SENTINEL LDK

MIGRATION GUIDE: HASP4 TO SENTINEL LDK

Trademarks, Copyrights, and Third-Party Software

Disclaimer

All information herein is either public information or is the property of and owned solely by and/or its subsidiaries or affiliates who shall have and keep the sole right to file patent applications or any other kind of intellectual property protection in connection with such information.

Nothing herein shall be construed as implying or granting to you any rights, by license, grant or otherwise, under any intellectual and/or industrial property rights of or concerning any of and any of its subsidiaries and affiliates (collectively referred to herein after as “Thales”) information.

This document can be used for informational, non-commercial, internal and personal use only provided that:

> The copyright notice below, the confidentiality and proprietary legend and this full warning notice appear in all copies.

> This document shall not be posted on any network computer or broadcast in any media and no modification of any part of this document shall be made.

Use for any other purpose is expressly prohibited and may result in severe civil and criminal liabilities.

The information contained in this document is provided "AS IS" without any warranty of any kind. Unless otherwise expressly agreed in writing, Thales makes no warranty as to the value or accuracy of information contained herein.

The document could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Furthermore, Thales reserves the right to make **any change or** improvement in the specifications data, information, and the like described herein, at any time.

Thales hereby disclaims all warranties and conditions with regard to the information contained herein, including all implied warranties of merchantability, fitness for a particular purpose, title and non-infringement. In no event shall Thales be liable, whether in contract, tort or otherwise, for any indirect, special or consequential damages or any damages whatsoever including but not limited to damages resulting from loss of use, data, profits, revenues, or customers, arising out of or in connection with the use or performance of information contained in this document.

Thales does not and shall not warrant that this product will be resistant to all possible attacks and shall not incur, and disclaims, any liability in this respect. Even if each product is compliant with current security standards in force on the date of their design, security mechanisms' resistance necessarily evolves according to the state of the art in security and notably under the emergence of new attacks. Under no circumstances, shall Thales be held liable for any third party actions and in particular in case of any successful attack against systems or equipment incorporating Thales products. Thales disclaims any liability with respect to security for direct, indirect, incidental or consequential damages that result from any use of its products. It is further stressed that independent testing and verification by the person using the product is particularly encouraged, especially in any application in which defective, incorrect or insecure functioning could result in damage to persons or property, denial of service or loss of privacy.

Product Version: 8.3 and later (2209-1)

Contents

Introduction

About Sentinel LDK

Sentinel® License Development Kit (*Sentinel LDK*) is a Software Digital Rights Management (DRM) solution that delivers strong copy protection, protection for Intellectual Property and secure and flexible licensing. Sentinel LDK 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 LDK 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-based or software-based. Sentinel LDK hardware-based protection, which utilizes Sentinel HL keys, provides the safest and strongest level of protection.

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

About This Guide

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

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

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

The guide provides the following:

- An overview and guidelines for a three-stage migration path from HASP4 to Sentinel LDK, starting with an install base that includes only HASP4 keys and HASP HL keys.
- Migration procedures
- Tables that list the tools and functionalities of HASP4 and their counterparts in Sentinel LDK

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

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

> **NOTE:** The procedures and terminology employed in this guide are based on the assumption that you want to migrate from HASP4 to Sentinel HL keys. However, since Sentinel HL and Sentinel SL keys are supported by the same set of tools and APIs, the procedures for migrating to Sentinel SL keys are similar (migrating to SL keys requires an activation process). If you want to migrate to Sentinel SL keys, refer to the *Sentinel LDK Software Protection and Licensing Guide* or contact Thales Technical Support.

About Sentinel HL Keys

The following types of Sentinel HL keys are available, replacing the HASP HL keys that were provided until now:

- Sentinel HL (Driverless configuration) keys

  These keys make use of HID drivers (included in the Windows operating system) instead of Thales drivers. When used as standalone keys, these keys can be used without installing the Run-time Environment. (Network keys require the Run-time Environment.) However, these keys are not backward-compatible with applications protected with Sentinel LDK 6.1 or earlier, Sentinel HASP, HASP HL 1.x, or HASP4. To use these keys, your protected application must include the Licensing API libraries from Sentinel LDK v.7.1 or later, and you must be working with the backend from Sentinel LDK v.7.1 or later.

