Sentinel HASP®

Moving from SmartKey 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 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- 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

This migration guide is intended for users of SmartKeys. The guide's main focus is for users who wish to continue using a hardware-based protection solution, but want to migrate to the more comprehensive HASP HL key protection and advanced licensing provided by Sentinel HASP. The guide assumes that the reader has a good understanding of both the SmartKey and the Sentinel HASP systems and provides the following:

- Three migration paths from SmartKey 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 SmartKey documentation, or the *Sentinel HASP Software Protection and Licensing Guide*, *Sentinel HASP Installation Guide*, or Help documentation.
- Tables comparing Sentinel HASP and SmartKey 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 SmartKey, refer to the relevant SmartKey documentation.

Available Migration Paths

Three migration paths are available. In Migration Paths 1 and 2, the stages are not interdependent, meaning it is possible to begin at Stage 2. (Note that Stage 2 is identical in both of these migration paths.) Similarly, the time that you wait before moving from one stage to the next is entirely at your discretion.

- **Migration Path 1** provides a gradual move towards improved security for your products in a very short time by merely adding Sentinel HASP as a complementary system to your current protection, and converting to the complete Sentinel HASP protection system at your convenience.

  Using Migration Path 1, you introduce Sentinel HASP alongside your current SmartKey protection, allowing a gradual adjustment at your own pace to the enhanced functionality offered by Sentinel HASP. When you are ready, you can phase SmartKey out and fully implement the superior protection of the Sentinel HASP solution.

  For more information, see *Migration Path 1—Sentinel HASP Complementing SmartKey* Implementation on page 6.

- **Migration Path 2** provides a way to phase out your installation base of SmartKeys over time—without necessitating the recall and replacement of SmartKeys, and without having to continue their distribution.

  Using Migration Path 2, and creating a version of your software that recognizes both SmartKey and HASP HL keys, you can start distributing HASP HL keys to new customers while existing customers continue using their SmartKeys. You can then gradually replace your install base of SmartKeys with HASP HL keys.

  For more information, see *Migration Path 2— Sentinel HASP and SmartKey Combined* Implementation on page 9.

  Simultaneous migration of both paths is possible, to create a three-stage solution of full Sentinel HASP implementation.

  - Implement Stage 1 of Migration Path 1 to add increased security to your current SmartKey protection using Sentinel HASP Envelope. Implementing this stage can provide an immediate solution to SmartKey emulators.
  - Implement Stage 1 of Migration Path 2 for a gradual migration that does not require the distribution of both a Sentinel HASP protection key and a SmartKey. This migration works well in markets that are less prone to piracy.

  Implementation and distribution according to steps 1 and 2 may be performed simultaneously, depending on the requirements of your market.

  - Implement Stage 2 of Migration Path 2 to completely remove SmartKeys and to upgrade to a full implementation of Sentinel HASP protection, utilizing the strongest security and accomplishing the highest licensing flexibility.

- **Migration Path 3** enables a gradual transition from SmartKey GSS to Sentinel HASP. A SmartKey GSS-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 3—Gradual Migration from SmartKey to Sentinel* HASP on page 13.

