User Guide to

Lybero Escrow Managing System

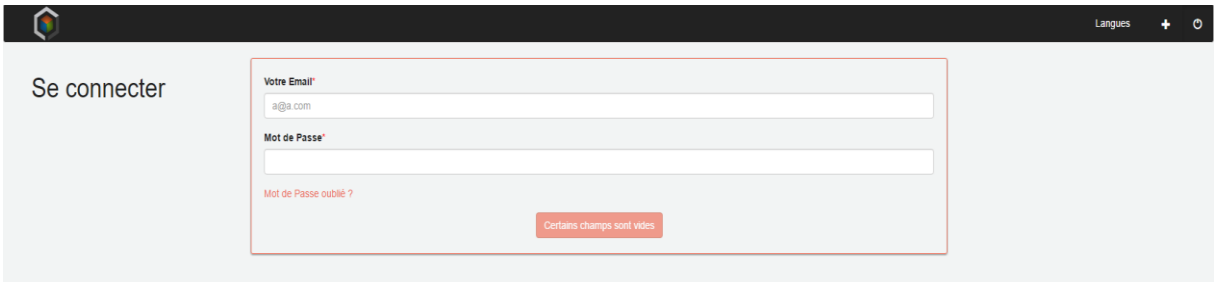
Be sceptical, it is our core business

Applies to : LEMS version 2.1.1

Does this document apply to you?

Go to https://saas.lybero.net/saas/

If your screen looks like this, then you're just where you're supposed to be.



This User Guide contains what you need to get those secrets you're itching to hide and finally stop being scared of data breach.

Using this, you'll learn how to:

- The basics of LEMS algorithm (you may skip this part if you do not want to know the gory details)
- Create an account using our Web platform
- How to create a domain
- How to configure our JNLP Platform for your domain
- How to deposit a secret

Table of content

Welcome to LEMS

Introduction

The Lybero Escrow Managing System, or as we call it LEMS, is a system born from the need of Bertrand Wallrich, the Chief Security Officer of Inria during 10 years and research teams specialized in encryption and protocol security.

The problem of Bertrand was the following: he realized that the theft of portable computers was not at all random but concentrated in some research teams with critical activities. Their computers were routinely disappearing during transportations, in conferences, … So he decided to ask for a systematic encryption of the hard-drives of these team members.

Concerning the theft, it was very efficient. However, after 3 weeks of holidays when people were coming back, they had forgotten their passwords, and when the drive is encrypted, well, there is not much that may be done (except restoring a backup if you have one).

As Inria is a distributed research center, the use of physical safe to store the key and passwords or passphrases, was not very practical, so Bertrand was looking for a digital escrow system, which he did not find. So he decided to write one in following the wise advices of high level security researchers.

This is the LEMS that you are using now. It is here to allow anybody to store secrets, to protect them for the organization and to allow the retrieval in a very secure way.

LEMS elements

The system allows to store a secret (text or file of any size) on a server and to retrieve it when a previously registered quorum of secret administrators asks for it.

The system is built of the following parts:

- A front-end server in charge of all communications with the different actors.
- A storage server.
- System administrator: he is in charge of creating the "domain" which associates secrets and a given list of secret administrators. These operations are done with a web interface.
- Secret administrators: they are in charge of allowing the retrieval of a secret, they have a dedicated software (LEMS client software).
- Users: they can store secrets or ask for the retrieval of them. They have a dedicated client software.

We are going to describe the sequence to create a domain, to store and to retrieve a secret in the system.

Properties and constraints of LEMS

All cryptographic operations are done on the client software. No secret key is available on the server, neither in memory nor in a file, nor rebuilt at any time. The "Perfect Forward Secrecy" is respected on the server.

All random value generations (session keys, asymmetric keys, …) are generated on the client software. These operations are difficult to attack as they are distributed.

Even in case a server is hacked, it does not lead to any secret divulgation.

To recover a secret, a quorum of secret administrators is mandatory. If an attacker wants to steal a secret, he will need to corrupt not one persons but many (the quorum number).

A user does not need to be identified to store a secret. However, he cannot retrieve the secret. To get it back, he must request it from the secret administrators[1].

[1]The mechanisms for requesting the recovery of a secret, the validation of the identity of the person requesting a secret, the legitimacy of the request and of its transmission to the administrators are not treated in this document.

