Sentinel RMS Envelope v1.0

ReadMe for Linux (32-bit and 64-bit)

gemalto
security to be free

Document Revision History

Revision	Action/Change	Date
A	Sentinel RMS Envelope v1.0	December 2017

Disclaimer and Copyrights

All information herein is either public information or is the property of and owned solely by Gemalto NV. and/or its subsidiaries 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 Gemalto's 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, Gemalto 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, Gemalto reserves the right to make any change or improvement in the specifications data, information, and the like described herein, at any time.

Gemalto 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 Gemalto 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.

Gemalto 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 Gemalto be held liable for any third party actions and in particular in case of any successful attack against systems or equipment incorporating Gemalto products. Gemalto 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: Sentinel RMS Envelope v1.0

Document Number: 007-014004-001, Rev. A

Release Date: December 2017

CONTENTS

About Sentinel RMS Envelope

Sentinel® RMS Envelope (also referred to as "RMS Envelope") is a wrapping application that protects your native C language applications[1] with a secure shield. This application offers advanced protection features to enhance the overall level of security of your software.

RMS Envelope protects Linux (32-bit and 64-bit) executables and shared objects—providing a means to counteract reverse engineering and other anti-debugging measures.

By using RMS Envelope, you establish a link between the protected application and a Sentinel RMS license. This link is broken whenever the protected application cannot access the RMS license. While protecting an application, you can apply protection options that are controlled by the engines running RMS Envelope.

Currently, RMS Envelope is available as a command-line application only. The RMS Envelope protection process is described in the following diagram:

You can also use the evaluation version of RMS Envelope for protecting your applications. For more information about evaluation, see Using RMS Envelope in Evaluation mode.

[1]The terms "program" and "application" are used throughout this document as a generic reference to the various types of programming code that can be protected using RMS Envelope, regardless of whether they are executables, binaries, assemblies or libraries.

What's Included in the Package

The table below lists the files included in the Sentinel RMS Envelope Linux package:

> **Note:** For information regarding the complete package, refer to the ReadMeFirst.pdf document included with the order e-mail. This document is also available here.

File/Folder Name	Description
RuntimeEnvironment	The Sentinel® LDK Run-time Environment. It is required for communication with the Sentinel LDK Developer key. **Installer packages** You can find the following Sentinel LDK Run-time installer packages in this directory: • **aksusbd_7.60-1_i386.deb** - Sentinel LDK Run-time Environment DEB Installer for Linux. For more information, refer to the corresponding ReadMe file (*readme_deb.html*) included in this directory. • **aksusbd-7.60.1-i386.tar.gz** - Sentinel LDK Run-time Environment installer script for Linux. For more information, refer to the corresponding ReadMe file *(readme_tar_gz.html*) included in this directory. • **aksusbd-7.60-1.i386.rpm** - Sentinel LDK Run-time Environment RPM Installer for Linux. For more information, refer to the corresponding ReadMe file *(readme_rpm.html)* included in this directory. **Compatibility package installation script** The Sentinel LDK Run-time Environment is compatible with 32-bit (x86) Linux operating systems. 32-bit compatibility packages are available that enable the Run-time Environment to operate under a 64-bit Linux operating system. However, each Linux distribution requires a different 32-bit compatibility package. The following script automates the process of identifying and obtaining the required compatibility package: • **install_32bit_compatibility_package_for _x64.sh** For more information, refer to the corresponding ReadMe file (*install_32bit_ compatibility_package_for_x64_readme.html*) included in this directory.
VendorTools	Contains RMS Envelope executable - SentinelRMSEnvelope.
Sentinel RMS Envelope ReadMe.pdf	This file.