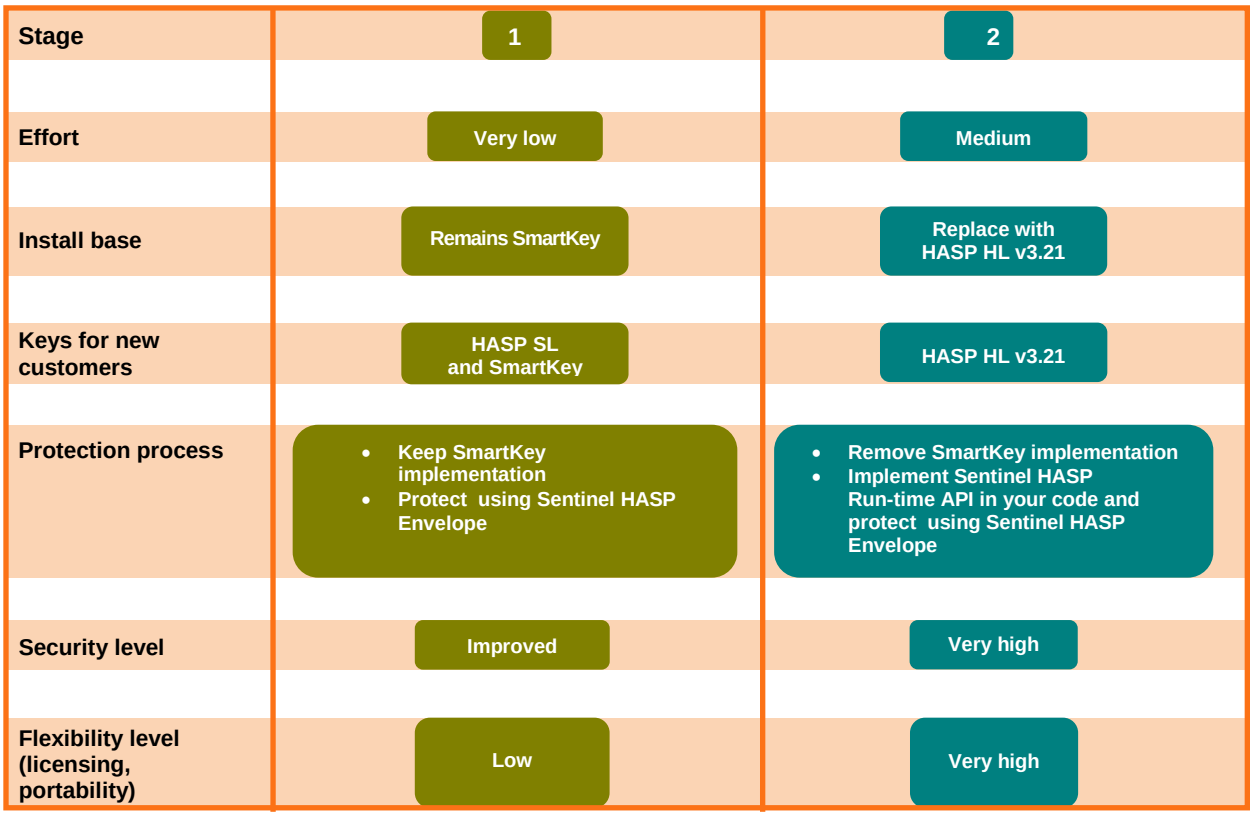
Migration Path 1—Sentinel HASP Complementing SmartKey Implementation

This two-stage migration path enables you to improve your security in a very short time by implementing Sentinel HASP Envelope protection, and locking your protected application to a software-based HASP SL key that employs product activation. The activation process can be performed manually (using software utilities), or automatically via the Sentinel HASP Run-time and Sentinel HASP Activation APIs. The manual approach deploys quickly since no additional code must be written. However, it may be less convenient when dealing with larger installation bases. In such cases, it may be preferable to choose automatic activation, which will require integration of the APIs.

Stage 1 presents an opportunity for you to enhance your existing SmartKey protection. While maintaining your trusted current protection, you have only to add Sentinel HASP as a complementary system. This gradual change from SmartKey to Sentinel HASP means that the entire installation base is not forced to change all at once. While your clients adjust to Sentinel HASP protection, you can easily transition to Stage 2, which offers a much higher level of security and provides more portability. Stage 2 is ideal for new customers and/or when distributing new versions of your software.

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: Initial Implementation of Sentinel HASP Functionality

Stage 1 enables you to easily implement basic functionality of the Sentinel HASP system, while retaining SmartKeys as your installation base. By supplying your customers with a HASP SL key in addition to their SmartKey, they gain increased security and licensing capabilities.

Implementing Stage 1

The following procedure details the steps required to implement Stage 1 of the SmartKey-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. Using Sentinel HASP Business Studio™, create the following:
   a. A Feature that represents the protected application
   b. A Provisional Product containing the Feature you created, with licensing terms stating that the license will expire in a specific number of days. The license period can be defined for any period between one and 90 days.
   c. A Bundle of Provisional Products (as a V2C file) containing the Provisional Product you created
   d. A Sentinel HASP Run-time Environment (RTE) Installer containing the Bundle of Provisional Products you created
   e. A Sentinel HASP Remote Update System (RUS) utility (using the RUS branding option)
3. Integrate the Sentinel HASP RTE Installer with embedded license data into your application.