Preliminary phase

Step 0: escrow domain creation

Before storing a first secret, the following configuration parameters must be chosen:

- The escrow domain name (a character string). It allows to do different secret administrators groups with different quorums, and different cryptographic parameters.
- The total number of secret administrators for this domain.
- The quorum (smaller than the total number of administrators)
- The login name of the secret administrators (without space character in them).

All these parameters are sent to the front-end server.

Step 1 : bi-key generation for each secret administrator

Each administrator will connect to the web interface and then to the client software. Each secret administrator must choose a password.

The client software generates randomly a public/private key pair for an asymmetric Elgamal cryptographic algorithm for this administrator. Both public and private keys are sent to the central server. The private key is symmetrically encrypted with the AES algorithm and the secret administrator's password.

Once all administrators did that, the domain initialization process can continue.

Step 2 : exchange of cryptographic parameters

The client software gets all information of the secret administrators (login, public key, encrypted private key, secret administrator rank).

Each secret administrator connects to the interface, gives his password which allows to decrypt his secret key.

Each administrator generates a polynomial of degree quorum minus one in a discrete integer space:

$$f(x) = \sum_{i=0}^{quorum-1} a_i mod(q).x^i$$

A part of the escrow domain public key is also worked out by each secret administrator:

$$A_0 = g^{(a_0)} mod(p).$$

(p is fixed for a given escrow domain).

Each secret administrator works out for each other the following number $S_{rank} = f(rank)$. He stores centrally then the following information:

A_O

and

$list of enc(S_{rank}) encrypted\ with\ the\ corresponding\ secret\ administrator\ public\ key$

Step 3 : domain public key generation

Once all A_O *and* $enc(S_{rank})$ are received by the server, the server works out the public key of the escrow domain.

$$Y = (\prod_{i=1}^{max} A_{0(i)}) mod(p)$$

The server then makes it available to all client software when asked.

Storage of a secret

The client software gets all informations necessary among which the domain public key *Y.*

The user then enters the string or the files to store.

The client software creates randomly a session key *m*.

The client software encrypts the provided data with this session key and an AES symmetric encryption algorithm.

The client software encrypt the session key m with the domain public key with Elgamal algorithm. It also integrates a random number *r*

$$m \rightarrow (\alpha, \beta) = (g^r mod(p), (Y^r . m) mod(p))$$

The software client sent to the server the Elgamal encrypted data (α, β) and non encrypted information allowing to identify the stored secret.

The server answers the success or failure of the secret storage.

Recovery of a secret

The recovery of the secret is done thanks to a quorum of administrators operating independently but in a given configurable temporal window (for example one hour).

Step 0: partial secret administrator secret decryption

The client software gets all information from the domain among which *Y* the public key of the domain and all $enc(S_{rank})$.

The software client also knows the rank *i* of its connected administrator.

With the password of the secret administrator, it can decrypt the private key of the administrator and then all S_{ji} corresponding to this secret administrator and

provided by the other secret administrators It then works out $S = \sum_{j=1}^{max} S_{ji} mod(q)$. S is the secret of the current secret administrator i.

Step 1: retrieval of the encrypted data

The secret administrator gets all information corresponding to the secret which is currently retrieved. It includes α.

Step 2: over-encryption

The software client works out $C = \alpha^S mod(p)$. He sends it to the server, encrypted for each other administrator with their public key. The server stores all C_{ji}(C worked out by secret administrator j and encrypted for the administrator i). The software client waits for these values from the other administrators.

Once the quorum is reached, the server sends back the C_{ji}(for i from 1 to max),β and the data encrypted with m the session key.

Step 3: local decryption

The client software works out $m = \beta . (\sum_{i \in \tau} C_j^{\lambda_j^{\tau}} mod(p))^{-1}$.

j being the rank of the currently connected secret administrator.

$\tau = \{i_1, \ldots, i_{quorum}\}$ is the rank of the different secret administrators that did a partial decryption.

$\lambda_j{}^{\tau}$ is the Lagrange polynomial coefficient : $\lambda_j{}^{\tau} = \prod_{k \in \tau \setminus \{j\}} \frac{k}{k-j} mod(q)$.

m is the session key which is necessary to decrypt the encrypted session data. Once decrypted these data can be provided to the secret administrator.

In the previous description, we have considered that the person asking for a secret is one of the secret administrator. However it may be another person. In this case, β is encrypted with the public key of the requestor, and the requestor does the final step of the decryption (getting m and deciphering the data). The secret administrators cannot decrypt the data.

