White Paper

Network Licensing WP2

0-15Apr014ks(WP02_Network).odt

Subject: Network Licensing with the CRYPTO-BOX
Version: Smarx OS PPK 5.90 and higher
Last Update: 28 April 2014
Target Operating Systems: Windows 8/7/Vista (32 & 64 bit), XP, Linux, OS X, iOS, Android
Target Processor Platforms: Intel x86, ARM
Supported Programming Tools: see chapter 3.3.6 in this document
Applicable for Product: CRYPTO-BOX® SC / XS / Versa

Network Licensing with the CRYPTO-BOX®

Network licensing with the CRYPTO-BOX is ideal for cost-effective software licensing in (corporate) networks. The software vendor determines how often the application is allowed to run in a network - with just one CRYPTO-BOX per network.

Furthermore, it allows software licensing not only for PCs and laptops, but also for environments without the possibility to connect a dongle directly:

- Mobile devices (Tablets, Smartphones)
- Virtual machines (Windows/Citrix Terminal Server)

This document provides an overview about network licensing and the components on the client and server side.

White Paper

Network Licensing

Table of Contents

1. Network Licensing – An Overview

1.1. Introduction

Hardware-based protection with the CRYPTO-BOX requires that the protected applications have a corresponding CRYPTO-BOX attached to the computer in order to function properly. If the CRYPTO-BOX is attached, the program will communicate with it, performing more detailed verification of information stored in the device, such as:

- Verification of the CRYPTO-BOX Serialnumber (BoxName) or Developer ID
- Using the hardware-based encryption engine to decrypt information during application runtime
- Using the internal memory of the CRYPTO-BOX to store licensing information of the application

All these, as well as many other unique CRYPTO-BOX features, can be used to build a very reliable protection strategy.
However, in some cases a CRYPTO-BOX attached to the local computer is not desired or possible, for example:

- At places where the CRYPTO-BOX can get lost or stolen.
- Where a centralized license management is required.
- For mobile devices (Tablets, Smartphones) with no USB connector.
- For virtual machines (eg. Windows or Citrix Terminal Server solution) offering no USB support.

1.2. Advantages of Network Licensing

Network licensing provides the following advantages over the local connection of the the CRYPTO-BOX to the USB port of the computer:

- Complete control over licenses, respectively running instances of the protected application, in the entire network.
- Protection and license management for environments where a local USB port is not available/accessible like restricted PCs, Smartphones, Tablets (iOS and Android, support for Windows RT and Windows Phone will be available soon).
- No administrative access required for both installation and during runtime of the protected application (access to a locally connected CRYPTO-BOX requires administrative rights at least for installation of the CRYPTO-BOX drivers)
- Suitable for licensing in server-based computing environments (Microsoft, VMWare, Citrix, etc.).
- Cost-efficiency: multiple license counters can be defined to protect several applications with one CRYPTO-BOX.
- Integrated tool to monitor and check the status of connected clients.

1.3. License Control System (LCS) for Configuring the Maximum Number of Seats

The License Control System (LCS) is available for all CRYPTO-BOX models (except CRYPTO-BOX Versa). It allows to configure the CRYPTO-BOX with a seat limit between 0 and 254. The value 255 is reserved for unlimited use. The programming of the network license counter can be done with the Smarx OS Application Framework, or the command line tool 'SmrxProg' (both are included in the Professional Protection Kit).

LCS is available as an option. Please refer to www.marx.com → Solutions → LCS for further details on pricing.

The network license counter is a data object which is stored in a special area of the CRYPTO-BOX memory. It can hold network license information for multiple partitions (different applications). That means you can define independent network license counters for each partition (application) in the CRYPTO-BOX.

For more details on CRYPTO-BOX partitions, we strongly recommend to read chapter 11.6 in the Smarx Compendium. It's important to understand this concept.

When a program tries to login to a proper partition, first Smarx OS will search for a LCS record for this partition. If the record is found, LCS checks the number of permitted licenses and the number of seats currently logged in. If there are free licenses available, access to the partition is granted, otherwise it is denied. For a CRYPTO-BOX Versa, the number of permitted network seats is always unlimited.

The Remote Update Management System (RUMS) allows the programming/update of the license counter directly by the end user, providing subsequent transactions through the sale of additional licenses for the software distributor.

More details on RUMS can be found in the RUMS Application Notes.

