Keyrona
User Manual

unified key management solution
for hard disk and file encryption systems

Manual version 0.5
Dedicated Keyrona version 1.0-rc4

Keyrona User Manual – Table of Content:

Table of Content

1. Introduction

The Keyrona project aims at developing a unified key management solution for hard disk and file encryption systems such as cryptsetup-luks, eCryptFS, and TrueCrypt. The key management layer shall fulfill requirements of end users, SME's, and public authorities, as well as supporting different authentication systems such as passwords, smartcards, and Trusted Platform Modules (TPM).

2. Installation

In order to use Keyrona, you will need a GNU/Linux distribution (e.g., Fedora, Debian or Gentoo) as well as a recent Linux kernel. This kernel needs to have support for crypto and device-mapper as well as TPM-support. Currently keyrona supports dm-crypt, ecryptfs, truecrypt, encfs and cryptsetup-luks. You must install at least one of them to make any use of keyrona:

- cryptsetup
- cryptsetup-luks
- dm-setup
- truecrypt
- ecryptfs-utils
- encfs
- OpenSSL
- TrouSerS

Additionally, you will need the following Linux tools:

- scons
- make
- unzip
- patch
- ldconfig
- dos2unix

2.1 Download Keyrona

You can get Keyrona either as a .tar.gz-file from:

https://projects.sirrix.com/trac/TPM-KM/browser/release

and unpack it via:

 # tar -xzf keyrona-<ver>.tar.gz

or checkout the Subversion-repository:

 # svn co https://projects.sirrix.com/svn/TPM-KM

2.2 Install Keyrona

In order to install Keyrona, use the 'install.sh'-script.
(Pre-compiled packages for Fedora and Debian / Ubuntu are also available).
You need to be root to execute the script:

 # sudo ./install.sh

The script first checks the dependencies, that are required for keyrona to work.
If everything works out correctly you should see :

=> Done... ;)

The next important step is adding the local users to the keyrona group to allow them to execute the *keyrona_mount* command:

 # sudo useradd -G keyrona username_xy

Alternatively, you can modify the */etc/group* file directly.

2.3 Add Keyrona to runlevel

After installation, an init-script is available in */etc/init.d*. In order to automatically start the Keyrona keyprovider upon system boot, you have to add it to the according runlevel. The following sections describe, how this is done for various distributions:

2.3.1 Gentoo

 # rc-update add keyrona

2.3.2 Fedora

 # chkconfig --add keyrona

Note: Fedora is by default started with the SE-Linux *enforcing*-mode enabled. The current version of Keyrona does not provide a necessary ruleset specifying the required permissions. Please switch to *permissive*-mode instead in */etc/selinux/config*.

2.3.3 Debian

 # update-rc.d keyrona defaults

3. Configuration

After the installation of Keyrona was successful, the following files have been installed:

Program libraries:
/usr/include/keyrona_cryptlib.h
/usr/lib/libkeyronacl.so

Program binaries:
/usr/bin/keyrona_mount
/usr/bin/keyrona_manager
*/usr/sbin/keyrona_keyprovider*d

Configuration folder and storage for Keyronas options
/etc/keyrona/
/etc/keyrona/database/
/etc/keyrona/keys/
/etc/keyrona/scripts/

Keyrona configuration file
/etc/keyrona/keyrona.cfg

Socket folder
/var/run/keyrona/

Logfile
/var/log/keyrona.log

3.1 Configuration File

You can find the main configuration file of keyrona here:

/etc/keyrona/keyrona.cfg

Its main purpose is the definition of the path to the programs. If you wish to change the storage directories for keys or Keyronas databases, you can edit them here. Please note that changes to the default values can prevent the program from running properly. So make sure, that all folders exists and have the correct permission set (user: root, group: keyrona).

Keyrona can run in two different modes:

KEYRONA_MODE = ”ENTERPRISE”
KEYRONA_MODE = ”PRIVATE”

The private mode will only prompt for username and password during creating of a user. The Enterprise mode will ask for complete X.509 certificate entries.

