2 digits after the slash represent the number of consecutive ´1´ bits in the desired netmask. ‘24’ in this example is equivalent to the netmask ‘255.255.255.0’, whereas the netmask ‘255.255.252.0’ corresponds to ‘22’.

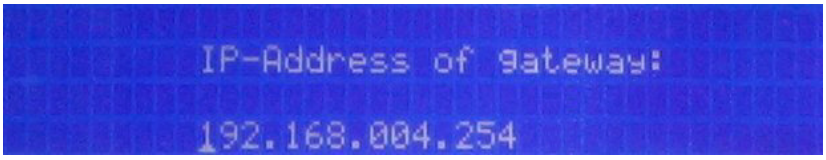
With the 🡐 and 🡒 keys you can select a digit. With the 🡑 and 🡓 keys you can adjust the value of a digit.

If you have entered the desired address press the “OK” key and confirm the following query. The new address is immediately valid.

Any previous DHCP configurations for the selected interface are lost after confirmation of the settings for a static IP address.

4.2.1.2 Default Gateway

With this menu point you can configure the default gateway. The display then starts the dialog as follows:



With the arrow keys you can change the IP address of the gateway (see also above). After pressing the “OK” key you must explicitly confirm the new address. The gateway is added immediately to the kernel routing table.

4.2.1.3 DHCP

The Dynamic Host Configuration Protocol (DHCP) is used to obtain a unique IP address and other parameters such as subnet mask, etc. After delivery the DHCP settings is turned off for every installed interface. If several network interfaces are used a submenu will be displayed where the appropriate interface can be chosen.

DHCP can be separately configured for every installed interface. After enabling the configuration a DHCP broadcast is sent to the network that is connected to the selected interface. The local DHCP server sends back the needed settings and the network is immediately configured with the received values. If there is no response from a local DHCP server within approximately one minute the configuration is interrupted.

If the DHCP settings are once enabled and are going to be disabled, the settings for the static IP configuration (see 4.2.1.1) becomes active immediately after confirmation of the menu..

4.2.1.4 SSH Daemon

The SSH Daemon is divided into two submenu entries for enabling / disabling the SSH Daemon and generation of new RSA and DSA keys.

Configuration

In the default configuration no remote access to the CryptoServer LAN via SSH (secure shell) is allowed. But in case you need the possibility of remote administration for the system, you can enable the SSH Daemon.

After selecting this function the SSH Daemon can be set to “enabled” or “disabled”. If it is enabled, additionally an IP address and a netmask have to be entered that define the clients which are allowed to use the SSH Daemon. The IP address and mask have to be entered in the form **nnn.nnn.nnn.nnn/mm**. The two digits after the slash represent the number of consecutive ´1´ bits in the desired net mask. Example:

Setting	Clients that can use the SSH Daemon
192.168.004.077/32	Client with IP address 192.168.4.77
192.168.004.000/24	All clients with IP addresses that start with 192.168.4
192.168.000.000/16	All clients with IP addresses that start with 192.168
000.000.000.000/00	All clients.

Generate new keys

The SSH protocol uses a public key mechanism to authenticate the server against the client. These mechanisms are RSA for SSH-protocol version 1 and RSA or DSA for SSH-protocol version 2. The keys are pre-installed during the installation at Utimaco. It is recommended to generate own keys if SSH is used in the customer network.

To generate new RSA and DSA key pairs this menu must be used. The keys are generated locally and are stored on the local flash disk of the CSLAN. Every previously generated keys are moved to the directory ‘/etc/ssh/bak’ on the CSLAN for possible manually restore. Every new generation of key pairs will override the keys in the ‘bak’ directory.

4.2.1.5 Card Reader

Many functions require a PIN-Pad connected either to the CryptoServer LAN’s COM port or directly to the COM port of the CryptoServer. There are different PIN-Pad types that can be used. The correct PIN-Pad type has to be configured in order to use it.

This function first shows the PIN-Pad type that is currently configured for the CryptoServer LAN and the CryptoServer. Then the configuration can be set to one of the following PIN-Pads:

- Ingenico SafePad
- Omnikey CardMan 8630
- ACR 80
- Autodetect (between Ingenico SafePad and ACR 80)

4.2.1.6 Export csxlan.conf

The current configuration file ‘csxlan.conf‘ for the CSXLAN daemon is copied either to a floppy disk or to an USB memory stick.

4.2.1.7 Import csxlan.conf

A new configuration file ‘csxlan.conf’ is read from a DOS formatted floppy disk or from an USB memory stick. The imported file is checked for syntax errors. The new configuration is not used until the CSXLAN daemon is restarted. You can use the ‘LAN Box administration > Reboot’ menu item to restart the system.

4.2.1.8 Keyboard layout

To change the keyboard layout of the keyboard that is directly connected to the CSLAN this menu must be entered. Choose the preferred layout with the display menu. The chosen layout will become active immediately after confirmation of the selected layout. The default layout after delivery is ‘german’.

4.2.1.9 SNMP

In this menu the SNMP support for the CryptoServer LAN can be enabled or disabled. For further information about SNMP read section 5.2.8.

4.2.2 State

4.2.2.1 Show Version

This menu item prints the version numbers of all installed software packages. At least two packages are installed:

Package	Description
base	base system, contains operating system
cs2	CryptoServer software: LAN daemon, menu system, driver, etc.

4.2.2.2 List Clients

A list of all active TCP connections is printed. Each line of this list consists of the following elements:

- port number of the server
- IP address of the client
- port number of the client

4.2.3 Diagnostic

4.2.3.1 Trace Level

The CryptoServer LAN knows about several log levels. Each level is independent from the others (the Verbose level does not include the Info level).

- **Info:** informational messages like connection establishment, termination and the execution of functions of the control module are written to the log file.
- **Verbose**: details about the state machine are logged.
- **Packet data:** the content of request and reply packets is logged (max. 256 bytes per packet).

In this menu all of the above levels can be selected by pressing the “OK” key. To confirm the selection choose the ‘accept’ menu item.

4.2.3.2 Export Trace File

Here you can export the log files to a floppy disk or USB memory stick. All log files found in the directory ‘/var/log’ and starts with csxlan.log are concatenated and compressed.

The default settings of 1MB for MaxSize and 3 for MaxFiles result in 5 Megabytes total space. These settings can be done in the configuration file /etc/ulogd.conf.

4.2.3.3 ifconfig

The selection of this item displays the network interface status like IP configuration and interface statistics. If several network interfaces are used a submenu will be displayed where the appropriate interface can be chosen. You can scroll through the result with the arrow keys. “OK” and “EXIT” keys return you to the ‘Diagnostic’ sub menu.

4.2.3.4 ping

You are prompted for an IP address to send ICMP messages to. The network address is used as initial address. If you confirm this default, each host on your network will answer. This is the simplest way to check the network reachability.

By using the arrow keys you can change the address.

You can scroll within the results by using the arrow keys.

4.2.4 Update

Software updates from Utimaco are provided in form of compressed archives. Their filename is something like ‘pkgname-xx.yy.zz.pkg.tar.gz’ where xx.yy.zz is the version number of the update.

Updates are installed from a DOS formatted floppy disk or a USB memory stick, so if you received the update(s) by e-mail, you must copy them to such storage device. Essentially an update is a compressed tape archive which is unpacked and copied to the disk. If several updates are stored on the storage device, they are automatically installed one after the other in alphabetically order according to their version numbers. If errors are encountered during the installation of an update, the process is aborted. Error messages are shown on the display.

4.2.5 Self Test

The system starts a self test and displays the results of each step.

The following tests are performed:

- Ethernet adapter
- Networking
  - Reachability of the default gateway
  - Reachability of workstations in the local network
- CryptoServer
  - Driver for CryptoServer loaded?
  - CSXLAN daemon running?
  - Direct access via PCI and all configured listener ports are probed.

After the completion of the self test a report is printed to the display.

4.2.6 Reboot

Sometimes it is necessary to reboot the CryptoServer LAN to activate new settings (e. g. after a software update). After selecting this menu item, the CryptoServer LAN is rebooted.

4.2.7 Shutdown

After a confirmation the CryptoServer LAN system is shut down. After approximately 15 seconds the system will power off.

4.3 CryptoServer Administration

This chapter describes the implemented functions to administrate the CryptoServer hardware security module in the CryptoServer LAN environment.

If there is more than one CryptoServer security module installed inside the CryptoServer LAN, you have to choose one to be administrated after entering the function 'CryptoServer Administration'. For this, a menu appears on the display listing all installed CryptoServer devices. After selecting one device, all subsequent administration functions are executed on this device until you return to the main menu.

4.3.1 Introductory Remarks

4.3.1.1 Error Display

If an error is encountered while executing a command a comprehensive error message is shown on the display:

```
Last operation failed and returned
Error B0870001
     CryptoServer module ADM
     File Open Error
Press OK key to return
```

4.3.1.2 Functions That Need the Initialization Key

Some of the described functions (e. g. 'Load File(s)', 'Delete File(s)') need the *Initialization Key* to sign the command. The Initialization Key is stored on a smartcard and can be used via a PIN-Pad connected to the COM port below the display. If a function needs the key the following message will be displayed:

```
Please mind output of PinPad
```

If an error occurs while executing the PIN-Pad functions a detailed error message is printed to the display.

4.3.1.3 Representation of Text on the Display

Some of the described functions have a rather large output that exceeds the length and the width of the display. However this output can be accessed by using the cursor keys of the display. The output is always concluded by the text:

If this text is not to be seen on the display a part of the text is still hidden.

4.3.2 Admin – Module Functions

At the end of the boot process the CryptoServer's operating system module SMOS is started by the bootloader.

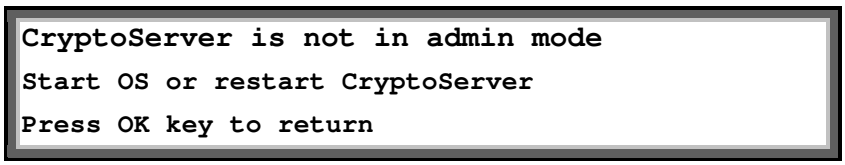
After its own initialization SMOS starts all other firmware modules found inside the CryptoServer. The operating system SMOS runs parallel to other active firmware modules and does not terminate until the CryptoServer is shutdown. To reach the *administration mod*e the following four essential base firmware modules must be loaded and started:

Name	Functionality
SMOS	**Small Multitasking Operating System** Operating system of the CryptoServer, responsible for receiving and sending byte streams via the physical interfaces of the CryptoServer, task scheduling, access to the CryptoServer hardware and memory.
UTIL	**Utilities** Provides RTC access and random number generation (using the CryptoServer hardware noise generator and a pseudo random number generator) to other firmware modules
CMDS	**Command Scheduler** External interface: administration of user database (user name, given permission status, authentication method) Internal functionality: responsible for distribution of received (external) commands to the appropriate firmware modules, checks authentication of received commands
ADM	**Administration** External interface for loading, replacing and deleting firmware modules, gives RTC access and status and other information

For more information on CryptoServer firmware modules see [CS2ADMIN]. After the initialization of these base modules the CryptoServer is administrable, i. e. in particular, all functions of the menu 'Admin – Module Functions' are available.

Before the submenu for this group of functions is accessed, a 'Get Status' command is sent to the CryptoServer. If the CryptoServer is not accessible at this time, there will be no reaction of the display or to any key that is pressed. After the CryptoServer timeout expired an error-message is shown on the display.

Most functions in this menu require the CryptoServer to be in administration mode. Each function of the Admin Menu that needs the SMOS to be running tests if the CryptoServer is in administration mode. If this is not the case an error message is shown on the display:


CryptoServer is not in admin mode
Start OS or restart CryptoServer
Press OK key to return

4.3.2.1 Get Status of CryptoServer

This function shows the status of the CryptoServer. In the following a typical output of the 'Get Status' command (if the CryptoServer is in administration mode) is shown:

```
CryptoServer started-Admin Modules loaded
state    = 00000005 OPERATIONAL
temp     = 37,0 [°C]
alarm    = OFF
BL vers. = 01000500
HW vers. = 01000200
UID: 09000006 80f51e01
adm1: UTIMACO CS000063
adm2: Bernd CS000063
adm3:
Press OK key to return
```

See [CS2ADMIN] for a detailed description of the status output.

The CryptoServer can be in administration mode or in bootloader mode to execute this function.

The entries exceed the width of the display. To read the whole entry the cursor-keys must be used to browse through the text.

4.3.2.2 List Files

This command lists all files that are stored in the CryptoServer.

The file system of the CryptoServer is divided into the FLASH- and the SYS-directory:

The SYS-directory usually holds the back-up copies of the base firmware modules, public RSA keys that are needed for administration, the boot loader's configuration file and some log files.

In the FLASH-directory the regular set of all firmware modules is stored, in particular all customer specific modules, and any other kind of application specific data (databases, configuration or log files, ...).

In operational mode the files in the SYS-directory can neither be deleted nor updated. Only the files in the FLASH-directory can be deleted or updated.

Executing 'List files' the following submenu is printed on the display:

```
Please select directory
Files in FLASH – directory
All files
Files in SYS – directory
```

After selecting one of the entries and pressing the "OK" key a listing of the files of the chosen directory is printed on the display:

```
Files in directory SYS
Prod.KeyRsaPub              168    –
Init.KeyRsaPub              168    –
MdlSg.KeyRsaPub             168    –
bl.ini                       72    –
smos.mtc  1.0.3.9 0x00    95452  SMOS  SMOS
cmds.mtc  1.0.6.0 0x83    66324  CMDS  Command Scheduler
util.mtc  1.0.6.0 0x86    40860  UTIL  Utility Module
adm.mtc   1.0.2.0 0x87    34508  ADM   Admin Module
temp.log                     40    –
alarm.log                    24    –
    10 files in SYS 234112 bytes used
```

```
Files in directory FLASH
user.db                      20    -
alarm.log                    24    -
temp.log                     40    -
smos.mtc  1.0.3.9 0x00   95452  SMOS  SMOS
cmds.mtc  1.0.6.0 0x83   66324  CMDS  Command Scheduler
util.mtc  1.0.6.0 0x86   40860  UTIL  Utility Module
adm.mtc   1.0.2.0 0x87   34508  ADM   Admin Module
vdes.mtc  1.0.0.3 0x81   24092  VDES  DES Module
lna.mtc   1.0.3.0 0x8e   70508  LNA   LNA
vrsa.mtc  1.0.4.0 0x84   74760  VRSA  RSA Module
hash.mtc  1.0.1.0 0x89   51676  HASH  Hash Module
db.mtc    1.0.1.1 0x88   38300  DB    Database module
pp.mtc    1.0.5.0 0x82   42692  PP    PinPad Driver
sc.mtc    1.0.0.5 0x85   31164  SC    Smartcard Module
asn1.mtc  1.0.1.2 0x91   14660  ASN1  Asn1 Module
    15 files in FLASH 575436 bytes used

Press OK key to return
```

Exemplary description of the entry for the CMDS firmware module:

`cmds.mtc 1.0.6.0 0x83 66324 CMDS Command Scheduler`

cmds.mtc	file name of the module as stored on the CryptoServer
1.0.6.0	version number of this module
0x83	function code FC of the module (module ID)
66324	size of the module (in bytes)
CMDS	abbreviation of the module name
Command Scheduler	long name or description of the module

The entries exceed the width of the display. To read the whole entry the cursor-keys must be used to browse through the text.