   (See *Sentinel HASP Software Protection and Licensing Guide*, chapter “Distributing Sentinel HASP with your Software.”)
4. Protect your program using the SmartKey API, but do not implement GSS protection.
5. Use Sentinel HASP Envelope to protect your program.
6. Continue distributing a SmartKey with each copy of your software.
7. Request the customer to run the Sentinel HASP RUS utility to generate a C2V file for the machine intended to host the license of the protected software, and to send you the file.

   Note: The C2V file can also be generated using the Sentinel HASP Run-time API.
8. In Sentinel HASP Business Studio, create and execute a Product Key-based order.
9. In the Sentinel HASP Business Studio—Customer Services window, select **Activate Product** and use the C2V from the customer and the Product Key you just created to generate a V2C file.
10. Send the V2C file to the customer with instructions to activate the HASP SL key using this file and the Sentinel HASP RUS utility.

    Note: Steps 8-10 can be performed using the Sentinel HASP Activation API.

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 HASP HL keys.

Implementing Stage 2

The following procedure details the steps required to implement Stage 2 of the SmartKey-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 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.

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

   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 that is to be protected.
2. Replace all calls to SmartKey in the code with calls to HASP HL keys.
   See Table 3: *Comparison of SmartKey API Functions and Sentinel HASP Run-time API Functions* on page 18 for a list of SmartKey functions and their Sentinel HASP equivalents.

   (See *Sentinel HASP Software Protection and Licensing Guide*, appendix “Sentinel HASP Run-time API Reference”.)
3. Protect your software using Sentinel HASP Envelope.

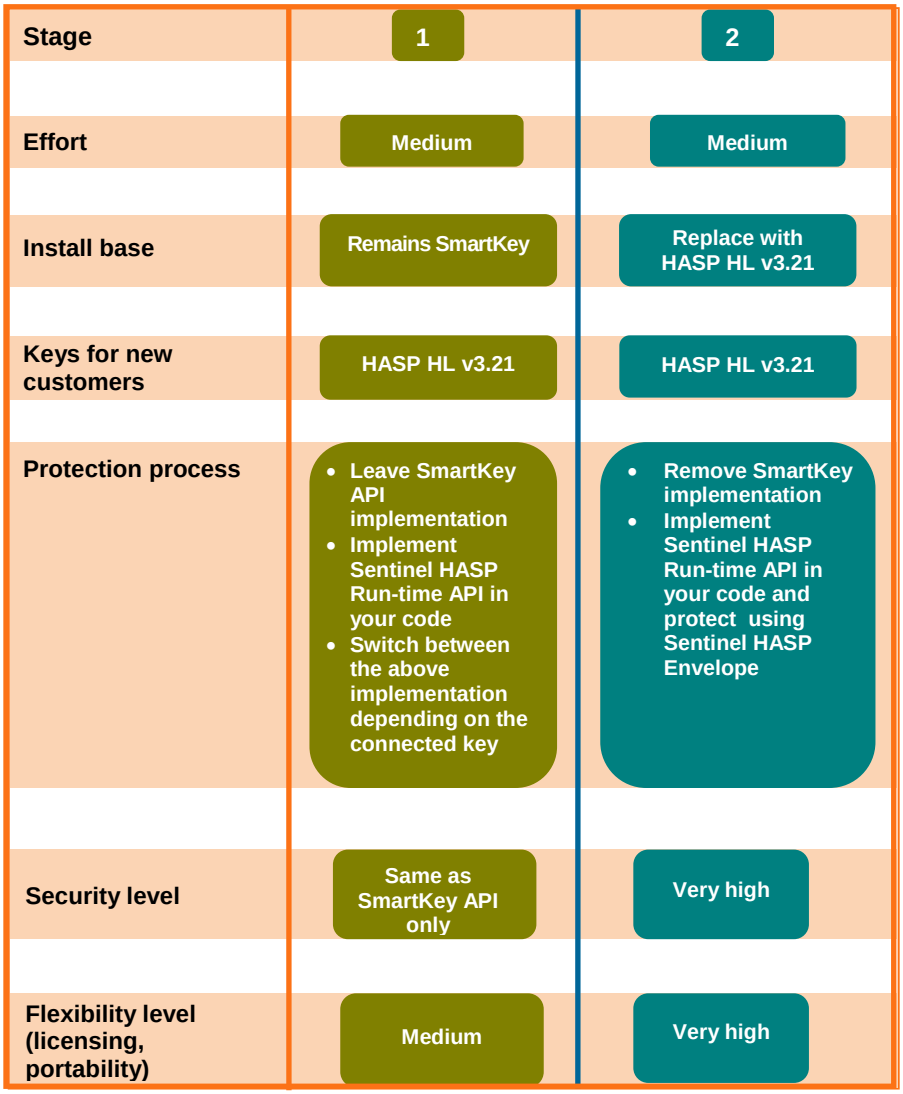
   (See *Sentinel HASP Software Protection and Licensing Guide*, chapter “Sentinel HASP Envelope Protection”.)
4. Follow the instructions in the *Sentinel HASP Software Protection and Licensing Guide* to distribute your software (see chapter “Distributing Sentinel HASP with your Software”).
5. Ensure that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