- Sentinel HL (HASP configuration) keys

  These keys are fully compatible with existing HASP HL keys and with older generations of HASP keys (and with Hardlock/HASP4 keys). These keys can work with your existing API libraries and Run-time Environment, and you can work with your current backend environment. These keys can be upgraded at the customer site to Sentinel HL (Driverless configuration) keys and can thus provide all the benefits provided by the Driverless-configuration keys.

> **NOTE:** Occurrences of the term **Sentinel HL key** in this guide generally refer to the Sentinel HL (Driverless configuration) key.

Requirements for Run-time Environment

You are required to install the Sentinel LDK Run-time Environment on at least some of your machines for the following types of Sentinel protection keys:

- Sentinel SL-AdminMode keys (the type of Sentinel SL key recommended for one of the migration paths in this book).

- Sentinel HL (HASP configuration) standalone keys. The Run-time Environment is required on the computer where the protected application is executed and the key is attached.

- Sentinel HL network keys.

  This includes the following keys:

  - Sentinel HL Net and NetTime (HASP configuration) keys
  - Sentinel HL Net and NetTime (Driverless configuration) keys

- Any Sentinel HL (Driverless configuration) key (other than Basic) with a concurrency license

  The Sentinel HL network key is connected to any computer in the network.

  The Run-time Environment is required on the computer where the network key is attached. The protected application can execute on different computers in the network.

The standalone Sentinel HL (Driverless configuration) keys do not require the Run-time Environment.

For more information, see “Protection Keys That Require Sentinel LDK Run-time Environment” in the *Sentinel LDK Software Protection and Licensing Guide*.

Migration Use Cases

The table below lists various use cases that can be implemented using the procedures in this migration guide.

	Existing Installed Hardware Base	**Existing Protection Implementation**	**Planned Hardware**	**Planned Protection Implementation**	**Planned Backend**	**Notes**
1	HASP4 or HASP	HASP4	HASP (since already deployed)	Sentinel LDK	Sentinel LDK-EMS	Provides better backend, better security
2	Basic key: HASP4 or HASP	HASP4	HASP Basic key	Sentinel LDK	None	Provides better security
3	HASP4 or HASP	HASP4	SHL(HASP)	Sentinel LDK	Sentinel LDK-EMS	Provides better backend, better security. No backward compatibility in LDK; new keys are usable by existing applications and can be applied with legacy H4 licenses via legacy backend
4	Basic keys: HASP4 or HASP	HASP4	SHL(HASP) Basic key	Sentinel LDK	None	Provides better security
5	HASP4 or HASP	HASP4	SHL(Driverless)	Sentinel LDK	Sentinel LDK-EMS	Provides better backend, better security No backward compatibility
6	Basic keys: HASP4 or HASP	HASP4	SHL(Driverless) Basic key	Sentinel LDK	None	Provides better security No backward compatibility

Table Legend:

HASP4	HASP4 protection key
HASP	HASP HL protection key
SHL(HASP)	Sentinel HL (HASP configuration) protection key
SHL(Driverless)	Sentinel HL (Driverless configuration) protection key

Compatibility with HASP4

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

In addition, most of the Sentinel Licensing 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 Sentinel HL keys.

The table that follows indicates the compatibility of HASP4 keys (FW 1.x, 2.x, 3.x, 4.x) with Sentinel LDK hardware and software.

Protection Type	Driver and License Manager	Result	Notes
HASP4 API/ Envelope	HASP4 Run-time Environment	Compatible	Standalone and Network keys (concurrency)
Sentinel Licensing API (Used in HASP4 login access)*	Sentinel LDK Run-time Environment and License Manager**	Compatible	Only Standalone keys
Sentinel LDK Envelope	Sentinel LDK Run-time Environment and License Manager	Not compatible	
* Sample code can be seen using Sentinel LDK ToolBox. ** Sentinel License Manager and Admin Control Center cannot see HASP4 keys.			

Note: Thales no longer supports HASP4 legacy software.

You can begin supplying Sentinel HL (HASP configuration) keys to new customers immediately, with no change to your existing software. When you decide to migrate your entire customer base to Sentinel HL (Driverless configuration) keys, it will not be necessary to replace the HASP configuration keys that have already been deployed. These keys can be upgraded to Sentinel HL (Driverless configuration) keys at the customer site.

Shortcut to Enhanced Protection

Sentinel SL “Unlocked License” is a mechanism by which the protection applied to an application can be significantly enhanced without affecting the current protection and licensing process.

You use Sentinel LDK Envelope to apply a sophisticated protection wrapper over any existing HASP4 protection and licensing scheme. This wrapper protects your application against reverse engineering and theft of intellectual property.

