Sentinel HASP®

Moving from Hardlock to Sentinel HASP

Migration Guide

SafeNet®

Copyrights and Trademarks

Cross-Locking, Hardlock, Hasp, HASP4, Method-Level Protection, Sentinel, Sentinel HASP, Sentinel HASP HL, Sentinel HASP SL, Sentinel HASP Business Studio, Sentinel HASP Reporting Module, Sentinel HASP Trialware, Sentinel SuperPro, and Sentinel UltraPro are either registered in United States Patent and Trademark Office or are trademarks of SafeNet, Inc. and its subsidiaries in the United States and/or other countries, and may not be used without written permission.

All other trademarks are property of their respective owners.

Patents

HASP® hardware and/or software products described in this document are protected by one or more of the following Patents, and may be protected by other United States and/or foreign patents, or pending patent applications: US 5,359,495, US 5,898,777, US 6,189,097, US 6,073,256, US 6,272,636, US 6,009,525, US 6,044,469, US 6,055,503, US 6,334,213, US 6,434,532, US 6,285,985, US 6,334,214, US 6,009,401, US 6,243,692, US 6,363,356, US 7,149,928, US 7,065,652, US 6,915,425, US 6,898,555, US 7,065,650, US 7,225,336, US 7,191,325, EP 1220075, EP 1318451, EP 1271310, EP 1353259, EP 1387235 and EP 1439446.

Disclaimer

We have attempted to make this document complete, accurate, and useful, but we cannot guarantee it to be perfect. When we discover errors or omissions, or they are brought to our attention, we endeavor to correct them in succeeding releases of the product. SafeNet, Inc., is not responsible for any direct or indirect damages or loss of business resulting from inaccuracies or omissions contained herein. The specifications contained in this document are subject to change without notice.

.

November 2010 Revision 0910-1

Contents

Introduction

About Sentinel HASP

Sentinel HASP® is a Software Digital Rights Management (DRM) solution that delivers strong copy protection, protection for Intellectual Property, and secure and flexible licensing. Sentinel HASP is an all-in-one solution that enables you to choose a hardware- or software-based protection key, based on business considerations. Sentinel HASP software engineering and business processes are completely separate to ensure:

- Effective and efficient product development
- Quick time to market
- Immediate addressing of customer and market needs
- Comprehensive support throughout the software product's protection and licensing life cycle

The level of protection for your software is determined by the locking type you choose—hardware- or software-based. Sentinel HASP hardware-based protection, which utilizes HASP HL keys, provides the safest and strongest level of protection.

About This Guide

This guide is intended for Hardlock users who wish to continue using a hardware-based protection solution, but who want to migrate to the improved HASP HL key protection and advanced licensing options provided by Sentinel HASP.

Note: If you want to implement Sentinel HASP software-based protection, refer to the *Sentinel HASP Software Protection and Licensing Guide*.

The guide assumes that the reader has a good understanding of both the Hardlock and the Sentinel HASP systems. It provides the following:

- An overview and guidelines for a two-stage migration path from Hardlock to Sentinel HASP, starting with an install base consisting only of Hardlock keys
- Procedures relating to the migration that are not documented in either the Hardlock documentation, or the *Sentinel HASP Software Protection and Licensing Guide*, and Help documentation
- Tables that list the tools and functionalities of Hardlock and their counterparts in Sentinel HASP

