# Sentinel HASP®

Moving from Sentinel Hardware Keys 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 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 a software product's protection and licensing life cycle

The level of protection for your software is determined by the locking type you choose—hardware-based or software-based. Sentinel HASP hardware-based protection, which utilizes HASP HL keys, provides the safest and strongest level of protection. Sentinel HASP software-based protection, which utilizes HASP SL keys and software activation, provides electronic software and license distribution. Both keys are supported by the same set of tools and APIs, and the transition between them is transparent.

## About this Guide

SafeNet provides you with a broad spectrum of DRM products. Without losing security, you can use this guide to migrate from Sentinel Hardware Keys to the Sentinel HASP product and gain the capabilities of having a licensing solution that provides strong security against piracy (by using HL keys) and the flexibility of software-based licensing (SL keys) within the same implementation.

The primary factors for migrating from Sentinel Hardware Keys to Sentinel HASP are:

- Business logic is more flexible using Sentinel HASP Business Studio™.
- The same implementation of Sentinel HASP provides access to both software-based (SL) and hardware-based (HL) license protection.
- No loss of license protection security in moving from Sentinel Hardware Keys to Sentinel HASP

The guide assumes that the reader has a good understanding of both the Sentinel Hardware Keys and the Sentinel HASP systems and provides the following:

- Migration paths from Sentinel Hardware Keys to Sentinel HASP, each with an overview, guidelines, and discussion of advantages and disadvantages.
- Procedures relating to the migration that are not documented in either the Sentinel Hardware Keys documentation or the *Sentinel HASP Software Protection and Licensing Guide* and Help documentation
- Tables comparing Sentinel HASP and Sentinel Hardware Keys, tools, and API functions.

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 Sentinel Hardware Keys, refer to the *Sentinel Hardware Keys Developer's Guide*.

## Available Migration Paths

Both migration paths contain two stages; each designed to facilitate your move from Sentinel Hardware Keys to the protection and licensing functionalities of Sentinel HASP.

The stages are not interdependent, meaning it is possible to begin at Stage 2. Similarly, the time that you wait before moving from one stage to the next is entirely at your discretion.

**Migration Path 1** provides a way to phase out your installation base of Sentinel Keys over time—without necessitating the recall and replacement of Sentinel keys, and without having to continue their distribution.

Using Migration Path 1, and creating a version of your software that recognizes both Sentinel and HASP HL keys, you can start distributing HASP HL keys to new customers while existing customers continue using their Sentinel keys. You can then gradually replace your install base of Sentinel keys with HASP HL keys.