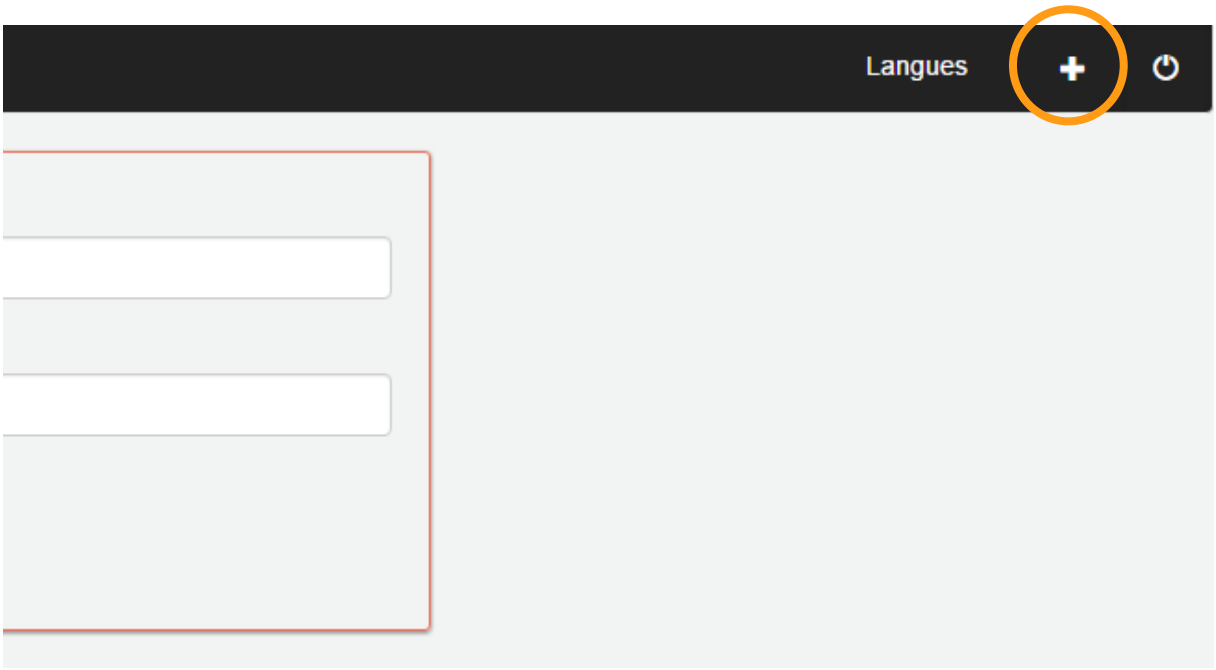
Next: Signing up

Signing up

While signing up might seem pretty easy, we never know and wouldn't like you to be lost.

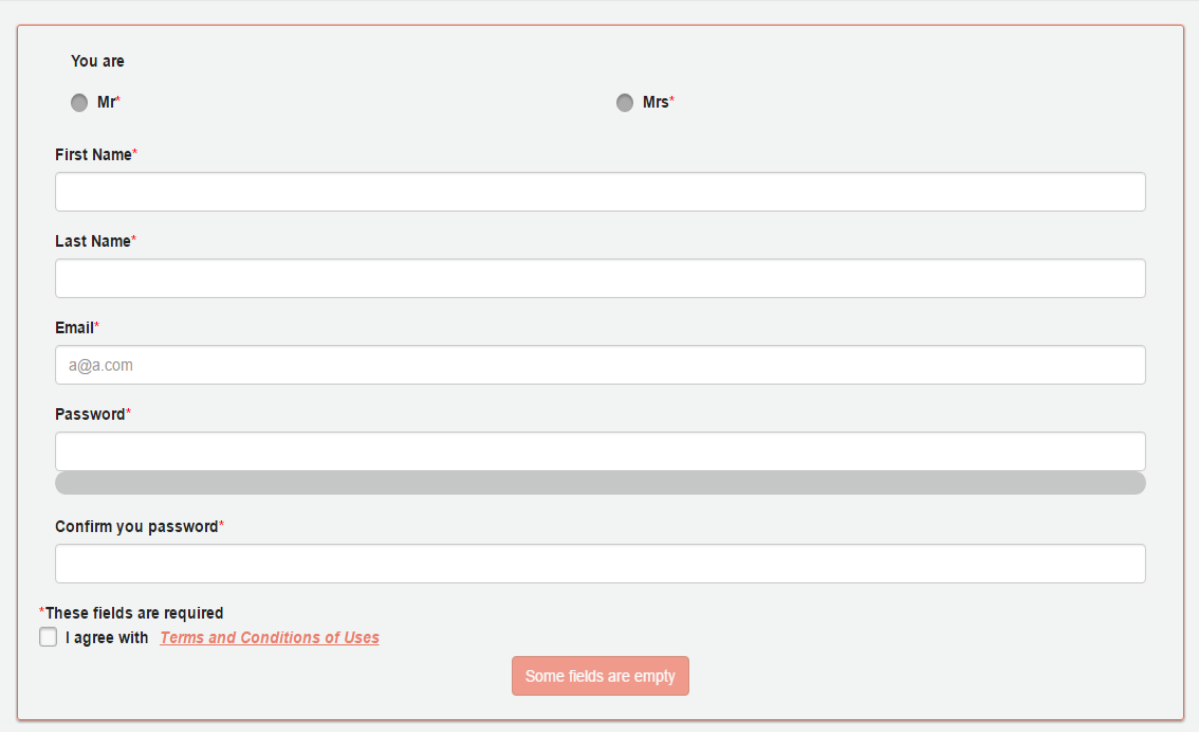
Starting from the platform homepage, you'll notice a plus sign in the navbar. This plus sign will allow you to access to our sign up page directly.