Prerequisites

This section describes RMS Envelope prerequisites:

- For Protecting Applications
- For Using RMS Envelope Protected Applications (Redistributables)
- For Generating Licenses

For Protecting Applications

The following requirements must be met on the system where you want to protect applications using RMS Envelope:

Supported Platforms

RMS Envelope supports the following Linux (32-bit and 64-bit) operating systems for both running RMS Envelope and using the protected applications:

- Red Hat Enterprise Linux (RHEL) 6.x(Kernel v2.6.32)
- Red Hat Enterprise Linux (RHEL) 7.x(Kernel v3.10.0 -121)
- SUSE Linux Enterprise Server (SLES) 11 SP4(Kernel v2.6.18)
- SUSE Linux Enterprise Server (SLES) 12(Kernel 3.0.13)
- OpenSUSE 13.1 (Kernel v3.7.10)
- OpenSUSE 13.2 (Kernel v3.11.6)
- OpenSUSE 42.x (Kernel 4.4.27-2-default)
- Debian Linux 8.x (Kernel v3.2.0.4)
- Debian Linux 9.x (Kernel v4.9.0-3-amd64)
- Ubuntu Linux 16.x (Kernel v4.10)

Sentinel RMS Licensing Libraries

The following Sentinel RMS licensing libraries (v9.2.1 or later) are available. You can choose from these depending upon your requirements. You must place the chosen library in the RMS Envelope directory before protecting an application.

Note: Both the Sentinel RMS SDK and RMS Envelope should have the same serial number.

Architecture	Type	Library	Availability
32-bit	Standalone	libnonet.so	The standalone licensing library. This is available under the Sentinel RMS (v9.2.1 or later) installation directory.
	Network	libls.so	The network licensing library. This is available under the Sentinel RMS (v9.2.1 or later) installation directory.
	Integrated	liblssrv.so	The integrated licensing library that allows an application to switch between standalone and network licensing. This is available under the Sentinel RMS (v9.2.1 or later) installation directory.
	SCP Integrated	liblssrvscp.so	The library for deploying applications in the Cloud Served - Lease standalone mode. This is included with the SCL Add-on for RMS (not available under the Sentinel RMS installation directory).
64-bit	Standalone	libnonet64.so	The standalone licensing library. This is available under the Sentinel RMS (v9.2.1 or later) installation directory.
	Network	libls64.so	The network licensing library. This is available under the Sentinel

Architecture	Type	Library	Availability
			RMS (v9.2.1 or later) installation directory.
	Integrated	liblssrv64.so	The integrated licensing library that allows an application to switch between stand-alone and network licensing. This is available under the Sentinel RMS (v9.2.1 or later) installation directory.
	SCP Integrated	liblssrvscp64.so	The library for deploying applications in the Cloud Served - Lease Standalone mode. This is included with the SCL Add-on for RMS (not available under the Sentinel RMS installation directory).

Sentinel LDK Developer Key

The Sentinel LDK Developer key is a hardware key required for protecting applications/shared objects. This key is shipped separately to you. For more information, refer to the *ReadMeFirst.pdf* available with the order email.

However, the Sentinel LDK Developer key is not required for:

- Protecting applications in evaluation mode.
- Running protected applications

Sentinel LDK Runtime

Sentinel LDK Runtime v7.6.0 (or later) is required for communication with the Sentinel LDK Developer key.

General Recommendations for Protecting Applications

The following recommendations should be followed while protecting applications using RMS Envelope:

- Allow debugging and memory dumping when you protect applications that have the 'exec' command.
- Do not use pthread_exit() in your main thread. If you do, protected applications may not be terminated properly, and you may have to kill the process explicitly.
- Flags for dlopen can be later promoted if a shared library has been loaded with dlopen. To promote a flag, RTLD_NOLOAD is used with other flags. For example, a library that was previously loaded with RTLD_LOCAL can be reopened with RTLD_NOLOAD | RTLD_GLOBAL.
- For anti-debugging, the RTLD_NOLOAD flag cannot be used without the RTLD_NODELETE flag. For example, a protected library that was previously loaded with RTLD_LOCAL must be reopened with RTLD_NOLOAD | RTLD_NODELETE | RTLD_GLOBAL.
- Do not protect a custom locked shared library with a custom locked license.

For Using the Protected Applications (Redistributables)

RMS Envelope automatically copies the resources required by the protected application in its directory. You need to redistribute all required resources together with the protected application. This redistributable package typically consists of:

- **Your Protected application**
  RMS Envelope protected application/shared object.
- **Customized library**
  If the `customLib` option is used while protecting the application, the customized library should be shipped with the protected application.
- For standalone licenses, the license file must be available with the application in the same folder.

> **Note:** In addition, you may need to explicitly include the SCP configuration file to use the protected application in the Cloud Served - Lease Standalone mode. This is NOT copied automatically by RMS Envelope. Place it in the same directory as the protected application. For more information, see the Standalone Mode section of the SCP Installation and Configuration Guide.