Migration Path 2— Sentinel HASP and SmartKey Combined Implementation

This two-stage migration path enables you to phase out your installation base of SmartKeys over time, without necessitating immediate recall and replacement of the SmartKeys and without having to continue their distribution. To achieve this status, you create a version of your software that is able to identify both SmartKey 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 SmartKeys.

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 2.



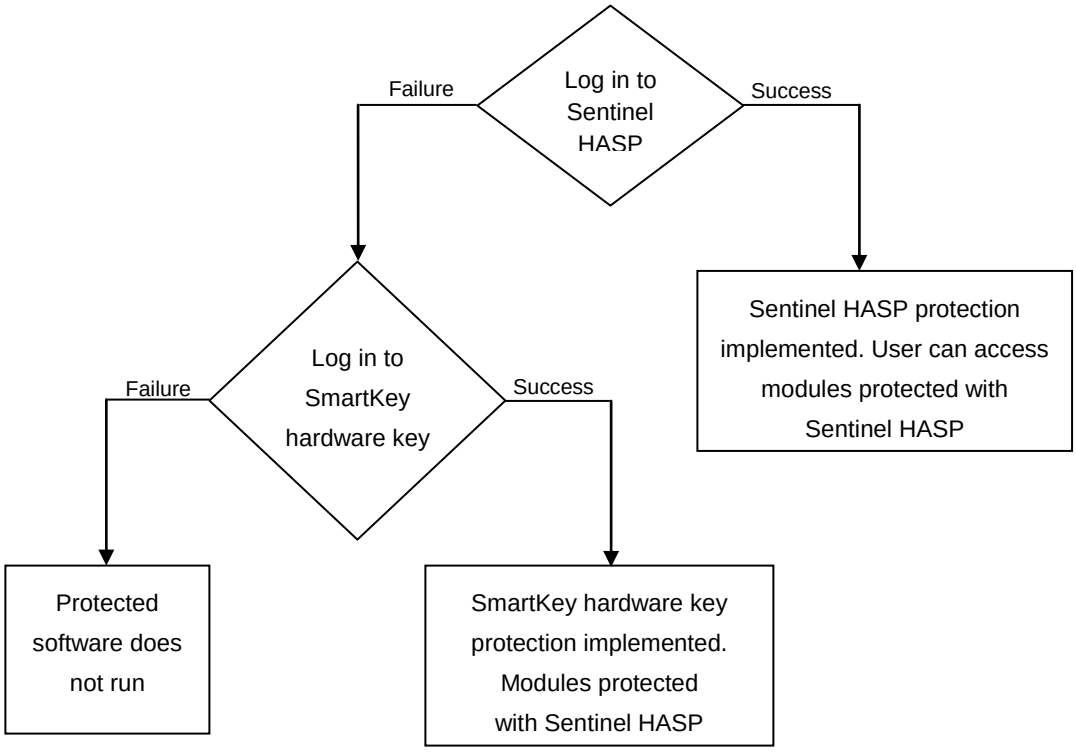
Stage 1: Combining SmartKey with Sentinel HASP Protection

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 SmartKey. If a SmartKey is detected, SmartKey protection is used.

