LEOSAC SAS



LEOSAC KEY MANAGER

Table of Contents

[User Guide 1](#__RefHeading___Toc1154_2924010706)

[LEOSAC KEY MANAGER 2](#__RefHeading___Toc1156_2924010706)

[Prerequisites 2](#__RefHeading___Toc1158_2924010706)

[Software 2](#__RefHeading___Toc1160_2924010706)

[Hardware 2](#__RefHeading___Toc1162_2924010706)

[Home 3](#__RefHeading___Toc1164_2924010706)

[Dark Mode 4](#__RefHeading___Toc1166_2924010706)

[Log Console 4](#__RefHeading___Toc1168_2924010706)

[About 5](#__RefHeading___Toc1170_2924010706)

[Maintenance plan 5](#__RefHeading___Toc1172_2924010706)

[Key Store 6](#__RefHeading___Toc1174_2924010706)

[Key List 7](#__RefHeading___Toc1176_2924010706)

[Search a Key 7](#__RefHeading___Toc1178_2924010706)

[Create a new Key 7](#__RefHeading___Toc1180_2924010706)

[Manual definition 7](#__RefHeading___Toc1182_2924010706)

[Cryptogram Import 9](#__RefHeading___Toc1184_2924010706)

[Symmetric Key 9](#__RefHeading___Toc1186_2924010706)

[Key Version 9](#__RefHeading___Toc1188_2924010706)

[Checksum value 9](#__RefHeading___Toc1190_2924010706)

[Key Generation 10](#__RefHeading___Toc1192_2924010706)

[Key Link 13](#__RefHeading___Toc1194_2924010706)

[Print 13](#__RefHeading___Toc1196_2924010706)

[Asymmetric Key 14](#__RefHeading___Toc1198_2924010706)

[Key Store Favorite 14](#__RefHeading___Toc1200_2924010706)

[Favorite Edition 15](#__RefHeading___Toc1202_2924010706)

[Close Key Store 15](#__RefHeading___Toc1204_2924010706)

[Key Store Type 15](#__RefHeading___Toc1206_2924010706)

[File 15](#__RefHeading___Toc1208_2924010706)

[NXP SAM AV2 / AV3 16](#__RefHeading___Toc1210_2924010706)

[HSM PKCS#11 19](#__RefHeading___Toc1212_2924010706)

[Favorites 20](#__RefHeading___Toc1214_2924010706)

# LEOSAC KEY MANAGER

Leosac Key Manager software allows you to manage key stores.

It provides a generic user interface for key generation, creation, edition and cryptography operations for symmetric and asymmetric key types.

## Prerequisites

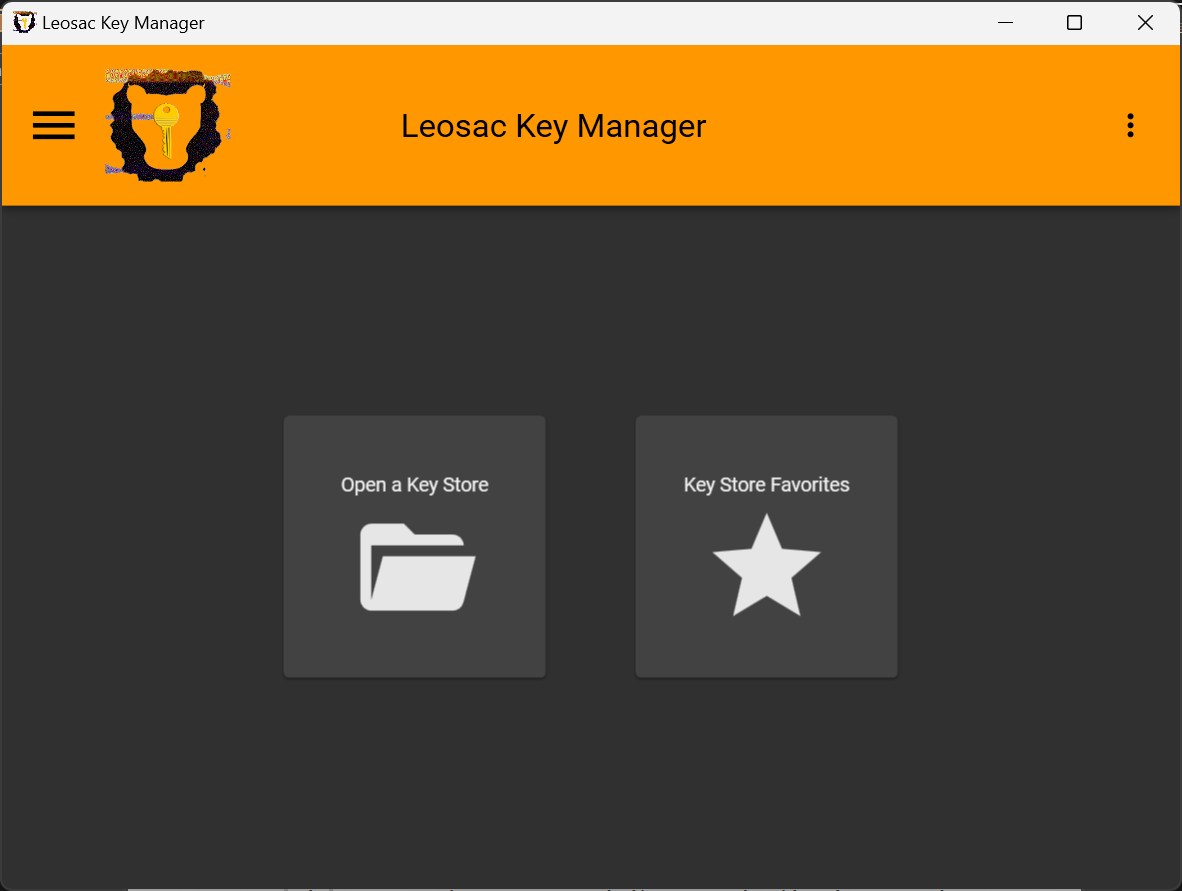
### Software

* **.NET Desktop 8.0[[1]](#footnote-2)**
* **VC++** **Redistributable[[2]](#footnote-3)**

### Hardware

|  |  |
| --- | --- |
| Processor | 2 GHz |
| RAM | 4096 MB |
| Minimal Disk Space | 4.5 GB |

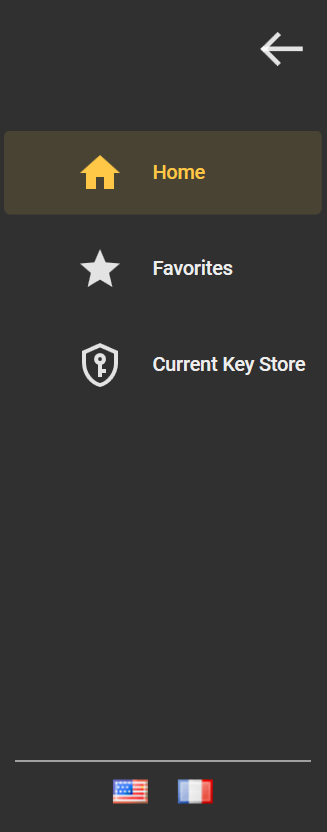
## Home



The main Leosac Key Manager window provides shortcut to main software features.

Two tiles allows you to open an existing key store or to list your favorites key store.

On the left side, a floating menu  guide you permanently to the main software features as well.