Notice you can also change the languages. For now, only French and English are available, but we never know, this might change someday.



We only ask you a couple of information such as your gender, first name, last name, email and password. Those information are use as such:

- Gender, first name and last name are used to make the mail, interface more friendly.
- Email address is used to confirm your account, and send notification email (probably no more than two). It is also used as your login username.
- Password is used to confirm your identity. You will need to remember this one.



An email will then be sent to you in order to confirm your address. Just click on or copy-paste the link given in your browser, in order to confirm your account.

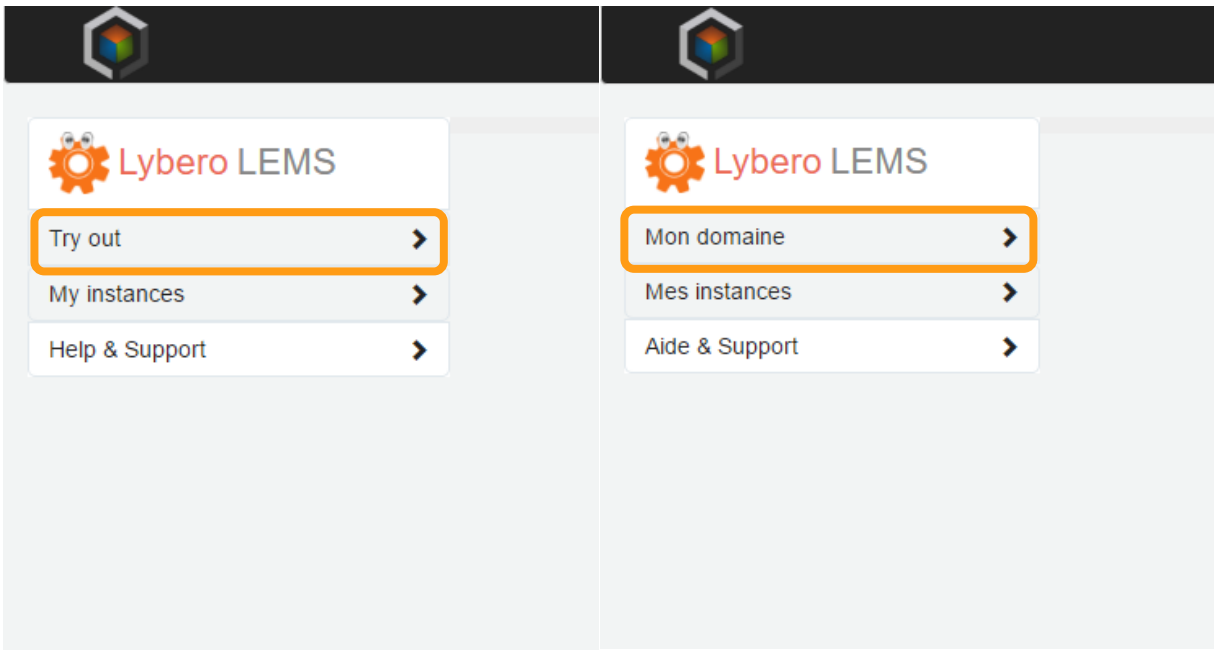
That's it, you can now log in our platform. **Welcome**!

