eToken Linux Middleware version 2.0 Beta Guide

August 2005

Contact Information

Support

If you have any questions regarding this package, its documentation and content or how to obtain a valid software license you may contact your local reseller or Aladdin's technical support team:

Country / Region	Telephone
USA	1-212-329-6658 1-800-223-4277
EUROPE: **Austria, Belgium, France, Germany, Netherlands, Spain, Switzerland, UK**	00800-22523346
Ireland	0011800-22523346
Rest of the World	+972-3-6362266 ext 2

If you want to write to the eToken Technical Support department, please go to the following web page:

http://www.Aladdin.com/forms/eToken_question/form.asp

Website

http://www.Aladdin.com/eToken

COPYRIGHTS AND TRADEMARKS

DISCLAIMER

ALADDIN KNOWLEDGE SYSTEMS LTD.

DEVELOPER'S LICENSE AGREEMENT

IMPORTANT INFORMATION - PLEASE READ THIS AGREEMENT CAREFULLY BEFORE OPENING THE PACKAGE AND/OR USING THE CONTENTS THEREOF AND/OR BEFORE DOWNLOADING OR INSTALLING THE SOFTWARE PROGRAM. ALL ORDERS FOR AND USE OF THE ETOKEN PRODUCTS (including without limitation, the Developer's Kit, libraries, utilities, diskettes, CD-ROM, eToken® keys and the Developer's Guides) (hereinafter "Product") SUPPLIED BY ALADDIN KNOWLEDGE SYSTEMS LTD. (or any of its affiliates - either of them referred to as "ALADDIN") ARE AND SHALL BE, SUBJECT TO THE TERMS AND CONDITIONS SET FORTH IN THIS AGREEMENT. BY OPENING THE PACKAGE CONTAINING THE PRODUCTS AND/OR BY DOWNLOADING THE SOFTWARE (as defined hereunder) AND/OR BY INSTALLING THE SOFTWARE ON YOUR COMPUTER AND/OR BY USING THE PRODUCT, YOU ARE ACCEPTING THIS AGREEMENT AND AGREEING TO BE BOUND BY ITS TERMS AND CONDITIONS.

IF YOU DO NOT AGREE TO THIS AGREEMENT DO NOT OPEN THE PACKAGE AND/OR DOWNLOAD AND/OR INSTALL THE SOFTWARE AND PROMPTLY (at least within 7 days from the date you received this package) RETURN THE PRODUCTS TO ALADDIN, ERASE THE SOFTWARE, AND ANY PART THEREOF, FROM YOUR COMPUTER AND DO NOT USE IT IN ANY MANNER WHATSOEVER.

1. Title & Ownership

THIS IS A LICENSE AGREEMENT AND NOT AN AGREEMENT FOR SALE. The software component of Aladdin's eToken Software Development Kit, including any revisions, corrections, modifications, enhancements, updates and/or upgrades thereto, (hereinafter in whole or any part thereof defined as: "Software"), and the related documentation, ARE NOT FOR SALE and are and shall remain in Aladdin's sole property. All intellectual property rights (including, without limitation, copyrights, trade secrets, trademarks, etc.) evidenced by or embodied in and/or attached/connected/related to the Product, are and shall be owned solely by Aladdin. This License Agreement does not convey to you an interest in or to the Software, but only a limited right of use revocable in accordance with the terms of this License Agreement. Nothing in this Agreement constitutes a waiver of Aladdin's intellectual property rights under any law.

2. License

Subject to payment of applicable license fees, Aladdin hereby grants to you, and you accept, a personal, nonexclusive and fully revocable limited License to use the Software, in executable form only, as described in the Software accompanying user documentation and only according to the terms of this Agreement:

2.1 You may install the Software and use it on computers located in your place of business, as described in Aladdin's related documentation.

2.2 You may merge and link the Software into your computer programs for the sole purpose described in the Developer's Guide; however, any portion of the Software merged into another computer program shall be deemed as derivative work and will continue to be subject to the terms of this Agreement. The Software shall not be used for any other purposes.

3. Sub-Licensing

After merging the Software in your computer program(s) according to section 2, you may sub-license, pursuant to the terms of this Agreement, the merged Software and resell the hardware components of the eToken® keys which you purchased from Aladdin, to distributors and/or users. Preceding such a sale and sub-licensing, you shall incorporate by reference in your contracts with such distributors and/or users, and otherwise provide for all distributors and/or users to be bound by, the warranties, disclaimers, and license terms specified by Aladdin in this Agreement.