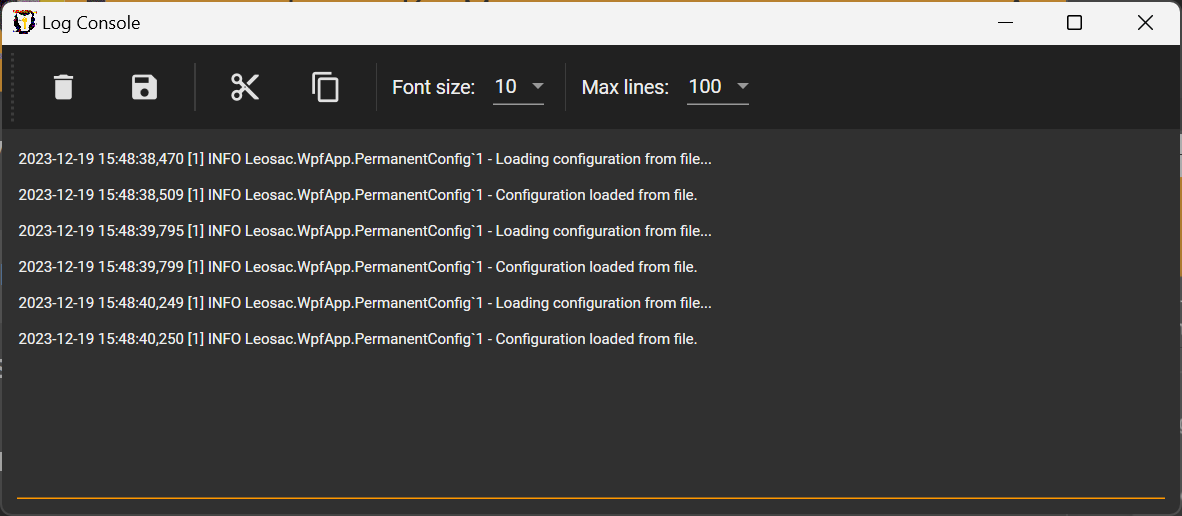
On the right side an additional menu  gives you access to optional and more advanced application parameters.



## Dark Mode

The option  switch the user interface from light to dark mode. This choice is locally saved.

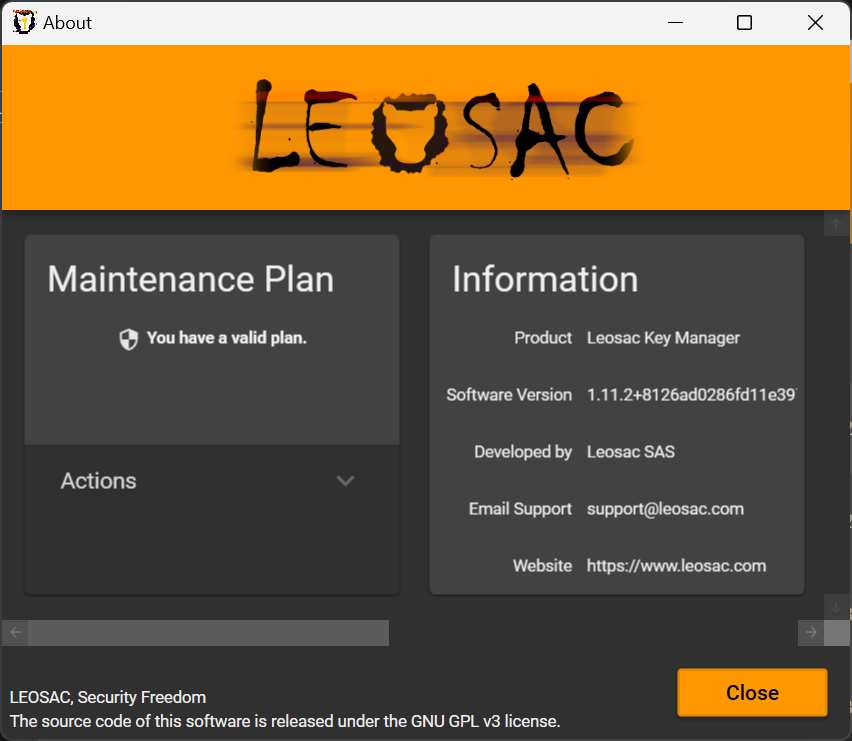
## Log Console



Details of executed operations with associated trace are by default saved into a local log file.

Such logs can be displayed in real time directly inside the application through the “Log Console” window.

## About



La fenêtre « à propos » fournit des informations complémentaires sur le programme, l’éditeur et les bibliothèques dépendantes.

C’est également à partir de cette fenêtre que la vérification des mises à jour peut être effectuée et son automatisation activée/désactivée.

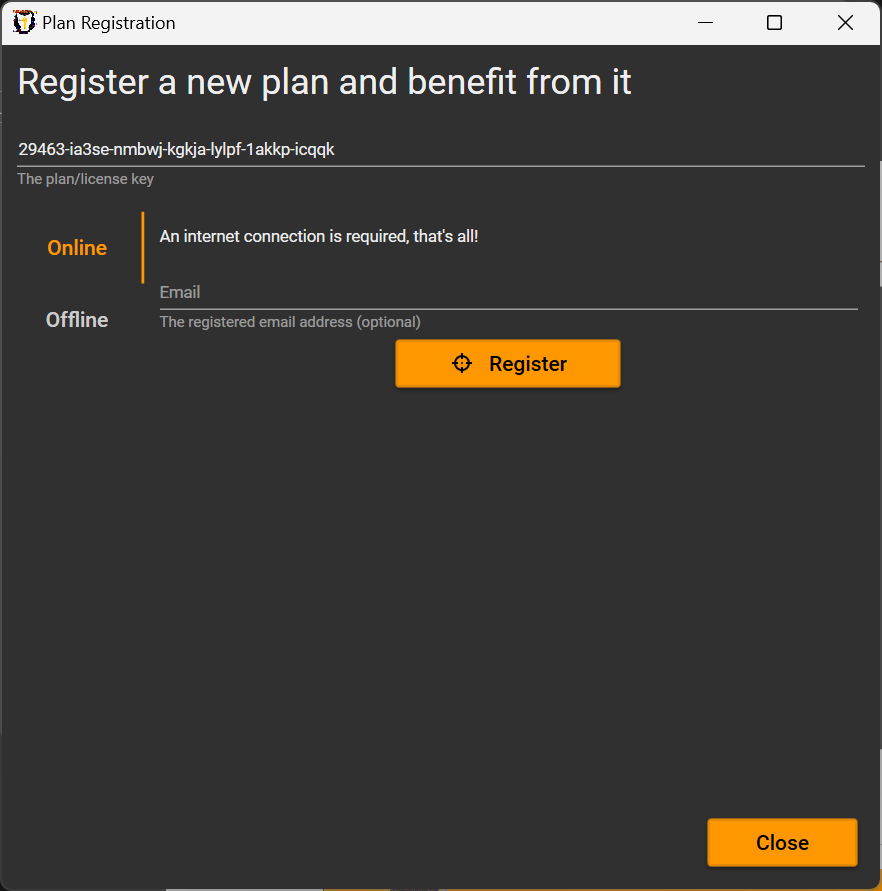
Des détails sur le plan de maintenance actuel sont par ailleurs affichés.

