Sentinel HASP®

Moving from HASP4 to Sentinel HASP

Migration Guide

SafeNet®

Copyrights and Trademarks

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 Software DRM 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.

The backward compatibility of HASP HL keys to HASP4 keys enables you to migrate to Sentinel HASP in stages.

About This Guide

This guide is intended for HASP4 users who wish to continue using a hardware-based protection solution, who also want to take advantage of the supreme strength of HASP HL key protection and of the advanced licensing options provided by Sentinel HASP.

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

This guide assumes that the reader has a good understanding of both the HASP4 and the Sentinel HASP systems.

The guide provides the following:

- An overview and guidelines for a three-stage migration path from HASP4 to Sentinel HASP, starting with an install base comprising only HASP4 keys
- Migration procedures, which are not documented in either the HASP4 documentation, or the *Sentinel HASP Software Protection and Licensing Guide* and Sentinel HASP Help documentation
- Tables that list the tools and functionalities of HASP4 and their counterparts in Sentinel HASP

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

For detailed information and procedures relating to HASP4, refer to the relevant HASP4 documentation.

Compatibility With HASP4

HASP HL keys are fully backward compatible with HASP4 keys. This means that deployed applications currently protected by HASP4 can work with HASP HL keys without the need to update software or drivers.

In addition, most of the Sentinel HASP Run-time API functions can also communicate with HASP4 keys. This enables you to create a version of your software protected by either type of hardware key. You can then supply the same software version to end users who already have HASP4 keys and to new customers who will receive HASP HL keys.

Three-Stage Migration Path

The three-stage migration path is designed to facilitate a gradual move towards improved security for your products, implementing the protection and licensing functionalities of Sentinel HASP. The stages are not interdependent, and it is possible to begin at Stage 2 or even at Stage 3, or to move from Stage 1 directly to Stage 3. Similarly, the time that you wait before moving from one stage to the next is entirely at your discretion. The following diagram summarizes the three-stage migration path.

Stage	1	2	3
Effort	–	+	++
Install base	Remains HASP4	Remains HASP4	Replace all with HASP HL v3.21
Keys for new users	HASP HL v3.21	HASP HL v3.21	HASP HL v3.21
Protection process	No change to code. Software remains protected with HASP4	Leave HASP4 API, add Sentinel HASP API. Use HASP4 Envelope.	Fully implement Sentinel HASP API and Envelope
Security level	+	++	++++

Stage 1: Using HASP HL keys as HASP4 keys

HASP HL keys are fully backward compatible with HASP4 keys, so that your software can work with either key. At this initial stage, it is not necessary to make any changes to your software or drivers. This enables you to start to ship HASP HL keys to your customers and gradually replace your install base of HASP4 keys with HASP HL firmware v.3.21 keys at your convenience. When you decide to move to Stage 2 or 3 of the migration process and protect your software with Sentinel HASP, you will already have deployed HASP HL keys.

Proceed as follows:

1. Leave your install base with the HASP4 keys they are currently using.
2. Start distributing HASP HL keys with new purchases. At this stage you do not need to make any changes to your software, which remains protected by HASP4 security.

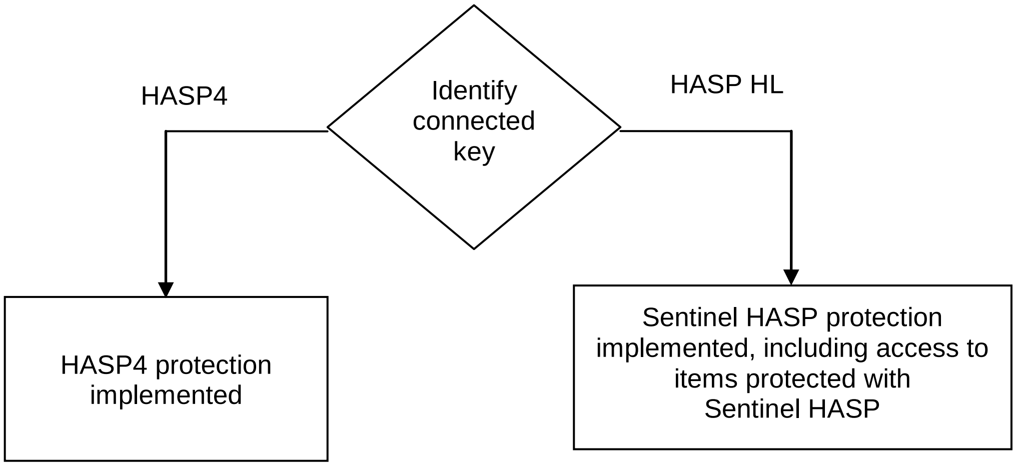
Stage 2: Combining HASP4 and Sentinel HASP Protection

In this stage, you create a version of your software that adds Sentinel HASP protection to the existing HASP4 protection. This stage is usually implemented with a new release of the protected product.