Previous: Welcome to LEMS

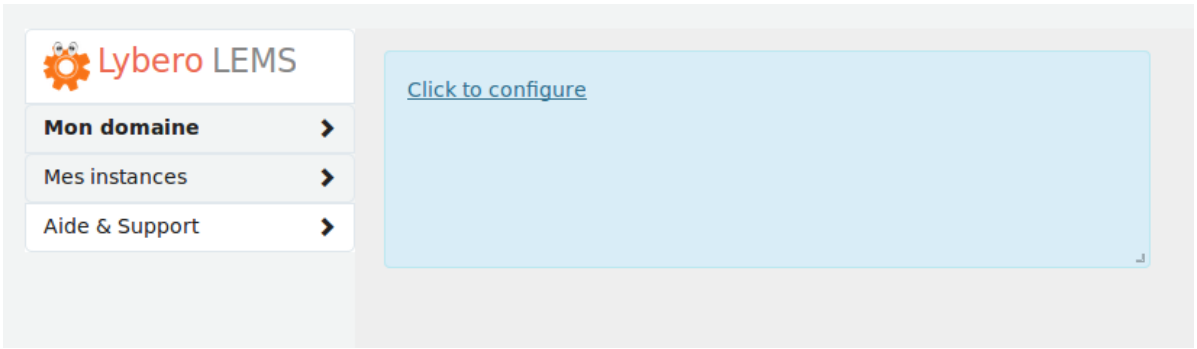
Next: Creating a domain

Creating a domain

Once you log in, a menu will appear on the left.



The "try out" ("Mon domaine" in French) entry allows you to configure your **demo domain**. For now, only one domain is available. But you can put as many secrets as you want in it.



Click on "Click to configure" to start the wizard installation of your domain.

Several information will be asked from you:

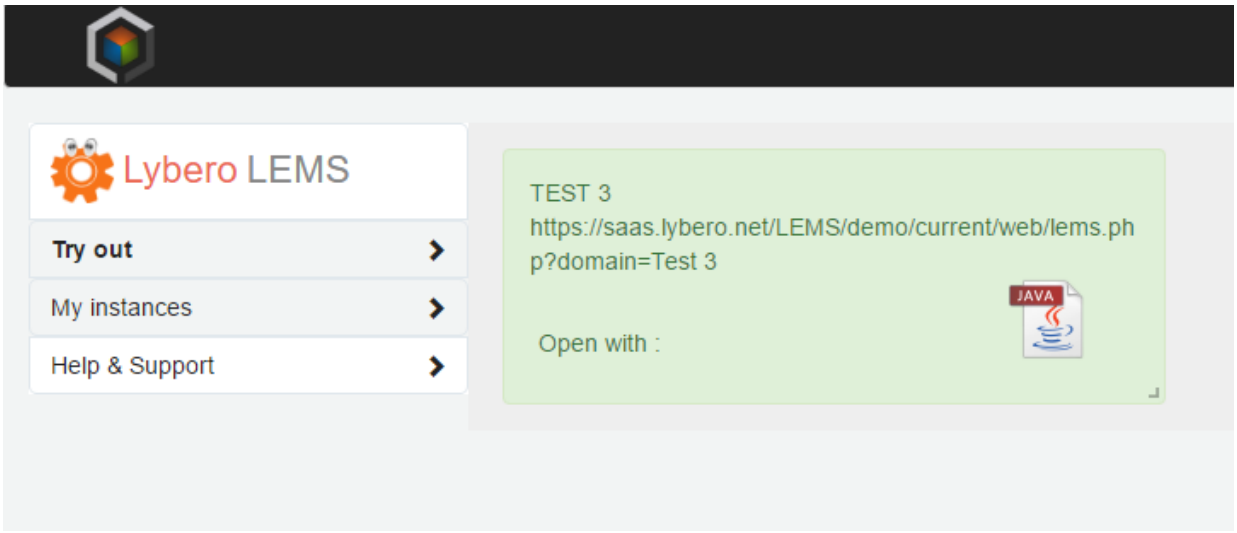
- **Domain name**
- **A small description**
- **Administrator list**
- **Quorum wished**

The administrator list, is a list of login separated with comma. You will need to remember those login and pass it down to the administrator of your domain. Indeed, the administrator might not be only you.