4.3.2.3 Load File(s)

This function loads or updates files to the FLASH-directory from a floppy disk or USB memory stick.

Requirements for execution of this command:

- Installed PIN-Pad on COM interface below display.
- DOS formatted floppy disk or USB memory stick with files to be loaded (in the root directory).
- Smartcard with Initialization Key.

On execution the following text is printed on the display:

```
Please insert USB stick / floppy and press OK
key
```

After inserting the storage device in the CryptoServer LAN and pressing the **OK** key, the names of the files in the root directory of the storage device are displayed:[2]

```
Please select files to load
➔*adm.mtc
  cmds.mtc
  smos.mtc
```

With the keys ↑ and ↓ the directory can be searched. With the key ➔ files can be selected or unselected. In front of selected files a * is printed. Pressing the **OK** key will start the loading process of the files into the CryptoServer. As these commands must be signed with the Initialization Key, the PIN-Pad dialog is executed. While loading a file the last loaded file is displayed:

```
Please wait...
adm.mtc loaded
```

Finally a summary of all loaded files is displayed:

[2] For loading files from archives see below.

```
The following files were loaded
adm.mtc
smos.mtc
Press OK key to return
```

If any error occurs during execution of the command, the loading operation is interrupted even with files pending to be loaded. The file that produced the error is displayed:

```
Error loading file
adm.mtc
Press OK key to return
```

After confirmation of this error message a detailed error description is displayed.

Load Files from Archives

It is also possible to load several files contained in an archive like TAR or ZIP.

These files and/or modules can be packed into one archive which can be selected over the display menu. After confirming the load procedure this archive is unpacked to the local CryptoServer LAN system and each file contained in the archive will be loaded into the security module CryptoServer after successful authentication.

Archives are detected on their filename extension. At the moment the following extensions are supported:

.tar - for uncompressed TAR archives

.tar.gz - for gzip compressed TAR archives

.tgz - for gzip compressed TAR archives

.zip - for zip compressed archives (like e.g. WinZip)

If one of the selected files has such a file name extension it will be treated like an archive and its content will be loaded into the CryptoServer.

The extensions are case sensitive and must be written in lower case!

Any subdirectories contained in the archive will also be checked for loadable files or modules.

4.3.2.4 Delete File(s)

This function deletes files in the FLASH-directory.

Requirements for execution of this command:

- Installed PIN-Pad on COM interface below display.
- Smartcard with Initialization Key.

On execution the names of the files in the FLASH-directory are displayed:

```
Please select files to delete
➔*adm.mtc
  cmds.mtc
  smos.mtc
```

With the keys ↑ and ↓ the directory can be searched. The ➔ key can select or deselect files. In front of selected files a * is printed. Pressing the “OK” key will start the deleting process of the files. As this command must be signed, the PIN-Pad dialog is executed.

While deleting the selected files the last deleted file is displayed:

```
Please wait...
adm.mtc deleted
```

Finally a summary of the deleted files is displayed:

```
The following files were deleted
adm.mtc
smos.mtc
cmds.mtc
```

If any error occurs during execution of the command, the deleting operation is interrupted even with files pending to be deleted. The file that produced the error is displayed:

```
Error deleting file
adm.mtc
Press OK key to return
```

After confirmation of this error message a detailed error description is displayed.

4.3.2.5 List Currently Active Modules

This function lists all firmware modules that are loaded by the operating system SMOS while booting the CryptoServer. This modules register with the SMOS after start-up.

```
  0 SMOS                1.0.3.9 OK
 83 CMDS                1.0.6.0 OK
 86 UTIL                1.0.6.0 OK
 87 ADM                 1.0.2.0 OK
 81 VDES                1.0.0.3 OK
 8E LNA                 1.0.3.0 OK
 84 VRSA                1.0.4.0 OK
Press OK key to return
```

Exemplary description of the entry for the firmware module CMDS:

```
83  CMDS     1.0.6.0  OK
```

83	function code FC of the module (module ID)
CMDS	abbreviation of the module name
1.0.6.0	version number of this module
OK	status of this module

A module's status can be one of the following values:

Output	Description
INTERNAL	The module has finished its internal initialization.
DEP_OK	The module has successfully completed the check of dependencies on other modules.
OK	The initialization of the module is complete. Possible calls to services from other modules are done successfully.
FAILED	The initialization of the module failed. Services provided by this module are not available.

If one of the module's states is not "OK" the command "Show bootlog" gives a detailed description of the reason why the module did not start properly.

4.3.2.6 Get Time

'Get Time' shows the time of the CryptoServer:

```
Date and time
2005.07.26 09:34:37.428
Press OK key to return
```

The time format is:

year.month.day hour:minute:second.milliseconds

4.3.2.7 Set Time to System Time

This function sets the CryptoServer's internal time to the system time of the communication unit CryptoServer LAN.

Requirements for execution of this command:

- Installed PIN-Pad on COM interface below display.
- Smartcard with Initialization Key.

Since this command must be signed by the Initialization Key, the PIN-Pad dialog is executed. If the operation is finished successfully the following text is printed to the display:

```
Last operation succeeded
Press OK key to return
```

This function sets the real time clock of the CryptoServer, not the clock of the PC!

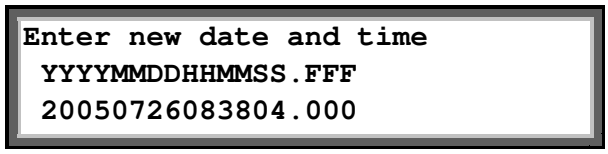
4.3.2.8 Set Time Manually

With this command the CryptoServer's time can be set manually.

Requirements for execution of this command:

- Installed PIN-Pad on COM interface below display.
- Smartcard with Initialization Key.

On execution the following dialog is printed to the display:

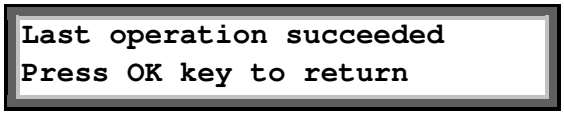


Abbreviations:

Y	Year
M	Month
D	Day
H	Hour
M	Minute
S	Second
F	Fraction of Second (Millisecond)

The underscore shows which digit is active and can be changed with the ↑ and ↓ keys. After changing the value of the date-time-string, pressing the key "OK" will start the PIN-Pad dialog (for signing the command with the Initialization Key), whereas pressing the "EXIT" key will abort.

If the operation is finished successfully the following text is printed to the display:



This function sets the real time clock (RTC) of the CryptoServer, not the clock of the PC!

4.3.2.9 List Users

This function lists the users that are registered in the user data-base of the CryptoServer.

```
Name       Permission   Mechanism   Flags
ADMIN       22000000    RSA sign    no_login + sma
Press OK key to return
```

In the following table the fields of the output are described:

Field	Description
Name:	Here are list the 8 characters long User names.
Permission	Permission mask. Consists of 8 nibbles, each can be in the range 0_H to 3_H The first nibble corresponds to the user group 7, the last to user group 0. The value of the nibble is the authentication level of the user within the corresponding user group. Example: '00100010' means, the user has authentication level 1 in groups 1 and 5.
Mechanism	Authentication mechanism: ■ RSA sign ■ Clear password ■ SHA-1 password ■ Smartcard signature
Flags	**no_login:** only single command authentication is allowed for this user (whereas static login is not allowed!). **allow_login**: static login is allowed for this user. **sm:** user is allowed to establish secure messaging sessions (for secure data transfer) **sma:** user can establish secure messaging sessions, but only with authentication.

See [CS2ADMIN] for more detailed information about user's permissions, the various authentication mechanisms, static login or secure messaging.

4.3.2.10 Show Bootlog

During start-up of the CryptoServer a boot log file is created by the operating system SMOS. This command shows the boot log file on the display.

```
SMOS Ver. 1.0.3.9 started

module    0x83    (CMDS)    initialized successfully
module    0x8e    (LNA)     initialized successfully
module    0x89    (HASH)    initialized successfully
module    0x82    (PP)      initialized successfully
module    0x86    (UTIL)    initialized successfully
module    0x81    (VDES     initialized successfully
module    0x87    (ADM)     initialized successfully
module    0x88    (DB)      initialized successfully
module    0x85    (SC)      initialized successfully
module    0x84    (VRSA)    initialized successfully
module    0x91    (ASN1)    initialized successfully

Press OK key to return
```

4.3.2.11 Show Memory Information

This function shows the storage occupancy of the SYS-, FLASH-directory and of the NVRAM.

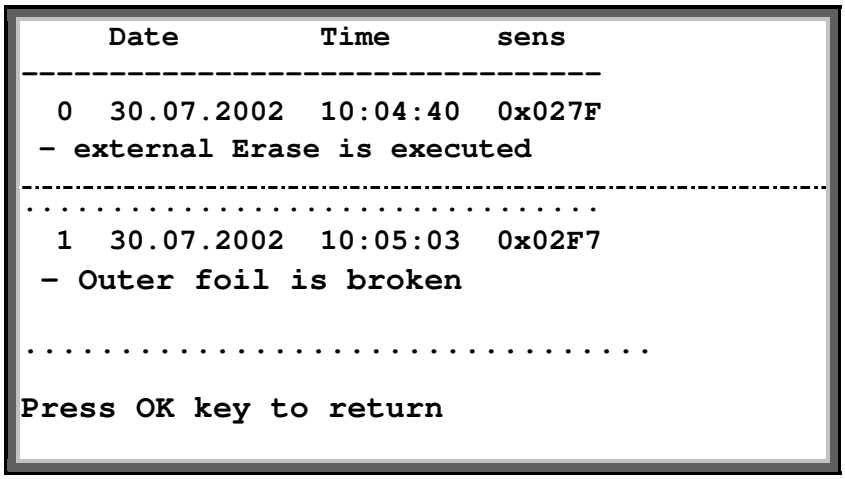
```
SYS
Total        = 377600
Used         = 236944
Free         = 140656
-------------------------------------------------------
Available    = 140656

FLASH
Total        = 14968064
Used         = 637200
Free         = 14330864
Available    = 14330864

NVRAM
Total        = 468224
Used         = 472
Free         = 467752
Available    = 468224

Press OK key to return
```

Here means

■ Total:	overall capacity
■ Used:	allocated storage space
■ Free:	unorganized capacity
■ Available:	storage space that can be used

4.3.2.12 View Alarm Log

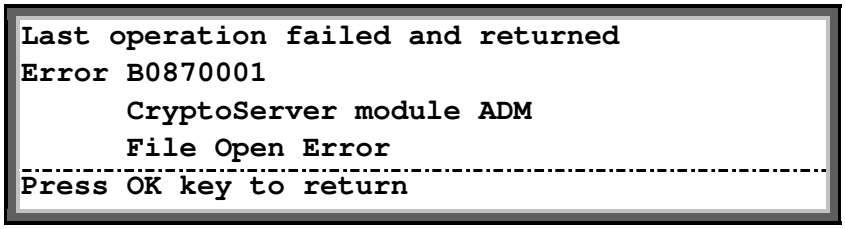
Whenever an alarm occurs, date, time and alarm cause are recorded to the file ‘alarm.log’. This alarm log is shown on the display:



Possible causes for alarms are:

- Temperature too low
- Temperature too high
- Inner foil is broken
- Outer foil is broken
- Manual Erase is executed
- Power is too high
- Power is too low
- Invalid Master Key (reason for this is usually an empty battery)
- External Erase is executed

If no alarm log was generated (for instance because no alarm has occurred yet) the following error-notification will be displayed:



4.3.2.13 View Temperature Log

If the CryptoServer's internal temperature is less than 5°C or more than 58°C the CryptoServer is powered down. An entry is then written to the file 'temp.log'.

This file can be displayed with this function:

```
     Date         Time       Temp
-------------------------------
  0  30.07.2002   10:04:40   58,5
  1  30.07.2002   10:05:03   58,0
Temperature limits
Critical temperature: 58,0
Low temperature: 5.0
High temperature: 58,0
Press OK key to return
```

If no temperature log was generated yet, the following error-notification will be displayed:

```
Last operation failed and returned
Error B0870001
      CryptoServer module ADM
      File Open Error
Press OK key to return
```

4.3.2.14 View Time Log

Whenever the internal clock of the CryptoServer is set, an entry is written into the time log file ‘time.log’. Each entry contains the “old” date and time before changing the clock, and the new date and time the clock is set to.

The Time Log File can be displayed with this function:

```
    Date        Time      new Date    new Time
-----------------------------------------------
  0 19.04.2004 14:16:13 21.04.2004 17:20:00
  1 20.05.2004 09:32:07 20.05.2004 09:34:10
-----------------------------------------------
Press OK key to return
```

If no time log was generated yet, the following error-notification will be displayed:

```
Last operation failed and returned
Error B0870001
      CryptoServer module ADM
      File Open Error
-----------------------------------------------
Press OK key to return
```

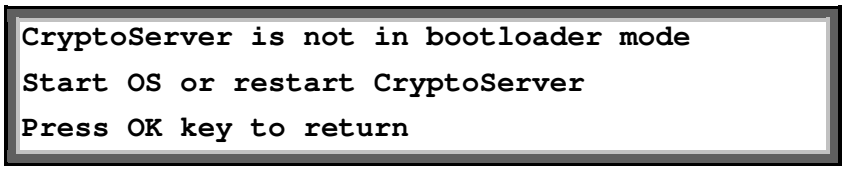
4.3.3 Bootloader Functions

This command group addresses CryptoServer's boot loader, which offers functionality for base (initial) administration.