4. Prohibited Uses

Except as specifically permitted in Sections 1, 2 and 3 above, you agree not to:

4.1 Use, modify, merge or sub-license the Software or any other of Aladdin's Products, except as expressly authorized in this Agreement and in the Developer's Guide.

4.2 Sell, license (or sub-license), lease, assign, transfer, pledge, or share your rights under this License with/to anyone else.

4.3 Modify, disassemble, decompile, reverse engineer, revise or enhance the Software or attempt to discover the Software's source code.

4.4 Place the Software onto a server so that it is accessible via a public network.

4.5 Use any back-up or archival copies of the Software (or allow someone else to use such copies) for any purpose other that to replace an original copy if it is destroyed or becomes defective. If you are a member of the European Union, this agreement does not affect your rights under any legislation implementing the EC Council Directive on the Legal Protection of Computer Programs. If you seek any information within the meaning of that Directive you should initially approach Aladdin.

5. Limited Warranty

Aladdin warrants, for your benefit alone, that:

5.1 The Software, when and as delivered to you, and for a period of three (3) months after the date of delivery to you, will perform in substantial compliance with the Developer's Guide, provided that it is used on the computer hardware and with the operating system for which it was designed.

5.2 The eToken® key, for a period of twelve (12) months after the date of delivery to you, will be substantially free from significant defects in materials and workmanship.

6. Warranty Disclaimer

ALADDIN DOES NOT WARRANT THAT ANY OF ITS PRODUCT(S) WILL MEET YOUR REQUIREMENTS OR THAT ITS OPERATION WILL BE UNINTERRUPTED OR ERROR-FREE. TO THE EXTENT ALLOWED BY LAW, ALADDIN EXPRESSLY DISCLAIMS ALL EXPRESS WARRANTIES NOT STATED HERE AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NO ALADDIN'S DEALER, DISTRIBUTOR, RESELLER, AGENT OR EMPLOYEE IS AUTHORIZED TO MAKE ANY MODIFICATIONS, EXTENSIONS, OR ADDITIONS TO THIS WARRANTY. If any modifications are made to the Software or to any other part of the Product by you during the warranty period; if the media and the eToken® key is subjected to accident, abuse, or improper use; or if you violate any of the terms of this Agreement, then the warranty in Section 5 above, shall immediately be terminated. The warranty shall not apply if the Software is used on or in conjunction with hardware or program other than the unmodified version of hardware and program with which the Software was designed to be used as described in the Developer's Guide.

7. Limitation of Remedies

In the event of a breach of the warranty set forth above, Aladdin's sole obligation shall be, at Aladdin's sole discretion:

7.1 To replace or repair the Product, or component thereof, that does not meet the foregoing limited warranty, free of charge.

7.2 To refund the price paid by you for the Product, or component thereof. Any replacement or repaired component will be warranted for the remainder of the original warranty period or 30 days, whichever is longer. Warranty claims must be made in writing during the warranty period and within seven (7) days of the observation of the defect accompanied by evidence satisfactory to Aladdin. All Products should be returned to the distributor from which they were purchased (if not purchased directly from Aladdin) and shall be shipped by the returning party with freight and insurance paid. The Product or component thereof must be returned with a copy of your receipt.

8. Exclusion of Consequential Damages

The parties acknowledge that Product is inherently complex and may not be completely free of errors. ALADDIN SHALL NOT BE LIABLE (WHETHER UNDER CONTRACT, TORT (INCLUDING NEGLIGENCE) OR OTHERWISE) TO YOU, OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE (INCLUDING INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES), INCLUDING, WITHOUT LIMITATION, ANY LOSS OR DAMAGE TO BUSINESS EARNINGS, LOST PROFITS OR GOODWILL AND LOST OR DAMAGED DATA OR DOCUMENTATION, SUFFERED BY ANY PERSON, ARISING FROM AND/OR RELATED WITH AND/OR CONNECTED TO ANY USE OF THE SOFTWARE AND/OR ANY COMPONENT OF THE PRODUCT, EVEN IF ALADDIN IS ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

9. Limitation of Liability

IN THE EVENT THAT, NOTWITHSTANDING THE TERMS OF THIS AGREEMENT, ALADDIN IS FOUND LIABLE FOR DAMAGES BASED ON ANY DEFECT OR NONCONFORMITY OF ITS PRODUCT(S), ITS TOTAL LIABILITY FOR EACH DEFECTIVE PRODUCT SHALL NOT EXCEED THE PRICE PAID TO ALADDIN FOR SUCH DEFECTIVE PRODUCT.