2. Smarx®OS Network Server

2.1. Introduction

Smarx OS Networking allows protected applications to access a CRYPTO-BOX attached to the USB port of any computer in a network.

A special program called **Smarx OS Network Server** (or CBIOS Network Server), running on a computer in the network, manages remote connections to the CRYPTO-BOX attached to this computer. Any Smarx OS based application trying to access a CRYPTO-BOX will be recognized as a **Smarx OS Network Client** (see chapter *3)*.

The Smarx OS Network Server is available for Windows (32/64Bit), Linux (32/64Bit) and Mac OS X platforms.

See item "Drivers & Tools" → "Network Server" in the Protection Kit Control Center for more details.

2.2. Server Installation

The CRYPTO-BOX Network Server is available for the following platforms:

- Windows 32 and 64 bit versions (Windows XP and up)
- Linux 32 and 64 bit
- Mac OS X

Check our website www.marx.com → Support → Downloads → Network Utilities to get the latest version of the CRYPTO-BOX Network Server for your platform.

For the Windows platform, the server is provided as Windows Installer setup (.msi) and Windows Installer Merge Module (.msm) for 32 or 64 bit. See item "Drivers & Tools" → "Network Server" in the PPK Control Center for more information.

The Windows Installer setup (.msi) also installs the CRYPTO-BOX device driver during installation, so it is not required to install the drivers separately with CBUSetup.exe.

During installation, you can select if you want to start the server as a service under Windows. This can be changed later if required, see chapter *2.3* for details.

2.3. Server Administration

2.3.1. General Issues

After the Smarx OS Network Server is installed (see chapter *2.2*), it can be administrated either locally (on the same PC where it is running) or remotely, using the Network Administrative Console.
When started, the Network Server icon appears in the system tray. To open the Server console select **"Open"**, to stop/start the Server select **"Stop/Start"** and to shutdown the Server select **"Exit"**.

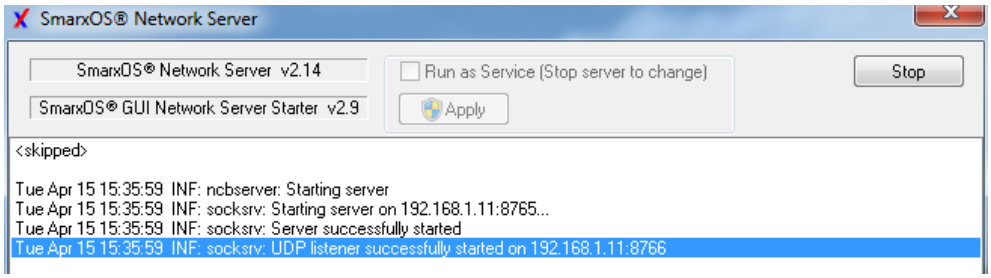

Figure 2.1:
Server Console

2.3.2. Change Server Settings

There are two ways to change the settings of the Smarx OS Network Server:

- Using the Administrative Console (AdminApp.exe), which allows you to manage the server remotely using the TCP/IP protocol.
- Editing the CBIOSSrv.cfg file in the server directory.

The following settings can be changed only by editing the CBIOSSrv.cfg file:

- **IP:** IP address of the server (useful if multiple network adapters are available) "*" means: automatic settings
- **Port:** TCP port for server access, default value: 8765
- **UDP Port:** port for UDP access, default value: 8766
- **Debug Level:** specifies output and status messages from the server, written to log file (CBIOSSrv.log)

  Level 0 - quiet, nothing is written to log file
  Level 1 - critical errors
  Level 2 - warnings
  Level 3 - general information (default)
  Level 4 - debug information

The following settings can be changed either by editing CBIOSSrv.cfg file or using Administrative Console:

Server administration password: This password is required to access the server with the Administrative Console. It can be changed either by editing the "Password" entry in CBIOSSrv.cfg file or by clicking the "Change password" button in the Administrative Console.

Connection timeout: Timeout for an inactive socket-level connection in situations when a connection (socket) to the server is opened but client does not send any data. The connection will be closed on reaching the timeout. After that, the server will be ready to process subsequent requests.
At the client side, a Connection Timeout message (transport layer error of the last CBIOS API call) will be received. The client has to repeat the last call to resume work. Default setting is 30 seconds.