The quorum is the minimum number of administrator needed in order to decrypt a secret.

In the end, clicking on Configure will send to our server all of those information and asking our system to build your domain.

If everything went well, you would get this kind of screen.



Congratulations! Your domain is now created.

Previous: Signing up

Next: Configure the JNLP

Configure your JNLP

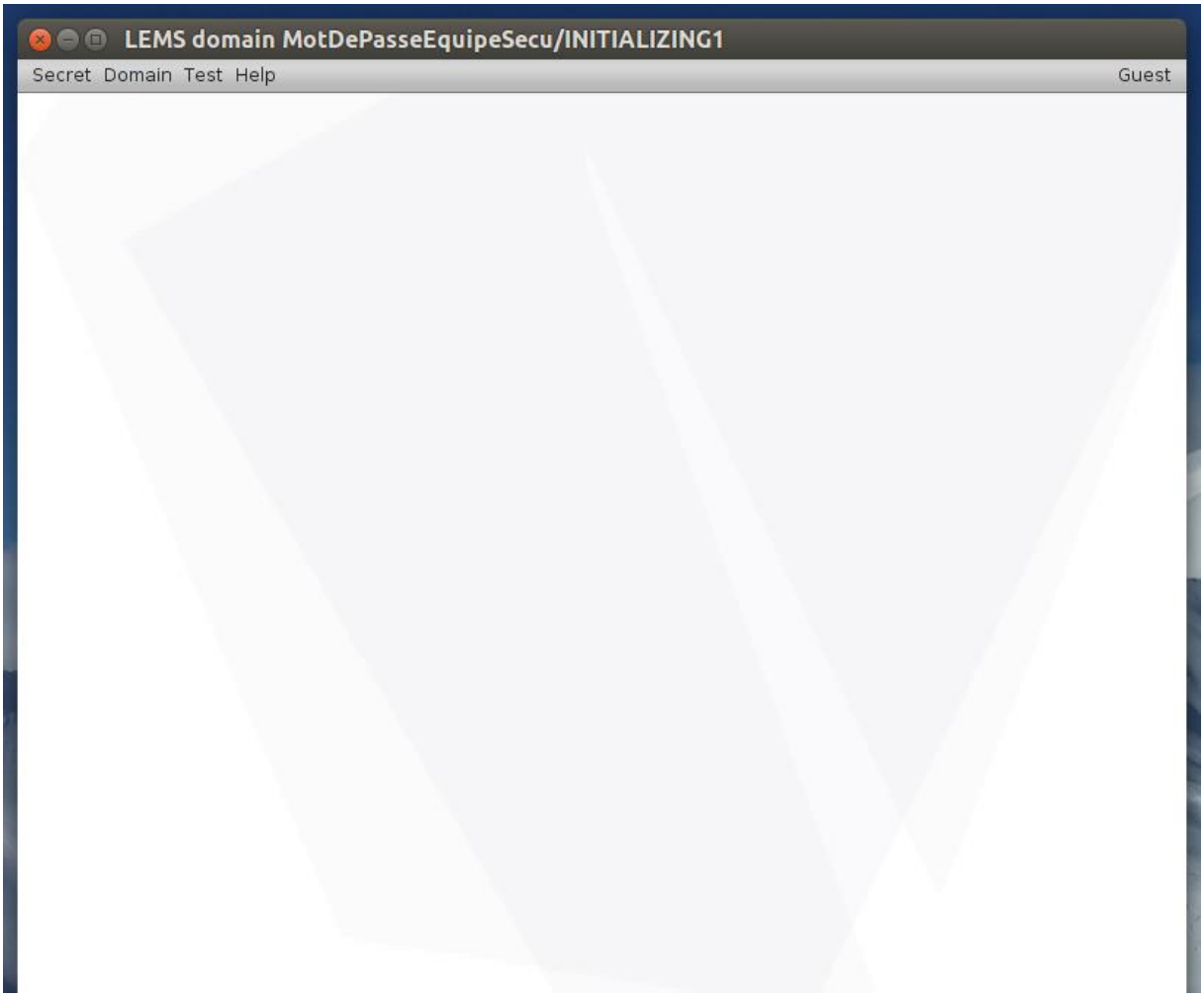
In order to access the JNLP, you need to have a Java Web Start application.

Why Java?