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 HASP4 key. If a HASP4 key is found, HASP4 protection is implemented.

In order to maximize security and implement the higher level of protection provided by Sentinel HASP concurrently with HASP4 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 HASP4. The Sentinel HASP-protected items can only be activated with a HASP HL key. In this case, if a HASP4 key is used, the modules protected with HASP4 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 2:


HASP4
Identify connected key
HASP HL
HASP4 protection implemented
Sentinel HASP protection implemented, including access to items protected with Sentinel HASP

Implementing Stage 2

You can maintain your legacy HASP4 protection simultaneously with the added protection and licensing capabilities of Sentinel HASP. This is possible because the HASP HL key is divided into two segments, each of which functions as a separate key. The legacy key segment provides full HASP4 functionality. The Sentinel HASP key segment provides the advanced Sentinel HASP capabilities. When you combine HASP4 and Sentinel HASP functionality, the two separate key segments require a dual process in the production phase and may also require a dual process at the end-user site, depending on the functionality applied.

Following is a breakdown of the required steps when you choose to retain legacy HASP4 protection in your code concurrently with Sentinel HASP functionality. Where relevant, you are pointed to additional information in the *Sentinel HASP Software Protection and Licensing Guide.*

1. Installing Sentinel HASP and Introducing your Vendor Keys

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. (For more information, see the *Sentinel HASP Installation Guide*.)

2. Linking to Sentinel HASP Run-time API Libraries

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 HASP4 library files. The Sentinel HASP Run-time API `.lib` files contain both Sentinel HASP and HASP4 functionalities.
- If you link to your customized Sentinel HASP Run-time API `.dll` files, do **not** remove the link to the HASP4 library files.
- Include your customized Sentinel HASP Run-time API header files in your project. Do **not** remove included HASP4 headers.

3. Enabling Your Software to Work With HASP4 or Sentinel HASP Protection

To enable your software to work with HASP4 or Sentinel HASP protection, implement the decision tree illustrated on page 7 of this document, as follows:

a. Use the Sentinel HASP Run-time API `hasp_get_info()` function to identify a key. If a HASP HL key is identified, invoke Sentinel HASP protection; if a HASP HL key is **not** identified, invoke HASP4 protection.

(See the *Sentinel HASP Software Protection and Licensing Guide*, appendix “Sentinel HASP Run-time API Reference”.)

b. If neither a HASP HL nor a HASP4 key is detected, invoke the behavior of the application when no key is detected.

c. To increase security, we recommend that you envelope your application. In Stage 2 of the migration process, the protected software must be able to run with either HASP4 or HASP HL keys. Therefore, you cannot use Sentinel HASP Envelope to wrap the entire application. Wrap your application using the latest HASP4 Envelope.

4. Protecting Selected Items With Sentinel HASP

Optionally, you can 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.

When using the Sentinel HASP Run-time API, note the following:

- For 32-bit applications, you can leave the existing HASP4 API calls in your code.
- If you migrate your application to 64-bit, remove the HASP4 API calls and replace them with corresponding Sentinel HASP Runtime API calls where necessary.

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

5. Licensing, Production and End-User License Management

The Sentinel HASP and legacy HASP4 segments in a HASP HL key are managed separately, and the licensing, production and updating of each segment are implemented using the relevant tools.

The following table details the tools and definitions you must use to manage each of the key segments.

Management of…	Sentinel HASP Segment	Legacy HASP 4 Segment
Licensing	Sentinel HASP Business Studio™	HASPEdit
Production and Creation of Remote Update Files	Sentinel HASP Business Studio	♦ DOS RUS—Vendor Utility ♦ Win32 RUS—API sample
Remote Update at End-User Site	♦ Sentinel HASP Remote Update System (RUS) ♦ Sentinel HASP Run-time API `hasp_update()` function	♦ DOS RUS—Customer Utility ♦ Win32 RUS—API sample
Key Monitoring	Sentinel HASP Admin Control Center (part of the Sentinel HASP Run-time Environment)	Aladdin Monitor
License Management at End-User Site	HASP License Manager, interfaced through the Sentinel HASP Admin Control Center (part of the Sentinel HASP Run-time Environment)	HASP License Manager

6. Replacing HASP4 Drivers With Sentinel HASP Run-time Environment

Replace the HASP4 drivers that are included in your application setup with Sentinel HASP Run-time Environment.