Inactivity timeout (scan rate): This is the session-level inactivity timeout. If the client does not send a “keep alive” packet for the previously opened session before this timeout is reached, the session will be closed and all appropriate resources (network licenses, etc.) will be released.
On the client side, it means that the connection to the server is OK, but the current encrypted session with the server is disconnected. So the client should call CBIOS_OpenBy... again.
In case of connection problems (client receives CBIOS_ERR_CONN_REFUSED), the client application can do the following:

- Repeat the last API call (maybe twice).

- If this does not help: Try to reopen the CRYPTO-BOX with CBIOS_OpenBy... call.
- If that also fails: Close the session completely with CBIOS_Close() and start from the beginning with CBIOS_Connect().

Also refer to chapter *3.3.5* for more information on how to handle typical situations such as connection breaks.
Default setting is 3 seconds (3000 milliseconds in CBIOSSrv.cfg).

Inactivity scan rate: This is the rate at which the session table is scanned for inactive sessions (those sessions for which the "keep alive" packet is not received in time). Default setting is 180 seconds.

2.3.3. Administrative Console

The administrative application **AdminApp.exe/AdminApp64.exe** is used for Smarx OS Network Server administration. It provides the following functions:

- Displays information about the CRYPTO-BOX and its partitions available on the server-side.
- Displays information about the network applications attached to the Server and the licenses available.
- Permits to disconnect all network clients (by restarting the server).
- Allows to change Smarx OS Server settings (to configure the server).
- Permits to change the password needed for Server Administration.
- Allows to restart or shutdown the server remotely.

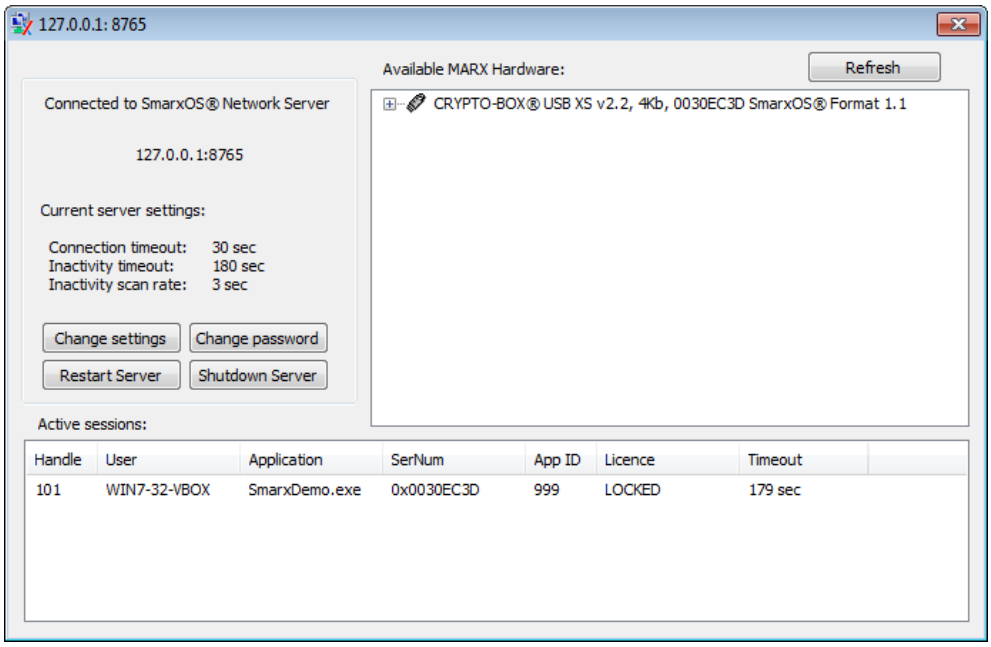

Figure 2.2:
Administrative Console

Connect to the Server

The Administrative Console connects to the server via TCP-IP protocol. Server Name (IP address – 127.0.0.1 by default, which means the local computer), Port (8765 by default) and Password (admin by default) need to be submitted on the connect dialog window.

Server Information

After connecting to the Server, information on Server settings, the attached CRYPTO-BOX and all active applications are displayed.

Change Server Settings

To change the server settings, click the "Change settings" button. Set the Server Connection timeout, the Inactivity timeout, and the Inactivity scan rate (see chapter *2.3.2* for more information).

Restart or Shutdown Server

To restart/shutdown the server press the "Restart Server" or "Shutdown Server" button.

Remember that if the server is shut down, it can only be restarted locally on the computer where it is installed!