10. Termination

Your failure to comply with the terms of this Agreement shall terminate your license and this Agreement. Upon termination of this License Agreement by Aladdin:

10.1 The License granted to you in this Agreement shall expire and you, upon termination, shall discontinue all further use of the Software and other licensed Product(s);

10.2 You shall promptly return to Aladdin all tangible property representing Aladdin's intellectual property rights and all copies thereof and/or shall erase/delete any such information held by it in electronic form. Sections 1, 4, 6, 7, 8, 9, 10 and 11 shall survive any termination of this Agreement.

11. Governing Law & Jurisdiction

This Agreement shall be construed and governed in accordance with the laws of Israel (except for conflict of law provisions) and only the courts in Israel shall have jurisdiction in any conflict or dispute arising out of this Agreement. The application of the United Nations Convention of Contracts for the International Sale of Goods is expressly excluded. The failure of either party to enforce any rights granted hereunder or to take action against the other party in the event of any breach hereunder shall not be deemed a waiver by that party as to subsequent enforcement of rights or subsequent actions in the event of future breaches.

12. Government Regulation and Export Control

You agree that the Product will not be shipped, transferred, or exported into any country or used in any manner prohibited by applicable law. It is stipulated that the Product is subject to certain export control laws, rules, and/or regulations, including, without limiting the foregoing, to the United States and/or Israeli export control laws, rules, and/or regulations. You undertake to comply in all respects with the export and reexport restriction as set forth herein and any update made thereto from time to time.

13. Third Party Software

If the Product contains any software provided by third parties, such third party's software is provided "As Is" without any warranty of any kind and Sections 2, 3, 4, 6, 8, 9-12 of this Agreement shall apply to all such third party software providers and third party software as if they were Aladdin and the Product respectively.

14. Miscellaneous

This Agreement represents the complete agreement concerning this License and may be amended only by a written agreement executed by both parties. If any provision of this Agreement is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

I HAVE READ AND UNDERSTOOD THIS LICENSE AGREEMENT AND AGREE TO BE BOUND BY ALL OF THE TERMS.

FCC Compliance

eToken USB has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation.

This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one of the following measures:

a. Reorient or relocate the receiving antenna.

b. Increase the separation between the equipment and receiver.

c. Connect the equipment to an outlet on a circuit different from that to which the receiver is connected.

d. Consult the dealer or an experienced radio/TV technician.

FCC Warning

Modifications not expressly approved by the manufacturer could void the user authority to operate the equipment under FCC rules.

All of the above applies also to the eToken USB.

FCC authorities have determined that the rest of the eToken product line does not contain a Class B Computing Device Peripheral and therefore does not require FCC regulation.

CE Compliance

The eToken product line complies with the CE EMC Directive and related standards*. eToken products are marked with the CE logo and an eToken CE conformity card is included in every shipment or upon demand.

*EMC directive 89/336/EEC and related standards EN 55022, EN 50082-1.

UL Certification

The eToken product line successfully completed UL 94 Tests for Flammability of Plastic Materials for Parts in Devices and Appliances. eToken products comply with UL 1950 Safety of Information Technology Equipment regulations.

ISO 9002 Certification

The eToken product line is designed and manufactured by Aladdin Knowledge Systems, an ISO 9002-certified company. Aladdin's quality assurance system is approved by the International Organization for Standardization (ISO), ensuring that Aladdin products and customer service standards consistently meet specifications in order to provide outstanding customer satisfaction.

Certificate of Compliance

Upon request, Aladdin Knowledge Systems will supply a Certificate of Compliance to any software developer who wishes to demonstrate that the eToken product line conforms to the specifications stated. Software developers can distribute this certificate to the end user along with their programs.

Table of Contents

Chapter 1

Introduction

About This Chapter

This Guide introduces Aladdin's eToken™ middleware version 2.0 Beta for Linux.

The eToken middleware explores the standard API (PKCS#11), which makes it usable for application development as well as for using with PKCS#11-enabled ready made applications. In particular, eToken middleware has been tested with the following applications:

- Mozilla Firefox
- Mozilla Thunderbird

This document is intended primarily for use by developers and project managers, and explains in detail how to use the middleware for developing applications that customize the use of eToken for specific organization or client requirements. It does not cover the settings that are required in the ready made applications in order to work with eToken.