You can apply this protection immediately as a short-term or long-term solution while you develop your process to migrate to Sentinel HL keys. For maximum security, Thales recommends that you obtain a batch code for this purpose that is different from the batch code that you will use for your Sentinel HL keys.

For more information regarding Unlocked Licenses, see the *Sentinel LDK Software Protection and Licensing Guide*. For pricing information for Unlocked Licenses, contact your Thales sales representative.

Support Contacts

To obtain assistance in using Sentinel products, feel free to contact our Thales Support team:

https://supportportal.thalesgroup.com/csm?id=sentinel

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 LDK. 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 table summarizes the three-stage migration path.

	Stage 1	Stage 2	Stage 3
Implementation effort	-	+	+ +
Install base	Remains HASP4	Remains HASP4 or Sentinel HL (HASP configuration) keys	• Replace HASP4 with Sentinel HL (Driverless configuration) keys. • Upgrade Sentinel HL (HASP configuration) keys to Sentinel HL (Driverless configuration) keys.
Keys for new customers	Sentinel HL (HASP configuration) keys	Sentinel HL (HASP configuration) keys	• Sentinel HL (Driverless configuration) key
Protection process	No change to code. Software remains protected with HASP4	Leave HASP4 API, add Sentinel Licensing API. Use HASP4 Envelope.	Fully implement Sentinel Licensing API and Sentinel LDK Envelope
Security level	+	+ +	+ + + +
Additional benefits	Sentinel HL keys received by new customers will not have to be replaced at later stages.	Sentinel HL keys received by new customers will not have to be replaced at later stages.	Driverless deployment

Stage 1: Using Sentinel HL keys as HASP4 keys

Sentinel HL (HASP configuration) 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 Sentinel HL (HASP configuration) keys to your customers and gradually replace your install base of HASP4 keys with Sentinel HL (HASP configuration) keys at your convenience. When you decide to move to Stage 2 or 3 of the migration process and protect your software with Sentinel LDK, you will already have deployed Sentinel HL keys.

Proceed as follows:

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

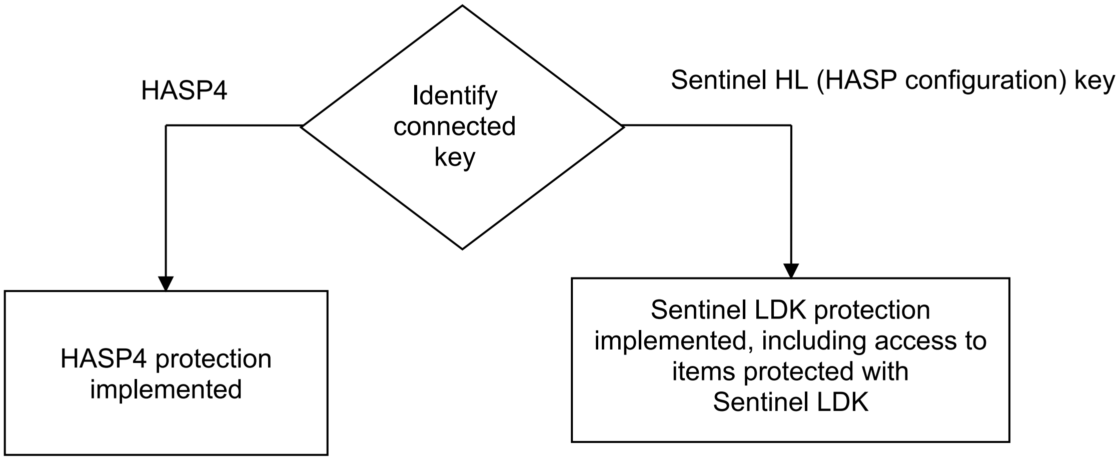
Stage 2: Combining HASP4 and Sentinel LDK Protection

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

When the software runs, it tries to log into a Sentinel HL (HASP configuration) key. If a Sentinel HL key is found, Sentinel LDK protection is implemented. If no Sentinel 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 LDK concurrently with HASP4 protection of your software, you can protect selected files or modules with Sentinel LDK. Sentinel LDK-protected items will have greater security than those only protected by HASP4. The Sentinel LDK-protected items can only be activated with a Sentinel HL (HASP configuration) key. In this case, if a HASP4 key is used, the modules protected with HASP4 will function, but modules protected with Sentinel LDK will not run.