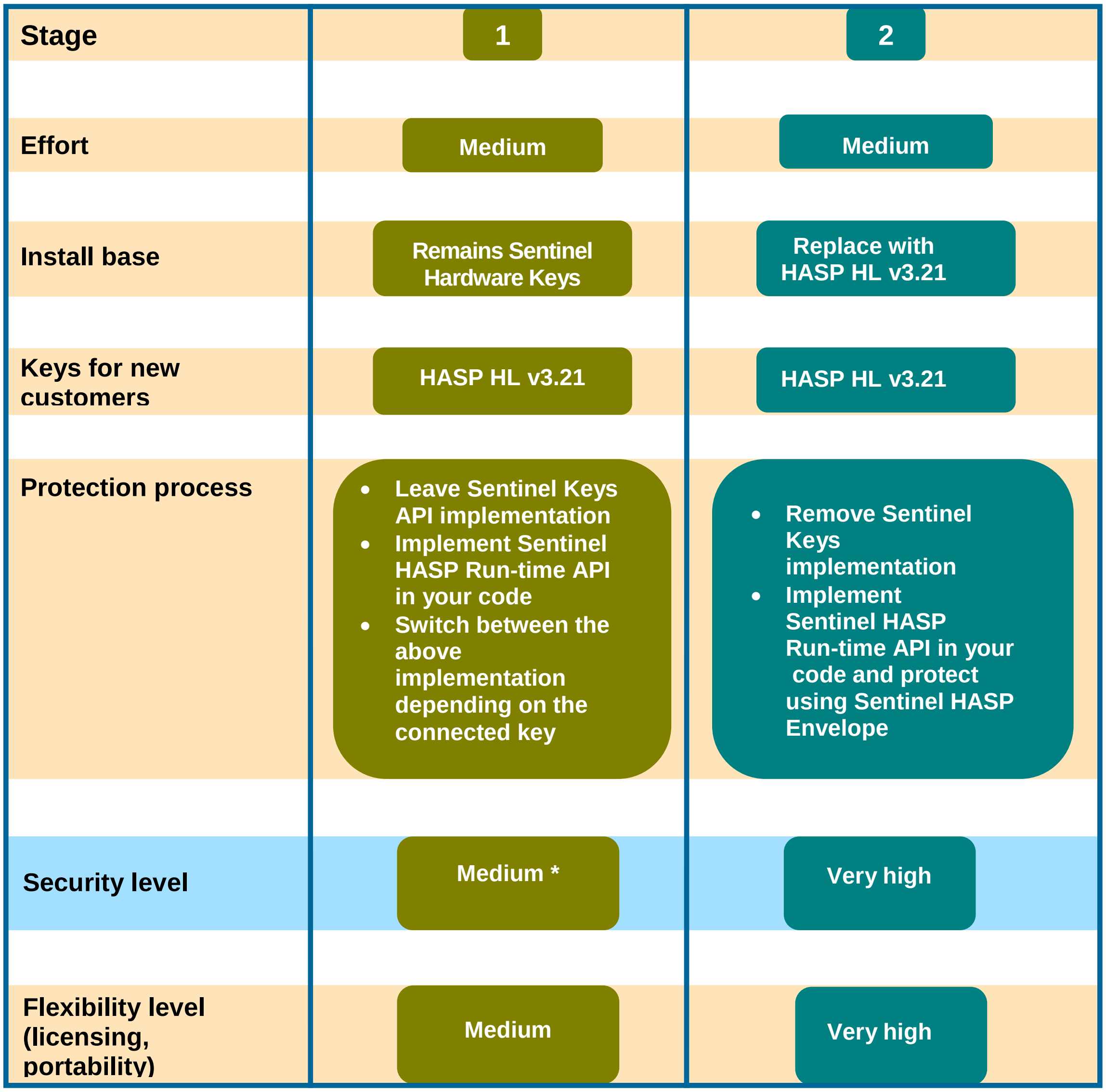
**Migration Path 2** enables a gradual transition from Sentinel Keys to Sentinel HASP. A Sentinel Keys-protected version and a Sentinel HASP-protected version of your software are distributed, together with a launcher application. The launcher detects whether a Sentinel HASP protection key is connected to the computer and launches the appropriate version of the program. For more information, see *Migration Path 2—Gradual Migration from Sentinel Keys to Sentinel HASP* on page 10.

# Migration Path 1— Sentinel HASP and Sentinel Keys Combined Implementation

This two-stage migration path enables you to phase out your installation base of Sentinel Keys over time, without necessitating immediate recall and replacement of the Sentinel keys and without having to continue their distribution. To achieve this status, you create a version of your software that is able to identify both Sentinel and HASP HL keys. This could be a completely new version of your software or the current version (with just the migration path changes implemented), 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 Sentinel keys.

The time that you wait before moving from one stage to the next is entirely at your discretion. You can even skip Stage 1 and proceed directly to Stage 2.

The following diagram summarizes the two stages for Migration Path 1.



| Stage | 1 | 2 |
|---|---|---|
| Effort | Medium | Medium |
| Install base | Remains Sentinel Hardware Keys | Replace with HASP HL v3.21 |
| Keys for new customers | HASP HL v3.21 | HASP HL v3.21 |
| Protection process | • Leave Sentinel Keys API implementation<br>• Implement Sentinel HASP Run-time API in your code<br>• Switch between the above implementation depending on the connected key | • Remove Sentinel Keys implementation<br>• Implement Sentinel HASP Run-time API in your code and protect using Sentinel HASP Envelope |
| Security level | Medium * | Very high |
| Flexibility level (licensing, portability) | Medium | Very high |