The bootloader is the first program code that is executed inside the CryptoServer after start-up, before the operating system module SMOS is called and is up and running. As long as the bootloader is active (and thus SMOS is not running yet), the CryptoServer is said to be in *bootloader mode*. Some commands can only be executed in bootloader mode.

Before the submenu for this group of functions 'Bootloader Functions' is accessed a 'Get Status' command is sent to the CryptoServer. If the CryptoServer is not accessible at this time, there will be no reaction of the display or to any key that is pressed. After the CryptoServer's timeout expired an error-message is shown on the display.

Most functions in this menu require the CryptoServer mandatory to be in bootloader mode, i. e. the bootloader must be running. Each function of this menu that needs the bootloader to be running tests if the CryptoServer is in bootloader mode. If this is not the case an error message is shown on the display:



4.3.3.1 Get Status of CryptoServer

This function shows the status of the CryptoServer. Following a typical output of the 'Get Status' command (if the CryptoServer is in bootloader mode) is shown:

```
CryptoServer is in Bootloader Mode
State     = 00000004 INITIALIZED
Temp      = 37,0 [°C]
Alarm     = OFF
-------------------------------------------------------
BL vers. = 01000500
HW vers. = 01000200
UID: 09000006 80f51e01
Adm1: UTIMACO CS000063
Adm2: Bernd CS000063
Adm3:
Press OK key to return
```

See [CS2ADMIN] for a detailed description of the status output.

The CryptoServer can both be in administration mode or in bootloader mode to execute this function.

The entries exceed the width of the display. To read the whole entry the cursor-keys must be used to browse through the text.

4.3.3.2 Reset CryptoServer to Bootloader

The CryptoServer gets a reset command immediately followed by a dummy command, it therefore stays in bootloader mode. The bootloader remains active until a 'Start OS' command is sent.

4.3.3.3 Start OS (Normal Mode)

The ‘Start OS’-command loads the file ‘smos.mtc’ from the FLASH-directory and starts the operating system SMOS of the CryptoServer. SMOS will then load and start all other firmware modules from the FLASH-directory.

This is the preferable way to start the operating system and thus to let the CryptoServer enter the administration mode. Only if the CryptoServer is not working properly after an update or inadvertent deletion of crucial firmware modules it is recommended to use the ‘Start OS (Recovery Mode)’ function.

4.3.3.4 Start OS (Recovery Mode)

This command loads the file ‘smos.mtc’ from the SYS-directory and starts the operating system SMOS of the CryptoServer. SMOS will then load and start all other firmware modules that are found in the SYS-directory, i. e. the back-up copies of the base firmware modules.

It may be necessary to use this command if it is not longer possible to start the regular set of firmware modules (which are stored in the working directory \FLASH), for example because one or more indispensable modules (SMOS, CMDS, ADM, UTIL) have been deleted by mistake:

The files in the SYS-directory can not be deleted or updated except a ‘Clear CryptoServer’ command was performed beforehand. Thus the stable firmware modules that are loaded into the SYS-directory before delivery (back-up copies of the base firmware modules) assure a safe administration of the CryptoServer even in case that some base firmware modules in the FLASH-directory are missing or defect.

4.3.3.5 Clear CryptoServer

The ‘Clear CryptoServer’ command clears the CryptoServer to the state “INITIALIZED”. This means that the operating system SMOS and all other firmware modules are deleted in both the FLASH-directory and in the SYS-directory, as well as all customer’s data.

This clear function must be executed before or after changing the Initialization Key.

Requirements for execution of this command:

- Installed PIN-Pad on COM interface below display.
- Smartcard with Initialization Key.

Since this command must be signed by the Initialization Key, the PIN-Pad dialog is executed after the function is called.

All data inside the CryptoServer will be deleted s. t. the CryptoServer enters the ‘initialized’ state. In this state only the following data remain in the CryptoServer:

- Bootloader code
- Production Key $\mathbf{K_{PROD\text{-}PUB}}$
- Module-Signature Key $\mathbf{K_{MDLSIG\text{-}PUB}}$
- Initialization Key $\mathbf{K_{INIT\text{-}PUB}}$
- Alarm Log
- Temperature Log

4.3.3.6 Load Base Firmware Modules

This function loads files to the SYS- and to the FLASH-directory from a floppy disk or USB memory stick.

Since storage space in the SYS-directory is limited, this function should only be used to load the most important base firmware modules (SMOS, ADM, CDMS, UTIL), needed to get a minimal working system. Loading further firmware modules should be done with the 'Load File(s)' command in administration mode (see 4.3.2.3 and 4.3.3.3).

Requirements for execution of this command:

- Installed PIN-Pad on COM interface below display.
- DOS formatted floppy disk or USB memory stick with files to be loaded (files must be stored in the root directory).
- Smartcard with Initialization Key.

On execution the following text is printed on the display:

After inserting the storage device in the CryptoServer LAN and pressing the **OK** key, the names of the files in the root directory of the storage device are displayed.

With the arrow keys ↑ and ↓ the directory can be searched. With the key ➔ files can be selected or unselected. In front of selected files a * is printed. Pressing the **OK** key will start the loading process of the files to the CryptoServer. As this command must be signed with the Initialization key, the PIN-Pad dialog is executed now.

While loading a file the last loaded file is displayed.

Finally a summary of the loaded files is displayed:

```
The following files were loaded
adm.mtc
smos.mtc
Press OK key to return
```

If any error occurs during execution of the command, the loading operation is interrupted even with files pending to be loaded. The file that produced the error is displayed:

```
Error loading file
adm.mtc
Press OK key to return
```

After confirmation of this error message a detailed error description is displayed.

In case of a full SYS-directory the attempt to load a file will result in an error message. Nevertheless the file will be loaded to the FLASH-directory.

4.3.3.7 Set RTC to System Time

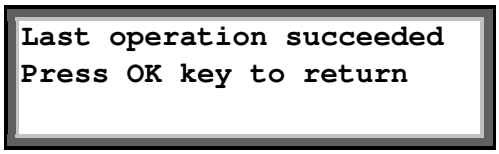
This function sets the real time clock (RTC) of the CryptoServer to the system time of the CryptoServer LAN.

Requirements for execution of this command:

- Installed PIN-Pad on COM interface below display.
- Smartcard with Initialization Key.

Since this command must be signed by the Initialization Key, the PIN-Pad dialog is executed after the function is called.

If the operation is finished successfully the following text is shown on the display:



This function sets the real time clock of the CryptoServer, not the clock of the PC!

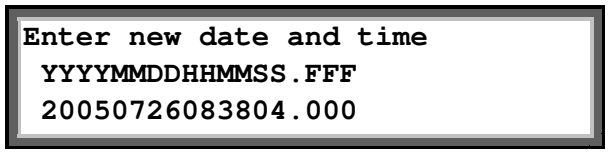
4.3.3.8 Set RTC Manually

With this command the CryptoServer's time (real time clock, RTC) can be set manually.

Requirements for execution of this command:

- Installed PIN-Pad on COM interface below display.
- Smartcard with Initialization Key.

On execution the following dialog is printed to the display:

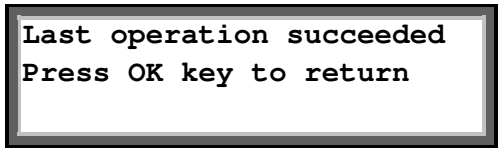


Abbreviations:

Y	Year
M	Month
D	Day
H	Hour
M	Minute
S	Second
F	Fraction of Second (Millisecond)

The underscore shows which digit is active and can be changed with the ↑ and ↓ keys. After changing the value of the date-time-string, pressing the "OK" key will start the PIN-Pad dialog (for signing the command with the Initialization Key), whereas pressing the "EXIT" key will abort.

If the operation is finished successfully the following text is shown on the display:



This function sets the real time clock (RTC) of the CryptoServer, not the clock of the PC!

4.3.3.9 Reset Alarm

This command manually resets the *alarm state* if the physical reason for the alarm is no longer present:

If an alarm occurs on the CryptoServer, nearly all data will be erased and an entry into the log file ‘alarm.log’ will be made. The occurrence of the alarm will be stored as a special alarm state. Even if the physical reason for an alarm has been removed (e. g. a low battery was changed, or the temperature is back to the allowed range), the alarm state is still active and has to be manually reset by the user: In this alarm state only the ‘Get Status’ and ‘Reset Alarm’ commands can be executed. This approach assures that no alarm goes unnoticed. The ‘Get Status’ command shows detailed information about the actual alarm.

If the physical reason for the alarm is still pending (e. g. the foil is broken), any attempt to reset the alarm state will fail.

Requirements for execution of this command:

- Installed PIN-Pad on COM interface below display.
- Smartcard with Initialization Key.

Since this command must be signed by the Initialization Key, the PIN-Pad dialog is executed after the function is called.

If the operation is finished successfully the following text is printed to the display:

```
Last operation succeeded
Press OK key to return
```

Possible causes for an alarm are:

- Temperature too low
- Temperature too high
- Inner foil is broken
- Outer foil is broken
- Manual Erase is executed
- Power is too high
- Power is too low
- Invalid Master Key (reason for this is usually an empty battery)
- External Erase is executed

4.3.3.10 Change Initialization Key

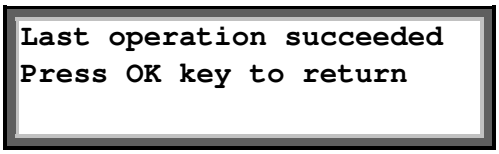
This function changes the public part of the Initialization Key of the CryptoServer. The new key is stored as 'Init.KeyRsaPub' in the SYS directory, thus overwriting the old Initialization Key.

Requirements for execution of this command:

- Installed PIN-Pad on COM interface below display.
- Smartcard with old Initialization Key.
- Smartcard with new Initialization Key.

After calling the function a PIN-Pad dialog will be started. First the smartcard with the new Initialization Key will be asked for. The public key will be read from this smartcard. Then the smartcard with the old Initialization Key is needed to sign the command.

If the operation is finished successfully the following text is shown on the display:



4.3.3.11 View Alarm Log

This function is described in section 4.3.2.12.

4.3.3.12 View Temperature Log

This function is described in section 4.3.2.13.

4.3.3.13 View Time Log

This function is described in section 4.3.2.14.

4.3.4 Generic Commands

The generic commands in this section are directed to the PCI driver of the CryptoServer.

To differentiate the commands ‘Restart’ and ‘Reset’ a basic knowledge of the boot process is necessary:

After any reset (power-up or hardware reset) the CryptoServer starts with the boot loader firmware code. The boot loader will do the first necessary start procedures: The internal RAM is cleared, several self-tests are executed and the alarm state and temperature is checked. After approximately 3 seconds commands of the host can be received, for a period of 10 seconds.

- If the bootloader receives any command within this defined time window of 5 seconds, the CryptoServer remains in bootloader mode until a ‘Start OS’ command is received.
- Otherwise, if no command is received in the time window of 5 seconds, the operating system SMOS will be started automatically and the CryptoServer enters the administration mode (i. e. it is operational).

This process of entering the administration mode (getting operational) after a reset can be sped up by using the ‘Restart CryptoServer’ command (see below). The recommended method to reset the CryptoServer is therefore to use the ‘Restart’ command.

The CryptoServer may be in any mode (bootloader mode or administration mode) to execute the driver commands. No authentication towards the CryptoServer (e. g. with the Initialization Key) is required.

4.3.4.1 Restart CryptoServer

The CryptoServer gets a reset command immediately followed by a ‘Start OS’ command. Thus the 5 seconds time window (in which bootloader commands could be received) is skipped and the operating system module is started.

After the successful completion of the command the CryptoServer is immediately available in administration mode. In particular the commands described in 4.3.2 are available.

4.3.4.2 Reset CryptoServer to Bootloader

The CryptoServer gets a reset command immediately followed by a dummy command and therefore stays in bootloader mode. In particular the commands described in 4.3.3 are available now.

The bootloader mode remains active until a ‘Start OS’ command is sent.

4.3.4.3 Reset CryptoServer

This command executes a hardware reset of the CryptoServer at PCI-level. All PCI-registers are cleared.

After the reset, the CryptoServer is able to receive bootloader commands during the above described time window, before the operating system module is started. The CryptoServer will stay in bootloader mode if any command is sent in the above mentioned time window of 7 seconds after reset.

The recommended method to reset the CryptoServer is to use the ‘Restart CryptoServer’ command.

4.3.4.4 Show Driver Info

Show driver info lists PCI-registers and PCI-driver status information of the CryptoServer. In the following table the output and the description are represented:

Output				Description
Vers	2.0.4			
Slot	0000:01:0c.0			
tx	idle			
rx	idle			
batt	ok			Battery status
txrt	0	0		
rxrt	0	0		
mbr	00100021	00000068		Mailbox register
bar0	00000000	07cf6000	00	Master Write Address0
bar0	00000000	00000068	08	MW Count Status0
bar0	00000000	00000000	10	Master Write Address1
bar0	00000000	00000000	18	MW Count Status1
bar0	00000000	000f0100	20	Misc
bar0	0e040000	00000000	28	Misc
bar0	00000008	00000000	30	I2O
bar0	00000000	00000000	38	Reserved
bar0	ffffffff	ffffffff	40	I2O
bar0	00000000	07d04000	48	Master Read Address0
bar0	00000000	00000010	50	Master Read Count Status0
bar0	00000000	00000000	58	Master Read Address1
bar0	00000000	00000000	60	Master Read Count Status1
bar0	08080808	08080808	68	Misc
bar0	00400020	00000068	70	PCI Outgoing MailBox Reg.
bar0	00100021	xxxxxxxx	78	PCI Incoming MailBox Reg.
bar0	0000feff	00000000	80	Misc
bar0	00050100	00000000	88	Misc
bar0	00000000	05000000	90	Misc
bar0	00000000	00000000	98	Reserved
bar0	00000000	00000000	a0	Reserved
bar0	00000000	00000000	a8	Reserved
bar0	00000000	00000000	b0	Reserved
bar0	00000000	00000000	b8	Reserved
bar0	00000000	00000000	c0	Reserved
bar0	00000000	00000000	c8	Reserved
bar0	00000000	00000001	d0	Misc
bar0	00000000	00000000	d8	Reserved
bar0	00000000	00000000	e0	Reserved
bar0	00000000	00000000	e8	Sw rst / hw rst
bar0	00000000	00000000	f0	Reserved
bar0	00000000	00000000	f8	Reserved
Press OK key to return				

4.3.5 PIN-Pad-Applications

One or more firmware modules can contain special functions to access the PIN-Pad immediately via the CryptoServer. These functions can for instance be used for loading a key directly into the CryptoServer. These functions are registered with the CMDS firmware module and can be accessed by the PIN-Pad applications menu.