The following flowchart shows the sequential flow when the protected software executes in Stage 2:


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

Implementing Stage 2

You can maintain your legacy HASP4 protection simultaneously with the added protection and licensing capabilities of Sentinel LDK. This is possible because the Sentinel HL (HASP configuration) key is divided into two segments, each of which functions as a separate key. The legacy key segment provides full HASP4 functionality. The Sentinel LDK key segment provides the advanced Sentinel LDK capabilities. When you combine HASP4 and Sentinel LDK 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 LDK functionality. Where relevant, you are pointed to additional information in the *Sentinel LDK Software Protection and Licensing Guide.*

1. **Installing Sentinel LDK and Introducing your Vendor Keys**

   If you have not already done so, install Sentinel Vendor Suite and Sentinel LDK-EMS, and introduce your Sentinel Vendor keys. As part of the Vendor key introduction process, Sentinel LDK generates customized Sentinel Licensing API libraries for your Vendor Code. For more information, see the *Sentinel LDK Installation Guide*. (Sentinel LDK-EMS is not required for Sentinel HL Basic keys.)

2. **Linking to Sentinel Licensing API Libraries**

   Link the customized Sentinel Licensing API libraries to the protected files as follows:

   - If you link to your customized Sentinel Licensing API `.lib` files:
     - Remove the existing link to the HASP4 library files.
     - Replace any `hasp (…)` call with the corresponding `hasp_*` call (for example, `hasp_login`, `hasp_logout`). You can use Sentinel LDK ToolBox to simplify this process.

     (The Sentinel Licensing API `.lib` files contain both Sentinel LDK and HASP4 functionalities, but the names of the Sentinel LDK and HASP4 functions are not identical.)
   - If you link to your customized Sentinel Licensing API `.dll` files, do **not** remove the link to the HASP4 library files.
   - Include your customized Sentinel Licensing API header files in your project. Do **not** remove included HASP4 headers.

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

   To enable your software to work with HASP4 or Sentinel LDK protection, implement the decision tree illustrated on page 10 of this document, as follows:

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

      (See the help system for the Sentinel Licensing API.)

   b. If both a Sentinel HL (HASP configuration) key and a HASP4 key are not detected, invoke the behavior of the application when no key is detected.

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

4. Protecting Selected Items With Sentinel LDK

Optionally, you can enhance the security of selected items in your software by protecting them with Sentinel LDK. You can protect individual files using Sentinel LDK Envelope or Sentinel Licensing API. You can protect code snippets and other data using the Sentinel Licensing API. These protected items are only accessible when a Sentinel HL (HASP configuration) key is connected.

When using the Sentinel Licensing 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 Licensing API calls where necessary.

(See the help system for the Sentinel Licensing API.)

5. Licensing, Production and End-User License Management

The Sentinel LDK and legacy HASP4 segments in a Sentinel HL (HASP configuration) 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 LDK Segment	Legacy HASP4 Segment
Licensing	Sentinel LDK-EMS™	HASPEdit
Production and Creation of Remote Update Files	Sentinel LDK-EMS	♦ DOS RUS—Vendor Utility ♦ Win32 RUS—API sample
Remote Update at End-User Site	♦ Sentinel Remote Update System (RUS) ♦ Sentinel Licensing API hasp_update() function	♦ DOS RUS—Customer Utility ♦ Win32 RUS—API sample
Key Monitoring	Sentinel Admin Control Center (part of the Sentinel LDK Run-time Environment)	Aladdin Monitor
License Management at End-User Site	Sentinel License Manager, interfaced through the Sentinel Admin Control Center (part of the Sentinel LDK Run-time Environment)	Sentinel License Manager

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

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

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

7. Distributing Your Software

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

(See *Sentinel LDK Software Protection and Licensing Guide*, chapter “Distributing Sentinel LDK 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 LDK, you must supply customers who will be using these modules with Sentinel HL (HASP configuration) 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 Sentinel HL (HASP configuration) keys.

Stage 3: Full Implementation of Sentinel LDK

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

The following procedure details the steps you take in order to implement Stage 3 of the HASP4-to-Sentinel LDK migration process. Where relevant, you are pointed to additional information in the *Sentinel LDK Software Protection and Licensing Guide*.

Implementing Stage 3

Fully implement the advanced Sentinel Licensing API by integrating Sentinel LDK functionalities into your code.