This introductory chapter includes the following sections:

- eToken Middleware for Linux Architecture on page 2 which details the elements included in the Linux middleware.
- Hardware and Software Requirements on page 5 describes the necessary essentials needed to use the middleware.
- eToken Middleware Kit on page 6 itemizes the prerequisites needed for and the contents included in the middleware kit.

eToken Middleware for Linux Architecture

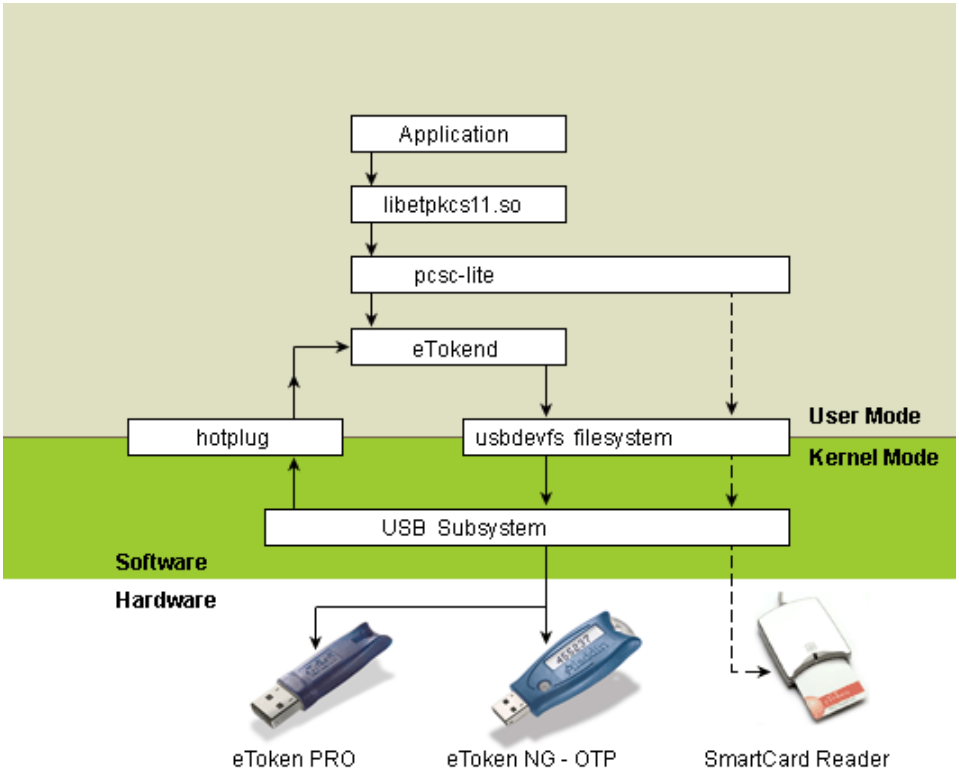
The Architecture of the eToken middleware for Linux consists of the following:

- Hardware Layer
- Kernel Layer
- PC/SC Layer
- PKCS#11 API

The first two layers belong to the operating system layer and are beyond the scope of this document.

Figure 1-1 is a graphical representation of this eToken architecture.

Figure 1-1 eToken Middleware Linux Architecture



Hardware Layer

The hardware layer comprises the tokens, smartcards, smartcard readers and the computer's USB controller.

The current version of eToken software supports only CardOS/M4 eTokens. These are the eToken PRO and eToken NG-OTP tokens. R2 tokens are not supported.

Note:
The eToken CardOS/M4 is commonly referred to as the eToken PRO.

PC/SC Layer

PC/SC standard was created by the PC/SC workgroup (http://pcscworkgroup.com/) in order to promote smartcard interoperability and to facilitate the development of smartcard applications.

The M.U.S.C.L.E. project (http://www.linuxnet.com/) aims to promote the use of smartcards in the Linux environment. This project provides a light-weight implementation of the PC/SC standard, which is called PC/SC Lite. This project also provides a range of other tools for using smartcards.

Aladdin's eToken middleware uses the following components in order to work with the PC/SC Lite infrastructure:

1. The etokend daemon – This component is responsible for keeping track of which tokens are inserted into the computer, and for communicating with them. etokend relies on hotplug events and the upper parts of the PC/SC layer for detecting insertion or removal of tokens.
2. aksifdh.so – This is a small shared library that is loaded by M.U.S.C.L.E.'s pcscd daemon and used for communicating with the token (through etokend).

PKCS#11 API

The middleware supports the PKCS#11 industry standard.

For additional information please refer to the following sources:

1. The RSA Labs website at www.rsalabs.com for the PKCS#11 specification.
2. The PKCS#11 section on page 14 for more details on PKCS#11 support for Linux, and details about interoperability issues with the Windows version of the library.

Hardware and Software Requirements

The Linux kernel can operate on many different hardware platforms and it comes packaged as part of many different GNU/Linux distributions.

Only a subset of these hardware/distribution combinations has been well tested with the eToken middleware. These are specified below as part of the requirements for the eToken middleware.