## Maintenance plan

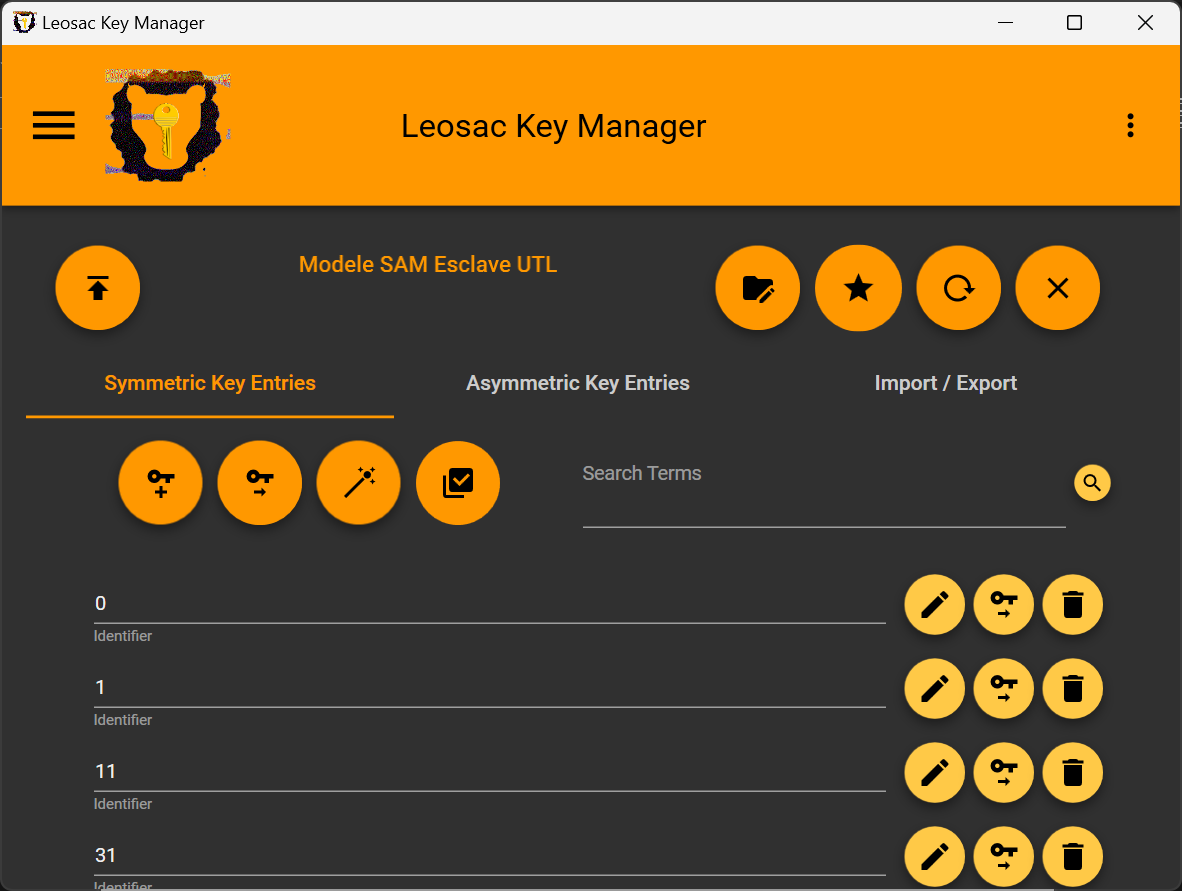
It is highly recommended to subscribe to a maintenance plan to guarantee minimal installation support, to get access to all features and services, and to support the continuous software development effort.

Visit [https://leosac.com/key-manager/](https://leosac.com/fr/key-manager/) for more information about maintenance plan.

Once a plan subscribed, you can directly proceed to its registration from the application.



# Key Store

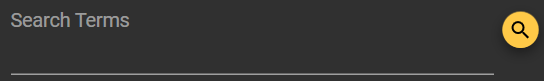


## Key List

On Key Store opening, the list of keys available on the key store is automatically displayed.

Depending the key store capacities, you will be able to edit an existing key entry , import a cryptogram  or remove the key entry .

### Search a Key



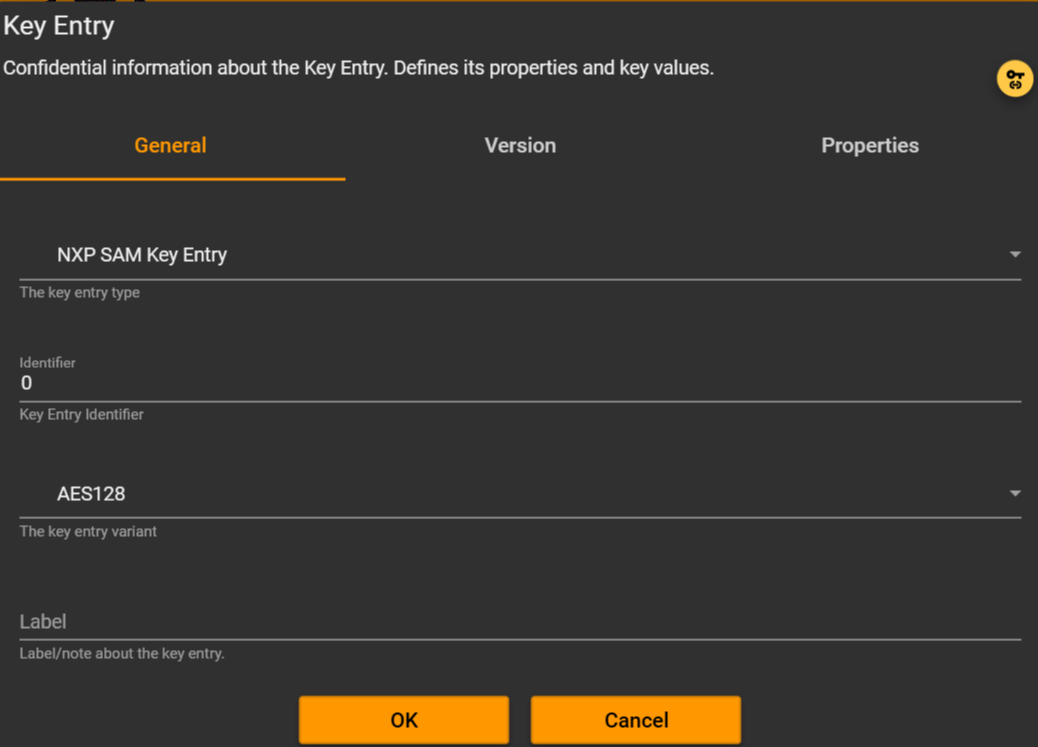
When a key store contains a lot of keys, you may want to search for a specific key using a search filter available on top of the key list.

## Create a new Key

### Manual definition

To create a new key defined locally, use the button .

A window will ask you to define expected parameters for the new key.



#### Key Entry type

The new key entry type to define.

It is recommended to directly select a type matching the targeted technology, incompatibilities or missing properties could arise otherwise.

#### Key Entry identifier

An identifier, usually unique, which can be used to reference the key later one.

Some Key Store may expect specific formatting.

#### Variant

Usually the cryptography algorithm to use.

#### Label

Optional field, not always permanently saved.

#### Versions

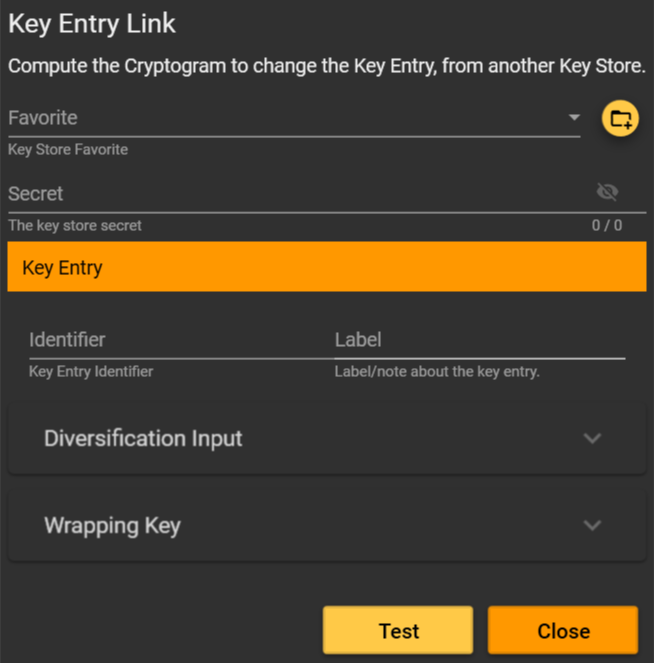
Several key versions with different key values could exists for the same key entry.

#### Properties

Additional parameters specific to the targeted technology. More information on Key Store Types section.

#### Link

You can link a key entry globally to another key entry from another key store using the button .