The behavior of this 'PIN-Pad-Applications' function depends on the number of registered PIN-Pad applications. If only one PIN-Pad application is registered it will be executed immediately. More than one registered PIN-Pad applications result in a menu to choose the appropriate one:

```
Please select PinPad application
➔Enter working key over PIN-Pad
 Administrate CryptoTimeStamp
```

With the arrow keys ↑ and ↓ the wanted application can be selected. Pressing the **OK** key will start the selected PIN-Pad application.

In both cases the called function is displayed with the reminder to mind the PIN-Pad:

```
PinPad application to be executed
Enter working key over PIN-Pad
Please mind output of PinPad
```

The PIN-Pad dialog is starting now.

In the case that no PIN-Pad application is registered with the CMDS firmware module an error message is displayed:

```
No PinPad applications available
Press OK key to return
```

5 Advanced Configuration of the CryptoServer LAN

5.1 System Administration

The LINUX operating system is installed on the communication processor. If screen and keyboard are connected, then it is possible to log on to the system under the user name 'root' (the preconfigured password is 'utimaco').

The preconfigured root password is 'utimaco'.

We highly recommend you to change the password as soon as possible!

After a successful login you can change the password by executing the 'passwd' command. You are prompted for the new password. The password should consist of at least 6 characters and you should use one or more digits and punctuation marks.

5.2 Configuration Settings

The base configuration like setting the IP address and the default gateway address can be done with the help of the display and menu system (see section 4)

The configuration of the CSXLAN daemon and the display can be done either by editing the file '*/etc/csxlan.conf*', or by exporting this file to a floppy disk or USB stick and after changing the desired parts importing the file back to the CryptoServer LAN. See section 5.2.6 for the format of the configuration file and sections 4.2.1.6 and 4.2.1.7 for import/export of this file.

Possibilities to change the configuration files are described in section 5.2.6.

For more complex changes to the system it is necessary to connect a screen and keyboard and to log on to the operating system as 'root' user.

For security reasons, access via SSH is normally prohibited.

You should be familiar with the UNIX standard editor vi.

After you made your changes the system must be reinitialized. The simplest way to do this is to execute the following command:

```
# /etc/init.d/cs2 restart
```

5.2.1 Changing the IP Address

The CryptoServer LAN IP address (not the multicast address) can be altered either by using the menu system (see section 4 Display and Menu System, 4.2.1.1) or directly by changing the appropriate variables in the configuration file '/etc/sysconfig/networking'.

Example file '`/etc/sysconfig/networking`':

```
# Begin /etc/sysconfig/networking

NETCONFIG="_0"

NET_DEV_0=eth0
IP_ADDR_0=192.168.4.205
NETWORK_0=192.168.4.0
NETMASK_0=255.255.255.0
BRDCAST_0=192.168.4.255

GATEWAY=192.168.4.254

# End /etc/sysconfig/networking
```

After changing any values in the file'/etc/sysconfig/networking' the new settings must be activated with the command

`Cryptoserver:/#` */etc/init.d/network restart*

5.2.2 Special network configuration

The network interfaces of the CSLAN are configured using auto negotiation to detect proper speed and settings from the local Ethernet. Sometimes it is necessary to force the speed and duplex mode of the interfaces even when the different network components cannot work together. To alter the settings of each interface the command line tool *ethtool* is used. The detailed command options are described in the manual page of *ethtool* (*http://linuxcommand.org/man_pages/ethtool.8.html*).

The most common parameters are:

Parameter	Description
speed 10 | 100 | 1000	Set network interface speed to 10, 100 or 1000 Mbit/s
duplex half | full	Set network interface mode to half or full duplex
autoneg on | off	Turns the auto negotiation of the network interface on or off

The parameters for the network interfaces must be set at runtime when the operating system is started. The settings of the parameters are not permanent. So this configuration must be done after every reboot of the CSLAN.

To perform these settings automatically on system startup the configuration file *'/etc/sysconfig/networking'* can contain these settings in the configuration item *'ETHTOOL_X'* where *X* is the number of the according network interface.

Example file '`/etc/sysconfig/networking`':

```
# Begin /etc/sysconfig/networking

NETCONFIG="_0"

NET_DEV_0=eth0
IP_ADDR_0=192.168.4.205
NETWORK_0=192.168.4.0
NETMASK_0=255.255.255.0
BRDCAST_0=192.168.4.255
ETHTOOL_0="speed 100  duplex half  autoneg off"

GATEWAY=192.168.4.254

# End /etc/sysconfig/networking
```

After changing any values in the file'/etc/sysconfig/networking' the new settings must be activated with the command

`Cryptoserver:/#` */etc/init.d/network restart*

5.2.3 Setting a Default Gateway

The CryptoServer LAN default gateway can be altered either by using the menu system (see section 4 Display and Menu System, 4.2.1.2) or directly by executing the command `route_config.sh` with the new gateway address as argument.

Example: Setting '192.168.4.254' as the default gateway address:

```
Cryptoserver:/# route_config.sh 192.168.4.254
Shutting down network device eth0.              [  OK  ]
Setting up network device eth0.                 [  OK  ]
Setting up routing                              [  OK  ]
Disabling IP-Spoofing.
Disabling Syn flood attack by sending cookies.
Suppressing icmp echo broadcasts
Suppressing redirects
Suppressing source routing
```

You can display the current route settings with the command `route`:

```
CryptoServer:/# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.4.0     *               255.255.255.0   U     0      0        0 eth0
default         192.168.4.254   0.0.0.0         UG    0      0        0 eth0
```

If you want to configure advanced routing settings this can be done in the configuration file /etc/route.conf. The configuration is explained inside this file. To activate this settings the network has to be restarted.

`Cryptoserver:/#` /etc/init.d/network restart

5.2.4 Enabling SSH

In the default configuration there are no services running on a CryptoServer LAN other than those listed in the configuration file `/etc/csxlan.conf`.

If you need remote administration for the system, you can activate the Secure Shell daemon (SSH) by editing the file '/etc/sysconfig/ssh'. To enable the SSH service, set the configuration variable `START_SSHD` to 'yes'.

Additionally you must enable the selected service in the file `/etc/hosts.allow`. This file lists the access permissions of clients for certain services. Each entry has the following format:

```
daemon_list: client_list
```

`daemon_list` is a list of one or more process names, `client_list` is a list of several host addresses or host names. If a host address ends with a dot, only the leading portion of the address is checked.

For instance the line

```
sshd: 192.168.1.1 192.168.4.
```

grants all clients from the net 192.168.4 and the client 192.168.1.1 access to the service Secure Shell.

The activation of the SSH service and also the settings for the file '/etc/hosts.allow' can also be altered by using the menu system (see section 4 Display and Menu System, 4.2.1.3)

5.2.5 Check for Incoming TCP Requests

It is possible to configure the CSXLAN daemon in that way that only configured hosts can get connected. For this the configuration entry `HostsAllow` can be added in the `[Global]` section of the configuration file `csxlan.conf` (see section 5.2.6 below).

Example:

```
HostsAllow = 0 # Default setting
HostsAllow = 1 # Hosts must be enabled via the file hosts.allow
```

Generally, configuration of the hosts will be done in the configuration file `/etc/hosts.allow` on the CryptoServer LAN. The hosts will be checked during connection establishment against their IP addresses.

If the configuration entry `HostsAllow` is activated (second case in example above), then the hosts must be explicitly configured in the file `hosts.allow`. Here an example for such an entry:

```
csxlan: 192.168.1.1 192.168.4.
```

The keyword `csxlan` is mandatory.

For more information about the configuration options of the `hosts.allow` file, we refer to the Linux/Unix manual page hosts_access(5).

5.2.6 The Configuration File 'csxlan.conf'

All the other CryptoServer LAN settings (apart from the IP address) are made in the configuration file `/etc/csxlan.conf`. This configuration file consists of several sections. There is exactly one [Global] section, and one or more [CryptoServer] and [Listener] sections.

Each section consists of several variable assignments. Comment lines can be placed everywhere inside the file, starting with a # character and extending to the end of line. Empty lines are ignored.

A simple assignment looks like

```
VARIABLE = VALUE
```

A list assignment looks like

```
VARIABLE = {
      value1
      value2
      ...
}
```

The most important variables are as follows:

Variables of the **[Global] Section**:

Variable	Description
`Debug`	Debug level. Must be set to '0x00' (no debug).
`Watchdog`	Enables the hardware watchdog timer if a positive value is given. The default value for this variable is to disable the watchdog.
`MaxConnections`	Number of concurrent connections that can be served by the CSXLAN daemon (default setting: 20 connections).

`AuthReset`	Indicates if the CSADM commands 'Reset', 'ResetToBL' and 'Restart' must be authenticated (see also [CS2ADMIN], and 7.1). If any commands will be executed locally, no authentication is needed.
	`AuthReset = 0 # no authentication needed`
	`AuthReset = 1 # authentication needed`

`HostsAllow`	Indicates if any incoming connection request shall be checked against the file '`hosts.allow`' on the CryptoServer LAN (see also 5.2.5 above).
	`HostsAllow = 0 # no check`
	`HostsAllow = 1 # check against hosts.allow`

`DenyLock`	Disable the LOCK command for control commands (see also 7.19)
	`DenyLock = 0 # LOCK command is allowed`
	`DenyLock = 1 # LOCK command is disabled`
	If the DenyLock item is not available the LOCK command is allowed

For each integrated physical CryptoServer device there has to be one **`[CryptoServer]` Section:**

Variable	Description
`Label`	Unique symbolic name for the current CryptoServer (see variable `Route_to` below).
`Device`	Filename of the device file that is assigned to the current CryptoServer device, e.g. `/dev/cs2a` for the first device.
`Timeout`	Amount of time in milliseconds before the device driver aborts a command because the CryptoServer returned no answer. Optional, the default value is 10 minutes.

For each port the CryptoServer LAN should serve, there is one **`[Listener]` Section**:

Variable	Description
`Address`	The local interface address the socket is bound to. If omitted, the server socket is bound to all local interfaces. This is useful if the Cryptoserver LAN is shipped with several network interfaces or if there are configured alias addresses. Commands that are received from various adapters can be processed each by a specific CryptoServer. In case that the multicast flag is set (and the protocol selected is UDP, see below) it gives the multicast group address.
`Port`	Port number the CSXLAN daemon listens for connections. If omitted, port 288 is used by default.
`Protocol`	TCP or UDP (see also section 5.3.1). Optional (default is TCP).
`Multicast`	Multicast flag, ‘0’ or ‘1’. If using Multicast, the protocol has to be UDP.
`Keepalive`	Enable or disable TCP keep alive-packets to detect dead connections. Set to ‘0’ or ‘1’.
`Priority`	Priority assigned to requests to this port. The priority is a value between 1 (highest) and 100 (lowest priority). The default is 100.
`Route_to`	This mandatory option takes the label of a `[CryptoServer]` section and ties the server socket to that specific CryptoServer device. You can also supply a list of labels. The listener then acts as a load balancing port.

For the configuration of the display the **`[DisplayAdmin]` section** is used:

`Device`	This name corresponds to a `[Listener]` section and is used for the communication of the menu system with the CryptoServer The name is formed by concatenating protocol, address and port: *protocol:port@address* Protocol and port specification is optional and defaults to TCP and 288. **Examples:** *TCP:288@localhost, TCP:localhost, 288@localhost, localhost* If there are more than one CryptoServer hardware security modules installed inside the CryptoServer LAN, a list assignment has to be used.
`StateDevice`	The device used for status queries. The same syntax as for the `Device` variable applies.
`Display`	The device name of the serial port connected to the display module. The default value is `/dev/ttyS0`

`PINPad`	Specifies a connected PIN-Pad. The name has the following form:
	`:smartcard-id:pinpad-id:serial-port`
	For more information of this format see also the CryptoServer Administration Guide [CS2ADMIN].
	For the delivered applications this string should be
	`:cs2:cp8:/dev/ttyS1`

`Timeout`	Duration in milliseconds after the last key was pressed before the menu system skips one level up. The default value is 60 seconds.
`CS2_Timeout`	Duration in milliseconds after the currently executed command to the CryptoServer will be aborted. The default value is 10 minutes.
`Logo`	A string giving the first line which is displayed on the status screen of the display (optional, default is “CryptoServer 2000”). If there are more than one CryptoServer hardware security modules installed inside the CryptoServer LAN, a list assignment has to be used.

Example of /etc/csxlan.conf:

```
# csxlan configuration file

[Global]
# no debug output
Debug =   0x00

MaxConnections =  20

# enable the watchdog timer (=1)
Watchdog = 1

# disable LOCKing for control commands
DenyLock = 1


[CryptoServer]
Label  =   CS1
Timeout=   3600000
Device =   PCI:/dev/cs2


# Standard Port
[Listener]
Protocol  = tcp
Port = 288
Keepalive = 1
Route_to  = CS1

# Port for StateDevice
[Listener]
Address = localhost
Protocol  = tcp
Port = 288
Route_to  = CS1


# Multicast Port
[Listener]
Address   = 224.1.0.1
Port      = 288
```

```
Protocol  = udp
Multicast = 1
Route_to  = CS1

[DisplayAdmin]
Device      = localhost
StateDevice = UDP:localhost
Display     = /dev/ttyS0
PINPad      = :cs2:cp8:/dev/ttyS1
Timeout     = 30000
```

5.2.7 Load Balancing

For a load balancing system, you need at least two CryptoServer hardware security modules. To accomplish load balancing, requests have to be sent to a special port which is used by several available CryptoServers.

Example for load balancing (excerpt from a `/etc/csxlan.conf` configuration file):

```
.
# local CryptoServer device
[CryptoServer]
Label  =   CS1
Device =   PCI:/dev/cs2a


# a remote device, Crypto2 is the hostname
[CryptoServer]
Label   = CS2
Device  = TCP:Crypto2


# Listen at port 288/tcp
[Listener]
Route_to = CS1
Priority = 10


# Listen at port 4711/tcp, distribute requests
# to CS1 and CS2
[Listener]
Port     = 4711
Priority = 50
Route_to = {
CS2
CS1
}
.
```

This excerpt from a configuration file defines two CryptoServer devices CS1 and CS2 and two Listeners. Requests to port 288 are always executed on the local CryptoServer labelled CS1. Requests sent to port 4711 are executed on either CS2 (in this case a remote system) or CS1.