3.2 Scripts

Keyrona needs several scripts in order to communicate with the different encryption schemes. The scripts are located under:

/etc/keyrona/scripts/

There are three scripts for each cryptosystem supported by Keyrona. The performed operations are:
-create
-mount
-unmount

In the current version the create and the mount scripts select the default encryption for each encryption system. E.g. for truecrypt it is just AES with RIPEMD-160, etc. You may change this in order to fit your demands. Therefore you have to change the parameter of the create and the mount script. For the exact parameter please refer to the manpage/helppage of the cryptosystem. However this action may prevent the use of other keyrona devices that were not created with your system.

4. The Keyrona Manager

After the installation and the configuration, the Keyrona system has to be initialised. This can be done via the command line tool "*keyrona_manager*".
If you start the manager without any parameters, it will show you a help-screen with a detailled description of the available functions. Additionally, you can get verbose output, if you append

 # keyrona_manager --verbose

4.1 Administrators

The first step of the initialisation is to create an administrator. This can be done by invoking:

 # keyrona_manager --init createAdmin
 # keyrona_manager -i ca

With the first administrator, Keyrona automatically creates a new group called "Admingroup". Every admin is part of this group. Every Volumekey will be encrypted with the public key of this group.

To list all existing administrators type:

 # keyrona_manager --init listAdmins
 # keyrona_manager -i la

To delete an administrator use the following command line:

 # keyrona_manager --init deleteAdmin
 # keyrona_manager -i da

Note, that after the first creation at least one administrator must exist to prevent the loss of the Admingroup key. To remove the last administrator as well you have to delete the database in /etc/keyrona/database.

In case you want to change the password of an admin, please use:

 # keyrona_manager --user changeUserCrendential
 # keyrona_manager -u cuc

4.2 Users

To add a new user invoke:

 # keyrona_manager --user createUser
 # keyrona_manager -u cu

In case a user already exists you can import them as well by typing:

 # keyrona_manager --user importUser
 # keyrona_manager -u iu

To list all existing users, type the following command line:

 # keyrona_manager --user listUsers
 # keyrona_manager -u lu

To delete a user you have to invoke:

 # keyrona_manager --user deleteUser
 # keyrona_manager -u du

Note, that even with all users deleted the administrators can still access a volume.
If you delete a user, the user will be removed from Keyronas database. But this can still be recovered, since all created keys are renamed. If you want to remove a user completely from the system, make sure to manually remove the backup keys from */etc/keyrona/keys*

To change a user's password, type:

 # keyrona_manager --user changeUserCrendential
 # keyrona_manager -u cuc

4.3 Token

To add a new token invoke:

 # keyrona_manager --token createToken
 # keyrona_manager -t ct

Here you can select, whether the token shall be only usable with a password or whether it is enough, to “have” the token in place. Additionally, please make sure to mount your e.g., USB-token to a specific place in the Linux filesystem (e.g., */mnt/token*) before using it, since Keyrona does not automatically take care of this.

To list all existing token, type the following command line:

 # keyrona_manager --token listToken
 # keyrona_manager -t lt

To delete a token you have to invoke:

 # keyrona_manager --token deleteToken
 # keyrona_manager -t dt

4.4 Groups

Users and token can be grouped. Therefore, it is possible to add a group to a volume, such that one member of the group is able to access a volume.

To create a new group, type:

 # keyrona_manager --group createGroup
 # keyrona_manager -g cg

To delete a group, invoke:

 # keyrona_manager --group deleteGroup
 # keyrona_manager -g dg

To list groups, use the following command line:

 # keyrona_manager --group listGroups
 # keyrona_manager -g lg

For each group, 3 commands exist to manage the subject's memberships.
To add a subject (i.e., a user or a token) to a group, type:

 # keyrona_manager --group addSubjectToGroup
 # keyrona_manager -g astg

To list all the subjects in a group invoke:

 # keyrona_manager --group listSubjectsInGroup
 # keyrona_manager -g lsig

Finally to delete a subject from a group, use:

 # keyrona_manager --group deleteSubjectFromGroup
 # keyrona_manager -g dsfg

4.5 Platforms

The platform option can be used to bind a specific Volume to this platform. In order to do so, you need TPM support enabled in Linux. Make sure, that you have your TPM device driver loaded and that */dev/tpm0* exists. Additionally, make sure that the TrouSerS *tcsd* daemon is running.

To create a new platform use this commandline:

 # keyrona_manager --platform createPlatform
 # keyrona_manager -p cp

This will seal the Volume data to this platform. Since Keyrona does not know your SRK password, make sure to set it to the WELL_KNOWN_SECRET during TakeOwnership of your TPM.