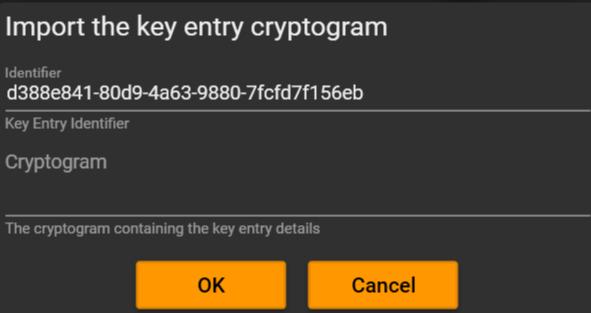
You will then have to define the referenced Key Store to use for the link and the targeted key entry on this key store.

Links at this stage are purely declarative, the execution will be done only at key store publication. Such operations are usually performed from a “File Key Store” acting as a template to populate a “NXP SAM AV2/AV3 Key Store” or “HSM PKCS#11 Key Store”.

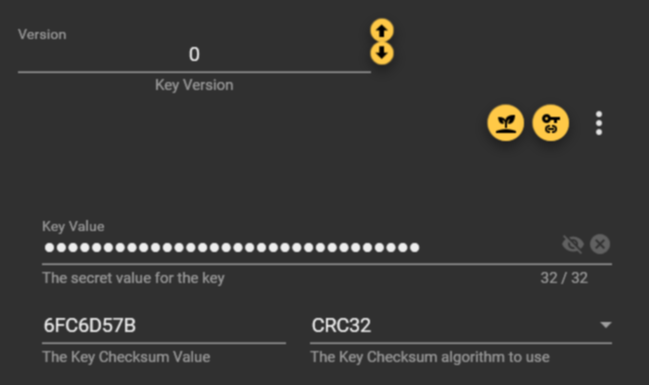
For security reason you may be expected to setup a Wrapping Key. This key will have to exists on both the linked Key Store and the targeted Key Store at publication time.

### Cryptogram Import

If you have an offline cryptogram, you can import its raw data to a key entry using the button .



## Symmetric Key



### Key Version

When the key store support Key Version, you will be able to define the version value.

This information is purely informational but could be used for further key rotation automation.

### Checksum value

Checksum value are useful to ensure the entered key doesn’t contains any typo.

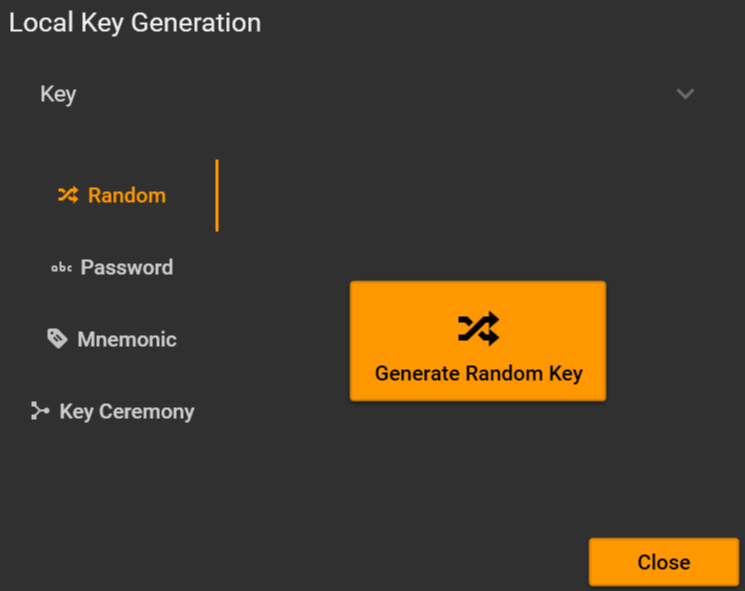
For DES and AES keys, it is recommended to use KCV algorithm. For others, you can use CRC32 or a simple SHA256 hash.

### Key Generation

Open the Key Generation window using the button .

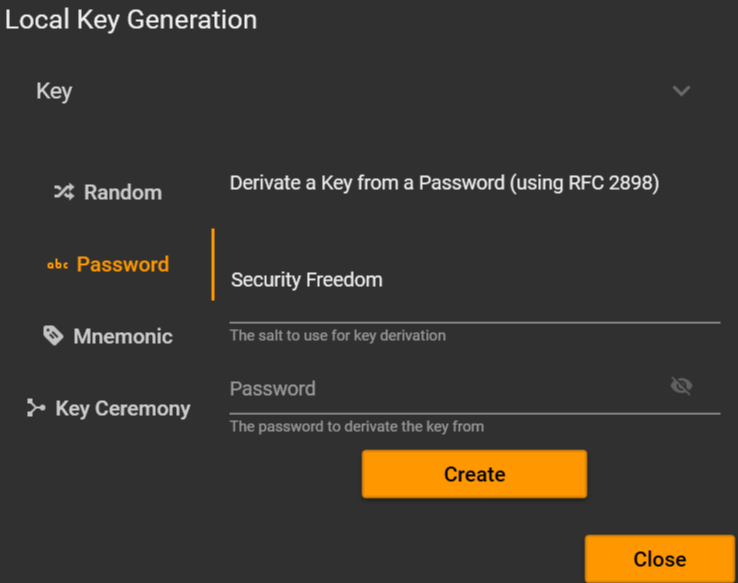
Several methods are available.

#### Random



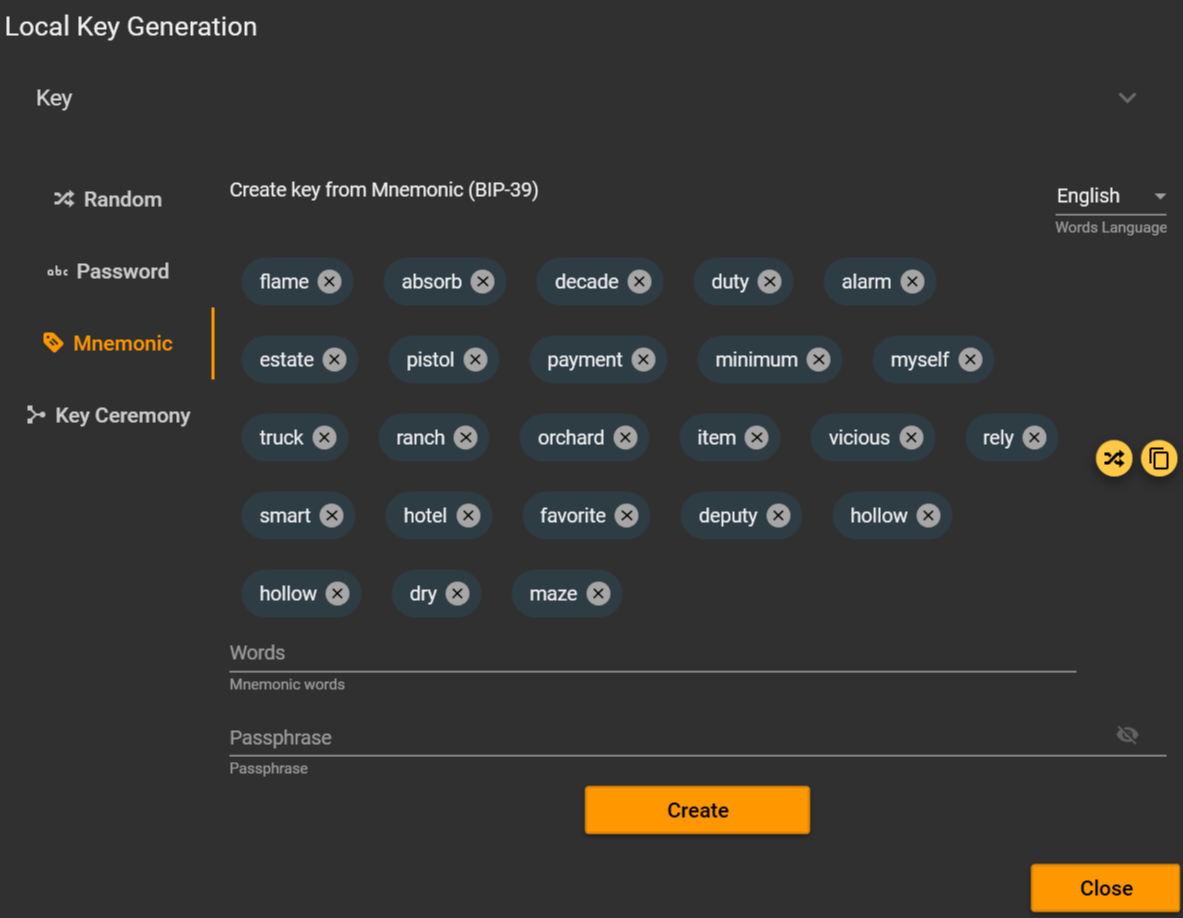
Generate a pseudo-random key using the button « Generate Random Key ».