Each port has a queue for requests. As soon as a CryptoServer is idle, all queues assigned to this CryptoServer are checked for waiting requests. If several entries are present, the system picks the one with the highest priority. The priority of the others is increased by one.

It should be noted that load balancing is done only if there are at least two clients connected to a load balancing port and if more than one request is in the queue. All requests in one queue have the same priority. Requests are always sent to the first idle CryptoServer, even if more than one is idle. A “Round Robin” scheme does not apply here.

It is advisable to configure a Listener with a high priority for administration tasks.

5.2.8 SNMP

To support the Simple Network Management Protocol (SNMP) some parts of the Net-SNMP package (http://net-snmp.sourceforge.net) are installed on the CryptoServer LAN. The examples below uses command line tools (`snmpget`, `snmptable`) from this package. After delivery the SNMP support is disabled. To enable SNMP the display menu (see section 4) must be used.

Currently only SNMPv1 and SNMPv2 is supported.

The supported Object Identifier (OID) are derived from the Management Information Base (MIB) NET-SNMP-EXTEND-MIB.txt that is located in the directory '/etc/snmp/mibs' on the CSLAN. The information that can be gathered from the CryptoServer and CryptoServer LAN via SNMP are listed below

The default community string to access the SNMP variables is 'cslan'. This can be changed in the configuration file '/etc/snmp/snmpd.conf' on the CSLAN.

Description	CryptoServer Temperature in °C
Type	String
OID	.1.3.6.1.4.1.8072.1.3.2.4.1.2.11.84.101.109.112.101.114.97.116.117.114.101.1
Example	`snmpget -Oqv -v 1 -c cslan <IP Address> <OID>`
Ex. Output	`38.5`

Description	Firmware module status
Type	String
OID	.1.3.6.1.4.1.8072.1.3.2.4.1.2.11.77.111.100.117.108.101.83.116.97.116.101.1
Example	`snmpget -Oqv -v 1 -c cslan <IP Address> <OID>`
Ex. Output	`OK`

Description	CryptoServer operation state
Type	String
OID	.1.3.6.1.4.1.8072.1.3.2.4.1.2.8.71.101.116.83.116.97.116.101.1
Example	`snmpget -Oqv -v 1 -c cslan <IP Address> <OID>`
Ex. Output	`OPERATIONAL`

Description CryptoServer alarm state
Type String
OID .1.3.6.1.4.1.8072.1.3.2.4.1.2.8.71.101.116.65.108.97.114.109.1
Example `snmpget -Oqv -v 1 -c cslan <IP Address> <OID>`
Ex. Output `OFF`

Description CryptoServer LAN load average in %
Type String
OID .1.3.6.1.4.1.8072.1.3.2.4.1.2.7.71.101.116.76.111.97.100.1
Example `snmpget -Oqv -v 1 -c cslan <IP Address> <OID>`
Ex. Output `40.4`

Description CryptoServer LAN process table
Type Table
OID .1.3.6.1.4.1.2021.2
Example `snmptable -v 1 -c cslan <IP Address> .1.3.6.1.4.1.2021.2`
Example Output

```
prIndex   prNames prMin prMax prCount prErrorFlag prErrMessage prErrFix
      1    csxlan     2     3       2           0                     0
      2 dsp_admin     1     1       1           0                     0
      3     ulogd     1     1       1           0                     0
```

To list a summary of the CryptoServer variables (aside from the process table) the OID .1.3.6.1.4.1.8072.1.3.2.4 can be used:

```
snmpwalk -OQs -v 1 -c cslan <IP Address> .1.3.6.1.4.1.8072.1.3.2.4

nsExtendOutLine."GetLoad".1 = 0.0
nsExtendOutLine."GetAlarm".1 = OFF
nsExtendOutLine."GetState".1 = OPERATIONAL
nsExtendOutLine."ModuleState".1 = OK
nsExtendOutLine."Temperature".1 = 41.5
```

For a detailed list of the CryptoServer variables (aside from the process table) the OID .1.3.6.1.4.1.8072.1.3.2.3 can be used:

```
snmpwalk -OQs -v 1 -c cslan <IP Address> .1.3.6.1.4.1.8072.1.3.2.3

nsExtendOutput1Line."GetLoad" = 0.0
nsExtendOutput1Line."GetAlarm" = OFF
nsExtendOutput1Line."GetState" = OPERATIONAL
nsExtendOutput1Line."ModuleState" = OK
nsExtendOutput1Line."Temperature" = 41.5
nsExtendOutputFull."GetLoad" = 0.0
nsExtendOutputFull."GetAlarm" = OFF
nsExtendOutputFull."GetState" = OPERATIONAL
nsExtendOutputFull."ModuleState" = OK
nsExtendOutputFull."Temperature" = 41.5
nsExtendOutNumLines."GetLoad" = 1
nsExtendOutNumLines."GetAlarm" = 1
nsExtendOutNumLines."GetState" = 1
nsExtendOutNumLines."ModuleState" = 1
nsExtendOutNumLines."Temperature" = 1
nsExtendResult."GetLoad" = 0
nsExtendResult."GetAlarm" = 0
nsExtendResult."GetState" = 0
nsExtendResult."ModuleState" = 0
nsExtendResult."Temperature" = 0
```

5.2.9 Web Interface

The CryptoServer LAN is shipped with a web interface over that several status information can be viewed in a normal web browser. After delivery the web interface is disabled. To enable it choose “Configuration Settings” -> “Web Interface” via the display menu and confirm the question with “enable”. The Web Interface is activated immediately and can be accessed via the browser over HTTP port 80. As URL the normal IP address of the CryptoServer LAN must be entered.

With the Web Interface only status information can be viewed. It is not possible to configure any settings on the CryptoServer or CryptoServer LAN

5.3 Communication with the CryptoServer LAN

Applications which wish to use the functions of the CryptoServer must send appropriate function calls to the CryptoServer LAN via Ethernet. For this purpose, the CryptoServer LAN has an IP address and a port number for TCP connections as well as a multicast address and an associated port number for UDP data transfers.

5.3.1 Communication over TCP

A client that wishes to communicate with the CryptoServer LAN must first establish a TCP connection. A CryptoServer command can then be sent over this TCP connection. The CryptoServer processes this command and sends a reply back via the same connection.

Once a connection has been established, any number of commands can be transmitted. Alternatively it is possible to establish a new connection for each individual command. Several connections to CryptoServer LAN (e. g. of different clients) can exist simultaneously. The maximum number of connections can be configured. The system is delivered with a default setting of 20 connections.

It is also possible to use UDP for the communication. However one should keep in mind that UDP is connectionless and is not a reliable protocol. The client must implement its own mechanisms to prevent data loss.

5.3.2 Communication via Multicasting

A command can also be sent to one or more CryptoServer LANs simultaneously via multicasting. For this purpose the client must send the command in a UDP packet to a multicast group address. All CryptoServer LANs in the network with the relevant multicast group address will process this command and send a reply back to the client.

It should be noted that UDP is an insecure datagram protocol. The sender is not informed if a packet does not reach the recipient. The client must therefore secure the communication with appropriate mechanisms such as timeouts, retries etc. It is also necessary to heed the maximum packet size under UDP.

5.4 The CryptoServer Command Structure

All the functions which are available from the CryptoServer via the command interface are requested through the transmission of a command block. The command block is a byte sequence which contains all the details of the requested function and all the parameters and data required to execute it. On termination of the function, the CryptoServer returns a reply block in the form of a byte sequence. This contains the reply data or, if appropriate, an error code.

A command block has the following basic structure:

$9C_H$	Length	FC	SFC	00_H	Command Data
1 byte	3 bytes	2 bytes	1 byte	1 byte	(Length – 8) bytes

A reply block has the following basic structure:

$9A_H$	Length	Reply Data
1 byte	3 bytes	(Length – 4) bytes

In the event of an error, a reply block is returned in this form:

$9E_H$	Length	Error Code	Reply Data (optional)
1 byte	3 bytes	4 bytes	(Length – 8) bytes

Field	Description
`Length`	Total length of the data block (including the header), coded in MSB – LSB order.
`FC`	Function Code of the firmware module of the requested function (module ID)
`SFC`	Sub Function Code of the requested function (ID of the function within the firmware module)

For special purposes other command block structures exist for the CryptoServer. They are explained in detail in the document "CryptoServer Application Interface" [CSAPI].

5.5 Logging

The CryptoServer LAN contains an own logging daemon that runs as a standalone program and receives messages from specific programs like *csxlan*. The daemon is named *ulogd* and can be configured via the configuration file *'/etc/ulogd.conf'*.

The default log file is named '/var/log/csxlan.log' and can be retrieved with the command *csadm CSLGetLogfile* or from the display menu *'LAN Box Administration -> Diagnostic -> Export Trace File'*.

Via the configuration file the user is able to configure the log entries on their own needs like logrotation, maximum number of log files that should be kept, date and time format of log entries, etc.

In the table below there is an overview about the possible configuration items.

The configuration file contains one section [Global] and for each log file a section [Logfile]

Section [Global]

Field	Description
`PID_File`	Absolute pathname of the file where the process id of the log daemon is written. Default is *'/var/run/ulogd.pid'*.
`Socket`	Absolute pathname of the socket from that the messages are read. Default is *'/dev/ulog'*.
`SerialFile`	Absolute pathname of the file, where the current sequence numbers are stored. This file is only created when the configuration item `Format` in the section `[Logfile]` contains at least a '`%s`'.

Section [Logfile]

Field	Description
`Name`	Filename of the log file. If the parameter '`NameStamp`' is also set, than this setting is appended to the filename. Default is '`ulog`'.
`Path`	Absolute path of the directory where the log files are stored. Default is '`/var/log`'.
`Mode`	Access rights of the file. These are masked with '*umask*'. Default is 0664.

`Format`	Specifies the format of a log entry. The string must contain placeholder that are replaced with data (like in *printf*). The following placeholder are possible: - `$m` - Text of the log message - `$s` - Sequence number - is increased by one for each entry - `$t` - Date / Time (ASCII) - `$p` - Program name (Log-Client) - `$u` - Date + Time (4 bytes, binary) After the placeholder `$s`, `$t` and `$p` the square bracket `[]` can contain a format string. For `$s` a format for an '*unsigned long*', for a `$p` a format for a string (like in *printf*), and for a `$t` a format string like in '*strftime*'. Default is '`$t[%d.%m.%y %T] $p: $m\n`'.
`MaxSerial`	Maximum value of the sequence number. This value is only considered if the Format-String contains a '`$s`'. Exceeds the Sequence number this value then it starts again from 0. Default is 99999999.
`StartStopLog`	Boolean value ('y', 'n', '1' or '0'). Indicates if on start or stop of the log daemon an entry is written to the log file. Default is 'n'.
`FlushInterval`	Time in seconds, after the log entries are written to the flash disk. A value of 0 means that all messages are written directly (unbufferd). Default is 3.
`NameStamp`	Format string for '*strftime*'. This string is evaluated for each message with the current time and is appended to the filename of the log file. When the time changes a new log file is created. E.g.: if the format specifier is '`%y%m%d`' than every day a new log file is created. If the format specifier is '`%y%m%d%H`' than every hour a new log file is created.
`MaxSize`	Maximum size of a log file in kB. This parameter is only evaluated if the parameter '`NameStamp`' is not given. Exceeds the size of the log file this value than the file is renamed and a new log file is created. Older log files are also renamed (like logrotate). On each log file name a number is appended starting by 0. Default value is 1024.
`MaxFiles`	Maximum number of log files (including the current one) that are kept. Are more files available than this value, the oldest ones are deleted. The minimum value is 2.
`ZipCmd`	This configuration item can contain the command that is used to compress all available log files (apart from the current one). E.g. `ZipCmd = "gzip -q -f"`.

`Input`	This parameter specifies a selection, what kind of messages are written into the specific log file. The format is: '`<program name>.<category>`'. Where `<program name>` is a String, that must be equal to the program name that logs via *ulogd* and `<category>` is a category (from 0 to 15), that must be correspond to the category of the message. For each a '*' can be given. Examples: - `Input = ntpd.2` Messages of the category 2 of the program 'ntpd'. - `Input = csxlan.*` All messages of the program '*csxlan*'. - `Input = *.5` All Messages of the category 5. The parameter '`Input`' can be given several times for one log file to specify different selections. Is this parameter is omitted then all messages for this log file are accepted. Every log message will be written into the first log file (in preceding order of the configuration file) where the message matches the criteria of the '`Input`' parameter. If no match is found the messages is discarded.

If the values have been changed in the configuration file ulogd.conf these changes become valid after either system restart or restart of the ulogd with the command

/etc/init.d/ulogd restart

5.6 External Erase

Every data inside the CryptoServer can be cleared manually by an external erase. The external erase can be executed by pushing the button (“1” in the picture below) located in the battery case of the CryptoServer LAN.

When the external erase button is pushed the CryptoServer LAN must be connected to the mains.

The external erase affects only the data inside the CryptoServer. The data on the CryptoServer LAN (operating system, configuration files, etc) are not erased.

6 Batteries of the CryptoServer LAN

The CryptoServer LAN contains two batteries to ensure that security relevant data are not lost even when the CryptoServer LAN is turned off. If the device is not used over a prolonged period, e. g. during storage or computer turned off, both batteries will have a durability of 2½ years at a minimum. If the CryptoServer LAN is permanently powered on, the batteries are not discharged and the lifetime of the batteries increases.

The CryptoServer LAN communication unit contains a size D 3.6 Volt lithium battery that is directly connected to the hardware security module CryptoServer and serves as the main power supply of the CryptoServer. This battery may power one or two CryptoServer modules and is called “external battery“. The external battery can also be exchanged by the customer.

Additionally a 3 Volt lithium battery is mounted on every CryptoServer (directly on the PCI board) to ensure that the erase circuit and the sensory are always working even when the CryptoServer is turned off (“carrier battery”). This battery is used only after the external battery is exhausted.

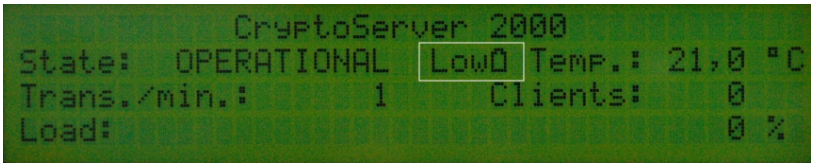
Only specialized staff of Utimaco is allowed to replace the carrier battery!

More information about both batteries and a detailed description of how to exchange the external battery can be found in [CS2LAN-OpManual].