Note: Depending on your TPM, the key generation might take some time, so in case you receive a timeout, just try again.

To delete an existing platform type

 # keyrona_manager --platfom deletePlatform
 # keyrona_manager -p dp

4.6 Volumes

To import a volume, enter:

 # keyrona_manager --volume importVolume
 # keyrona_manager -v iv

To create a new volume:

 # keyrona_manager --volume createVolume
 # keyrona_manager -v cv

NOTE THAT ALL DATA ON AN NEWLY CREATED VOLUME WILL BE LOST!!!
BE SURE YOU KNOW WHAT YOU ARE DOING!!!

The administrators will automatically have access to every volume added to Keyrona. Additionally, Keyrona stores the device name (e.g., /dev/sdf1), where the volume is currently attached to.
In case of portable devices (e.g., USB-sticks), this location may change (depending on the order of connected devices). Therefore, one can re-attach the volume to a different device via:

 # keyrona_manager --volume attachVolume
 # keyrona_manager -v av

To list all available Volumes, type:

 # keyrona_manager --volume listVolumes
 # keyrona_manager -v lv

To delete a volume, invoke:

 # keyrona_manager --volume deleteVolume
 # keyrona_manager -v dv

Note: Volumes will only be removed from the database, the physical Volume still exists. To delete it permanently you must format the physical volume.

4.6.1 Add Subject To Volume

To add a subject (i.e., a user or a token) directly to a Volume, type:

 # keyrona_manager --volume addSubjectToVolume
 # keyrona_manager -v astv

4.6.2 Add Group To Volume

To add a group to a Volume, type:

 # keyrona_manager --volume addGroupToVolume
 # keyrona_manager -v agtv

4.6.3 Delete Subject From Volume

To delete a user from a Volume, type

 # keyrona_manager --volume deleteSubjectFromVolume
 # keyrona_manager -v dsfv

4.6.4 Delete Group From Volume

To delete a group from a Volume, type:

 # keyrona_manager --volume deleteGroupFromVolume
 # keyrona_manager -v dgfv

4.6.5 List Subject In Volume

To list all the users , that have access to a certain Volume, invoke:

 # keyrona_manager --volume listSubjectInVolume
 # keyrona_manager -v lsiv

4.6.6 List Groups In Volume

To list all the groups, that have access to a certain Volume, invoke:

 # keyrona_manager --volume listGroupInVolume
 # keyrona_manager -v lgiv

4.7 Secret Sharing Schemes

Keyrona provides methods of generating a Secret-Sharing-Scheme (SSS), which makes it possible to grant access to a Volume, if and only if *n* out of *m* subjects (i.e. users or platforms) are available. The regular case is probably the use of a platform in combination with a user.

To create a new SSS for a Volume you have to invoke:

 # keyrona_manager --volume addSSSToVolume
 # keyrona_manager -v assstv

Note, that with an existing platform, you will automatically prompted, whether this platform shall be added to the SSS.

To delete a SSS from a Volume, you have to use:

 # keyrona_manager --volume deleteSSSFromVolume
 # keyrona_manager -v dsssfv

To List all the SSS of a Volume, type:

 # keyrona_manager --volume listSSSInVolume
 # keyrona_manager -v lsssiv

5. Working with keyrona

5.1 Starting Keyrona's Keyprovider

After managing your Keyrona configuration, you have to start the *keyrona_keyprovider*d.
This daemon performs the desired encryption and mounting of volumes. Additionally, it keeps track about the mounted devices.

You can start the daemon via script with (if not implicitly done within a runlevel):

 # sudo /etc/init.d/keyrona start

and stop it with:

 # sudo /etc/init.d/keyrona stop

Since the daemon keeps track about the mounted devices in a database, it is important to clear this database upon system startup. This is automatically done within the *init*-script. In case you want to manually prune the *keyrona_keyprovider*-database, append the "*--prune*" option, which will delete any existing entries in the database automatically.

 # sudo keyrona_keyprovider --prune

If you don't want it to go into background, append the *"--foreground"* option

5.2 Mounting

If the process started correctly you can mount a volume with:

 # keyrona_mount -m -u username -v volume -p /mount/destination

Note: You have 15 seconds to enter the password, otherwise the program will timeout and you have to enter the commandline again.