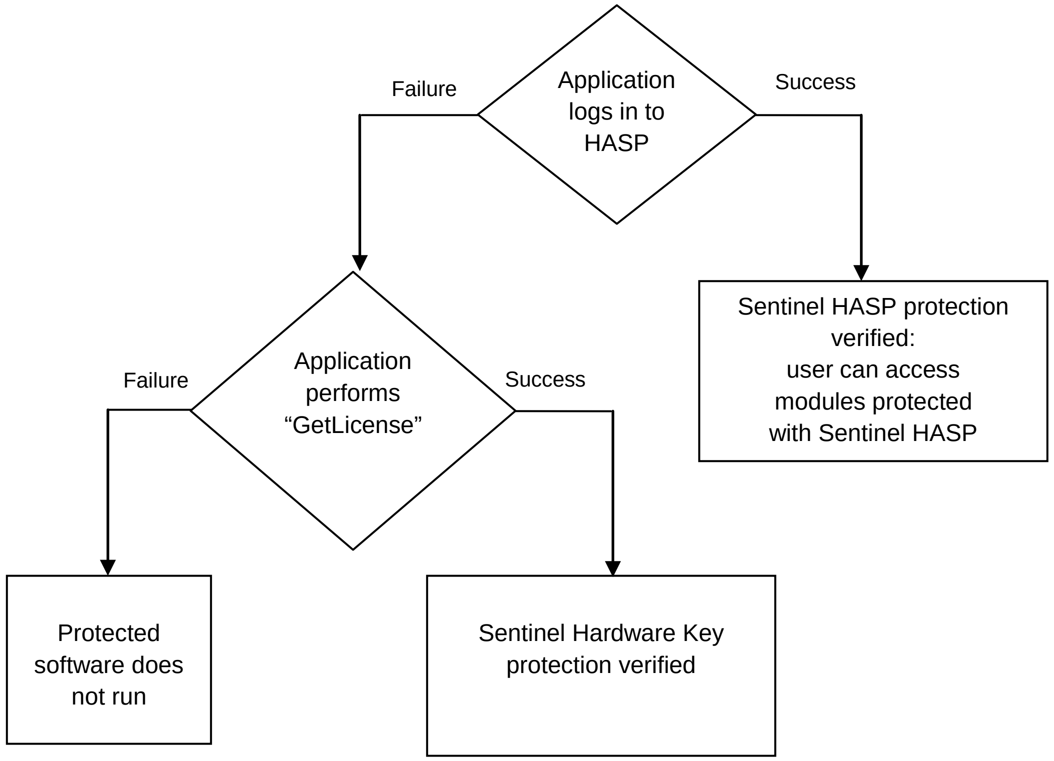
* Due to API-only implementation

# Stage 1: Combining Sentinel Keys with Sentinel HASP Protection

When your software runs, it attempts to log in to a HASP HL key. If a HASP HL key is detected, Sentinel HASP protection is used. If a HASP HL key is not detected, the software attempts to log in to a Sentinel key. If a Sentinel key is detected, Sentinel Keys protection is used.

During the period that your application supports both Sentinel Hardware Keys and Sentinel HASP, you are imposing an additional workload on your development and support departments. Therefore, an important objective during this stage is to provide incentive for your customers to switch from Sentinel Hardware Keys to Sentinel HASP. One way of accomplishing this is to protect new features in your application using only HASP HL keys. Customers that want to use the new features will be required to switch to HASP HL keys.

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


Application logs in to HASP
Failure
Success
Sentinel HASP protection verified: user can access modules protected with Sentinel HASP
Application performs "GetLicense"
Failure
Success
Protected software does not run
Sentinel Hardware Key protection verified

## Implementing Stage 1

The following procedure details the steps required to implement Stage 1 of the Sentinel Keys-to-Sentinel HASP migration process. Where relevant, you are pointed to additional information in the *Sentinel HASP Software Protection and Licensing Guide*.

**To implement both Sentinel HASP and Sentinel Keys functionality:**

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.

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

2. Integrate the Sentinel HASP RTE Installer into your application.

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

3. Include your Sentinel HASP Run-time API header files in your project. Do **not** remove included Sentinel Keys headers.