The eToken middleware should also work on many other Linux distributions, beside the ones below. Please contact eToken technical support if you need any assistance with using eToken on any of these other distributions.

Hardware Requirements

- Intel:CPU from the Intel i386 family of processors (includes 80386, 80486, the various Pentium processors and similar processors).
- At least one UHCI, OHCI or EHCI USB host controller.
- eToken CardOS/M4 (eToken PRO and eToken NG-OTP).

Supported Distributions

The eToken middleware supports the following distributions:

- Fedora Core 4
- Red Hat Enterprise Linux 4
- SuSe 9.3

Supported Tokens

The PKCS#11 library only supports the eToken CardOS/M4. These tokens are the eToken PRO and eToken NG-OTP.

eToken Middleware Kit

The eToken Middleware Kit for Linux contains everything you need to evaluate and start using eToken in the Linux environment and to develop your own applications.

eToken MIDDLEWARE for Linux Prerequisites

Before you install the eToken middleware, make sure you have the following:

- One of the supported Linux distributions.
- PC/SC Lite version 1.2.0 installed.
- At least one USB port.
- At least one eToken Security Key (either eToken NG-OTP or the eToken PRO (CardOS/M4)).

Chapter 2

Installation

About This Chapter

This chapter deals with the installation and removal of Linux middleware version 2.0 Beta on your system. All details needed for installation are provided and step-by-step instructions for loading PC/SC Lite, the middleware and its component parts – PKCS#11 and PC/SC support - are provided.

The chapter covers the following sections:

- Installing PC/SC Lite on page 8 details how to install PC/SC Lite correctly.
- Installing eToken Middleware for Linux on page 9 describes the process of installation for the required elements and lists the installed files.
- Uninstalling eToken Middleware for Linux on page 11 provides the procedure needed to uninstall the Linux middleware.

Installing PC/SC Lite

Before you install the eToken middleware you should install **PC/SC Lite version 1.2.0**.

➢ **To install PC/SC Lite:**

1. Download **PC/SC Lite version 1.2.0** from the M.U.S.C.L.E. website at http://www.linuxnet.com/middle.html.
2. Ensure you select the correct version and follow instructions for downloading and installing **PC/SC Lite version 1.2.0**.
3. Check that the pcscd daemon has started. If not, then start it with the command:
   /etc/init.d/*pcscd start*
4. If you need support for 2048-bit keys, you will need to define the macro for enhanced messaging before you compile pcsd.

Note:

If you need support for 2048-bit keys, you will need to define the macro for enhanced messaging before you compile pcsd as follows:

Build pcsc-lite with the macro PCSCLITE_ENHANCED_MESSAGING defined.

Before you start to build pcsc-lite, go to the file src/pcsclite.h and insert the following string

#define PCSCLITE_ENHANCED_MESSAGING

Before the #ifndef PCSCLITE_ENHANCED_MESSAGING

Then build the project as usual.

Installing eToken Middleware for Linux

Make sure you are the root user. If you are not the root user, then **type** "su" and enter the root password in order to become root.

➢ **To install the eToken Middleware:**

Issue the following commands:
tar –xzf etoken-3-60.-linux-i386.tar.gz*
cd etoken-3-60.-linux-i386.tar.gz*
./petoken install <number-of-readers>.

The asterisk * stands for the particular build number.

Notes:

1. The number of readers may vary between 1 and 8. If the number is not specified, the default number of readers is 1.

2. The etokend daemon should be started before the pcscd daemon.
If the pcscd daemon is already running when you install the PC/SC support, this installation script will stop pcscd before installing Aladdin's software and restart it afterwards.

3. Sometimes, when trying to start pcscd, it fails to begin. In such a case, delete the files
"/var/run/pcscd.pub and /var/run/pcscd.comm" and try again.

4. There is a bug in the current version of pcsc-lite (1.2.0), which causes a crash of the pcscd, while using the driver for Athena reader for smart card. To mitigate this problem, a patch is needed to be applied to the pcscd code. In the future, there may be an additional release of eToken software with pre-build patched pcscd binaries.

Installed Files

Table 2-1 lists all the files that are installed on the Linux platform. The asterisk stands for a specific version and build number.