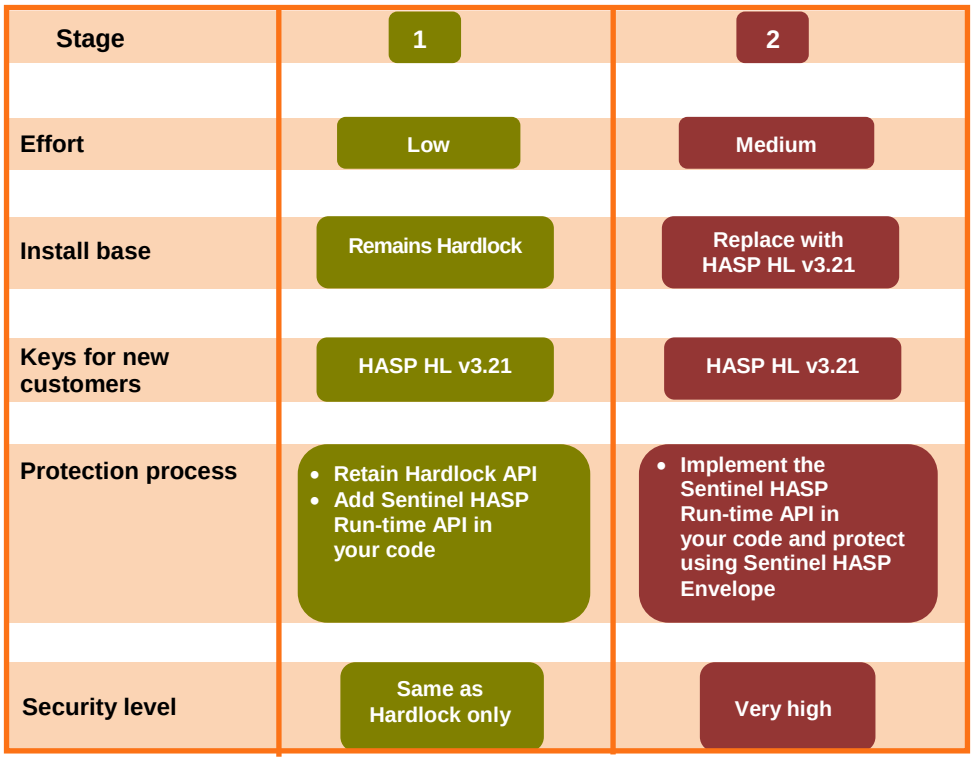
For detailed information and procedures relating to Sentinel HASP, refer to the, *Sentinel HASP Installation Guide*, *Sentinel HASP Software Protection and Licensing Guide* or to the relevant Sentinel HASP Help documentation.

For detailed information and procedures relating to Hardlock, refer to the relevant Hardlock documentation.

Migration Path 1—Gradual Migration from Hardlock to Sentinel HASP

This two-stage migration path enables you to improve your security and expand your licensing options by gradually implementing Sentinel HASP capabilities. The time that you wait before moving from one stage to the next is entirely at your discretion. You can also proceed directly to Stage 2.

The following diagram summarizes the two-stage migration path.



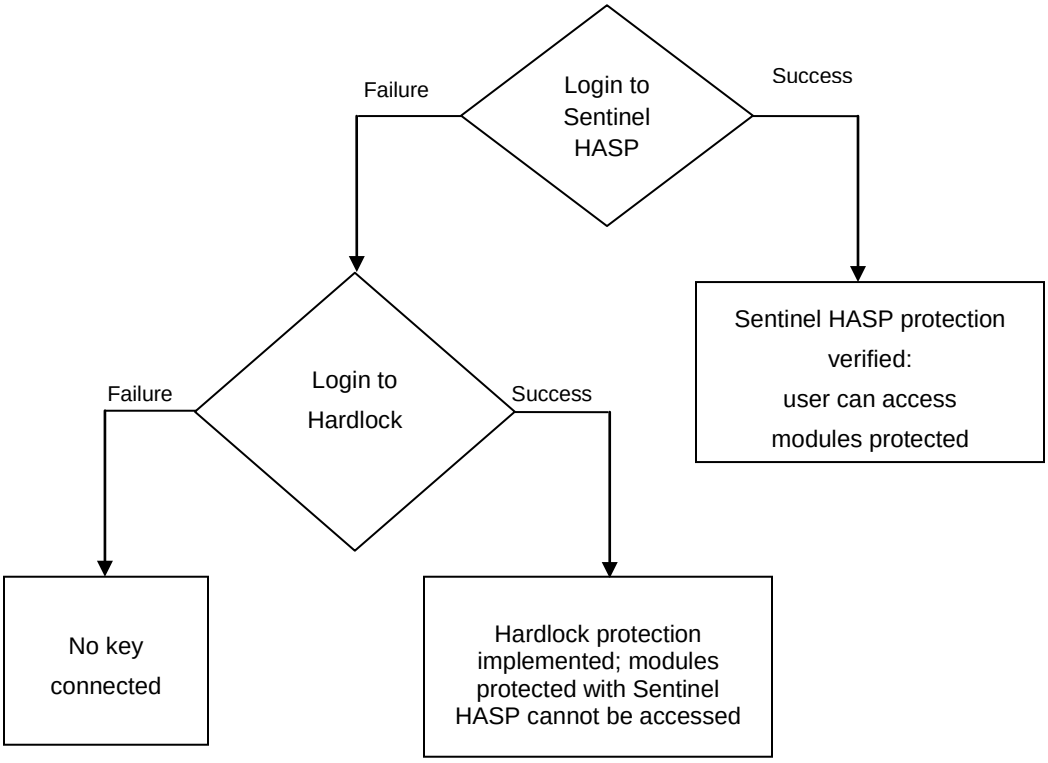
Stage 1: Combining Hardlock Protection with Sentinel HASP Protection