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

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

   a. Use the Sentinel HASP Run-time API to log in to a Sentinel HASP protection 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” and appendix “Sentinel HASP Run-time API Reference.”)

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

   c. If the Sentinel Keys “GetLicense” call fails, the behavior of the application when no key is detected is invoked.

   **Note:** You can optionally enhance the security of selected items in your application by protecting them using Sentinel HASP Envelope and using the encryption functions available in the HASP Run-time API. For maximum security, any file you choose to protect using the Sentinel HASP Run-time API, including a DLL, should also be protected using Sentinel HASP Envelope. You can also protect code snippets and other data using the API. These protected items will only be accessible when a HASP HL key is connected. (By protecting new functionality with Sentinel HASP only, you can provide incentive to existing customers to switch to HASP HL keys.)

   **Important:** Do not protect the entire application with Sentinel HASP Envelope, because doing so will disable the use of Sentinel 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 Sentinel keys with HASP HL keys, at your convenience.

# Stage 2: Full Implementation of Advanced Sentinel HASP Functionality

Stage 2 enables you to fully implement the functionalities of the Sentinel HASP system, thus gaining the benefit of its advanced business logic. After you implement full Sentinel HASP protection, all customers using this version of your software must use HASP HL keys.

## Implementing Stage 2

The following procedure details the steps required to implement Stage 2 of the Sentinel Keys-to-Sentinel HASP migration process. Where relevant, you are pointed to additional information in the *Sentinel HASP Software Protection and Licensing Guide*.

**To implement advanced Sentinel HASP functionality:**

- If you have not already implemented Stage 1, perform steps 1-3 of Stage 1 in order to complete the following:
  a. Install Sentinel HASP Vendor Suite and introduce your Sentinel HASP Vendor keys. As part of the Sentinel HASP 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*.)

  b. Integrate the Sentinel HASP RTE Installer into your application.

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

  c. Link the Sentinel HASP Run-time API library to the application to be protected.

- Replace all calls to Sentinel Keys in the code with calls to HASP HL keys.
  See Table 3: *Comparison of Sentinel Hardware Key API Functions and Sentinel HASP Run-time API Functions* on page 16 for a list of Sentinel Keys functions and their Sentinel HASP equivalents.

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

- Protect your software using Sentinel HASP Envelope.

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

- Follow the instructions in the *Sentinel HASP Software Protection and Licensing Guide* to distribute your software.

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

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

# Migration Path 2—Gradual Migration from Sentinel Keys to Sentinel HASP using a Launcher Application

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

The migration is achieved by creating two versions of your software—one protected using Sentinel Keys, 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 Sentinel Keys-protected version of your software is launched.

This migration path enables you to support existing users who already have Sentinel 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 3.

| Stage | 1 | 2 |
|---|---|---|
| Effort | Low | Medium |
| Install base | Remains Sentinel Hardware Keys | 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 Sentinel Keys, the other using Sentinel HASP Envelope<br>• Create a launcher application using the Sentinel HASP Run-time API to search for a Sentinel HASP protection key<br>• Switch between above binaries , depending on connected key | • Remove Sentinel Keys implementation<br>• Implement the Sentinel HASP Run-time API in your code and protect using Sentinel HASP Envelope |
| Security level | Same as Sentinel Keys only | Very high |
| Flexibility level (licensing, portability) | Medium | 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 Sentinel Keys, 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 Sentinel Keys-to-Sentinel HASP migration process. Where relevant, you are pointed to additional information in the Sentinel HASP documentation.