5.2.1 Mounting via Subject

Example:
 # keyrona_mount -m -u user1 -v myUSBStick -p /mnt/keyrona

or, if you use a token instead, make sure to write the username as, e.g., "*Token_myToken*":

 # keyrona_mount -m -u Token_myToken -v myUSBStick -p /mnt/keyrona

5.2.2 Mounting via Group

Example:
 # keyrona_mount -m -u Group_myGroup1 -v myUSBStick -p /mnt/keyrona

5.2.3 Mounting via Secret-Sharing-Scheme

In you decided to use a SSS, you can specify multiple user names, each one seperated with comma. For the use of a SSS that uses also the platform (called “this”), the command line should look like:

 # keyrona_mount -m -u username,Platform_this -v volume -p /mount/destination

Note: If multiple users try to login, each user has 15 seconds to enter his password.

5.3 Unmounting

To unmount the Volume, please use the exact command line, you used for mounting, with the only difference of exchanging the “*--mount*” parameter with “*--unmount*” (or short “*-m*” with “*-um*”)

 # keyrona_mount -um -u username,platform_this -p /mount/destination

Keyrona and /etc/fstab:

It is possible to automatically use Keyrona from within */etc/fstab*.
As a precondition, you need to make sure, that the script */sbin/mount.keyrona* is correctly installed and is executable (*# chmod 755 /sbin/mount.keyrona)*.

Now you can create entries in the */etc/fstab* to automatically mount Keyrona volumes via

 # mount volumename

The entries should e.g., look like this:

Volume Name	**Mount Path**	**keyrona**	**usernames**	**0**	**0**
volume1	*/mnt/vol1*	*keyrona*	*user1*	*0*	*0*
volume2	*/mnt/vol2*	*keyrona*	*user2*	*0*	*0*
volume3	*/mnt/vol3*	*keyrona*	*user1,user2*	*0*	*0*

Volume1/2 will then automatically prompt for the credential of user1/2. Volume3 is a secret-sharing scheme, that requires both user1 and user2 to be present.

In case you have a shared volume with different allowed users and you don't know which user will be sitting in front of the PC upon boot time, you can replace the username with the keyword '**KEYRONA_PROMPT**'. This will tell Keyrona to ask for the current username(s).

Volume Name	**Mount Path**	**keyrona**	**usernames**	**0**	**0**
volume1	*/mnt/vol1*	*keyrona*	*KEYRONA_PROMPT*	*0*	*0*

Note: The mount script currently doesn't update */etc/mtab* accordingly. In order to unmount a Keyrona volume, you still have to unmount it via the standard Keyrona unmount command.

6. Example

This part will give a short example, from the installation of Keyrona, to the actual mount process.

→ First thing to do is installation. Note, that compilation can take some time

 # sudo ./install.sh
[...]
=> Stripping binaries
=> Adding Keyrona group 'keyrona'
=> Please add the local Linux users to the Keyrona group
=> Assigning permissions
=> Done... ;)

→ Now add the local user to the Keyrona Group.

 # sudo useradd -G keyrona christoph

→ Create an administrator.

 # sudo keyrona_manager -i aa
[...]
Superuser successfully logged in!
Please enter username for new admin: deradmin
*Please enter passphrase for new key: *********
*Please verify password: *********
Successfully created new admin with SubjectID '1753451921'.
Checking existence of admin-group 'Admin': not present, creating new admin group...
Creating new key with label 'Admin'
Encrypting group password for admin: deradmin (1753451921)

→ Create a new user:

 # keyrona_manager -u cu
[...]
The function you requested requires admin authentication.
Only one administrator available, selecting admin 'deradmin'
*Please enter the according password for admin 'deradmin' (1753451921): *********
Admin successfully logged in!
Please enter username for new user: user1
*Please enter passphrase for new key: *****
*Please verify password: *****
Successfully created new user with SubjectID '1346747137'.
SubjectID: '1346747137'
SubjectName: 'user1'
SubjectType: 'User'
SubjectKeyfile: '/etc/keyrona/keys/[..].p15'
Subject is admin: 'false'
Subject's last login: Subject has not logged in, yet!
Subject's key information:
Key length: 2048

Key Algorithm: RSA

→ Create a Volume:

 # keyrona_manager -v cv
[...]

Number: Encryption Scheme:
1) DMCRYPT
2) ECRYPTFS
3) ENCFS
4) LUKS
5) TRUECRYPT

Please enter the desired encryption scheme: 4

Selected encryption scheme 'LUKS'

Number: Device:
1) /dev/sda
2) /dev/sda1
3) /dev/sda2
4) /dev/sda3
5) /dev/sdb
6) /dev/sdb1
7) /dev/sdc
8) /dev/sdd
9) /dev/sdd1
10) /dev/sdd2
11) /dev/sde
12) /dev/sdf

[Note: If the desired device has not been identified automatically, you can still enter the correct device (e.g., /dev/sdb1) below.]
Please select the device to be created: 6

Selected device '/dev/sdb1'

Please enter a unique identifier for device '/dev/sdb1': USB_STICK_1
WARNING: All data will be permanently lost on device '/dev/sdb1'
Number: Filesystem:
1) ext2
2) ext3
3) reiserfs
4) xfs

Please enter the desired filesystem for volume 'USB_STICK_1': 3

Selected filesystem scheme 'reiserfs'

Are you sure? (type uppercase 'YES'): YES
Command successful.
key slot 0 unlocked.

Command successful.
mkfs.reiserfs 3.6.21 (2009 www.namesys.com)

ATTENTION: YOU SHOULD REBOOT AFTER FDISK!
ALL DATA WILL BE LOST ON '/dev/mapper/keyrona_create'!
Continue (y/n):y
Initializing journal - 0%....20%....40%....60%....80%....100%
Syncing..ok
ReiserFS is successfully created on /dev/mapper/keyrona_create.

Successfully created volume '/dev/sdb1'.
Importing volume '/dev/sdb1' into Keyrona.
Successfully imported volume 'USB_STICK_1'.
Volume UUID: 'USB_STICK_1'
Currently attached to: '/dev/sdb1'
Encryption system: 'LUKS'

→ The last thing to do before mounting is adding the created user to the volume:

 # sudo keyrona_manager -v astv
[...]
We have the following volumes:
Volume UUID: 'USB_STICK_1'
Currently attached to: '/dev/sdb1'
Encryption system: 'LUKS'

Number: UUID:
1) USB_STICK_1

Please select the desired volume: 1

We have the following users (admins not included):
SubjectID: '68464446'
SubjectName: 'user1'
SubjectType: 'User'
SubjectKeyfile: '/etc/keyrona/keys/KeyronaUser_68464446_test1.p15'
Subject is admin: 'false'
Subject's last login: Subject has not logged in, yet!
Subject's key information:
Key length: 2048
Key Algorithm: RSA

Number: User:
1) user1

Please select the subject to be added: 1

We have the following admins:
SubjectID: '1416644769'
SubjectName: 'admin'
SubjectType: 'User'

SubjectKeyfile: '/etc/keyrona/keys/KeyronaUser_1416644769_admin.p15'
Subject is admin: 'true'
Subject's last login: 23.04.2009, 12:18:20
Subject's key information:
Key length: 2048
Key Algorithm: RSA
Current system messages for user 'admin':
23.04.2009, 12:16:10 Group 'Admin' has been added to volume 'USB_STICK_1'
23.04.2009, 10:29:25 Group 'Admin' has been added to volume 'test'

Only one administrator available, selecting admin 'admin'
*Please enter the according password for admin 'admin' (1416644769): ******
Successfully added subject 'user1' to volume 'USB_STICK_1'.

→ Start the *keyrona_keyprovider*d

*# sudo keyrona_keyprovider*d

→ And finally, mount the volume:

 # keyrona_mount -m -u user1 -v USB_STICK_1 -p /mnt/keyrona/
Running in private mode!
*Please enter passphrase for user 'user1': ******
========== KEYRONA SYSTEM MESSAGE ==========
To 'user1': you have the following new messages:
==========
09.04.2009, 15:55:54
You have been added to volume 'USB_STICK_1'
-> success!

Now the volume is mounted under the stated path (in this case /mnt/keyrona)

7. Errors

In case you identify an error or other strange behaviour please reference the log in */var/log/keyrona.log*

Please file bug reports in the TRAC-ticketing system available under:

http://www.keyrona.org

or write an email to

keyrona@sirrix.com