2.3.4. Running the Server as a Service under Windows

The Server can be started as a system service under Windows; it will be started automatically when Windows is launched. Simply follow these steps:

- Run the server (CBIOSSRV.EXE or CBIOSSRV64.EXE).
- Stop the server using the "Stop" button in the server console window.
- Enable the "Run as service" option.
- Click the "Apply" button.
- Start the server by clicking the "Start" button on the server console window.

After that, you can exit the server launcher application (on the system tray). You will be asked if you wish to stop the service or keep it running.

Later, after the system reboot, the service will be launched automatically during Windows startup (no login to Windows required).

Administrative rights are required to register/unregister the server as service.

2.3.5. Unregistering the Server as a Service

- Run the server (CBIOSSRV.EXE or CBIOSSRV64.EXE).

- Stop the server using the “Stop” button in the server console window.
- Disable the “Run as service” option.
- Start the server by clicking the “Start” button on the server console window.

After the Server has been registered as a service, CBIOSSRV can be also managed through *Administrative Tools -> Component Services* in Windows Control Panel.

2.3.6. Monitoring of Network Licenses

Since version 2.9, the Smarx OS Network Server (Windows version only) supports monitoring of network licenses. This functionality allows customers to get real statistics and information on their network licensing process. Standard System Event Log is used for logging: open the Windows Event Viewer (*Control Panel -> Administrative Tools -> Event Viewer*) and look for the "CBIOSServer" section. The main advantage is that it is easy to access System Event Log from any application (even remotely).

Any application can subscribe through standard API and then be notified on new records of "CBIOSServer" type. The Server sends the following notification codes and logs corresponding types of events:

- Server Started
- Server Stopped
- License Locked
- License Unlocked

See:
<SmarxOS PPK root folder>\SmarxOS\Network\Win\Samples\CBIOSServerEventLog_Sample

3. Smarx®OS Network Client

3.1. Introduction

The **“Smarx OS Network Client”** describes an application which connects to the **Smarx OS Network Server** (see chapter *2*) rather than direct access to a CRYPTO-BOX attached to the local USB port of the computer. Network support is provide for the automatic implementation with AutoCrypt (see chapter *3.2*) as well as the Implementation with API (see chapter *3.3*).

3.2. Automatic Software Protection with AutoCrypt

AutoCrypt provides protection of applications without any programming efforts. It supports network licensing for Windows platforms. All information required for network licensing, such as:

- Server IP address and port (alternatively, automatic search can be selected);
- Maximum number of network licenses (seats) for the application;
- Check for locally attached CRYPTO-BOX first;

can be set up during the steps of protecting the application.

The steps of protecting applications including network licensing and CRYPTO-BOX formatting are described in the AutoCrypt Application Notes.

The AutoCrypt logic for detecting the CRYPTO-BOX in the network by the protected application will work the following way:

- If an IP address for the server was specified, it will search for this IP address.
- If automatic search via UDP broadcasting was specified, it will try to locate the server automatically.
- If the CRYPTO-BOX was not found on the specified address or via auto-search, the protected application will open a dialog window asking for server settings.
- If the server was found, the settings will be stored (in the Windows registry) for next application start.

3.3. Implementation with API

3.3.1. Differences Between API Calls for Local and Network Access

The access to a remote CRYPTO-BOX connected to the CBIOS Network Server is provided with the same interface functions as for local access, plus some extra functions, which are required in network mode:

CBIOS_ScanNetwork	Search Smarx OS Network Servers
CBIOS_SetScanPort	Set/Get UDP port used for network searching
CBIOS_GetScanPort	(default port is used if not set)
CBIOS_GetServerInfo	Get short information about Smarx OS Network Server
CBIOS_Connect	Connects to Smarx OS Network Server
CBIOS_Disconnect	Disconnects from Smarx OS Network Server
CBIOS_LockLicense	Locks the network license for the opened application (partition)
CBIOS_ReleaseLicense	Releases the network license for the opened application (partition)
CBIOS_GetAppLicenses	Retrieves license info for the application (partition) from LMT
CBIOS_SetAppLicenses	Sets license info for the application (partition) in the LMT
CBIOS_CheckAppLicense	Verifies the network license for the application

For more details on all API calls please refer to cbios.h and to our Developer's Guides with API descriptions for different environments:

- CBIOS API reference for C/C++/Delphi/VB developers
  and
- CBIOS4NET Developer's Guide for C# developers.

Only standard mode functions are supported on the network, extended functions for CRYPTO-BOX configuration (as available in the XSMRX COM object) are not allowed.