Supported GNU libc (glibc) Version

- For 32-bit applications, glibc v2.4 (or later)
- For 64-bit applications, glibc v2.14 (or later)

See Also: Supported Platforms

For Generating RMS Licenses

You can generate RMS licenses using the tools described below. Contact Gemalto Sales Representative or Technical Support for assistance on obtaining these tools.

- lscgen - A command-line-based utility, available on Windows and Linux, that generates a license code.
- RMS License Code Generation Library API - The license code generation API functions help you to create your own custom license generator. For more information, refer to the Sentinel RMS SDK License Generation API Reference Guide.
- Sentinel EMS - The Sentinel license and Entitlement management solution.

> **Note:** RMS Envelope supports RMS license version 18 (or later).

Using RMS Envelope in Evaluation Mode

The command-line RMS Envelope provides the `--eval` option for protecting applications in this mode. To use RMS Envelope command-line application:

1. Open the terminal.
2. Go to the directory that contains RMS Envelope command-line application.
3. Use the following command to start RMS Envelope command-line application:

```
./SentinelRMSEnvelope --eval [options] <infile> <outfile>
```

For example:

```
./SentinelRMSEnvelope --eval -f:DOTS -v:1.0 -lib:<absolute path to the licensing library>
program program_protected
```

Notes

- The Sentinel LDK Developer Key and Sentinel LDK Runtime are not required for protecting applications in evaluation mode.
- Applications protected using evaluation mode of RMS Envelope display the following message at startup:
  - *This application is protected using demo version of Sentinel RMS Envelope.*
- In evaluation mode, applications protection period is restricted to the maximum of 90 days.
- The evaluation period starts from the date of application protection.
- To run the applications protected using evaluation mode, the vendor also requires the RMS license for a feature name and feature version combination specified at the time the application is protected. For more information, refer to the Prerequisite section.

Running RMS Envelope

RMS Envelope can be initiated using a command-line prompt. To use RMS Envelope command-line application:

1. Open the terminal.
2. Go to the directory that contains RMS Envelope command-line application.
3. Use the following command to start RMS Envelope command-line application:

```
./SentinelRMSEnvelope [options] <infile> <outfile>
```

For example:

```
./SentinelRMSEnvelope -f:DOTS -v:1.0 -lib:<Absolute path to licensing library> program program_protected
```

Where,

Item	Description
options	Protection options for additional security. The list of protection options is defined in the Mandatory Protection Options and Optional Protection Options sections.
infile	The application or shared object that needs to be protected. If the application/shared object is not available in the RMS Envelope directory, provide the absolute path of the application/shared object.
outfile	The resulting protected file. If an absolute path is not specified for storing the application/shared object, the file will be stored in the RMS Envelope directory.

Note: For Linux applications that were protected using Envelope: The installer for the protected application should determine if libXaw libraries are present on the end user's computer and, if not, install them.

Mandatory and Optional Protection Options

This section outlines the mandatory and customizable options that can be specified for protecting software with RMS Envelope:

Mandatory Protection Options

The following information must be provided in order to protect an application or shared objects using RMS Envelope:

- `-lib` - Absolute path of the Sentinel RMS licensing library.
- `-f: --fname` - Feature name
- `-v: --ver` - Feature version (required if a version is specified in the license)
- Input file location
- Output file location

Optional Protection Options

The list of protection options is defined in the Basic Protection Options and Advanced Protection Options sections. Except for the options included in the Mandatory Protection Options section, all other protection options are not compulsory.

Basic Protection Options

The table below describes the basic protection options that you can set while protecting your application using RMS Envelope:

Option	Description	Default Setting
-b: --bgchk:<time>	Enables you to specify the time interval for performing background checks. The protected application checks for the presence of a valid license after the specified time interval. If the background check value is greater than the key lifetime value of the license, the license check will be done according to the key lifetime value. Allowed values are: • 0 = Background checks disabled. In this case: – The license is released immediately after launching the application. – The license is not required for continuous use of the protected application. • Integers= Time intervals (in seconds) for background checks. Use of this option is recommended to periodically check that the licensing session is alive and is not bypassed at any point during the protected application execution. However, note the following limitations: **Limitations** • If the License Manager goes down while a protected console application that included a background check is running, the application is suspended. When the License Manager is restarted, the application resumes, but it goes into the background. The application can be brought to the foreground by using the shell built-in `fg` command from the same terminal where application had been launched. To bring a background application to the foreground: a. List all running jobs using command `jobs`. b. Choose the relevant job ID from the list. c. Enter `fg %[jobID]` to bring the application to the foreground. • Background license checks are not supported for interactive command-line applications (for example, FTP and SSH).	Default: 300 seconds.