#### Password



Derivate a key from a password using RFC 2898.

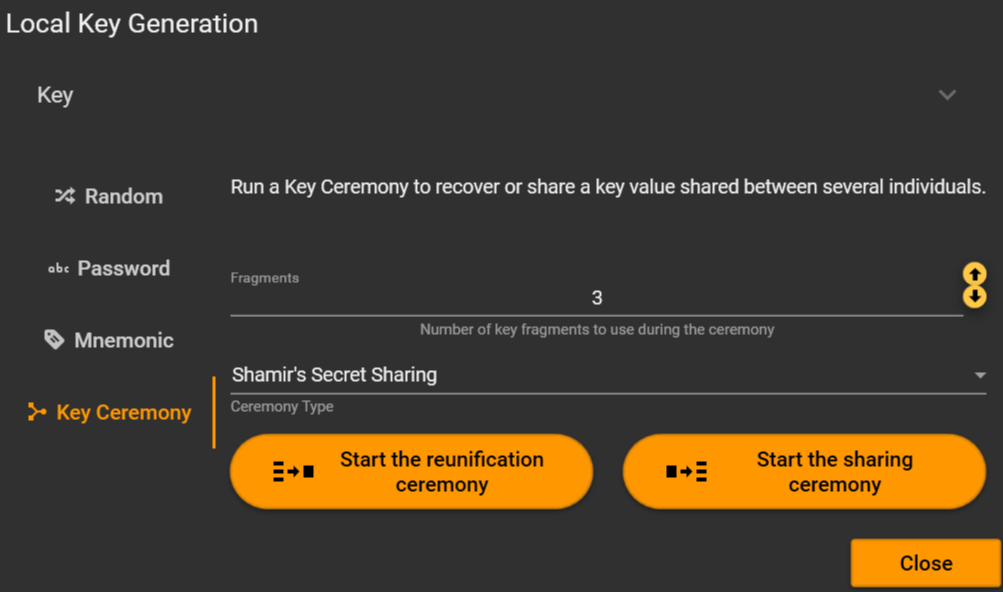
#### Mnemonic



Create a key from a list of Mnemonic words (BIP-39). It is recommended to generate a new word list using the button  then to define a passphrase.

This generation method has more interest for larger key size.

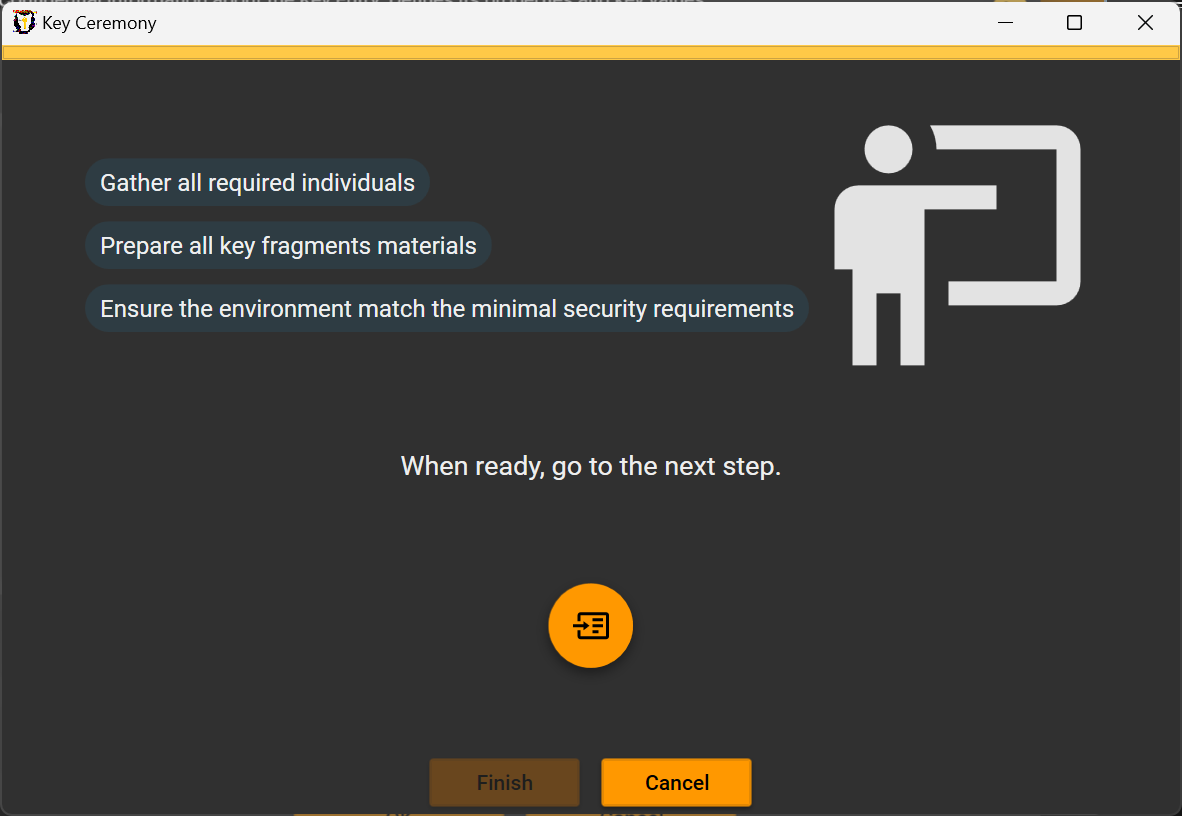
#### Key Ceremony



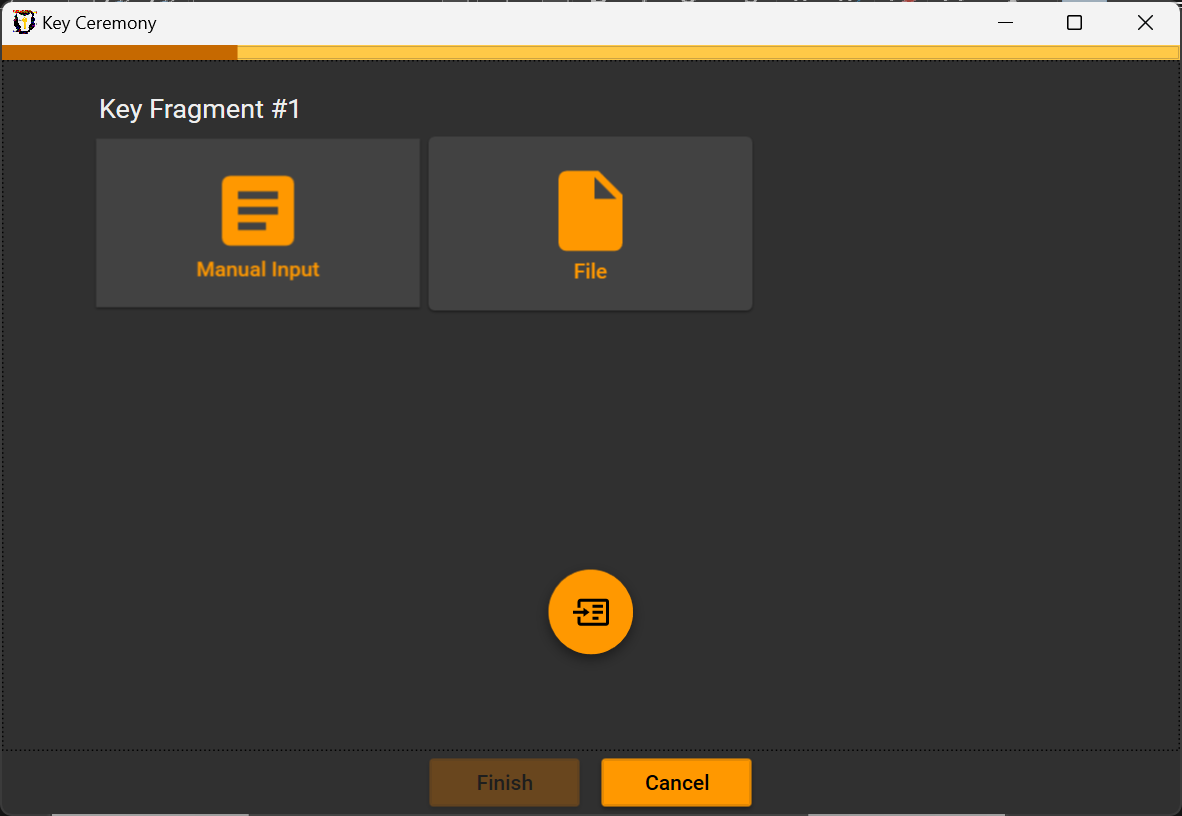
A Key Ceremony means here a reunification or sharing of several fragments into/from a key from/to several individuals.