3.3.2. License Control System – Defining Number of Network Licenses

Each CRYPTO-BOX contains a special License Management Table (LMT) which is placed in a separate system partition of the CRYPTO-BOX memory. Every record in this table contains information about the Application ID and the corresponding number of network licenses, defining how many clients are allowed to run the application simultaneously which accesses this Application ID (partition).

If there is no record for an Application ID, the number of network licenses for this Application ID is unlimited.

The License Management Table and the number of network licenses for each partition can be programmed using either the Smarx Application Framework (GUI-based) or the SmrxProg command line tool. Please refer to the Smarx Compendium, chapter 4.5 for more details.

When a program tries to login to a proper Application ID (partition), Smarx OS searches for the LCS record for this partition first. When found, the number of permitted licenses and the number of clients currently logged in, are checked. If there are free licenses available, access to the partition is granted, otherwise it is denied.

Every Smarx OS based application can access a local CRYPTO-BOX (via CBIOS API) as well as a remote CRYPTO-BOX (attached to the Smarx OS Network Server somewhere in the network).

The Smarx API (CBIOS) allows all clients to retrieve general information on the CRYPTO-BOX attached, including serial number, digital signature status, model info, etc., even if there are no free licenses available. The License Control System (LCS) is active only when the client application tries to **login** to an Application ID/partition.

The procedure of login/logout functionality in network mode is different than in local mode!
In local mode, all operations which require a login (read/write access to RAM1 or RAM2, access to encryption functions) have to placed in login/logout brackets, because after login the CRYPTO-BOX is can be used only exclusive by the current application/thread and is blocked for other threads. In network mode, the Network Server manages access from different applications. Therefore, before locking a license for one Application ID (partition), the application must login (and stay logged in until the license will be released. Please check the corresponding network licensing sample code for your compiler in the Protection Kit. It will help you to understand this concept.

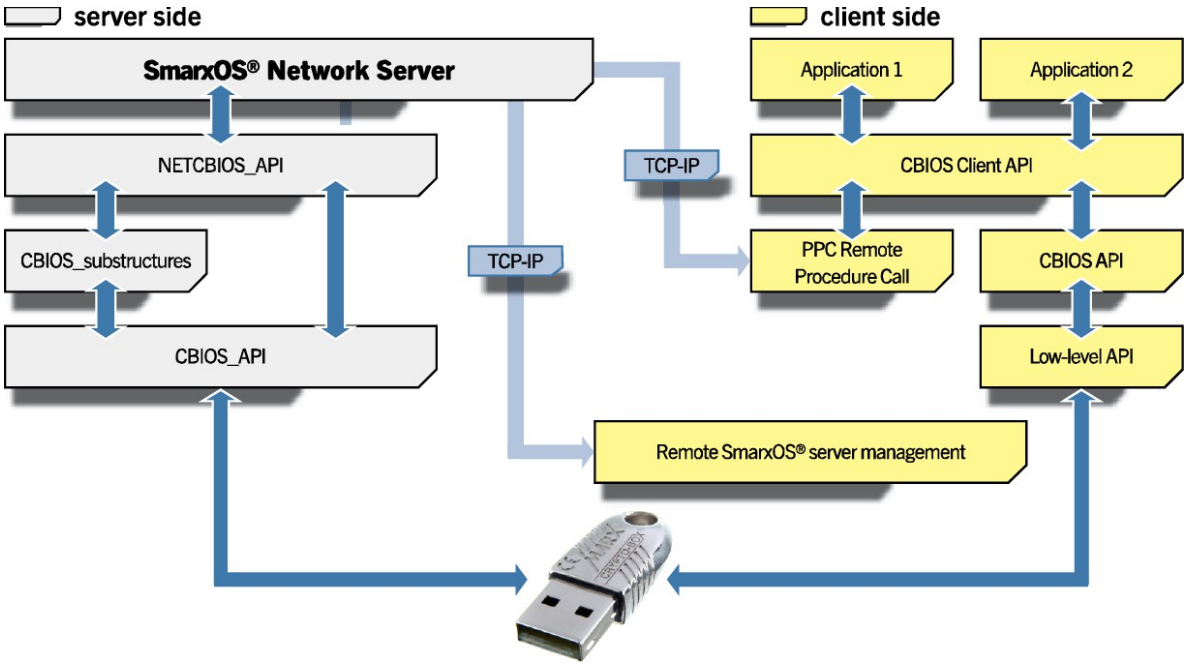

Fig. 3.1:
Architecture of Smarx® OS Networking

3.3.3. Typical Network Session Scenario

First of all, the protected application needs to connect to the Smarx OS Network Server. This can be done through the **CBIOS_Connect** function, if the Server Network Name or IP address is already known to the client. In other cases, the client can try to search for available Smarx OS Network Servers via **CBIOS_ScanNetwork**.