Stage 1 enables you to phase out your install base of Hardlock keys over a period of time, without necessitating immediate recall and replacement of the Hardlock keys. To achieve this, you create a version of your software that is able to identify both Hardlock and HASP HL keys. This could be a new version of your software, or the current version, with the ability to work with a HASP HL key. You can then start distributing HASP HL keys to all new customers, while existing users continue to use the Hardlock keys.

When the software runs, it tries to log into a HASP HL key. If a HASP HL key is found, Sentinel HASP protection is implemented. If no HASP HL key is found, the software then tries to log into a Hardlock key. If a Hardlock key is found, Hardlock protection is implemented.

In order to maximize security and implement the higher level of protection provided by Sentinel HASP concurrently with Hardlock protection of your software, you can protect selected files or modules with Sentinel HASP. Sentinel HASP-protected items will have greater security than those only protected by Hardlock. The Sentinel HASP-protected items can only be activated with a HASP HL key. In this case, if a Hardlock key is used, the modules protected with Hardlock will function, but modules protected with Sentinel HASP will not run.

The following flowchart shows the sequential flow when the protected software executes in Stage 1:



Implementing Stage 1

The following procedure details the steps you take in order to implement Stage 1 of the Hardlock-to-Sentinel HASP migration process. Where relevant, you are pointed to additional information in the Sentinel HASP documentation.

1. If you have not already done so, install Sentinel HASP Vendor Suite and introduce your Sentinel HASP Vendor keys. As part of the Vendor key introduction process, Sentinel HASP generates customized Sentinel HASP Run-time API libraries for your Vendor Code.

   (See *Sentinel HASP Installation Guide*, Part 1: *Installing the Sentinel HASP Software*.)

2. Integrate the Sentinel HASP Run-time Environment as part of your application setup.

   (See *Sentinel HASP Software Protection and Licensing Guide*, chapter *Distributing Sentinel HASP with Your Software*.)

   Note: If you are using Hardlock PCMCIA or the `hlvdd.dll`, install the **latest** legacy Hardlock driver, which incorporates the HASP HL driver.
   You can download the latest version from the Hardlock Support and Downloads Page.

3. Link the customized Sentinel HASP Run-time API libraries to the protected files as follows:

   - If you link to your customized Sentinel HASP Run-time API `.lib` files, remove the existing link to the Hardlock library files. The Sentinel HASP Run-time API `.lib` files contain both Sentinel HASP and Hardlock functionalities.
   - If you link to your customized Sentinel HASP Run-time API `.dll` files, do **not** remove the link to the Hardlock library files.
   - Include your customized Sentinel HASP Run-time API header files in your project. Do **not** remove included Hardlock headers.

   (See *Sentinel HASP Software Protection and Licensing Guide*, chapter *Sentinel HASP Run-time API Protection*)

4. To enable your software to work with Hardlock or Sentinel HASP protection, implement the decision tree on page 6 of this document, as follows:

   a. Use the Sentinel HASP Run-time API to log in to a key. If the login is successful, Sentinel HASP protection is invoked.

      (See *Sentinel HASP Software Protection and Licensing Guide*, chapter *Sentinel HASP Run-time API Protection*.)

   b. If the login to Sentinel HASP fails, log in using Hardlock functionality. If the Hardlock login is successful, Hardlock protection is invoked.

   c. If the login to Hardlock fails, the behavior of the application when no key is connected is invoked.

   Note: You can optionally enhance the security of selected items in your software by protecting them with Sentinel HASP. You can protect individual files using Sentinel HASP Envelope or Sentinel HASP Run-time API. You can protect code snippets and other data using the Sentinel HASP Run-time API. These protected items are only accessible when a HASP HL key is connected.

   Important: Do not protect the entire program with Sentinel HASP Envelope, because doing so will disable the use of HASP4 keys.

5. Supply all new customers with HASP HL keys. Only these customers can access modules protected with Sentinel HASP.
6. Gradually replace your install base of Hardlock keys with HASP HL keys, at your convenience.