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

7. Distributing Your Software

Sentinel HASP Run-time Environment is compatible with HASP4 keys, therefore there is no need to distribute HASP4 drivers in addition to the Sentinel HASP Run-time Environment.

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

8. Distributing End-User Keys

The new version of your software can still run with HASP4 keys, so that you do not need to recall your entire install base.

a. If you have protected selected modules with Sentinel HASP, you must supply customers who will be using these modules with HASP HL firmware v.3.21 keys. Existing customers who will not be using these modules can continue using HASP4 keys.

b. It is recommended that you supply all **new** customers with HASP HL firmware v.3.21 keys.

Stage 3: Full Implementation of Sentinel HASP

In this stage, fully implement the advanced functionalities of the Sentinel HASP system and gain the benefit of its increased security and licensing abilities. 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 3 of the HASP4-to-Sentinel HASP migration process. Where relevant, you are pointed to additional information in the *Sentinel HASP Software Protection and Licensing Guide*.

Implementing Stage 3

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

1. Installing Sentinel HASP and Introducing Your Vendor Keys

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.*)

2. Linking to Sentinel HASP Run-time Libraries

If you have not already done so, link the customized Sentinel HASP Run-time API library to the protected files, as described in step 2 of Stage 2.

3. Defining Sentinel HASP Feature IDs

If you used program numbers in HASP4, define new Feature IDs in Sentinel HASP Business Studio to replace the HASP4 program numbers.

(See *Sentinel HASP Software Protection and Licensing Guide*, chapter “Implementing Your Sentinel HASP Licensing Plan”.)

4. Protecting Your Software With Sentinel HASP

a. Insert calls to Sentinel HASP in your code. Refer to Table 4 on page 18 for a comparison of HASP4 API and Sentinel HASP Run-time API functions.

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

Note:

- For 32-bit applications, it is not necessary to remove the existing HASP4 API calls. These calls are supported by the Sentinel HASP Run-time API libraries.
- If you migrate your application to 64-bit, remove all HASP4 API calls.

b. Wrap the software using Sentinel HASP Envelope.

(See *Sentinel HASP Software Protection and Licensing Guide*, chapter “Sentinel HASP Envelope Protection”.)

5. Encrypting Data

In HASP4, a single proprietary encryption key was used to encrypt all data. Sentinel HASP applies a unique AES encryption key for each Feature ID, providing much stronger protection. In order to implement this protection, you must replace all data that was encrypted using HASP4 with data that is encrypted using Sentinel HASP as follows:

a. Decrypt the data using HASP4, or locate the original unencrypted data.

b. Encrypt the data using Sentinel HASP, using the Feature ID relevant to that data segment.

c. Ensure that the `hasp_login()`call for the data segment is defined with the same Feature ID that you used to encrypt the data. Replace the HASP4-encrypted data in your code with the Sentinel HASP-encrypted data.

(See *Sentinel HASP Software Protection and Licensing Guide*, chapter “Protection Strategies”.)

6. Writing Data to the HASP HL Key

The backward compatibility of Sentinel HASP with HASP4 enables you to continue writing data to the key using `Services 3,4,50,51,52, 53, 74,75,76` and `77`.

However, it is highly recommended that you take advantage of the secure communication channel that Sentinel HASP adds to the HASP HL key. To do this, write your existing data to one of the following Sentinel HASP memory files:

- Data in the `HASP_FILEID_RW` memory file enables you to write to, or read from, a key’s memory during run-time.
- Data in the `HASP_FILEID_RO` memory file enables you to read from a key’s memory during run-time. Data may be written to this memory file:
  - During your production process
  - When updating the key using the Sentinel HASP RUS
  - When updating the key using the Sentinel HASP Run-time API `hasp_update()` function

(See *Sentinel HASP Software Protection and Licensing Guide*, chapters “Implementing Your Sentinel HASP Licensing Plan” and “Sentinel HASP Remote Update System”.)

7. Wrapping Your Software With Sentinel HASP Envelope

Wrap the software using Sentinel HASP Envelope.

(See *Sentinel HASP Software Protection and Licensing Guide*, chapter "Sentinel HASP Envelope Protection".)

8. Replacing HASP4 Drivers With Sentinel HASP Run-time Environment

Replace the HASP4 drivers that are included in your application setup with Sentinel HASP Run-time Environment.

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

9. Licensing Your Software With Sentinel HASP

Use Sentinel HASP Business Studio to license your software.

(See *Sentinel HASP Software Protection and Licensing Guide*, Part 3: "Licensing".)