In the following subsection it is described how the status of the batteries can be checked. This should be done on a regular basis. If any of the batteries is in a critical state, it has to be exchanged. In case that the carrier battery has to be exchanged, please contact Utimaco Safeware AG.

6.1 Check State of the Batteries

If one of the two batteries of the CryptoServer LAN is low, as part of the status screen the word "LOW" and a battery symbol is displayed automatically in the center of the screen.



Use the menu on the display of the CryptoServer LAN to query the states of the different batteries:

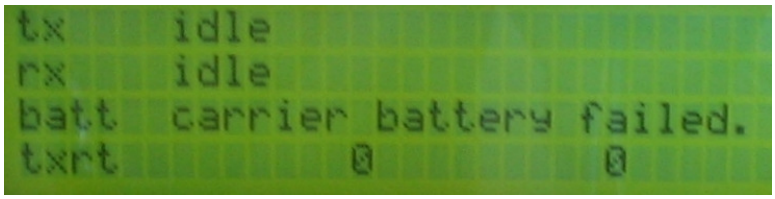
1. Press the **OK** key.
2. Choose the menu entry "CryptoServer administration" using the arrow key 🡻and confirm by pressing the **OK** key.
3. Choose the menu entry "Generic commands" using the arrow key 🡻and confirm by pressing the **OK** key.
4. Choose the menu entry "Show driver Info" using the arrow key 🡻and confirm by pressing the **OK** key.
5. Use the arrow key 🡻 to scroll down the contents of the display until the battery state is shown.

- **If both batteries contain sufficient charge** the display output contains the following line:

```
batt   ok
```

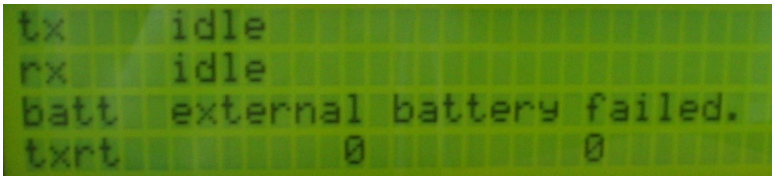
*If the message "**batt ?**" appears, the state of the batteries could not be determined because of concurrent data transfer to the CryptoServer. In this case (that should happen rarely) repeat the command "*Show driver info*" until the battery state is shown.*

- **Carrier battery in critical state:**



If this message is shown the carrier battery of the CryptoServer has low power.

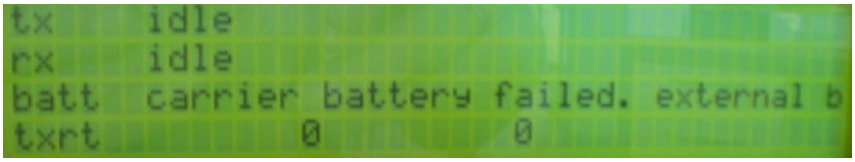
- **External battery in critical state:**



If this message is shown the external battery has low power.

- **Carrier and external batteries in critical state:**

*If both batteries have low power the two messages **"carrier battery failed"** and **"external battery failed"** are displayed back-to-back.*



7 Special CryptoServer LAN Control Commands

The following commands are not processed by the CryptoServer hardware security module but by the communication processor and serve to control the CryptoServer LAN.

The command and reply byte blocks generally have the same format as described in section 5.4:

They consist of 8 bytes (command block) respectively 4 bytes (reply block) header plus the pure command/reply data. Since these commands are not addressed to the CryptoServer (and any of its firmware modules inside) but to the communication processor CryptoServer LAN itself, as Function Code (FC) a constant value (00 00_H, “FC_CONTROL") has to be used. The Sub Function Codes (SFCs) are command-individual. As “Length” parameter always the length of the entire command/reply block has to be given, including the header.

7.1 Command Authentication

Some control commands have to be authenticated. This can usually be done by means of the root password (see 5.1).

If for command authentication a RSA signature mechanism shall be used, the CryptoServer LAN must store the public part of the respective RSA key. This key will be called CryptoServer LAN’s RSA key.

Generally, before a command can be authenticated, first the ‘Get Challenge’ command has to be called, see 7.6. This command returns a random number (‘challenge’) and other data which have to be used for the subsequent authentication.

Using this data a so-called *authentication token* (see below) can be calculated for a subsequent command. This byte block has to be sent as part of the command which has to be authenticated.

In the following subsection it is explained how to build and how to use the authentication token.

7.1.1 Authentication Token

For each command which needs to be authenticated it is mandatory to send an authentication token (a specific byte block calculated for authentication) within this command. To get the information needed to calculate this token the output of the command 'Get Challenge' must be used.

The structure of an authenticated command block is generally as follows (see also section 5.4):

$9C_H$	Length	FC	SFC	00_H	auth-token	data
1 byte	3 bytes	2 bytes	1 byte	1 byte	'tokenlength' bytes	n bytes

The construction of the authentication token `auth-token` depends on the used authentication mechanism (which can be based on password or RSA mechanism). The general token structure is like this:

tokenlength	tag	auth-data
2 bytes	1 byte	m bytes

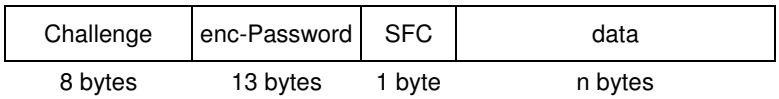
Field	**Description**	
`tokenlength`	total length of the authentication token (1 + m)	
`tag`	Tag indicating the authentication mechanism: ■ 01_H: Authentication based on the root password (using the UNIX-internal `crypt` command) ■ 02_H: Authentication based on the RSA algorithm	
`auth-data`	algorithm specific authentication data:	
	tag	**data**
	01_H	SHA-1 hash over command data, including the encrypted root password, see below
	02_H	RSA signature over SHA-1-hashed command data, calculated with the CryptoServer LAN's RSA key. See below.

The authentication token must indicate the same 'tag' as returned in the response to the preceding 'Get Challenge' command. If a different 'tag' will be used the authentication will fail.

The exact calculation of `auth-data` depends of the authentication mechanism, as indicated by the tag parameter:

- **Tag 01_H: Authentication based on the root password**

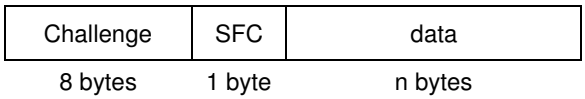
  `auth-data` is the result of a SHA-1 hash calculation over the following data:



Field	Description
`Challenge`	Random number from the preceding “Get Challenge” command
`Enc-Password`	Encrypted root password of the CryptoServer LAN. Encryption has to be done by the Unix-internal `crypt` function using the “salt” from the preceding “Get Challenge” command.
`SFC`	Sub-Function Code of this command, see above
`data`	Data in command block, see above (command specific)

- **Tag 02_H: Authentication based on the RSA algorithm**

  To calculate `auth-data` first a SHA-1 hash must be calculated over the following data:



  Then `auth-data` is the RSA signature over this SHA-1 hash value, calculated with the CryptoServer LAN’s RSA key.

Field	Description
`Challenge`	Random number from the preceding “Get Challenge” command
`SFC`	Sub-Function Code of this command, see above
`data`	Data in command block, see above (command specific)

7.2 Check Operational Readiness of CryptoServer LAN

This command returns information about all ports the CryptoServer LAN listens to. The command is especially suited for a multicast query and enables one to ascertain all the CryptoServer LAN devices available in the network.

The command block has the following structure:

$9C_H$	Length	FC	SFC	00_H
1 byte	3 bytes	2 bytes	1 byte	1 byte

Field	Value	Description
`Length`	$00\ 00\ 08_H$	Length of command block
`FC`	$00\ 00_H$	CryptoServer LAN Function Code, constant "FC_CONTROL"
`SFC`	01_H	Sub-Function Code, constant "SFC_CTRL_STATE"

Upon successful execution, the following reply is returned:

$9A_H$	Length	Record 1	Record 2	…	Record k
1 byte	3 bytes	n_1 bytes	n_2 bytes	…	n_k bytes

It includes one record with information data for every port that the CryptoServer LAN listens to (see also the description of the `[listener]`-sections in 5.2.6). The record length n_i is a multiple of 4 respectively.

Each record has the following format:

Record length	Protocol	IP address	Port	Device
4 bytes	4 bytes	4 bytes	2 bytes	n bytes

Field	Description
Length	total length of reply block ($4 + n_1 + n_2 + \ldots + n_k$)
Record Length	length n_i of this record including this field (14 + n Bytes)
Protocol	null terminated string “TCP” or “UDP”
IP address	interface address of the server (in network byte order).
Port	port number for TCP connections (in network byte order)
Device	device name (e.g. /dev/cs2a) padded with zeros s. t. the record length (14 + n) is a multiple of 4

Possible error codes:

E_CSLAN_CTRL_BADCMD	Wrong parameter count found in the command block.

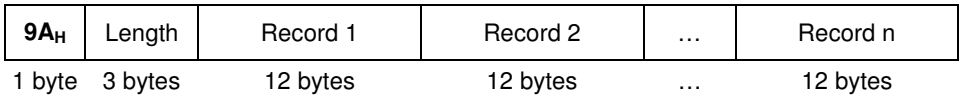
7.3 Get Current Connections

This command enables all currently existing connections to the CryptoServer LAN to be ascertained. The command block has the following structure:

$9C_H$	Length	FC	SFC	00_H
1 byte	3 bytes	2 bytes	1 byte	1 byte

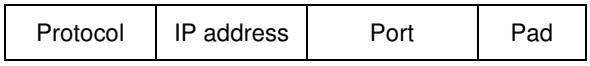
Field	Value	Description
Length	00 00 08_H	Length of command block
FC	00 00_H	CryptoServer LAN Function Code, constant "FC_CONTROL"
SFC	02_H	Sub-Function Code, constant "SFC_CTRL_GETCONN"

On successful execution, the following reply is returned, containing one record for each existing connection:



$9A_H$	Length	Record 1	Record 2	…	Record n
1 byte	3 bytes	12 bytes	12 bytes	…	12 bytes

Each record contains the following info data about the connection:



Protocol	IP address	Port	Pad
4 bytes	4 bytes	2 bytes	2 bytes

Field	Description
Length	total length of reply block (4 + n*12, where n is the number of connections)
Protocol	protocol: “UDP” or “TCP”, followed by a null byte.
IP address	IP address of the client (network byte order)
Port	port number of the client (network byte order)
Pad	2 fill-bytes

Possible error codes:

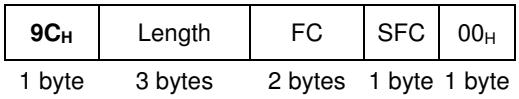
E_CSLAN_CTRL_BADCMD	Wrong parameter count found in the command block.

7.4 Get Communication Server Software Versions

This command is used to request the version numbers of the installed server software. At least the following information is returned:

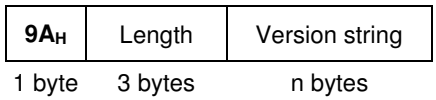
- version of the base system (base)
- version of the CryptoServer-relevant software (cs2)

The command block has the following structure:



Field	Value	Description
Length	00 00 08_H	Length of command block
FC	00 00_H	CryptoServer LAN Function Code, constant "FC_CONTROL"
SFC	03_H	Sub-Function Code, constant "SFC_CTRL_GETVER"

On successful execution, an ASCII character string containing the different version numbers is returned as reply:



Field	Description
Length	total length of reply block (4 + n)
Version string	ASCII character string including a terminating null character. Each version number is separated by a linefeed.

Possible error codes:

E_CSLAN_CTRL_BADCMD	Wrong parameter count found in the command block.
E_CSLAN_CTRL_MEM	memory allocation failed
E_CSLAN_CTRL_FILE	no such file (with version information)

7.5 Get Status of the Security Module's Driver

This command is used to output the status of the security module's PCI driver in the form of an ASCII character string. The command block has the following structure:

$9C_H$	Length	FC	SFC	00_H
1 byte	3 bytes	2 bytes	1 byte	1 byte

Field	Value	Description
`Length`	00 00 08_H	Length of command block
`FC`	00 00_H	CryptoServer LAN Function Code, constant "FC_CONTROL"
`SFC`	04_H	Sub-Function Code, constant "SFC_CTRL_CSSTATE"

On successful execution, an ASCII character string is returned:

$9A_H$	Length	Status data
1 byte	3 bytes	n bytes

Field	Description
`Length`	total length of reply block (4 + n)
`Status data`	ASCII character string, containing the status of the security module's PCI driver. It consists of several text lines separated by linefeed characters. See example below.

Possible error codes:

`E_CSLAN_CTRL_FAILED`	System call failure.
`E_CSLAN_CTRL_NO_ROUTE`	No relation between port where this command has been sent to (`[listener]` section in configuration file) and CryptoServer device.
`E_CSLAN_CTRL_BADCMD`	Wrong parameter count found in the command block.

Example:

```
vers  2.0.4
slot  0000:01:0c.0
tx    idle
rx    idle
batt  ok
txrt          0          0
rxrt          0          0
mbr   00100021  00000068
bar0  00000000  07c80000  00
bar0  00000000  00000068  08
bar0  00000000  00000000  10
bar0  00000000  00000000  18
bar0  00000000  000f0100  20
bar0  0e040000  00000000  28
bar0  00000008  00000000  30
bar0  00000000  00000000  38
bar0  ffffffff  ffffffff  40
bar0  00000000  07cc4000  48
bar0  00000000  00000010  50
bar0  00000000  00000000  58
bar0  00000000  00000000  60
bar0  08080808  08080808  68
bar0  00400020  00000068  70
bar0  00100021  xxxxxxxx  78
bar0  0000feff  00000000  80
bar0  00050100  00000000  88
bar0  00000000  05000000  90
bar0  00000000  00000000  98
bar0  00000000  00000000  a0
bar0  00000000  00000000  a8
bar0  00000000  00000000  b0
bar0  00000000  00000000  b8
bar0  00000000  00000000  c0
bar0  00000000  00000000  c8
bar0  00000000  00000001  d0
bar0  00000000  00000000  d8
bar0  00000000  00000000  e0
bar0  00000000  00000000  e8
bar0  00000000  00000000  f0
bar0  00000000  00000000  f8
```

7.6 Get Challenge

This command is used to get a random number (‘challenge’) from the CryptoServer LAN. The random number is used for the subsequent control command that must be authenticated (see also 7.1).