Option	Description	Default Setting
-cs: --csrv:<contact server>	This option is used for specifying the License Manager. **Notes** • For network licensing, specify the hostname or IP address of the machine where the Sentinel RMS License Manager is installed. • For standalone licensing, set NO-NET as the value of this option. • Alternatively, the License Manager name can be set using the LSHOST or the LSFORCEHOST environment variables.	If no License Manager name is set, the application looks for the license first on the local computer, and then it will make a broadcast for a license, looking for License Managers in the subnet.
--eval	Protects the application in This mode. The Sentinel LDK Developer key is not required for protecting applications in evaluation mode • If this option is used, the application will be protected in evaluation mode only, even if the Sentinel LDK Developer key is available. • In this mode, the protected application can be used for a period of up to 90 days, starting from the day it is protected. • For more information, refer to the Using RMS Envelope in Evaluation Mode section.	-
-f: --fname: <feature name>	Mandatory option. A feature identifies a suite of application, an application, a file, or a functionality of the software that needs to be licensed. The feature name can consist of alphanumeric characters, without spaces (in the ASCII range of 32-127). • The maximum length of the feature name is 24 characters. • The specified feature name should match the feature name that was specified in the license at the time of license generation.	-
-h --help	Displays user help.	-
-lib:<library absolute path>	Mandatory option. The absolute path pointing to the Sentinel RMS licensing library. • To protect a 32-bit application/shared object, provide the path of the 32-bit library • To protect a 64-bit application/shared object, provide the path of the 64-bit library.	-
--msg-out:<val>	Sets how the run-time user messages are displayed. Possible values are:	1

Option	Description	Default Setting
	• 0 = no message output • 1 = GUI • 4 = console • 5 = GUI and console	
-S1:<secret>...-S7:<secret>	Use this option to specify the secret strings for the challenge-response mechanism. The challenge-response mechanism is a technique used for authenticating the License Manager. The challenge strings (secrets) you define are encrypted within the license, with only the License Manager knowing how to decrypt them. The License Manager associates a secret with a feature, provided by the license code. The application also contains this secret. In the License Manager validation process, the protected application sends a "challenge" to the License Manager with a data string. The License Manager computes a response based on to the arranged algorithm, the values, the data string, and the secret, which it to the protected application. The protected application computes the expected response locally using data string and the secret, and verifies that the expected response matches the response returned by the License Manager. **Notes** • You can define up to 7 secrets (1 to 7) for the challenge-response mechanism. • Each secret can contain up to 12 printable characters. • The secrets specified here should match the secrets defined in the license. • If the license contains multiple secrets, you can specify fewer secrets in an exact sequence. For example, If the license contains 7 secrets (S1...S7), you can choose to specify only 3 of the secrets (S1, S2, and S3).	Disabled.
-t: --enable-ts:<val>	Enables the protected application to run on a Terminal Server/ Remote Desktop. The available options are: • 0 - Disable server(TS), Disable RDP • 1 - Disable server(TS), Enable RDP • 2 - Enable server(TS), Enable RDP	1

Option	Description	Default Setting
-v: --ver:<feature ver>	Feature version. Mandatory if a version is specified in the license. The maximum length of the version is 11 characters. • The specified feature version should match the feature version specified in the license. • Do not use this option if the license does not contain a feature version.	-

Advanced Protection Options

The table below describes the advanced protection options available.