10. Distributing Your Software

Follow the instructions in the *Sentinel HASP Software Protection and Licensing Guide* to distribute your software (chapter "Distributing Sentinel HASP with Your Software").

11. Distributing End-User Keys

Ensure that all customers who receive the Sentinel HASP-protected software also receive HASP HL keys.

12. Using HASP HL Net Keys

When Sentinel HASP is fully implemented and there are no HASP4 calls in your code, you do not need to have HASP4 license management tools installed (HASP License Manager or Aladdin Monitor). The Sentinel HASP Run-time Environment installation includes the Sentinel HASP Admin Control Center and the HASP License Manager that manages HASP HL Net and HASP HL NetTime keys.

Appendix

Table 1: Comparison of HASP4 With HASP HL Firmware v.3.21 Keys

HASP4 Keys		**HASP HL Keys**—Firmware v.3.21	
Key Type	**Memory Size**	**Key Type**	**Memory Size**
Standard	–	Basic	–
M1	112 Bytes	Pro	112 Bytes backward-compatible memory 112 Bytes Read/Write memory 112 Bytes ROM
M4	496 Bytes	Max	4 KB backward-compatible memory 4 KB Read/Write memory 2 KB ROM
Time	512 Bytes	Time	4 KB backward-compatible memory 4 KB Read/Write memory 2 KB ROM
–	–	Drive	512 MB / 2 GB Flash memory 4 KB backward-compatible memory 4 KB Read/Write memory 2 KB ROM
Net	496 Bytes	Net	4 KB backward-compatible memory 4 KB Read/Write memory 2 KB ROM
–	–	NetTime	4 KB Read/Write memory 2 KB ROM

Table 2: HASP4 Net Keys and Equivalent HASP HL Net Keys

HASP4 Net	HASP HL Net	HASP HL NetTime
Net 5	Net 10	NetTime 10
Net 10	Net 10	NetTime 10
Net 20	Net 50	NetTime 50
Net 50	Net 50	NetTime 50
Net 100	Net 250+	NetTime 250+
Net Unlimited	Net 250+	NetTime 250+

Table 3: HASP4 Tools and Functions and Their Sentinel HASP Counterparts

HASP4 Application / Functionality	Sentinel HASP Application / Functionality
Passwords	Sentinel HASP Vendor Code
Envelope *(automatic protection tool)*	Sentinel HASP Envelope (part of the Sentinel HASP Vendor Suite)
API	Sentinel HASP Run-time API
HASPDemo *(GUI-driven sample demonstrating API calls)*	♦ Sentinel HASP ToolBox (part of the Sentinel HASP Vendor Suite; ♦ API reference and Sentinel HASP Run-time code generator)
Driver	Sentinel HASP Run-time Environment
NetHASP License Manager	HASP License Manager, interfaced through the Sentinel HASP Admin Control Center (part of the Sentinel HASP Run-time Environment)
Aladdin Monitor	Sentinel HASP Admin Control Center (part of the Sentinel HASP Run-time Environment)
Aladdin DiagnositX	Sentinel HASP Admin Control Center (part of the Sentinel HASP Run-time Environment)
Aladdin DiagnostiX Memory Beamer	Sentinel HASP Remote Update System (RUS)—`hasprus.exe`
RUS—create remote update files: ♦ DOS RUS—Vendor Utility ♦ Win32 RUS—API sample	Sentinel HASP Business Studio (part of the Sentinel HASP Vendor Suite, generates V2C files)
RUS—remote update at end-user site: ♦ DOS RUS—Vendor Utility ♦ Win32 RUS—API sample	♦ Sentinel HASP RUS (`hasprus.exe`—generated by Sentinel HASP Business Studio and shipped to end users) ♦ `hasp_update()`, part of Sentinel HASP Run-time API