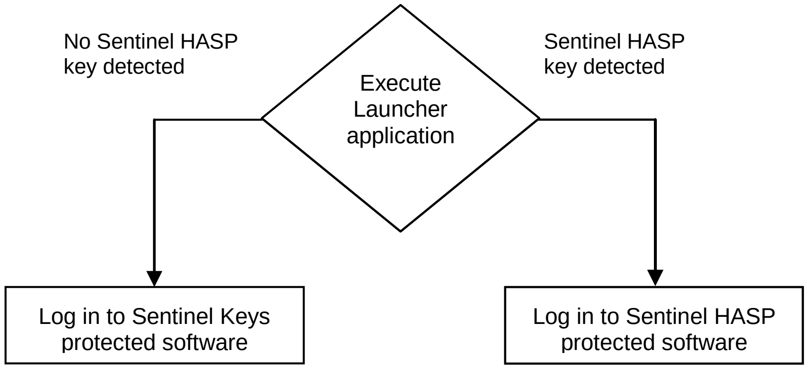
**To implement Sentinel HASP functionality:**

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

   (For more information, see the *Sentinel HASP Installation Guide*.)
2. Create a version of your software (for example, `program_shk.exe`) and implement Sentinel Keys protection using the Sentinel Shell, Sentinel API, or both.
3. Create a version of your software (for example, `program_hasp.exe`) and implement Sentinel HASP protection, using Sentinel HASP Envelope, the Sentinel HASP Run-time API, or both.
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_hasp.exe`.
   b. If a Sentinel HASP protection key is not detected, the launcher launches `program_shk.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 *Sentinel HASP Software Protection and Licensing Guide*, 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:


No Sentinel HASP key detected
Execute Launcher application
Sentinel HASP key detected
Log in to Sentinel Keys protected software
Log in to Sentinel HASP protected software

# 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 advances business logic. 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 Sentinel Keys-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. Replace all calls to Sentinel Keys in the code with calls to HASP HL keys. See Table 3: *Comparison of Sentinel Hardware Key API Functions and Sentinel HASP Run-time API Functions* on page 16 for a list of Sentinel Keys functions and their Sentinel HASP equivalents.

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

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 *Sentinel HASP Software Protection and Licensing Guide*, 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 A: Sentinel HASP and Sentinel Hardware Keys Comparison Tables

## Table 1: Comparison of HASP HL Firmware v.3.21 Keys and Sentinel Hardware Keys

| Model Type | HASP HL | Sentinel Hardware Keys |
|---|---|---|
| Basic<br>♦ No read/write memory functionality<br>♦ Perpetual license<br>♦ Locally connected key | HASP HL Basic | None |
| Memory<br>♦ Read/write and read-only memory<br>♦ Locally connected key | HASP HL Pro<br>♦ 112 bytes R/W + 112 bytes ROM<br>HASP HL Max<br>♦ 4 KB R/W + 2 KB ROM | Sentinel S<br>♦ 8 KB (1.75 KB accessible)<br>Sentinel X<br>♦ 12 KB accessible |
| Time<br>♦ Real-time clock<br>♦ Read/write and read-only memory<br>♦ Locally connected key | HASP HL Time<br>♦ RTC<br>♦ 4 KB R/W + 2 KB ROM | Sentinel ST<br>♦ RTC<br>♦ 8 KB (1.75 KB accessible) |
| Net<br>♦ Read/write and read-only memory<br>♦ Network-based licensing | HASP HL Net<br>♦ 4 KB R/W + 2 KB ROM<br>♦ Max no. of concurrent users: 10, 50, 250+ | Sentinel SNT<br>♦ 8 KB (1.75 KB accessible)<br>♦ Max no. of concurrent users: 3, 5, 10, 25, 50, 100, 250 |
| Net and Time<br>♦ Real-time clock<br>♦ Read/write and read-only memory<br>♦ Network-based licensing | HASP HL NetTime<br>♦ RTC<br>♦ 4 KB R/W + 2 KB ROM<br>♦ Max no. of concurrent users: 10, 50, 250 | Sentinel SN<br>♦ RTC<br>♦ 8 KB (1.75 KB accessible)<br>♦ Max no. of concurrent users: 3, 5, 10, 25, 50, 100, 250 |
| Drive<br>♦ Read/write and read-only memory<br>♦ Extended mass storage | HASP HL Drive<br>♦ 4 KB R/W + 2 KB ROM<br>♦ 2 GB or 4 GB USB Mass Storage drive | None |
| Vendor Site Production Key | Sentinel HASP Master Key | Sentinel I |
| Vendor Site Development Key | Sentinel HASP Developer Key | Sentinel I |
| Distribution Site Production Key | Distribution Channel functionality in Sentinel HASP Business Studio v.5 and later | Sentinel D |