In order to maximize security and implement the higher level of protection provided by Sentinel HASP Envelope, concurrently with the SmartKey protection of your software, you can protect SmartKey-protected files or modules using the Sentinel HASP Run-time API. Consider also using Sentinel HASP Envelope to protect any individual files that are not protected by SmartKey. Applications that are protected solely by Sentinel HASP can only be executed using a HASP HL key.

Sentinel HASP-protected applications have greater security than those protected by SmartKey alone. If a SmartKey is used, modules protected with SmartKey will continue to 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:



Note: The above diagram is relevant to all SmartKeys.

Implementing Stage 1

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

To implement both Sentinel HASP and SmartKey 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.

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

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 customized Sentinel HASP Run-time API header files in your project. Do **not** remove included SmartKey headers.

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

4. To enable your software to work with SmartKey or Sentinel HASP protection, implement the decision tree on page 10 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 SmartKey functionality. If the SmartKey login is successful, SmartKey protection is invoked.

   c. If the login to SmartKey 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. 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.

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

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 SmartKeys with HASP HL keys, at your convenience.

Stage 2: Full Implementation of Sentinel HASP Functionality

Stage 2 enables you to fully implement the advanced 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 HASP HL keys.

Implementing Stage 2

The following procedure details the steps required to implement Stage 2 of the SmartKey-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 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.

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

   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.
2. Replace all calls to SmartKey in the code with calls to HASP HL keys.
   See Table 3: *Comparison of SmartKey API Functions and Sentinel HASP Run-time API Functions* on page 18 for a list of SmartKey functions and their Sentinel HASP equivalents.

   (See *Sentinel HASP Software Protection and Licensing Guide*, appendix “Sentinel HASP Run-time API Reference.”)
3. Protect your 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 *Sentinel HASP Software Protection and Licensing Guide*, 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 3—Gradual Migration from SmartKey to Sentinel HASP using a Launcher Application

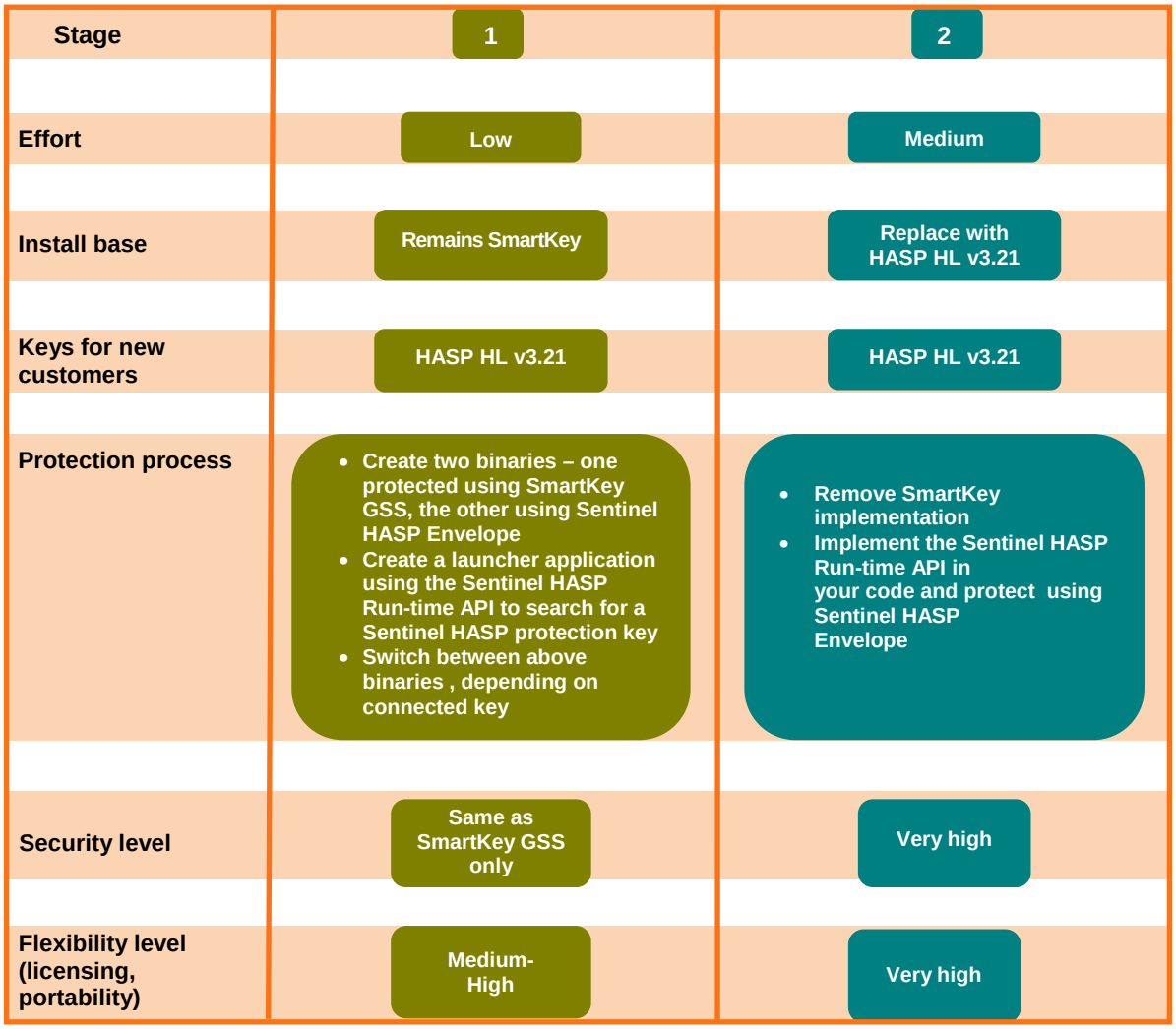
This migration path enables you to phase out your installation base of SmartKeys—without necessitating the recall and replacement of the SmartKeys, and without having to continue their distribution.