Table 2-1: Installation Files

File Name	
Distribution Archive	**Linux**
reader.conf	/etc/reader.conf
etokend.startup.script	/etc/init.d/etokend
etoken	/etc/hotplug.d/usb/etoken.hotplug
etokend	/usr/local/sbin/etokend
etckdump	Install directory
etckinit	Install directory
etsrvd	/usr/local/sbin/etsrvd
etsrvd.startup.script	/etc/init.d/etsrvd
libetokendll.so*	/usr/local/lib/libetokendll.so
libetpkcs11.so*	/usr/local/lib/libetpkcs11.so
pcscd.startup.script	/etc/init.d/pcscd
aksifdh.so.*	/usr/local/sbin/aksifdh.so
README	Install directory
LicenseAgreement.rtf	Install directory

Notes:

1. The installation packages include some files with the suffix -.startup.script. These files are copied to /etc/init.d/ and the suffix -.startup.script is dropped.

2. In the SuSe distribution, there is an additional script – etstart.suse – which is only used by the RTE during installation and should not be started by the user at a later stage.

Uninstalling eToken Middleware for Linux

Uninstalling the Linux middleware requires the removal of both PKCS#11 and PC/SC components. However you can remove just the PKCS#11 if desired.

➢ **To uninstall PKCS#11:**

1. cd etoken-3-60.*-linux-i386
2. change to the root user
3. run the command:

   ./petoken uninstall

Chapter 3

Developing Applications for eToken on Linux

About This Chapter

This chapter includes the following sections:

- PKCS#11 on page 14 describes the functionality and use of the PKCS#11 API including various tools and Windows interoperability.
- Redistribution on page 23 explains policy regarding redistribution of middleware elements.
- Troubleshooting on page 24 covers a number of problems that may arise and how to solve them.

PKCS#11

PKCS#11 is an API proposed by RSA Labs, which enables applications to access any security tokens that provide an implementation of the API.

libetpkcs11.so

libetpkcs11.so is Aladdin's implementation of PKCS#11 version 2.01 for Linux. The current version only supports the eToken CardOS/M4 tokens. eToken R2 tokens are not supported.

libetpkcs11.so is a dynamic link library that is installed in the /usr/local/lib/ directory.

It is recommended that you use dlopen() to explicitly load the library, and use dlsym() to obtain the address of C_GetFunctionList(). To obtain the table of other function pointers in the library, you should use C_GetFunctionList() and not use dlsym().

User Interactions

Generally, the PKCS#11 library should not require any user interaction. However in some cases, the eToken PKCS#11 provider may need a user password. In these cases, the eToken Windows RTE would open a pop-up dialog box asking for a password. The Linux PKCS#11 library will fail the function in these situations. There are two specific situations when a password would normally be required.

1. When an application tries to create a public object without first logging-in to the token. This behavior is allowed by the PKCS#11 specification, but is not allowed by the token. To avoid this situation, ensure that you log in before the creation or deletion of any object.
2. The C_InitToken function requires a password if the token was previously initialized. Refer to C_InitToken on page 18 to see how to work around this problem.

Object types

eToken supports a number of object types as detailed in Table 3-1

Table 3-1: Object Types Supported by eToken

Object Type	PKCS#11 Object Attributes	Comments
Data object	CKA_CLASS=CKO_DATA	
Certificate	CKA_CLASS=CKO_CERTIFICATE, CKA_CERTIFICATE_TYPE=CKC_X_509	Only X.509 certificates are supported
RSA private key	CKA_CLASS=CKO_PRIVATE_KEY, CKA_KEY_TYPE=CKK_RSA	See specifications of RSA private keys for different tokens
RSA public key	CKA_CLASS=CKO_PUBLIC_KEY, CKA_KEY_TYPE=CKK_RSA	Operations with RSA public keys are performed in software
DES/Triple-DES secret keys	CKA_CLASS=CKO_SECRET_KEY, CKA_KEY_TYPE=CKK_DES or CKK_DES2 or CKK_DES3	DES operations are performed in software

Public Object Write Access

PKCS #11 mandates that even when no one is logged in to the eToken, public token objects can be created or modified. Since the eToken permits only the legitimate user to modify eToken contents, it is not possible to implement this feature as specified.

Mechanisms

eTpkcs11 supports a subset of the entire PKCS #11 mechanism set. This subset is enough to enable eTpkcs11 to integrate seamlessly into PKI environments like Entrust PKITM.

eToken RTE does not support the CKM_SHA1_RSA_PKCS mechanism currently. If the application needs to use it (to achieve interoperability with standards or with other applications) some work-around needs to be made in the application code.

The full list of mechanisms supported is detailed in Table 3-2.