1. **Installing Sentinel LDK and Introducing Your Vendor Keys**

   If you have not already done so, install Sentinel Vendor Suite and Sentinel LDK-EMS, and introduce your Sentinel Vendor keys. As part of the Vendor key introduction process, Sentinel LDK generates customized Sentinel Licensing API libraries for your Vendor Code. See *Sentinel LDK Installation Guide.* (Sentinel LDK-EMS is not required for Sentinel HL Basic keys.)

2. **Linking to Sentinel Licensing Libraries**

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

3. **Defining Sentinel LDK Feature IDs**

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

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

4. **Protecting Your Software With Sentinel LDK**

   a. Insert calls to Sentinel LDK in your code. Refer to Table 4 on page 21 for a comparison of HASP4 API and Sentinel Licensing API functions.

      (See the help system for the Sentinel Licensing API.)

   b. Remove all existing HASP4 API calls.

   c. Wrap the software using Sentinel LDK Envelope.

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

5. **Upgrading to Sentinel HL (Driverless Configuration) Keys**

   Define an Update Product or Base Product that includes the **Upgrade to Driverless** attribute. This upgrades existing Sentinel HL (HASP configuration) keys to Sentinel HL (Driverless) keys. The attribute is ignored when applied to protection keys for which the attribute is not relevant (that is, Sentinel HL (Driverless configuration) keys). This Product should be applied to Sentinel HL keys that exist at the customer sites.

   For more information, see the appendix “How to Upgrade a Sentinel HL Key to Driverless Configuration” in the *Sentinel LDK Software Protection and Licensing Guide.*

6. **Encrypting Data**

   In HASP4, a single proprietary encryption key was used to encrypt all data. Sentinel LDK 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 LDK as follows:

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

   b. Encrypt the data using Sentinel LDK, 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 LDK-encrypted data.

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

7. **Writing Data to the Sentinel HL Key**

   Write your existing data to one of the following Sentinel LDK 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 RUS
     - When updating the key using the Sentinel Licensing API `hasp_update()` function

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

8. **Wrapping Your Software With Sentinel LDK Envelope**

   Wrap the software using Sentinel LDK Envelope.

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

9. **Replacing HASP4 Drivers With Sentinel LDK Run-time Environment**

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

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

10. **Licensing Your Software With Sentinel LDK**

    Use Sentinel LDK-EMS (or Sentinel License Generation API) to license your software.

(See *Sentinel LDK Software Protection and Licensing Guide*, Part 3: “Licensing”.)

11. **Distributing Your Software**

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

12. **Distributing or Upgrading End-User Keys**

    Ensure that all customers who are still working with HASP4 keys and who receive the Sentinel LDK-protected software also receive Sentinel HL (Driverless configuration) keys.

    The Sentinel HL (HASP configuration) keys that newer customers received will be upgraded to Sentinel HL (Driverless configuration) keys by the Product that you defined in step 5 above.

13. **Using Sentinel HL Network Keys**

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

APPENDIX A: Sentinel LDK and HASP4 Comparison Tables

Table 1: Comparison of HASP4 with Sentinel HL Keys

HASP4 Keys		**Sentinel HL Keys**	
Key Type	**Memory Size**	**Key Type**	**Memory Size**
Standard	–	Sentinel HL Basic	–
M1	112 Bytes	Sentinel HL Pro	112 Bytes backward-compatible memory* 112 Bytes Read/Write memory 112 Bytes ROM
M4	496 Bytes	Sentinel HL Max	4 KB backward-compatible memory* 4 KB Read/Write memory 2 KB ROM
Time	512 Bytes	Sentinel HL Time	4 KB backward-compatible memory* 4 KB Read/Write memory 2 KB ROM
–	–	Sentinel HL Drive	2 GB / 4 GB Flash memory 4 KB backward-compatible memory* 4 KB Read/Write memory 2 KB ROM
Net	496 Bytes	Sentinel HL Net	4 KB backward-compatible memory* 4 KB Read/Write memory 2 KB ROM
		Sentinel HL (Driverless configuration) key with concurrency license – all types except Basic	As described above.
–	–	Sentinel HL NetTime	4 KB backward-compatible memory* 4 KB Read/Write memory 2 KB ROM

*Backward-compatible memory is available in HASP configuration only.