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

This migration path enables you to support existing users who already have SmartKeys, 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: Initial Implementation of Sentinel HASP Functionality

During Stage 1 of the migration process, you create two versions of your software—one protected using SmartKey GSS, 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

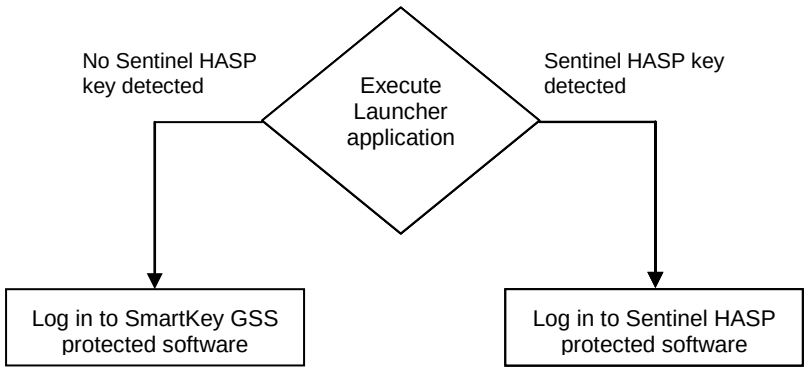
The following procedure details the steps required to implement the SmartKey GSS-to-Sentinel HASP migration process. Where relevant, you are pointed to additional information in the Sentinel HASP documentation.

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_smartkey.exe`) and implement SmartKey GSS protection using SmartKey GSS and/or the SmartKey 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_smartkey.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 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 SmartKey-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 SmartKey API, replace all calls to SmartKey in the code with calls to Sentinel HASP protection keys.
   See Table 3: *Comparison of SmartKey API Functions and Sentinel HASP Run-time API Functions* on page 18 for a list of SmartKey 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 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: HASP HL Key and SmartKey Comparison Tables