Stage 2: Full Implementation of Advanced Sentinel HASP Functionality

This stage enables you to fully implement the advanced functionalities of the Sentinel HASP system, and gain the benefit of its increased security and licensing capabilities. After you implement full Sentinel HASP protection, all customers using this version of your software must use HASP HL keys.

The following procedure details the steps you take in order to implement Stage 2 of the Hardlock-to-Sentinel HASP migration process. Where relevant, you are pointed to additional information in the Sentinel HASP documentation.

Implementing Stage 2

Fully implement the advanced Sentinel HASP Run-time API by integrating Sentinel HASP functionalities into your code.

1. If you have not carried out Stage 1, implement steps 1-3 of Stage 1 in order to complete the following:
   a. Install Sentinel HASP Vendor Suite and introduce your Vendor keys.
   b. Integrate the Sentinel HASP Run-time Environment as part of your application setup.
   c. Link the Sentinel HASP Run-time API library to the protected files.
2. Replace all calls in the code to Hardlock with calls to Sentinel HASP. Refer to Table 4 on page 16 for a list of Hardlock functions and their Sentinel HASP counterparts.
3. Protect the software using Sentinel HASP Envelope.

   (See *Sentinel HASP Software Protection and Licensing Guide*, chapter *Sentinel HASP Envelope Protection*.)
4. Follow the instructions in the *Sentinel HASP Software Protection and Licensing Guide* to distribute your software (see the chapter *Distributing Sentinel HASP with Your Software*).
5. Ensure that all customers who receive the Sentinel HASP-protected software also receive HASP HL keys.

Migration Path 2—Gradual Migration from Hardlock to Sentinel HASP Using a Launcher Application

This migration path enables you to phase out your installation base of Hardlock keys—without necessitating the recall and replacement of the Hardlock, and without having to continue their distribution.

The migration is achieved by creating two versions of your software—one protected using Hardlock Espresso, and the other protected using Sentinel HASP Envelope. The two versions of the software are bundled with a launcher application. If the launcher detects that a Sentinel HASP key is accessed, the Sentinel HASP Envelope-protected version of your software is launched. If a Sentinel HASP key is not detected, the Hardlock Espresso-protected version of your software is launched.

This migration path enables you to support existing users who already have Hardlock keys, and to provide new users with the added protection available with Sentinel HASP protection keys.

When you are ready to fully switch to Sentinel HASP protection and licensing functionality, many of your users will already be using Sentinel HASP protection keys.

The following diagram summarizes the two stages for Migration Path 2.

Stage	1	2
Effort	Low	Medium
Install base	Remains Hardlock	Replace with HASP HL v3.21
Keys for new customers	HASP HL v3.21	HASP HL v3.21
Protection process	• Create two binaries – one protected using Hardlock Espresso, the other using Sentinel HASP Envelope • Create a launcher application using the Sentinel HASP Run-time API to search for a Sentinel HASP protection key • Switch between above binaries , depending on connected key	• Remove Hardlock implementation • Implement the Sentinel HASP Run-time API in your code and protect using Sentinel HASP Envelope
Security level	Same as Hardlock Espresso only	Very high
Flexibility level (licensing, portability)	Medium-High	Very high

Stage 1: Initial Implementation of Sentinel HASP Functionality

During Stage 1 of the migration process, you create two versions of your software—one protected using Hardlock Espresso, and the other protected using Sentinel HASP Envelope. The two versions of the software are bundled with a launcher application. The launcher application detects which version of your software to use.

Implementing Stage 1

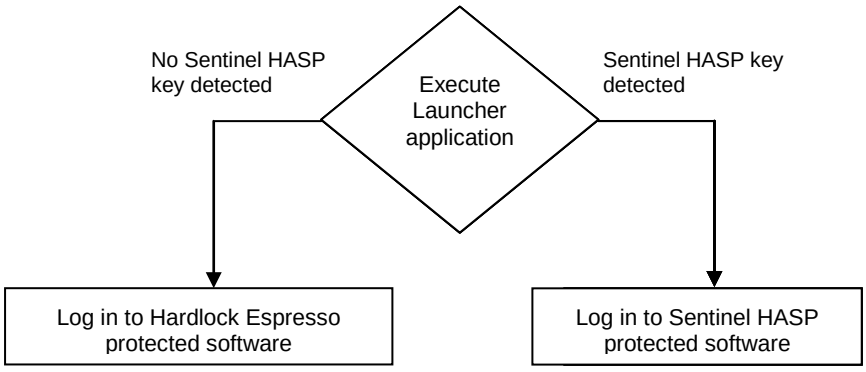
The following procedure details the steps required to implement the Hardlock Espresso-to-Sentinel HASP migration process. Where relevant, you are pointed to additional information in the Sentinel HASP documentation.