Define the expecting number of fragments to compute and the ceremony type.

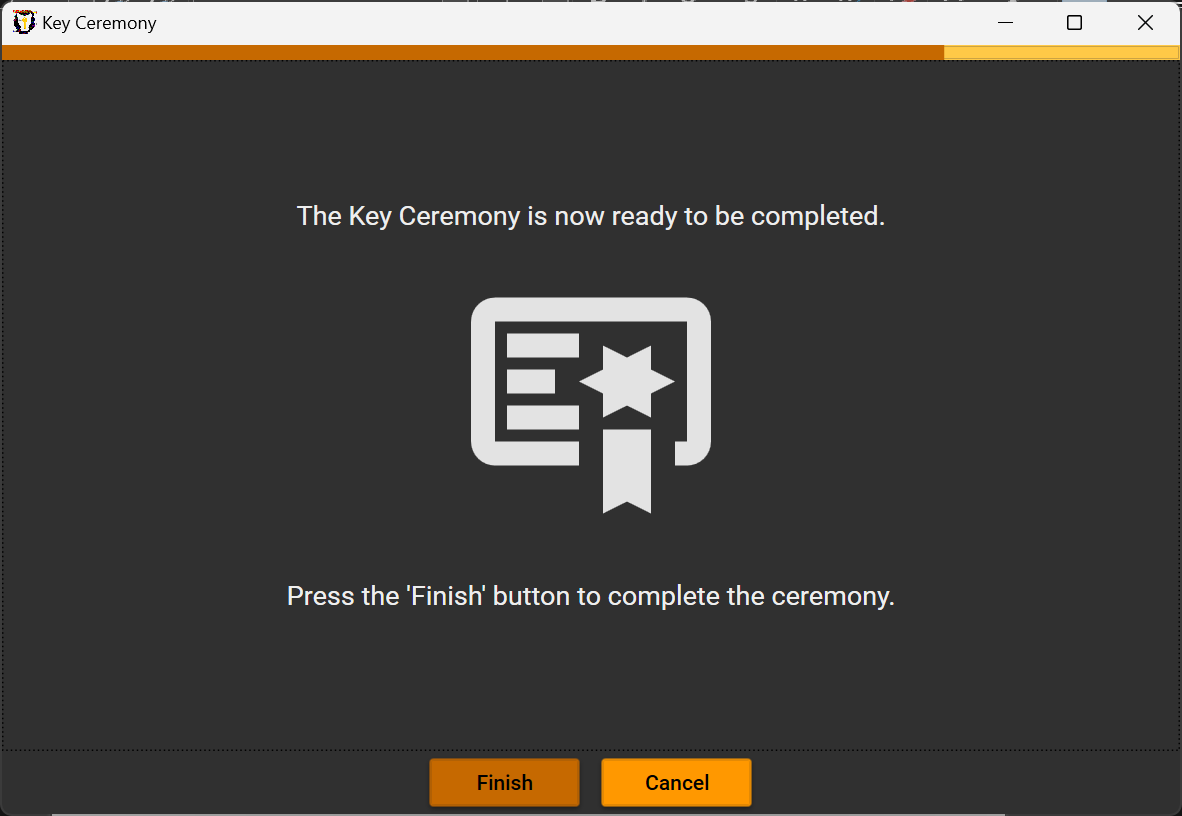
A basic key ceremony would be using XOR or concatenation of fragments. It is recommended to use a stronger key ceremony type providing parity in case of a fragment loose, like Shamir Secret Sharing key ceremony.



Gather individuals and prepare fragment supports, then when ready press button .



Go over the process of selecting expected support for each fragment then click .

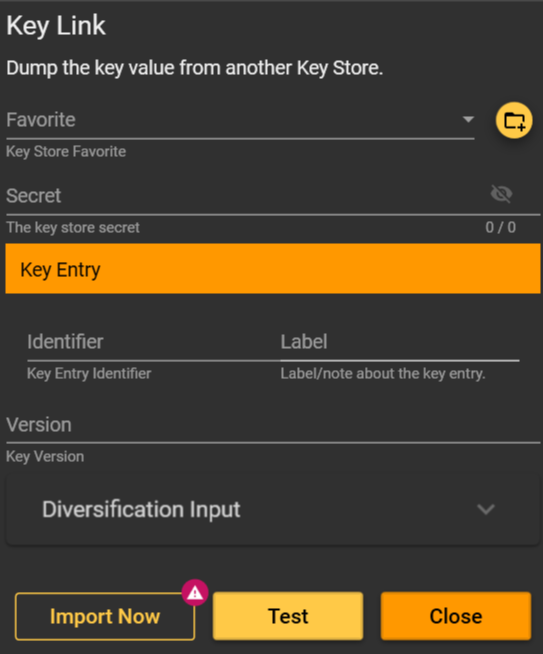


Once completed, press the button « Finish» for confirmation.

### Key Link

You can directly link a Key value from another Key Store using the button .

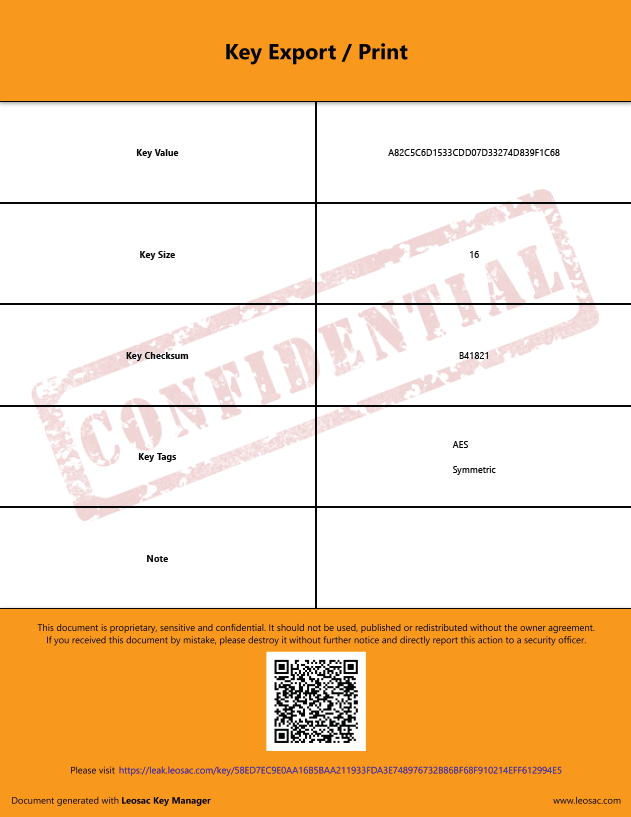
Such link is usally less secure than a link at a Key Entry level but match a specific use case. In practice, the key value is usually temporary extracted in clear from the source Key Store.