For more specification data regarding Sentinel HL keys, see the *Sentinel HL Data Sheet.*

Table 2: HASP4 Net Keys and Equivalent Sentinel HL Network Keys

HASP4 Net	Sentinel HL Net	Sentinel HL NetTime	Sentinel HL (Driverless configuration) key
Net 5	Net 10	NetTime 10	Sentinel HL (Driverless configuration) key other than Basic, with concurrency license
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 LDK Counterparts

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

HASP4 Application / Functionality	Sentinel LDK Application / Functionality
HASPEdit	
HASPEdit—String encryption/decryption	Sentinel LDK ToolBox—**Encrypt/Decrypt** functions
HASPEdit—FAS *(generating licenses)*	Sentinel LDK-EMS—**Entitlements** screen
HASPEdit—Programming the user memory area	♦ Sentinel LDK-EMS—define the data on the **Memory** tabbed page when creating a new Product. The data is written to the key during entitlement production. ♦ Sentinel LDK ToolBox—`hasp_write()` function
HASPEdit—Programming the memory area reserved for licensing (FAS area)	Sentinel LDK-EMS – define and manage license related information. Note that Sentinel HL keys employ Read-only memory which does not allow direct R/W access using Sentinel Licensing API. Only V2C updates created by Sentinel LDK-EMS or Sentinel License Generation API can overwrite or update this memory segment.
HASPEdit—setting the real-time clock	Key time set to UTC during key manufacture
HASPEdit—programming licenses to a key	Sentinel LDK-EMS—**Catalog > Products** and **Entitlements**
HASPEdit—reading the key memory	Sentinel LDK-EMS Studio—under **Entitlements**, **Check In C2V** or **Check In Key**
HASPEdit—reading the HASP ID Number	Sentinel LDK-EMS—view the connected key under **Entitlements**, **Check In C2V** or **Check In Key**
HASPEdit—Create programming utility for mass production of keys	Sentinel LDK-EMS— **Entitlements > New > Number of Keys**
Envelope	
Support for Demo key	Sentinel LDK Envelope—Demo key is default, change to non-demo in **Sentinel Batch Code** screen
DataHASP Tab—data file configuration	Sentinel LDK Envelope—**Protection Details** pane, **Enable data file encryption**
Using FAS	Sentinel LDK *Protect Once—Deliver Many* licensing technology. Sentinel LDK Envelope protects per Feature ID. Business logic and license production are implemented in Sentinel LDK-EMS.
Using HASP ID Number	Sentinel LDK Envelope options—**Protection Details** pane, **Advanced**, **Enable Custom Protection Key Login Scope**
Error Messages Tab	Sentinel LDK Envelope—**User Messages** pane
NetHASP parameters	Sentinel LDK Envelope ♦ Search mode—**Protection Details** pane ♦ Advanced search options—**Protection Details** pane, **Advanced**, **Enable Custom Protection Key Login Scope**

HASP4 Application / Functionality	Sentinel LDK Application / Functionality
Enhanced security Envelope switches	Sentinel LDK Envelope—**Protection Details** pane, **Advanced**
Driver	
`hinstall.exe` (command line)	`haspdinst.exe` (command line)
`hdd32.exe` (GUI driven)	`HASPUserSetup.exe` (GUI driven)
Driver installation API	Sentinel LDK Run-time Environment Installation API
License Manager, Monitor, DiagnostiX	
`lmsetup.exe`	Sentinel LDK Run-time Environment installers
LM application—`nhsrvw32.exe`	♦ Sentinel LDK Run-time Environment—Sentinel License Manager Service (LMS) ♦ `hasplms.exe`
LM application (service)—`nhsrvice.exe`	♦ Sentinel LDK Run-time Environment—Sentinel License Manager Service (LMS) ♦ `hasplms.exe`
Monitor Setup—`aksmon.exe`	Sentinel LDK Run-time Environment installers—install the Sentinel Admin Control Center
`nethasp.ini`—client configuration file	♦ Sentinel Licensing API—`hasp_login_scope()` ♦ Sentinel Admin Control Center ♦ `hasplm.ini`
`nhsrv.ini`—LM configuration file	♦ Sentinel Admin Control Center ♦ `hasplm.ini`
DiagnostiX setup—`aksdiag32_setup.exe`	Sentinel LDK Run-time Environment installers

Table 4: Comparison of HASP4 API and Sentinel Licensing API Functions

HASP4 API Service*	Sentinel Licensing 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 LDK. 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 LDK. 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 LDK. Sentinel Licensing API functions return all necessary data
`Service 43: Logout`	`hasp_logout()`
`Service 48: IdleTime`	Sentinel License Manager automatically detects idle sessions and releases them.

* Unlisted HASP4 functions are obsolete.