Table 3-2: Supported Mechanisms

PKCS #11 Mechanisms
CKM_RSA_PKCS_KEY_PAIR_GEN
CKM_RSA_PKCS
CKM_RSA_X_509
CKM_DES_KEY_GEN
CKM_DES_ECB
CKM_DES_CBC
CKM_DES_CBC_PAD
CKM_DES3_KEY_GEN
CKM_DES3_ECB
CKM_DES3_CBC
CKM_DES3_CBC_PAD
CKM_MD5
CKM_SHA_1
CKM_PBE_SHA1_DES3_EDE_CBC

On the eToken CardOS/M4, RSA key generation and operations are implemented on-token.

eToken CardOS/M4 and PKCS #11 Functions

The following PKCS #11 functions cannot be used with the eToken CardOS/M4:

- Export RSA private key.
- Wrap RSA private key.
- Get RSA private key attribute values.
- Copy RSA private key object.

Functions

This section discusses behavior specifics of various PKCS#11 functions used in eToken implementation. Please note that:

- <> are used to indicate that the particular field will be initialized with actual information.
- [] are used for flags that may or may not be set depending on the actual state.

C_GetInfo

The following fields of the returned structure will be set as shown:

cryptokiVersion	= 2.1
manufacturerID	= Aladdin Ltd.
flags	= none (0)
libraryDescription	= eToken PKCS#11

C_GetSlotInfo

The following fields of the returned structure will be set as shown:

slotDescription	= <reader name>
manufacturerID	= Aladdin Ltd.
flags	= CKF_REMOVABLE_DEVICE | CKF_HW_SLOT | [CKF_TOKEN_PRESENT]

hardwareVersion = 0.0

firmwareVersion = 0.0

C_GetTokenInfo

The following fields of the returned structure will be set as shown:

label	= <token name>
manufacturerID	= Aladdin Ltd.
model	= "eToken OTPNG" or "eToken CardOS/M4"
flags	= CKF_LOGIN_REQUIRED | CKF_RNG | [CKF_USER_PIN_INITIALIZED]
ulMaxSessionCount	= CK_EFFECTIVELY_INFINITE
ulMaxRwSessionCount	= CK_EFFECTIVELY_INFINITE
ulMaxPinLen	= MAX_PIN_SIZE
ulMinPinLen	= MIN_PIN_SIZE
hardwareVersion	= "3.0" for PRO and NG-OTP
firmwareVersion	= <token formware version>

C_InitToken

This function should initialize the eToken. According to PKCS#11 v2.01, the password it receives as a parameter should serve as the new Security Officer (SO) password. However there is no way to present credentials needed to re-initialize the eToken (later versions of the PKCS#11 standard changed the meaning of this parameter).

In Windows, using the current password will open a pop-up dialog box. In Linux it will fail. To avoid this, do the following:

- **Open the session with the eToken**. The eToken RTE will allow the session to be opened, even with a non-initialized eToken in order to work around this problem.
- **Login as the SO**. As previously described, if the eToken has an administrator password this should be used. If not, then the user password should be used. For a non-initialized eToken this call will be ignored.
- **Close the session**. The session must be closed since it is prohibited to perform *C_InitToken* if at least one session with the eToken is open. The eToken will be logged out, but eToken RTE will keep the password for a while (for the next step).
- **Perform *C_InitToken.*** The same formatting password is passed as a parameter.

It is still necessary to initialize the user password by using C_InitPIN.

C_InitPIN

This function behaves exactly as it is defined in the PKCS#11 specification. It may need to be issued for an eToken that has just been initialized.

If the eToken was initialized by the eToken Properties management tool in eToken RTE 3.60, the PIN may or may not have been initialized. This would depend on what initialization parameters were set (default behavior is to initialize the PIN)

C_Login

For an eToken without Administrator Password:
PKCS #11 mandates two entities that can log on to a eToken: the user and the security officer. Since the eToken CardOS/M4 (without an administrator) has a single legitimate user that can be authenticated to the eToken, it was decided that the eToken's legitimate user is also the security officer for that eToken.

For an eToken CardOS/M4 with Administrator Password:
Commencing with RTE 3.51 for Windows, *eTpkcs11* uses the eToken CardOS/M4's administrator password whenever it exists (the eToken CardOS/M4 was formatted with an administrator's password) and a security object is required.

For an eToken CardOS/M4 with Uninitialized Format Type:
Commencing with RTE 3.51, *eTpkcs11* can initialize an eToken CardOS/M4 that does not have the AKS AID (0x6666). In this particular case, eTpkcs11 builds the AKS AID with a default user password and an administrator password, as set using C_InitToken, C_InitPIN or C_SetPIN (after dummy C_Login as security object).

C_GetObjectSize

According to the PKCS#11 specifications, the *C_GetObjectSize* function returns an approximate object size and may be slightly inaccurate. As a result, when deleting or creating an object (for example using *C_CopyObject)*, the reported number of bytes may not be completely accurate and should be monitored.