To implement Sentinel HASP functionality:

1. If you have not already done so, install Sentinel HASP Vendor Suite and introduce your Sentinel HASP Vendor keys.

   (See *Sentinel HASP Installation Guide*, Part 1: *Installing the Sentinel HASP Software*.)
2. Create a version of your software (for example, `program_hardlock.exe`) and implement protection using Hardlock Espresso and/or the Hardlock API.
3. Create a version of your software (for example, `program_haspsrm.exe`) and implement Sentinel HASP protection, using Sentinel HASP Envelope and/or the Sentinel HASP Run-time API.
4. Create a launcher application using the Sentinel HASP Run-time API that will detect whether a Sentinel HASP protection key is accessible. Program the following behavior:
   a. If a Sentinel HASP protection key is detected, the launcher launches `program_haspsrm.exe`.
   b. If a Sentinel HASP protection key is not detected, the launcher launches `program_hardlock.exe`.
5. Package both versions of the software with the launcher application.
6. Follow the instructions in the *Sentinel HASP Software Protection and Licensing Guide* to distribute your software (see the chapter *Distributing Sentinel HASP with your Software*).
7. Ensure that all customers who receive the Sentinel HASP-protected software also receive HASP HL keys.

The following flowchart shows the flow when the application launcher executes:



Stage 2: Full Implementation of Sentinel HASP Functionality

Stage 2 enables you to fully implement the functionalities of the Sentinel HASP system, thus gaining the benefit of its increased security and licensing capabilities. After you implement full Sentinel HASP protection, all customers using this version of your software must use Sentinel HASP protection keys.

Implementing Stage 2

The following procedure details the steps required to implement Stage 2 of the Hardlock-to-Sentinel HASP migration process. Where relevant, you are pointed to additional information in the Sentinel HASP documentation.

To implement full Sentinel HASP functionality:

1. If you have a Hardlock API, replace all calls to Hardlock in the code with calls to Sentinel HASP protection keys. See Table 4 on page 16 for a list of Hardlock functions and their Sentinel HASP equivalents.
2. Protect your software using Sentinel HASP Envelope.

   (See *Sentinel HASP Software Protection and Licensing Guide*, chapter *Sentinel HASP Envelope Protection*.)
3. Follow the instructions in the *Sentinel HASP Software Protection and Licensing Guide* to distribute your software (see chapter *Distributing Sentinel HASP with your Software*).
4. Ensure that all customers who receive the Sentinel HASP-protected software also receive Sentinel HASP protection keys.

Appendix

Table 1: Comparison of Hardlock and HASP HL Firmware v.3.21 Keys

Hardlock		**HASP HL (SRM)—**Firmware v.3.21	
Key Type	**Memory Size**	**Key Type**	**Memory Size**
EYE, Twin	–	Basic	–
EYE, Twin with memory	32 Bytes R/W 96 Bytes ROM	Pro	112 Bytes R/W 112 Bytes ROM
		Max	4 KB R/W 2 KB ROM
–	–	Time	4 KB R/W 2 KB ROM
–	–	Drive	4 KB R/W 2 KB ROM 512 MB / 2 GB Flash memory
HL-Server	32 Bytes R/W 96 Bytes ROM	Net	4 KB R/W 2 KB ROM
–	–	NetTime	4 KB R/W 2 KB ROM

Table 2: HL-Server Keys and Equivalent HASP HL Net and NetTime Keys

HL-Server	**HASP HL Net**	**HASP HL NetTime**
HL-Server 5	Net 10	NetTime 10
HL-Server 10	Net 10	NetTime 10
HL-Server 20	Net 50	NetTime 50
HL-Server 50	Net 50	NetTime 50
HL-Server 250	Net 250+	NetTime 250+