After connecting successfully, the client application can retrieve information on all CRYPTO-BOX units and partitions available on the server-side, open a partition and login to a CRYPTO-BOX using standard CBIOS functions.

To lock a network license for the opened partition, use the **CBIOS_LockLicense** command. The license counter - taken from License Management Table (LMT, see chapter *3.3.2*), if present, will be decremented. If the network license for the application/partition is UNLIMITED, no actions will be performed. To release the network license of the open partition, use **CBIOS_ReleaseLicense**. The license counter will be incremented.

The **CBIOS_LockLicense** and **CBIOS_ReleaseLicense** functions can also be present in code written for local applications (if the same code is shared for local and network applications). In local mode, these functions can be used to prevent several copies of a protected application from being launched through a Terminal Server.

Before a Smarx OS network application may be closed, it must be disconnected from Smarx OS Network Server using the **CBIOS_Disconnect** function. If the application shows no signs of vitality (ignores keep-alive messages) within the timeout period defined in the server settings (see chapter *2.3*), it will be automatically

disconnected.

3.3.4. Support for License Binding

A special feature of the Smarx API is the ability to bind the protected software to a specific computer. For example, this can be beneficial if the use of the protected software outside the company is not permitted.

There are two possibilities of license binding in network mode:

- The protected software will be bound to the network server where the CRYPTO-BOX is attached
- The protected application will be bound to the client computer which issued the binding request.

For more details on license binding, please refer to the Smarx Compendium, chapter 13.2.

3.3.5. Network Licensing - Sample Code for Developers

The Smarx OS Protection Kit contains sample code which demonstrates CBIOS network licensing logic for all popular compilers and platforms. See section “Software Protection with Smarx API for Developers” in the Protection Kit Control Center for more details.

The following prototype for integrating CBIOS Network Licensing logic into your application is included in the Smarx OS PPK:

<SmarxOS PPK root folder>\ SmarxOS\API\Win\Samples\CBIOS Network Licensing\

Besides basic network licensing functionality it covers such typical situations as:

- Network connectivity losses and/or CBIOS Server stopped responding.
- Unplugging/plugging back the CRYPTO-BOX on the computer running CBIOS Server;
- and more.

The sample can serve as a prototype for developers integrating CBIOS Network Licensing logic to their applications. It is written in MFC C++ (Visual Studio 2005), but it serves as a reference code which can be adapted to other programming languages as well.

3.3.6. Network CBIOS API Calls

For a detailed description of all CBIOS API calls, see the Developer's Guides with API descriptions for different environments: CBIOS API reference for C/C++/Delphi/VB developers and CBIOS4NET Developer's Guide for C# developers.

3.4. Smarx Cloud Security – User Authentication and License Management for Web Applications

Smarx Cloud Security allows a web server to communicate via HTTP with the CRYPTO-BOX attached to the client’s computer or network. No customer specific software is needed on the client’s computer. It works with all standard web browsers.

The communication between the client and online-services is encrypted. This solution can be used for various scenarios:

- Secure user authentication for web sites/portals.
- Licensing of web based applications and services to end-users. Only users with a valid license in the CRYPTO-BOX, either attached to a local computer or in the network, will have access to the web content.
- Automation of remote updates of licensing information stored inside the CRYPTO-BOX, no manual processing is required.

A sample and more details are available at www.marx.com/en/solutions/cloud-security. Please scroll down the page to start the WEB API online demo which demonstrates how to access a CRYPTO-BOX at the end-user's side. It requires a CRYPTO-BOX with demo codes (contained in the Evaluation Kit). The sample includes the following features:

- Access to the CRYPTO-BOX connected either to the local USB port of the computer or on the network.
- Support for Notification (automatic detection if a CRYPTO-BOX is attached or removed).
- Binding the license to a specified computer or location (see chapter *3.3.4*).

The sample is written in PHP. Examples for Java/JSP and ASP.NET are also available.

Please refer to the Smarx Cloud Security White Paper for further details.