### Print

A key value can easily be printed using the buttun .

This operation should be considered sensitive as the key value will be extracted and printer in clear.



## Asymmetric Key

Asymmetric key support is currently partially implemented. The documentation will be updated on a further version.

## Key Store Favorite

Use the button  to save as a favorite the currently opened key store.

If the current key store is already on the favorite list and you want to remove it, use the button , or remove it from the dedicated Key Store Favorite screen.

## Favorite Edition

You can edit Key Store properties using the button .

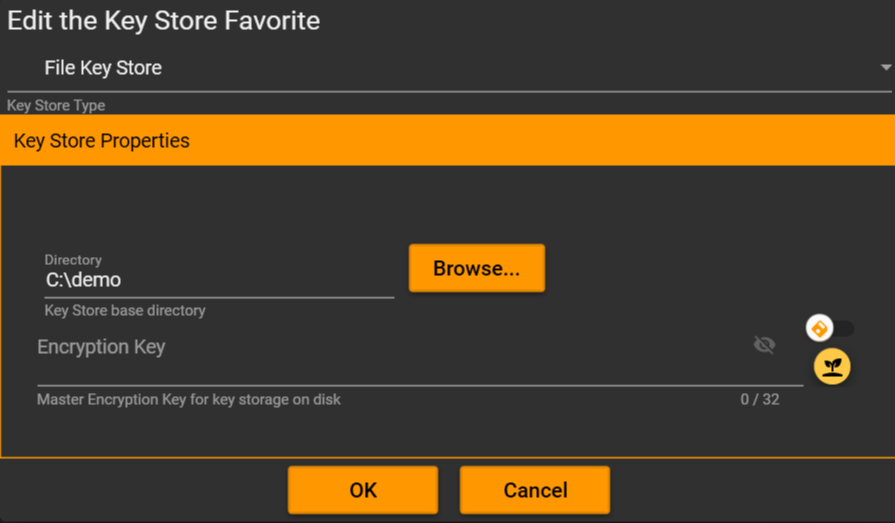
## Close Key Store

It is recommended to close the key store when you don’t need it anymore if you don’t close the application by itself. To close the current key store, use the button .