Table 3: Hardlock Tools and Functions and their Sentinel HASP Counterparts

Hardlock Tool / Functionality	**Sentinel HASP Tool / Functionality**
Encoding Hardlock keys	Keys are pre-encoded at SafeNet production site. Use your unique Sentinel HASP Vendor Code (stored in the Sentinel HASP Vendor keys)
Hardlock Bistro	Sentinel HASP Vendor Suite
Espresso	Sentinel HASP Envelope (part of Vendor Suite)
Cappuccino	Sentinel HASP Business Studio™ (part of Vendor Suite)
Latteccino	Sentinel HASP ToolBox

Hardlock Tool / Functionality	Sentinel HASP Tool / Functionality
	(part of Vendor Suite)
Hardlock Driver	Sentinel HASP Run-time Environment
Hardlock Server	Sentinel HASP Run-time Environment
Aladdin Monitor	Sentinel HASP Admin Control Center (part of the Sentinel HASP Run-time Environment)
Aladdin DiagnostiX	Sentinel HASP Admin Control Center (part of the Sentinel HASP Run-time Environment)
`Hlup.exe` (Hardlock upgrade)	Sentinel HASP Remote Update System (RUS)
Read Only memory	Read Only memory
–	Sentinel HASP key unique ID
Cappuccino	
Vendor Key manager	System uses signatures contained in Sentinel HASP Vendor keys
Insert order	Sentinel HASP Business Studio – Manage Orders
Creating licenses	Sentinel HASP Business Studio – Manage Products/Manage Orders
Programming the memory	Sentinel HASP Business Studio – Manage Products, or Sentinel HASP ToolBox
Reading the key memory	Sentinel HASP Business Studio – Manage Products, or Sentinel HASP ToolBox
Programming licenses to a key	Sentinel HASP Business Studio – Manage Orders/Produce Orders
Espresso	
Module address of Demo key	Demo key batch code is DEMOMA
Data Files node – data files filtering	Sentinel HASP Envelope – Enable data file encryption (DataHASP) check box in Protection Details pane
Data File node - data file encryption (`dfcrypt.exe`)	Sentinel HASP Envelope – Encrypt Data button in Protection Details pane
Using HL-RUS in Programs window	Sentinel HASP Envelope – Feature ID in Protection Details pane
Error Messages node	Sentinel HASP Envelope – User Messages pane
Local and network searches	Sentinel HASP Envelope – HASP search mode options in Protection Details pane
Sentinel HASP Run-time Environment	
`hldinst.exe` (command line)	`haspdinst.exe` (command line)
`Hldrv32.exe` (GUI driven)	`HASPUserSetup.exe` (GUI driven)
Driver installation API	Sentinel HASP Run-time Environment
Server, Monitor, DiagnostiX	
`Hlsw32.exe`	Sentinel HASP Run-time Environment
LM application – `hls32.exe`	Sentinel HASP Run-time Environment
LM application (service) –	Sentinel HASP Run-time Environment

Hardlock Tool / Functionality	Sentinel HASP Tool / Functionality
`hls32svc.exe`	
Monitor Setup – `aksmon32.exe`	Sentinel HASP Admin Control Center (part of the Sentinel HASP Run-time Environment)
DiagnostiX Setup – `Diagnostix.exe`	Sentinel HASP Admin Control Center (part of the Sentinel HASP Run-time Environment)

Table 4: Comparison of Hardlock API and Sentinel HASP Run-time API Functions

Hardlock API Function*	Sentinel HASP Run-time API Function
HL_LOGIN()	hasp_login() hasp_login_scope()
HL_LOGOUT()	hasp_logout()
HL_CODE()	hasp_encrypt() hasp_decrypt()
HL_MEMINFO()	hasp_get_sessioninfo() hasp_get_size()
HL_READID()	hasp_get_info()
HL_READ(); HL_READBL()	hasp_read()
HL_WRITE(); HL_WRITEBL()	hasp_write()
HL_PORTINF()	hasp_get_sessioninfo()
HL_ACCINF()	hasp_get_info()
HL_USERINF()	hasp_get_info()
HL_MAXUSER()	hasp_get_info()
HLM_WRITELICENSE()	hasp_update()
HLM_LOGIN()	In Sentinel HASP, hasp_update() does not require hasp_login() or hasp_logout()
HLM_LOGOUT()	
HLM_OCCUPYSLOT()	
HLM_FREESLOT()	
HLM_GETRUSINFO()	hasp_get_info()
HLM_CHECKCOUNTER()	hasp_get_info()
HLM_CHECKEXPDATE()	hasp_get_info()

* Hardlock functions that are not list are obsolete