C_DeriveKey

No mechanisms for key derivation are currently implemented.

Tools

The eToken middleware installation package for Linux contains two simple utilities for use with PKCS#11. Both utilities access the token only through the PKCS#11 interface.

etckinit

The purpose of this utility is to initialize the token for use with Aladdin's PKCS#11 library. It calls C_InitToken() and C_InitPin() in order to initialize the token and set its passwords.

The parameters are as follows:

<slot num> <format-password> <user-password>

You should also note the following points:

- etckinit will erase all the PKCS#11 objects from the token. If several tokens are inserted into the system, you should make sure that you entered the right slot number.
- etckinit assumes that the PKCS#11 library is installed in /usr/local/lib/libetpkcs11.so.

etckdump

etckdump prints the PKCS#11 objects contained on the token in a human-readable form.

It accepts the following parameters:

- --pin=<token's user PIN>

  or

  --pinhex=<token's user PIN>.

  The --pin option expects the PIN to be given literally as normal text. The --pinhex expects to get the PIN as a series of bytes in hexadecimal. The --pinhex option

should be used if your PIN contains non-printable characters.

- --slot=<slot number>
- -v1 - provides more information about the objects on the token. Without this option, the program will only print the number of objects it detected, but will not show their contents.
- --cklib=<path to the pkcs library>. This option allows you to specify another PKCS#11 library to use. The default is /usr/local/lib/libetpkcs11.so.

Examples:

1. Print all the token objects on a token that is inserted into the first slot and whose PIN is “1111”:
   etckdump --pin=1111 --slot=0 -v1
2. The same as above, using the --pinhex option (The ASCII code of the character “1” is 49, or “31” in hexadecimal):
   etckdump --pinhex=”31 31 31 31” --slot=0 -v1
3. The same as above, with a non-printable PIN:
   etckdump --pinhex=”a2 be ac bb” --slot=0 -v1

Note:
You may also use the eToken Explorer provided with eToken SDK 3.60 for Windows to explore and modify objects on the token.

Windows Interoperability

The same eToken CardOS/M4 can be used on both Windows and Linux operating systems. For the most part, the data that is created on the token on one platform can be read and used on the other and vice-versa. However, the following points should be noted:

1. Windows platforms often use Microsoft Cryptographic APIs called CAPI and CertStore. The current version of Aladdin's PKCS#11 library is capable of accessing data that was written using these interfaces (on Windows). The access is read-only, namely, no new CAPI objects can be created via PKCS#11, nor can they be erased.
2. There is no similar restriction in the opposite direction. Certificates that are downloaded directly to the token using PKCS#11, on either Windows or Linux, can be accessed through the Microsoft APIs.

Redistribution

You may redistribute the PC/SC and PKCS#11 components (or their contents) as part of your application.

Troubleshooting

Problem	Possible Diagnosis	Solution
eToken does not light up.	**1.** Your kernel does not support usb or the appropriate modules are not loaded. **2.** etokend daemon crashed.	**1.** Check that etokend daemon is up. Print on the command-line: ps ax|grep etokend if the etokend doesn't run, shut down the pcscd and etsrvd daemons, and restart them in the following order: etokend,pcscd,etsrvd. /etc/init.d/service_name start You can just reboot your machine to fix this situation.
pcsc_scan (part of the pcsc_tools package that is available from www.linuxnet.com) does not detect that a token was inserted or removed.	**1.** etokend is not running. **2.** pcscd is not running **3.** etokend and pcscd were not started in the right order.	Stop both services and restart them: /etc/init.d/pcscd stop /etc/init.d/etokend start /etc/init.d/pcscd start
pcscd does not start.	The files /var/run/pcsc.pub and /var/run/pcscd.comm were not deleted the last time that pcscd was stopped.	**1.** Check that pcscd is not running already **2.** Delete /var/run/pcsc.pub and /var/run/pcscd.comm **3.** Restart pcscd
Two tokens are inserted into the computer. Mozilla shows the same token twice and does not show the certificates from the other token.	Both tokens have the same label.	When you initialize the tokens, give them different labels.

Problem	Possible Diagnosis	Solution
If using Firefox and Thunderbird, a user certificate that was downloaded from Firefox cannot be used to sign and encrypt mails in Thunderbird.	The Root CA of the required Certification Authority cannot be found in the computer's Root Certification Authorities list.	Download the Root CA(s) from the correct CA site and import them to the trusted Root CA. Edit the Root CA in both Firefox and Thunderbird that has been imported to enable using them for signed mails and an SSL connection.