## Key Store Type

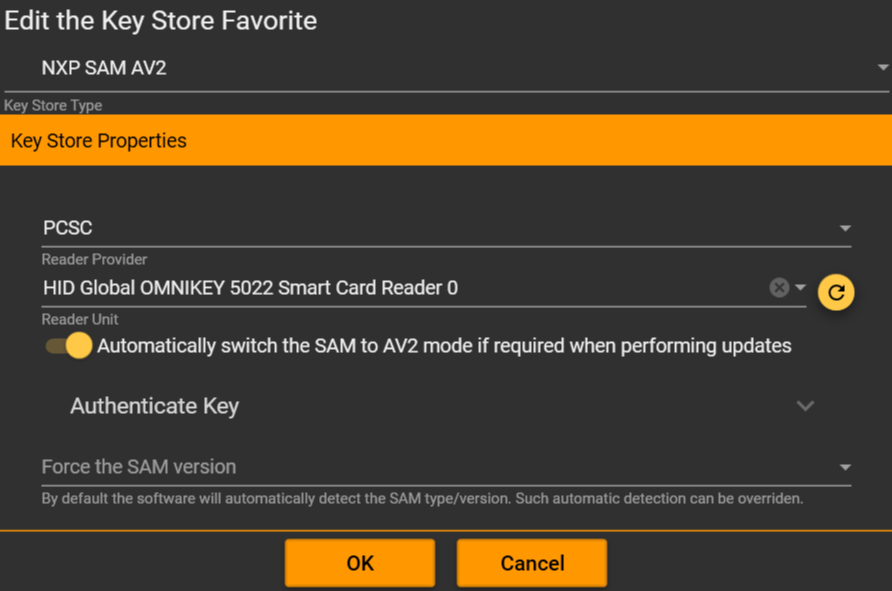
### File

#### Properties



### NXP SAM AV2 / AV3

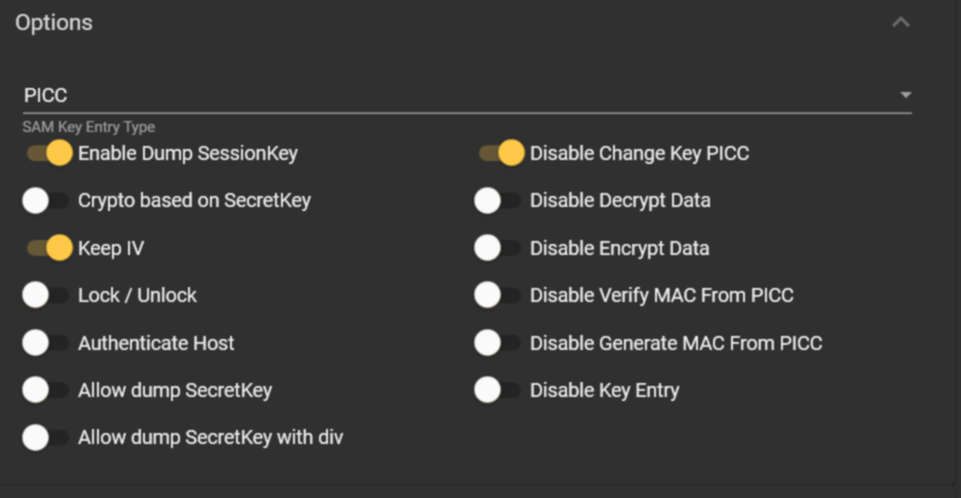
#### Properties



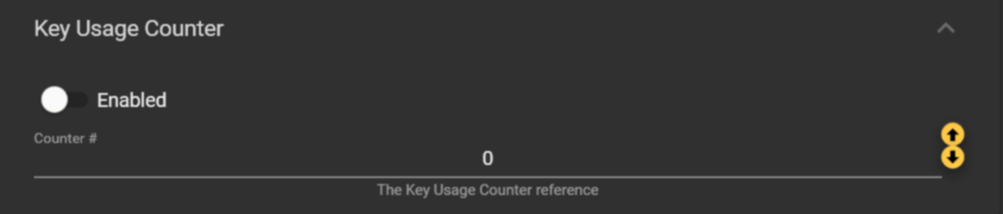
#### Key Types

##### DES / 3DES / 3K3DES / AES128

###### Options



###### Usage Counter



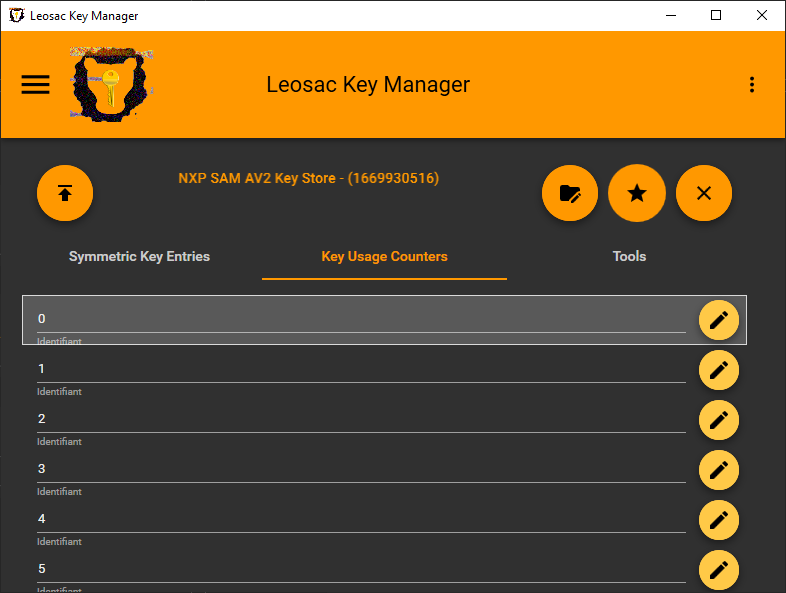
###### DESFire

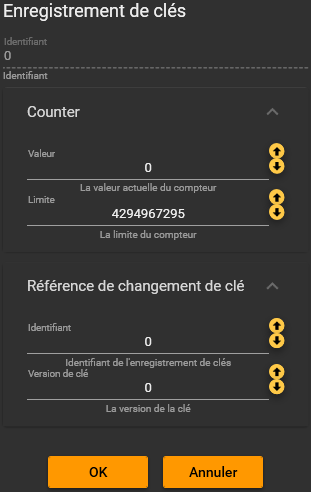


###### Change Key Reference

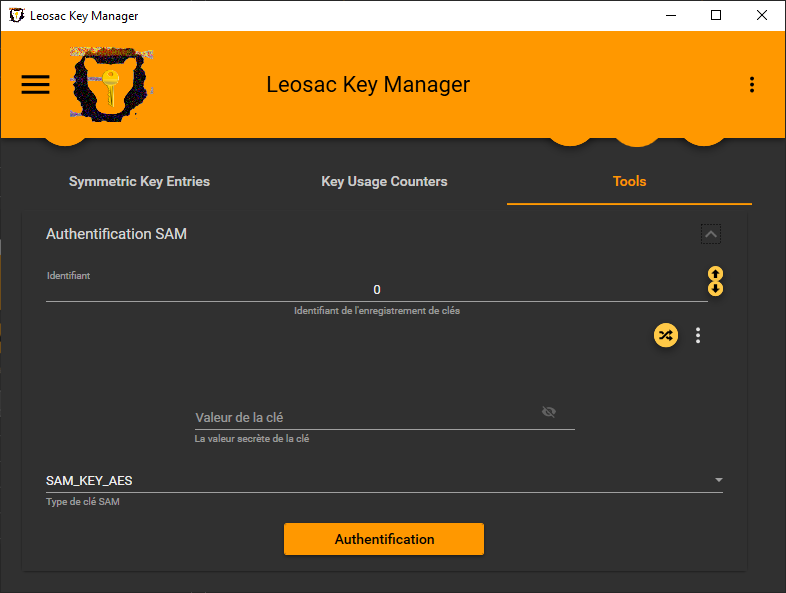


#### Counters



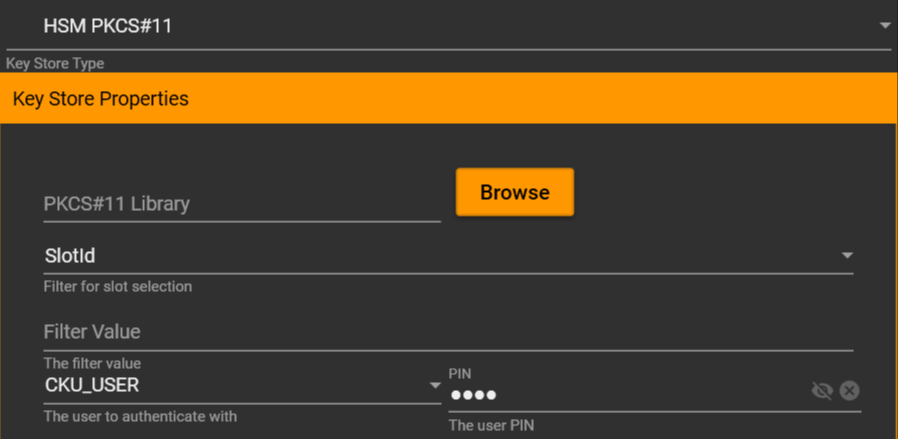


#### Tools

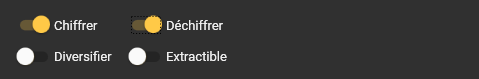


### HSM PKCS#11

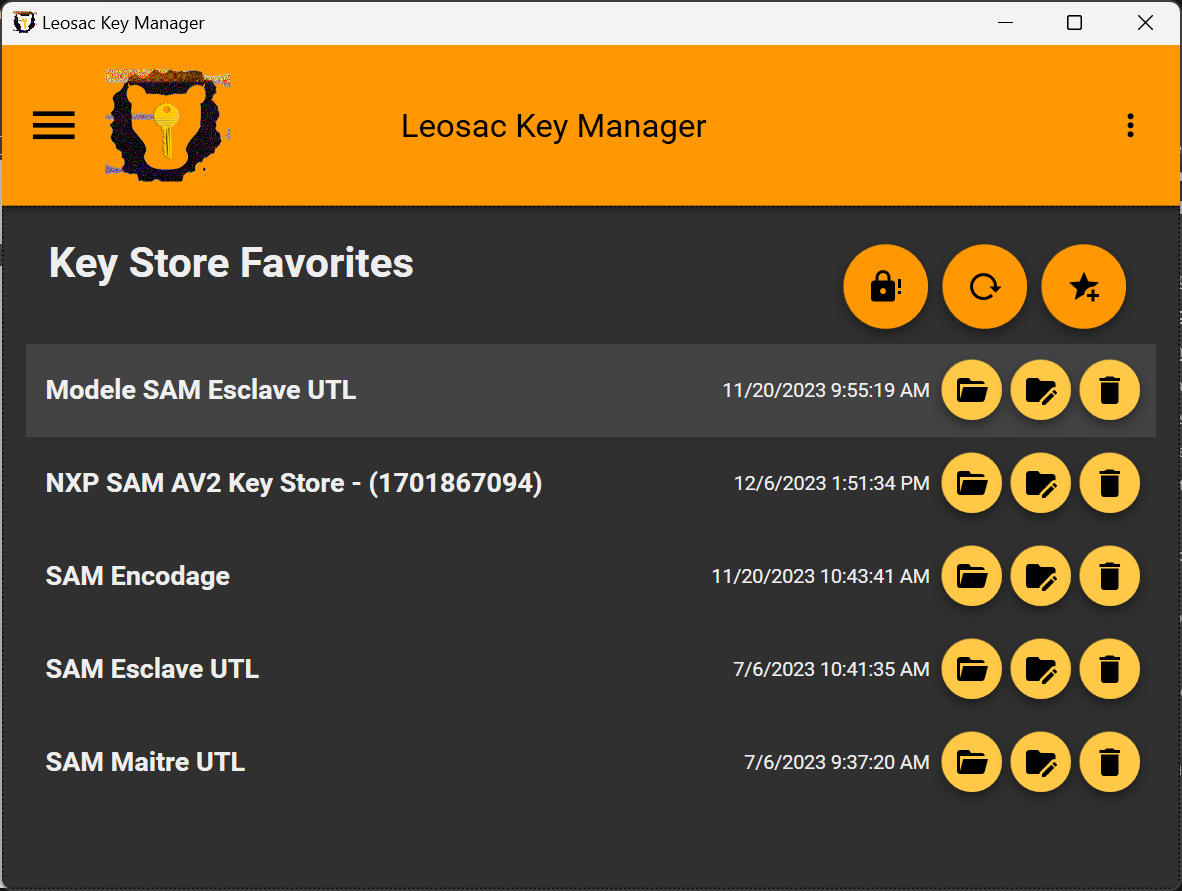
#### Properties



#### Key Types



# Favorites



You should save as favorites Key Stores that you are using frequently.

Using Favorites is also required to perform interactions between key stores (publication, links, …).

1. <https://dotnet.microsoft.com/en-us/download/dotnet/8.0> [↑](#footnote-ref-2)
2. <https://aka.ms/vs/17/release/vc_redist.x64.exe> [↑](#footnote-ref-3)