The command block has the following structure:

$9C_H$	Length	FC	SFC	00_H
1 byte	3 bytes	2 bytes	1 byte	1 byte

Field	Value	Description
`Length`	$00\ 00\ 08_H$	Length of command block
`FC`	$00\ 00_H$	CryptoServer LAN Function Code, constant "FC_CONTROL"
`SFC`	05_H	Sub-Function Code, constant "SFC_CTRL_GET_CHALLENGE”

Upon successful execution, one of the following replies is returned:

$9A_H$	Length	01_H	Challenge	Password-Salt
1 byte	3 bytes	1 byte	8 bytes	n bytes

or

$9A_H$	Length	02_H	Challenge
1 byte	3 bytes	1 byte	8 bytes

The returned tag parameter 01_H or 02_H depends on the authentication algorithm that has to be used (password or RSA mechanism). The algorithm is detected automatically during command execution (see below).

Field		Description
`Length`		total length of reply block (13+n respectively 13)
`Tag`	01_H	Authentication based on the root password (using the UNIX-internal `crypt` command which uses MD5 hash)
	02_H	Authentication based on the RSA algorithm
`Challenge`		Random number which has to be used to authenticate a subsequent control command. Will be used for the calculation of the authentication token, see 7.1.1.

`Password-Salt`	"Salt" parameter which has to be used to authenticate a subsequent control command (in case that the root password mechanism is used). The salt will be used as input parameter by the Unix-internal `crypt` function which will encrypt the root password. The in that way encrypted root password is used to calculate the authentication token, see 7.1.1.

A description of how to authenticate a control command (using either root password, 'salt' and 'challenge', or the CryptoServer LAN's RSA key and 'challenge') is also given with the respective commands.

Possible error codes:

`E_CSLAN_CTRL_BADCMD`	Wrong parameter count found in the command block.
`E_CSLAN_CTRL_DENIED`	Wrong permission for file operations.

7.7 Set Trace Level

This command is used to set the trace level (see 4.2.3.1) of the CryptoServer LAN communication processor.

The command block has the following structure:

$9C_H$	Length	FC	SFC	00_H	auth-token	Level
1 byte	3 bytes	2 bytes	1 byte	1 byte	n bytes	1 byte

Field	Value	Description
`Length`	9 + n	Length of command block
`FC`	00 00_H	CryptoServer LAN Function Code, constant "FC_CONTROL"
`SFC`	09_H	Sub-Function Code, constant "SFC_CTRL_TRACE"
`auth-token`		Authentication token needed to authenticate the command (see 7.1 and below)
`Level`		Bitmask for trace level, see also 4.2.3.1: ■ Trace Level **Info:** 80_H ■ Trace Level **Verbose**: 40_H ■ Trace Level **Packet data:** 20_H

The ‘auth-token’ parameter is used to authenticate the command.

To build the `auth-data` parameter which is part of the authentication token as described in 7.1.1, one of the following calculations has to be done:

- Root password mechanism:

  Calculate a SHA-1 hash over the following data:

Challenge	enc-Password	SFC	Level
8 bytes	13 bytes	1 byte	1 byte

- RSA key mechanism:

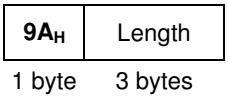
  To obtain `auth-data` first a SHA-1 hash must be calculated over the following data:

Challenge	SFC	Level
8 bytes	1 byte	1 byte

Then `auth-data` is the RSA signature over this SHA-1 hash value, calculated with the CryptoServer LAN's RSA key.

Field	Description
`Challenge`	Random number from the preceding "Get Challenge" command
`Enc-Password`	Encrypted root password of the CryptoServer LAN. Encryption has to be done by the Unix-internal `crypt` function, using the "salt" from the preceding "Get Challenge" command.
`SFC`	Sub-Function Code of this command (i. e. 09_H, see above)
`Level`	Trace level, see above

Upon successful execution, the following reply is returned:



Field	Description
`Length`	total length of reply block (00 00 04_H)

Possible error codes:

`E_CSLAN_CTRL_BADCMD`	Wrong parameter count found in the command block.
`E_CSLAN_CTRL_AUTH`	authentication failed.

7.8 Put Configuration File

This command is used to transfer a new configuration file to the CryptoServer LAN communication processor. The imported file is checked for syntax errors, semantic errors will not be detected. The new configuration is not used until the CSXLAN daemon is restarted.

The command block has the following structure:

$9C_H$	Length	FC	SFC	00_H	auth-token	Len	Filename	Configuration data
1 byte	3 bytes	2 bytes	1 byte	1 byte	'tokenlength' bytes	2 bytes	'Len' bytes	n bytes

Field	Value	Description
`Length`	10 + 'Len' + n + 'tokenlength'	Length of command block
`FC`	00 00_H	CryptoServer LAN Function Code, constant "FC_CONTROL"
`SFC`	$0C_H$	Sub-Function Code, constant "SFC_CTRL_CONFIG_PUT"
`auth-token`		Authentication token needed to authenticate the command (see 7.1 and below)
`Len`		Length of filename in MSB/LSB order
`Filename`		Name of the configuration file on the CryptoServer LAN.
`Configuration data`		Configuration file, see 5.2.6.

The 'auth-token' parameter is used to authenticate the command.

To build the `auth-data` parameter which is part of the authentication token as described in 7.1.1, one of the following calculations has to be done:

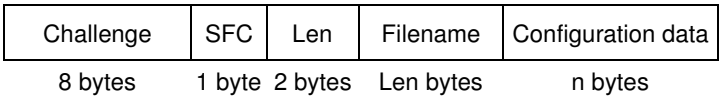
- Root password mechanism:

  Calculate a SHA-1 hash over the following data:

Challenge	enc-Password	SFC	Len	Filename	Configuration data
8 bytes	13 bytes	1 byte	2 bytes	Len bytes	n bytes

- RSA key mechanism:

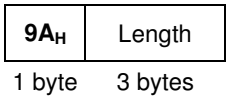
  To obtain `auth-data` first a SHA-1 hash must be calculated over the following data:



Then `auth-data` is the RSA signature over this SHA-1 hash value, calculated with the CryptoServer LAN's RSA key.

Field	Description
`Challenge`	Random number from the preceding "Get Challenge" command
`Enc-Password`	Encrypted root password of the CryptoServer LAN. Encryption has to be done by the Unix-internal `crypt` function using the "salt" from the preceding "Get Challenge" command.
`SFC`	Sub-Function Code of this command (i. e. $0C_H$, see above)
`Len`	see above
`Filename`	see above
`Configuration data`	see above

Upon successful execution, the following reply is returned:



Field	Description
`Length`	total length of reply block (00 00 04_H)

Possible error codes:

`E_CSLAN_CTRL_DENIED`	Wrong permission for file operations.
`E_CSLAN_CTRL_CONFIG`	Illegal configuration file.
`E_CSLAN_CTRL_BADCMD`	Wrong parameter count found in the command block.
`E_CSLAN_CTRL_AUTH`	authentication failed.

7.9 Get Configuration File

This command is used to transfer the configuration file (see 5.2.6) from the CryptoServer LAN communication processor to the client computer.

The command block has the following structure:

$9C_H$	Length	FC	SFC	00_H	Len	Filename
1 byte	3 bytes	2 bytes	1 byte	1 byte	2 bytes	'Len' bytes

Field	Value	Description
`Length`	10 + 'Len'	Length of command block
`FC`	00 00_H	CryptoServer LAN Function Code, constant "FC_CONTROL"
`SFC`	$0D_H$	Sub-Function Code, constant "SFC_CTRL_CONFIG_GET"
`Len`		Length of filename in MSB/LSB order
`Filename`		Name of the configuration file on the CryptoServer LAN

Upon successful execution, the following reply is returned:

$9A_H$	Length	Configuration data
1 byte	3 bytes	n bytes

Field	Description
`Length`	total length of reply block (n + 4)
`Configuration data`	Configuration file, see 5.2.6.

Possible error codes:

`E_CSLAN_CTRL_DENIED`	Wrong permission for file operations.
`E_CSLAN_CTRL_BADSIZE`	Corrupt length specification in the command block.
`E_CSLAN_CTRL_BADCMD`	Wrong parameter count found in the command block.
`E_CSLAN_CTRL_FILE`	no such file

7.10 Set Message Mode

During the execution of a function the CryptoServer could generate text messages. These messages are logged to a file by default. However a client can modify the behavior for this connection, by changing the message mode with this command. For more information about the block structure for messages see [CSAPI].

The possible values for the `Mode` parameter are as following:

0	messages are written to the logfile (default)
1	messages are discarded
2	messages are sent to the client

The command block has the following structure:

$9C_H$	Length	FC	SFC	00_H	Mode
1 byte	3 bytes	2 bytes	1 byte	1 byte	1 byte

Field	Value	Description
`Length`	$00\ 00\ 09_H$	Length of command block
`FC`	$00\ 00_H$	CryptoServer LAN Function Code, constant "FC_CONTROL"
`SFC`	10_H	Sub-Function Code, constant "SFC_ MSG_MODE"
`Mode`		New message mode, see above (0 = log, 1 = discard, 2 = client)

Upon successful execution, a reply frame with the previous mode is returned:

$9A_H$	Length	Mode
1 byte	3 bytes	1 byte

Field	Description
`Length`	total length of reply block ($00\ 00\ 05_H$)
`Mode`	previous message mode, see above

Possible error codes:

`E_CSLAN_CTRL_BADCMD`	Wrong parameter count found in the command block.
`E_CSLAN_CTRL_BADSIZE`	Corrupt length specification in the command block.

7.11 Set Priority

This command changes the priority of a listener (see also 5.2.6). The priority is a value between 1 (highest priority) and 100 (lowest priority).

The command block has the following structure:

$9C_H$	Length	FC	SFC	00_H	Port	Priority
1 byte	3 bytes	2 bytes	1 byte	1 byte	2 bytes	1 byte

Field	Value	Description
`Length`	00 00 $0B_H$	Length of command block
`FC`	00 00_H	CryptoServer LAN Function Code, constant "FC_CONTROL"
`SFC`	20_H	Sub-Function Code, constant "SFC_CTRL_PRIO_SET "
`Port`		Listener port in network byte order.
`Priority`		New priority (1-100)

Upon successful execution, a reply frame with the previous priority of the listener is returned:

$9A_H$	Length	Old priority
1 byte	3 bytes	1 byte

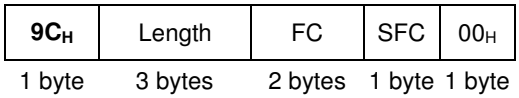
Field	Description
`Length`	total length of reply block (00 00 05_H)
`Old Priority`	previous priority (1-100)

Possible error codes:

`E_CSLAN_CTRL_BADCMD`	Wrong parameter count found in the command block.
`E_CSLAN_CTRL_BADSIZE`	Corrupt length specification in the command block.

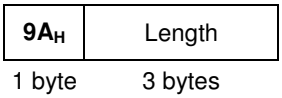
7.12 Check Connection

This command can be used by the host to check the connection to the CryptoServer LAN. The command block has the following structure:



Field	Value	Description
Length	00 00 08_H	Length of command block
FC	00 00_H	CryptoServer LAN Function Code, constant "FC_CONTROL"
SFC	22_H	Sub-Function Code, constant “SFC_CTRL_CHECK"

Upon successful execution, the following reply is returned:



In case that the connection does not work, no reply will be returned.

Field	Description
Length	total length of reply block (00 00 04_H)

Possible error codes:

E_CSLAN_CTRL_BADCMD	Wrong parameter count found in the command block.

7.13 Shutdown CryptoServer LAN

This command is used to shut down or reboot the operating system of the CryptoServer LAN communication processor.

The command block has the following structure:

$9C_H$	Length	FC	SFC	00_H	auth-token	Command
1 byte	3 bytes	2 bytes	1 byte	1 byte	'tokenlength' bytes	n bytes

Field	Value	Description
`Length`	8 + n + 'tokenlength'	Length of command block
`FC`	00 00_H	CryptoServer LAN Function Code, constant "FC_CONTROL"
`SFC`	$7E_H$	Sub-Function Code, constant "SFC_CTRL_SHUTDOWN"
`auth-token`		Authentication token needed to authenticate the command (see 7.1 and below)
`Command`		String "SHUTDOWN" or "REBOOT"

The 'auth-token' parameter is used to authenticate the command.

To build the `auth-data` parameter which is part of the authentication token as described in 7.1.1, one of the following calculations has to be done:

- Root password mechanism:

  Calculate a SHA-1 hash over the following data:

Challenge	enc-Password	SFC	Command
8 bytes	13 bytes	1 byte	n bytes

- RSA key mechanism:

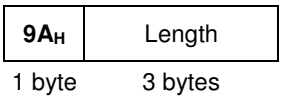
  To obtain `auth-data` first a SHA-1 hash must be calculated over the following data:

Challenge	SFC	Command
8 bytes	1 byte	n bytes

  Then `auth-data` is the RSA signature over this SHA-1 hash value, calculated with the CryptoServer LAN's RSA key.

Field	Description
`Challenge`	Random number from the preceding “Get Challenge” command
`enc-Password`	Encrypted root password of the CryptoServer LAN. Encryption has to be done by the Unix-internal `crypt` function using the “salt” from the preceding “Get Challenge” command.
`SFC`	Sub-Function Code of this command (i. e. $7E_H$, see above)
`Command`	See above

Upon successful execution, the following reply is returned:



Field	Description
`Length`	total length of reply block (00 00 04_H)

Possible error codes:

`E_CSLAN_CTRL_FAILED`	System call failure.
`E_CSLAN_CTRL_BADCMD`	Wrong parameter count found in the command block.
`E_CSLAN_CTRL_AUTH`	authentication failed.

7.14 Reset CryptoServer

This command is used to execute a reset of the security module CryptoServer. Here, various reset modes are possible:

- **Reset:** A hardware reset of the CryptoServer at PCI-level is executed. See 4.3.4.3.
- **Reset to BL:** The CryptoServer is reset and stays in bootloader mode. In particular the commands described in 4.3.3 are available now. See 4.3.4.2.
- **Restart:** The CryptoServer is reset and stays in administration mode. In particular the commands described in 4.3.2 are available now. See 4.3.4.1.

Depending on the setting, this command optionally has to be authenticated. This depends on the configuration of the CSXLAN (setting of `AuthReset` variable in the `[Global]` Section of the CSXLAN configuration file, see chapter 5.2.6).