HASP4 Application / Functionality	Sentinel HASP Application / Functionality
HASPEdit	
HASPEdit—String encryption/decryption	Sentinel HASP ToolBox—Encrypt/Decrypt functions
HASPEdit—FAS *(generating licenses)*	Sentinel HASP Business Studio—Manage Orders and Produce Orders
HASPEdit—Programming the user memory area	♦ Sentinel HASP Business Studio—define the data in the Define Memory Data tab, when creating a new Product. The data is written to the key during order production. ♦ Sentinel HASP ToolBox—`hasp_write()` function
HASPEdit—Programming the memory area reserved for licensing (FAS area)	HASP HL keys employ License On Chip technology, meaning that only the HASP chip may access the Sentinel HASP raw license data. Use Sentinel HASP Business Studio to define and manage license-related information.
HASPEdit—setting the real-time clock	Key time set to UTC during key manufacture
HASPEdit—programming licenses to a key	Sentinel HASP Business Studio—Manage Products and Manage Orders
HASPEdit—reading the key memory	Sentinel HASP Business Studio—under Manage Orders, Check in C2V and Key Data
HASPEdit—reading the HASP ID Number	Sentinel HASP Business Studio—view the connected key under Manage Orders, Check in C2V and Key Data
HASPEdit—Create programming utility for mass production of keys	Sentinel HASP Business Studio— under Manage Orders, New—specify the number of keys
Envelope	
Support for Demo key	Sentinel HASP Envelope—Demo key is default, change to non-demo in Sentinel HASP Profile pane
DataHASP Tab—data file configuration	Sentinel HASP Envelope—Protection Details pane, Enable data file encryption (DataHASP)
Using FAS	Sentinel HASP *Protect Once—Deliver Many* licensing technology. Sentinel HASP Envelope protects per Feature ID. Business logic and license production are implemented in Sentinel HASP Business Studio.
Using HASP ID Number	Sentinel HASP Envelope options—Protection Details pane, Advanced, Enable Custom HASP Login Scope
Error Messages Tab	Sentinel HASP Envelope—User Messages pane
NetHASP parameters	Sentinel HASP Envelope ♦ Search mode—Protection Details pane ♦ Advanced search options—Protection Details pane, Advanced, Enable Custom HASP Login Scope
Enhanced security Envelope switches	Sentinel HASP Envelope—Protection Details pane, Advanced

HASP4 Application / Functionality	Sentinel HASP Application / Functionality
Driver	
hinstall.exe (command line)	haspdinst.exe (command line)
hdd32.exe (GUI driven)	HASPUserSetup.exe (GUI driven)
Driver installation API	Sentinel HASP Run-time Installation API
License Manager, Monitor, DiagnostiX	
lmsetup.exe	Sentinel HASP Run-time Environment installers
LM application—nhsrvw32.exe	♦ Sentinel HASP Run-time Environment—HASP License Manager Service (LMS) ♦ hasplms.exe
LM application (service)—nhsrvice.exe	♦ Sentinel HASP Run-time Environment—HASP License Manager Service (LMS) ♦ hasplms.exe
Monitor Setup—aksmon.exe	Sentinel HASP Run-time Environment installers—install the Sentinel HASP Admin Control Center
nethasp.ini—client configuration file	♦ Sentinel HASP Run-time API—hasp_login_scope() ♦ Sentinel HASP Admin Control Center (part of the Sentinel HASP Run-time Environment) ♦ hasplm.ini
nhsrv.ini—LM configuration file	♦ Sentinel HASP Admin Control Center (part of the Sentinel HASP Run-time Environment) ♦ hasplm.ini
DiagnostiX setup—aksdiag32_setup.exe	Sentinel HASP Run-time Environment installers

Table 4: Comparison of HASP4 API and Sentinel HASP Run-time API Functions

HASP4 API Service*	Sentinel HASP Run-time API Function

`Service 5: HaspStatus`	`hasp_get_info()`
	`hasp_get_size()`

`Service 60, 88: HaspEncodeData`	`hasp_encrypt()`
`Service 61, 89: HaspDecodeData`	`hasp_decrypt()`
`Service 6, 46, 78: HaspID`	`hasp_get_info()`–(Format Template: Accessible Keys)
`Service 3, 44, 75: ReadWord/Byte`	`hasp_read()`
`Service 50, 52, 77: ReadBlock`	`hasp_read()`
`Service 4, 45, 74: WriteWord/Byte`	`hasp_write()`
`Service 51, 53, 76: WriteBlock`	`hasp_write()`
`Service 70: SetTime`	Not relevant in Sentinel HASP. Key time is set to UTC during key manufacture.

`Service 71: GetTime`	`hasp_get_rtc(),`
	`hasp_hasptime_to_datetime()`

`Service 72: SetDate`	Not relevant in Sentinel HASP. Key time is set to UTC during key manufacture.

`Service 73: GetDate`	`hasp_get_rtc()`
	`hasp_hasptime_to_datetime()`

`Service 42: Login`	`hasp_login()`
	`hasp_login_scope()`

`Service 40: LastStatus`	Not relevant in Sentinel HASP, Sentinel HASP Run-time API functions return all necessary data
`Service 43: Logout`	`hasp_logout()`
`Service 48: IdleTime`	Sentinel HASP License Manager automatically detects idle sessions and releases them.

* Unlisted HASP4 functions are obsolete