Java allows us to sign our code and portability. For now, it is the only language we found which permits both features. Therefore, we can assure you that the application you are using, is really ours.

Careful: You might need to allow the application to create a desktop launcher.

If everything went well, you would get this kind of screen.



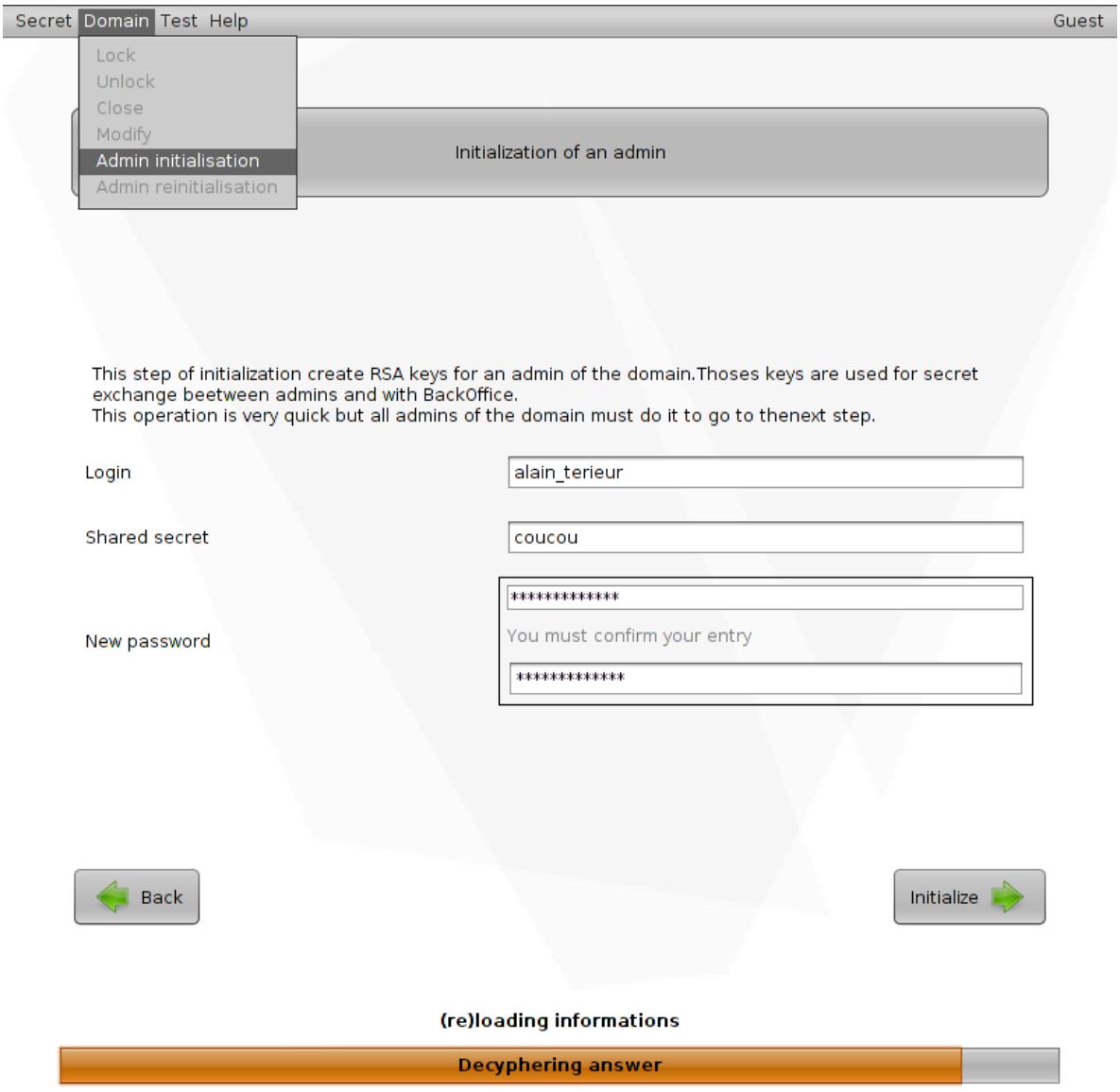
Step 1: Initializing all the admins

For each admin given in the administrator list when creating your domain, you have to create the RSA keys. Those keys are used for secret exchange between admins and with our BackOffice.

To do that, go to the Domain menu and click on Admin Initialization.

Fill the fields and click on initialize.