Option	Description	Default Setting
-customFunction:<name>	Custom lock function name. The maximum length of the custom function name is 32 characters. The custom function name should match the name defined in the custom library.	Optional.
-customLib:<name>	Absolute path of the customized 32/64-bit library. Use this option for locking licenses to a hardware device or software-based implementation, to generate a unique extended custom value for each machine. For More information about extended custom locking, refer to the Callback API section of the Sentinel RMS SDK API Reference Guide. **Notes** • Provide the path of the customized library. • Make sure that the customized library is available inside the RMS Envelope directory when the protected application is executed. • The maximum length of the custom library name is 32 characters. • To protect a 32-bit application/shared object, provide the path of the customized 32-bit library • To protect a 64-bit application/shared object, provide the path of the 64-bit customized library.	Optional.
-d --debug	Allows debugging of the protected application.	Enabled.
-ig: --ignore:<count>	Defines the number of times an application can be resumed in absence of a valid license. Possible values are: • 0= Abort or Retry • 1…254 - Ignore count value. For example, if the ignore count value is 5, your can ignore the license unavailability error 5 times. • 255= No limit  **Note:** If the value of--msg-out:<val> option is specified as *console*, this option is not supported for console applications.	0

Option	Description	Default Setting
--memdump	If this parameter is present, memory dumps can be generated for the protected application. **Notes** • If memory dumps are disabled, debugging is automatically disabled. This overrides the *--debug* parameter. • This parameter is applicable for both executables and shared objects. You can use either of two methods to take a memory dump: – kill signal - When using this method, specify --memdump to enable memory dumps for the protected application. – gcore (or other memory dump tools) - When using this method, you must specify both --memdump and --debug. Tools like gcore attempt to attach to the process as debuggers, so debugger detection must be disabled.	Disabled.
-q --quiet	Displays error and warning messages only.	Optional.
--wchar	Writes run-time errors as wide character strings. This option is required when you specify that messages will be output to a console (Specify 4 (stderr) in the *--msg-out:<val>* option).	Disabled.

Known Issues

The following known issues exist in RMS Envelope v1.0:

User Story/Service Request ID	Description
LDK-4545	Applications that do not link any object dynamically cannot be protected.

Frequently Asked Questions

The following are frequently-asked frequently asked questions related to RMS Envelope:

Related to Applications Supported for RMS Envelope Protection

Question: Which type of applications can be protected using RMS Envelope?

RMS Envelope can protect native C 32/64-bit ELF executables and shared objects.

Question: Can I protect Java executables using RMS Envelope?

No. RMS Envelope does not support protection of Java executables.

Related to Sentinel RMS SDK Compliance

Question: Which version of the RMS SDK is supported for using RMS Envelope?

RMS Envelope supports v9.2.1 (or later) of the RMS SDK.

Question: Which RMS license versions are supported by RMS Envelope?

RMS Envelope supports RMS license version 18 (or later).

Question: Does RMS Envelope support extended custom (CustomEx) locking?

Yes. You can lock licenses to a hardware device or to a software-based implementation to generate a unique fingerprint value not exceeding 64-bytes for each machine.

Supporting this requires you to implement the customized locking logic in your application first. For more information about the extended custom locking, refer to the Callback API section of the Sentinel RMS SDK API Reference Guide.

Related to Sentinel RMS Licenses

Question: How can I generate a license for an RMS Envelope-protected application?

See the topic: For Generating Licenses.

Question: How can I generate a license for an RMS Envelope-protected application using the Sentinel Entitlement Management System (Sentinel EMS)?

The Sentinel EMS users can perform product activation (license generation) using the instructions provided here.

Question: How does an RMS Envelope-protected application finds a license?

The license search mechanism is defined here. To enhance the license search mechanism for protected applications, do one of the following:

- Define the contact server while protecting an application.
- Use the LSHOST and LSFORCEHOST environment variable on the computer that is running a protected application.

Technical Support

You can contact us using any of the following options:

Business Contacts

To find the nearest office or distributor, use the following URL:
https://sentinel.gemalto.com/contact-us-sm/

Technical Support

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

- Customer Support Portal: (Preferred)
  - https://supportportal.gemalto.com/csm?id=sentinel
- Phone:
  - AMER: 800-545-6608 (US toll free), +1-410-931-7520 (International)
  - EMEA/APAC: https://supportportal.gemalto.com/csm?id=sentinel
    Click “Contact us”
- E-mail (only if having issue submitting the technical issue via portal)
  technical.support@gemalto.com

Downloads

You may want to explore updated installers and other components here:
https://sentinelcustomer.gemalto.com/sentineldownloads/