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

Model Type	HASP HL	SmartKey
Basic ♦ No read/write memory functionality ♦ Perpetual license ♦ Locally connected key	HASP HL Basic	SmartKey FX
Memory ♦ Read/write and read-only memory ♦ Locally connected key	HASP HL Pro ♦ 112 bytes R/W + 112 bytes ROM HASP HL Max ♦ 4 KB R/W + 2 KB ROM	SmartKey PR ♦ 64/128 bytes R/W SmartKey EP ♦ 64/128 bytes R/W SmartKey SP ♦ 896 bytes R/W
Time ♦ Real-time clock ♦ Read/write and read-only memory ♦ Locally connected key	HASP HL Time ♦ RTC ♦ 4 KB R/W + 2 KB ROM	None
Net ♦ Read/write and read-only memory ♦ Network-based licensing	HASP HL Net ♦ 4 KB R/W + 2 KB ROM ♦ Max. no. of concurrent users: 10, 50, 250	SmartKey NET ♦ 896 bytes R/W ♦ Max no. of concurrent users: 254
Net and Time ♦ Real-time clock ♦ Read/write and read-only memory ♦ Network-based licensing	HASP HL NetTime ♦ RTC ♦ 4 KB R/W + 2 KB ROM ♦ Max. no. of concurrent users: 10, 50, 250 users	None
Drive ♦ Read/write and read-only memory ♦ Extended mass storage	HASP HL Drive ♦ 4 KB R/W + 2 KB ROM ♦ 2 GB or 4 GB USB mass storage drive	SmartPico ♦ 896 bytes R/W ♦ from 256 Mb to 4 GB USB mass storage drive

Table 2: SmartKey and Sentinel HASP Tool Equivalents

SmartKey Tools	Sentinel HASP Tools
SmartKey Programming Central (SPC) – Setting label and password. ID code is programmed at production site.	Keys are pre-encoded at SafeNet production site. Use your unique Sentinel HASP Vendor Code (stored in the Sentinel HASP Vendor keys)
Global Security Setting (GSS)	Sentinel HASP Envelope (part of Sentinel HASP Vendor Suite)
SPC - Programming keys	Sentinel HASP Business Studio (part of Sentinel HASP Vendor Suite)
SPC (for function execution only)–Code samples provided in the SDK for the most common languages	Sentinel HASP ToolBox (part of Sentinel HASP Vendor Suite)
SmartKey Driver Installation (SDI)	Sentinel HASP Run-time Environment installer
SPCinfo and Skeymon (server monitor)	Sentinel HASP Admin Control Center (part of the Sentinel HASP Run-time Environment)
	Sentinel HASP Remote Update System (RUS)
Serial number	HASP ID

Table 3: Comparison of SmartKey API Functions and Sentinel HASP Run-time API Functions

SmartKey API Function*	Sentinel HASP Run-time API Function
`Locating mode msclink(), "L" command`	Performed automatically by `hasp_login()`
`Scrambling mode msclink(),smartlink(), "S" command`	Use `hasp_encrypt()` and `hasp_descrypt()` to perform encryption on data buffer
`Reading mode msclink(),smartlink(), "R" command`	Use `hasp_read()` to read Sentinel HASP memory
`Writing mode msclink(),smartlink(), "W" command`	Use `hasp_write()` to write Sentinel HASP memory
`Block Reading mode msclink(),smartlink(), "BR" command`	Use `hasp_read()` to read Sentinel HASP memory and set offset and length
`Block Writing mode msclink(),smartlink(), "BW" command`	Use `hasp_write()` to read Sentinel HASP memory and set offset and length
`Fixing Msclink(), "F" command`	Use the ROM memory of the Sentinel HASP instead of fixing the memory
`Programming Msclink(), "P" command`	Not required in Sentinel HASP. Login data is automatically managed.
`Comparing Msclink(), "C" command`	Performed automatically by `hasp_login()`

SmartKey API Function*	Sentinel HASP Run-time API Function
Model Reading msclink(),smartlink(), “M” command	hasp_get_sessioninfo()
Serial Number Reading msclink(),smartlink(), “N” command	hasp_get_sessioninfo()
Ext Model Reading msclink(),smartlink(), “M” command	hasp_get_sessioninfo()
Fix Reading msclink(),smartlink(), “X” command	Not present in Sentinel HASP
Fail Counter Reading msclink(),smartlink(), “A” command	Not present in Sentinel HASP
AES Set Msclink(), “G” command	Keys are pre-encoded at SafeNet production site
AES Scramble msclink(),smartlink(), “O” command	Performed automatically by hasp_login()
Open Mode smartlink(), “O” NET_command	Performed automatically by hasp_login() Logging in to a specific connected key is possible using hasp_login_scope()
Access Mode smartlink(), “A” NET_command	Not required in Sentinel HASP
User Number Mode smartlink(), “U” NET_command	hasp_get_sessioninfo()
Close Mode smartlink(), “C” NET_command	Hasp_logout ()