## Table 2: Sentinel Keys and Sentinel HASP Tool Equivalents

| Sentinel Keys Tools | Sentinel HASP Tools |
|---|---|
| Encoding Sentinel keys | Keys are pre-encoded at SafeNet production site. Use your unique Sentinel HASP Vendor Code (stored in the Sentinel HASP Vendor keys) |
| Sentinel Keys ToolKit | Sentinel HASP Vendor Suite |
| Shell or Cover Code | Sentinel HASP Envelope<br>(part of Sentinel HASP Vendor Suite) |
| License Manager | Sentinel HASP Business Studio<br>(part of Sentinel HASP Vendor Suite) |
| API Explorer | Sentinel HASP ToolBox<br>(part of Sentinel HASP Vendor Suite) |
| Sentinel Protection installer | Sentinel HASP Run-time Environment installer |
| Sentinel System Driver (legacy) | Sentinel HASP Run-time Environment installer |
| Sentinel Keys Server (legacy) | Sentinel HASP Run-time Environment installer |
| Sentinel Keys License Monitor (previously SuperPro Monitoring Tool and UltraPro License Monitor) | Sentinel HASP Admin Control Center (part of the Sentinel HASP Run-time Environment) |
| SafeNet Sentinel Medic | Sentinel HASP Admin Control Center (part of the Sentinel HASP Run-time Environment) |
| Secure Update utility | Sentinel HASP Remote Update System (RUS) |
| Serial number | HASP ID |

## Table 3: Comparison of Sentinel Hardware Key API Functions and Sentinel HASP Run-time API Functions

| Sentinel Hardware Key API Function* | Sentinel HASP Run-time API Function |
|---|---|
| SFNTCounterDecrement() | Not required in Sentinel HASP—performed automatically by hasp_login() when logging in to Features with execution counters. |
| SFNTDecrypt() | hasp_decrypt() |
| SFNTEncrypt() | hasp_encrypt() |
| SFNTSetContactServer() | Not required in Sentinel HASP—performed automatically by hasp_login().<br>Logging in to a specific server is possible using hasp_login_scope(). |
| SFNTGetLicense() | hasp_login() |
| SFNTGetFeatureInfo() | hasp_get_info() |
| SFNTGetLicenseInfo() | hasp_get_info() |
| SFNTQueryFeature() | Not required in Sentinel HASP—performed automatically by hasp_login().<br>**Note:** The SFNTQueryFeature function is implemented by a proprietary algorithm based on AES. If you want to implement a similar mechanism, you can replace the query/response table with a similar table generated using hasp_decrypt/hasp_encrypt. |
| SFNTEnumServer() | Not required in Sentinel HASP—performed automatically by hasp_login().<br>Logging in to a specific server is possible using hasp_login_scope(). |
| SFNTReadInteger() | hasp_read() |
| SFNTReadRawData() | hasp_read() |
| SFNTReadString() | hasp_read() |
| SFNTGetServerInfo() | hasp_get_sessioninfo() |
| SFNTReleaseLicense() | hasp_logout() |
| SFNTSetHeartbeat() | Not required in Sentinel HASP. |
| SFNTGetDeviceInfo() | hasp_get_info()or<br>hasp_get_sessioninfo() |
| SFNTSign() | Not required in Sentinel HASP. |
| SFNTVerify() | Not required in Sentinel HASP. |
| SFNTWriteInteger() | hasp_write() |
| SFNTReadRawData() | hasp_write() |
| SFNTWriteRawData() | hasp_write() |
| SFNTWriteString() | hasp_write() |
| SFNTSetConfigFile()<br>(added in Sentinel Hardware Keys version 1.3.0) | Run-time Environment configuration can be performed using the Admin Control Center user interface. The location of the configuration file is not modifiable. The file path is typically **Program Files\Common Files\Aladdin Shared\HASP\hasplm.ini**. |

* Sentinel functions that are not listed are obsolete.