The command block has the following structure:

Without authentication:

$9C_H$	Length	FC	SFC	00_H	Mode
1 byte	3 bytes	2 bytes	1 byte	1 byte	n bytes

With authentication:

$9C_H$	Length	FC	SFC	00_H	auth-token	Mode
1 byte	3 bytes	2 bytes	1 byte	1 byte	'tlength' bytes	n bytes

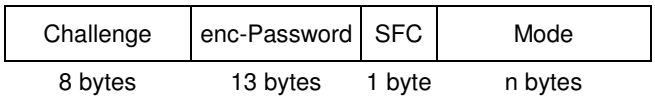
Field	Value	Description
`Length`	(8 + n) or (8 + n + 'tlength')	Length of command block
`FC`	00 00_H	CryptoServer LAN Function Code, constant "FC_CONTROL"
`SFC`	$7F_H$	Sub-Function Code, constant "SFC_CTRL_RESET"
`auth-token`		Authentication token needed to authenticate the command (see 7.1 and below)
`Mode`		Mode: String "RESET", "RESETTOBL" or "RESTART" (see above)

The (optional) 'auth-token' parameter is used to authenticate the command.

To build the `auth-data` parameter which is part of the authentication token as described in 7.1.1, one of the following calculations has to be done:

- Root password mechanism:

  Calculate a SHA-1 hash over the following data:



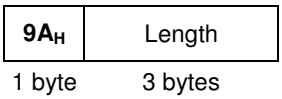
- RSA key mechanism:

  To obtain `auth-data` first a SHA-1 hash must be calculated over the following data:

  Then `auth-data` is the RSA signature over this SHA-1 hash value, calculated with the CryptoServer LAN's RSA key.

Field	Description
`Challenge`	Random number from the preceding "Get Challenge" command
`enc-Password`	Encrypted root password of the CryptoServer LAN. Encryption has to be done by the Unix-internal `crypt` function using the "salt" from the preceding "Get Challenge" command.
`SFC`	Sub-Function Code of this command (i. e. $7F_H$, see above)
`Mode`	See above

Upon successful execution, the following reply is returned:



Field	Description
`Length`	total length of reply block (00 00 04_H)

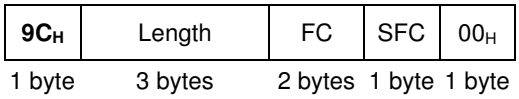
Possible error codes:

E_CSLAN_CTRL_NO_ROUTE	No relation between port where this command has been sent to ([listener] section in configuration file) and CryptoServer device.
E_CSLAN_CTRL_BADCMD	Wrong parameter count found in the command block.
E_CSLAN_CTRL_AUTH	authentication failed.

7.15 Get Serial Number

This command is used to obtain the serial number of the CryptoServer LAN communication unit. This serial number is normally located on the right hand side of the housing and starts with MD. The number is written to the file '/etc/version/serial' during the installation process and can be accessed with this command.

The command block has the following structure:



Field	Value	Description
Length	00 00 08$_H$	Length of command block
FC	00 00$_H$	CryptoServer LAN Function Code, constant "FC_CONTROL"
SFC	06$_H$	Sub-Function Code, constant " SFC_CTRL_GET_SERIAL"

Upon successful execution, the following reply is returned:

Field	Description
Length	total length of reply block (n + 4)
serial	Zero terminated serial number in ASCII format.

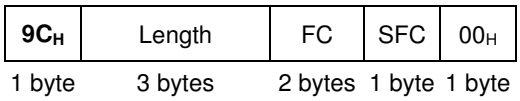
Possible error codes:

E_CSLAN_CTRL_BADCMD	Wrong parameter count found in the command block.
E_CSLAN_CTRL_NO_SER	No serial number available

7.16 Get Time

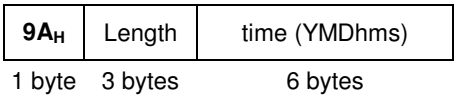
This command is used to get the local system time of the CryptoServer LAN communication unit (not of the CryptoServer!).

The command block has the following structure:



Field	Value	Description
Length	00 00 08$_H$	Length of command block
FC	00 00$_H$	CryptoServer LAN Function Code, constant "FC_CONTROL"
SFC	07$_H$	Sub-Function Code, constant " SFC_CTRL_GET_TIME"

Upon successful execution, the following reply is returned:



Field	Description
Length	total length of reply block (10)
time	Y – The number of years since 1900
	M – The number of months since January, in the range 1 to 12
	D – The day of the month, in the range 1 to 31
	h – The number of hours past midnight, in the range 0 to 23
	m – The number of minutes after the hour, in the range 0 to 59
	s – The number of seconds after the minute, in the range 0 to 59

Possible error codes:

E_CSLAN_CTRL_BADCMD	Wrong parameter count found in the command block.

7.17 Set Time

This command is used to set the local system time of the CryptoServer LAN communication unit (not of the CryptoServer!).

The command block has the following structure:

$9C_H$	Length	FC	SFC	00_H	auth-token	time (YMDhms)
1 byte	3 bytes	2 bytes	1 byte	1 byte	n bytes	6 bytes

Field	Value	Description
`Length`	14 + n	Length of command block
`FC`	00 00_H	CryptoServer LAN Function Code, constant "FC_CONTROL"
`SFC`	08_H	Sub-Function Code, constant " SFC_CTRL_SET_TIME"
`auth-token`		Authentication token needed to authenticate the command (see 7.1 and below)
`time`		Local system time to be set for CryptoServer LAN ■ Y – The number of years since 1900 ■ M – The number of months since January, in the range 1 to 12 ■ D – The day of the month, in the range 1 to 31 ■ h – The number of hours past midnight, in the range 0 to 23 ■ m – The number of minutes after the hour, in the range 0 to 59 ■ s – The number of seconds after the minute, in the range 0 to 59

The (optional) ‘auth-token’ parameter is used to authenticate the command.

To build the `auth-data` parameter which is part of the authentication token as described in 7.1.1, one of the following calculations has to be done:

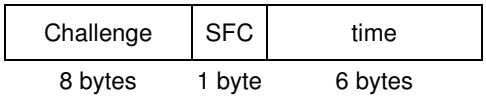
- Root password mechanism:

  Calculate a SHA-1 hash over the following data:

Challenge	enc-Password	SFC	time
8 bytes	13 bytes	1 byte	6 bytes

- RSA key mechanism:

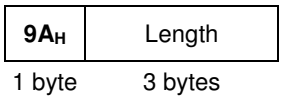
  To obtain `auth-data` first a SHA-1 hash must be calculated over the following data:



  Then `auth-data` is the RSA signature over this SHA-1 hash value, calculated with the CryptoServer LAN's RSA key.

Field	Description
`Challenge`	Random number from the preceding "Get Challenge" command
`enc-Password`	Encrypted root password of the CryptoServer LAN. Encryption has to be done by the Unix-internal `crypt` function using the "salt" from the preceding "Get Challenge" command.
`SFC`	Sub-Function Code of this command (i. e. 08_H, see above)
`time`	See above

Upon successful execution, the following reply is returned:



Field	Description
`Length`	total length of reply block (00 00 04_H)

Possible error codes:

`E_CSLAN_CTRL_BADCMD`	Wrong parameter count found in the command block.
`E_CSLAN_CTRL_AUTH`	authentication failed.

7.18 Get Load

This command is used to get the current working load of the hardware security module(s) CryptoServer in the CryptoServer LAN communication unit. If the communication unit contains more than one CryptoServer, the load for each CryptoServer is returned.

The load value returned by the command is not the working load at that time but is an average value of the last 60 seconds. It is recalculated and updated every 5 seconds.

The command block has the following structure:

$9C_H$	Length	FC	SFC	00_H
1 byte	3 bytes	2 bytes	1 byte	1 byte

Field	Value	Description
`Length`	00 00 08_H	Length of command block
`FC`	00 00_H	CryptoServer LAN Function Code, constant "FC_CONTROL"
`SFC`	$0f_H$	Sub-Function Code, constant " SFC_CTRL_GET_LOAD"

Upon successful execution, the following reply is returned:

$9A_H$	Length	load 1	...	load n
1 byte	3 bytes	2 bytes		2 bytes

Here, n is the number of hardware security modules CryptoServer included in the CryptoServer LAN communication unit (1 ≤ n ≤ 4). Usually, n = 1.

Field	Description
`Length`	total length of reply block (2*n + 4)
`load i`	Load of CryptoServer no. i in percent * 10, given as integer. **Example:** If the first CryptoServer has a load of 10.7%, as `load 1` the integer 107 will be returned.

The information of the working load of the CryptoServer is read from the display module of the CryptoServer LAN. If the display module is busy (i.e. an operator is working at the display) this information is not available and the command does not output any value.

Possible error codes:

E_CSLAN_CTRL_BADCMD	Wrong parameter count found in the command block.

7.19 Lock

This command locks the access to the CryptoServer e.g for maintenance. If the CryptoServer is locked no other command, beside of the connection which sent the lock command, can be send to the CryptoServer.

If the connection should be terminated unexpectedly without sending the unlock command, the lock will be automatically removed.

The command block has the following structure:

$9C_H$	Length	FC	SFC	00_H	mode	timeout	data
1 byte	3 bytes	2 bytes	1 byte	1 byte	1 byte	2 bytes	16 bytes

Field	Value	Description
`Length`	00 00 $0b_H$ or 00 00 $1b_H$	Length of command block
`FC`	00 00_H	CryptoServer LAN Function Code, constant "FC_CONTROL"
`SFC`	23_H	Sub-Function Code, constant " SFC_CTRL_LOCK"
`mode`	00 or constant LOCK_REJECT	If the CryptoServer is locked and a command is send over another transmission channel, this command will be rejected with the error code E_CSLAN_CTRL_LOCKED
	01 or constant LOCK_WAIT	If the CryptoServer is locked and a command is send over another transmission channel, this command will be queued and executed when the UNLOCK command was send. The application should be aware that a timeout can occur if the LOCK exists for a longer time.
`timeout`		Timeout in seconds when the LOCK should expire if no UNLOCK command is send. After reaching the timeout the lock will be removed automatically. If no timeout is needed this must be set to 0.
`data`		The data field is optional and can contain arbitrary data which will be send as data payload with the error code E_CSLAN_CTRL_LOCKED. E.g. this can contain detailed information why a CryptoServer was locked.

Upon successful execution, the following reply is returned:

$9A_H$	Length
1 byte	3 bytes

Field	Description
Length	total length of reply block (00 00 04_H)

Possible error codes:

E_CSLAN_CTRL_BADCMD	Wrong parameter count found in the command block.
E_CSLAN_CTRL_LOCKED	CryptoServer is already locked
E_CSLAN_CTRL_LOCK_DENIED	CryptoServer lock denied (on load balancing ports)
E_CSLAN_CTRL_NO_ROUTE	no relation between port and CryptoServer device

7.20 Unlock

This command unlocks a CryptoServer if it was locked before.

If the connection should be terminated unexpectedly without sending the unlock command, the lock will be automatically removed.

The command block has the following structure:

$9C_H$	Length	FC	SFC	00_H
1 byte	3 bytes	2 bytes	1 byte	1 byte

Field	Value	Description
`Length`	00 00 08_H	Length of command block
`FC`	00 00_H	CryptoServer LAN Function Code, constant "FC_CONTROL"
`SFC`	24_H	Sub-Function Code, constant " SFC_CTRL_UNLOCK"

If the CryptoServer is locked the UNLOCK command can only be send over the same transmission channel which has set the lock.

If no LOCK command was send before this command always succeeds.

Upon successful execution, the following reply is returned:

$9A_H$	Length
1 byte	3 bytes

Field	Description
`Length`	total length of reply block (00 00 04_H)

Possible error codes:

E_CSLAN_CTRL_BADCMD	Wrong parameter count found in the command block.
E_CSLAN_CTRL_LOCKED	CryptoServer is locked by a different transmission channel
E_CSLAN_CTRL_NO_ROUTE	no relation between port and CryptoServer device

7.21 Error Codes

Error Code	Value in hex	Description
E_CSLAN_CTRL_BADCMD	0xB90A0101	wrong parameter count found in the command block
E_CSLAN_CTRL_FAILED	0xB90A0102	system call failure
E_CSLAN_CTRL_DENIED	0xB90A0103	wrong permissions to access a file
E_CSLAN_CTRL_NO_ROUTE	0xB90A0104	no relation between port where this command has been sent to ([Listener] section in configuration file) and CryptoServer device
E_CSLAN_CTRL_FILE	0xB90A0105	no such file
E_CSLAN_CTRL_BADSIZE	0xB90A0106	corrupt length specification in the command block
E_CSLAN_CTRL_RESET	0xB90A0107	command aborted because a reset command has been executed
E_CSLAN_CTRL_AUTH	0xB90A0108	authentication failed
E_CSLAN_CTRL_MEM	0xB90A0109	memory allocation failed
E_CSLAN_CTRL_NO_SER	0xB90A010A	No serial number available
E_CSLAN_CTRL_LOCKED	0xB90A010B	CryptoServer is locked
E_CSLAN_CTRL_ALREADY_LOCKED	0xB90A010C	CryptoServer is already locked
E_CSLAN_CTRL_LOCK_DENIED	0xB90A010D	CryptoServer lock denied (on load balancing ports)
E_CSLAN_CTRL_NO_KSAPI	0xB90A0120	KSAPI compatibility listener not found
E_CSLAN_CTRL_CONFIG	0xB90A0201	attempt to import an invalid configuration file
E_CSLAN_QUEUE	0xB90A0301	queue for incoming requests is full
E_CSLAN_AVAIL	0xB90A0302	no CryptoServer online
E_CSLAN_LEN_MISMATCH	0xB90A0303	mismatch between real packet length and data length
E_CSLAN_NO_CONN	0xB90A0304	connection table full
E_CSLAN_MAPPING	0xB90A0305	KSAPI compatibility mapping not found

8 Legal Notice

The CryptoServer LAN (CSLAN) contains software based on the GNU public license (GPL) and other licenses. The license texts for each software package included in the CSLAN are available on the CSLAN in the directory `/usr/share/doc`.

To obtain a copy of the source code of these software packages please contact hsm@aachen.utimaco.de

9 References

No.	Title / Company	Doc.-No.
[CS2ADMIN]	CryptoServer – Administration Guide / Utimaco Safeware AG	2002-0021
[CSAPI]	CryptoServer Application Interface (CSAPI) / Utimaco Safeware AG	2002-0005
[CS2LAN-OpManual]	CryptoServer LAN – Operating Manual / Utimaco Safeware AG	2005-0001