- **Login** must be one of the login given in the administration list.
- **Shared secret** is a string that you have previously shared between admins.
- **Password**



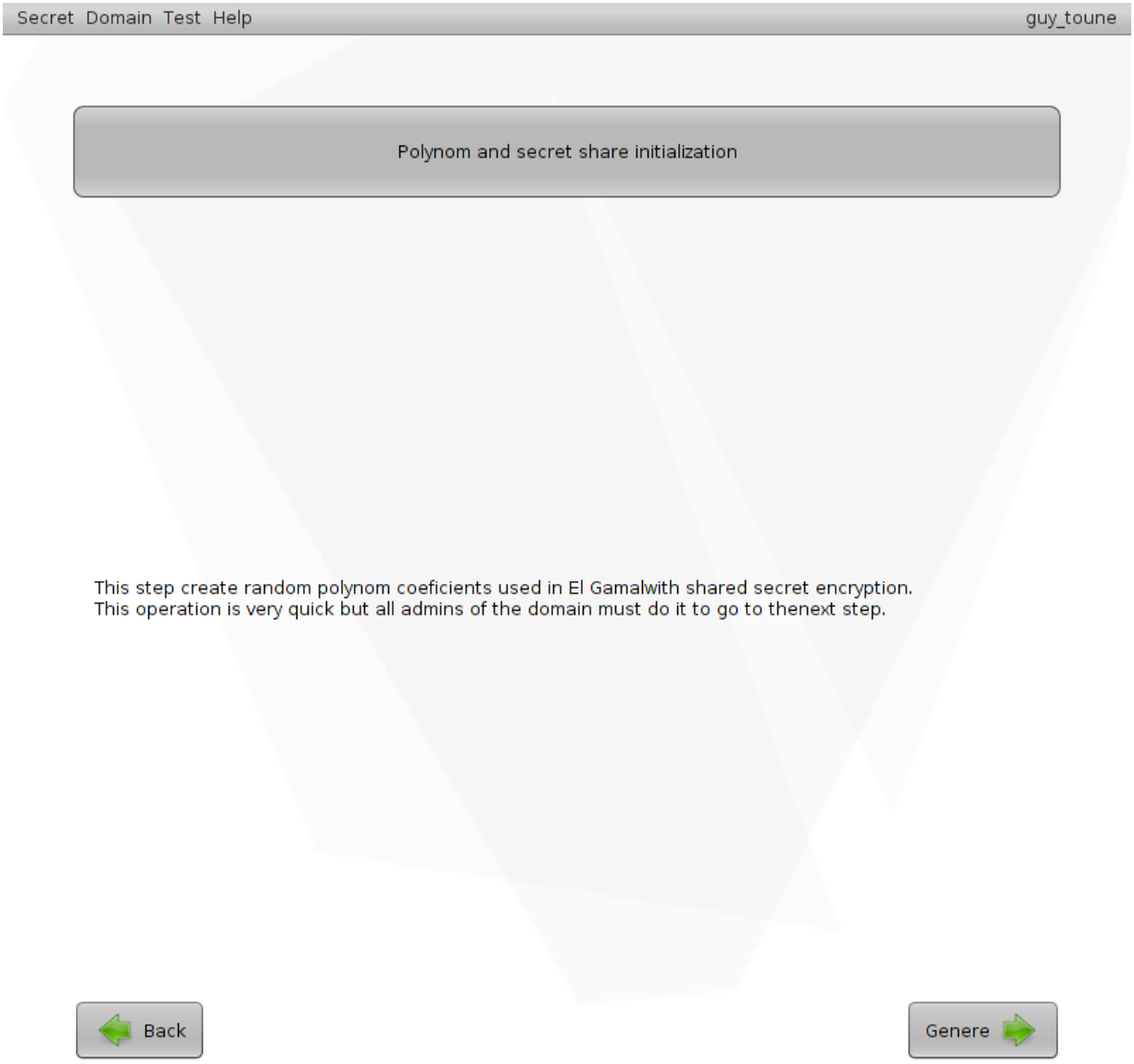
Note: The first admin that you will create **will be granted a SUPERVISOR role. He will be the one in charge of creating accounts for user.**

Step 2: Generate polynomial

For each admin given in the administrator list when creating your domain, you have to generate a polynomial with random coefficients as we are basing our security on El Gamal with shared secret encryption.

To do that, go to the Domain menu and click on Generate Polynom.

Again, each and all admins must do this step before being able to go to the next step.

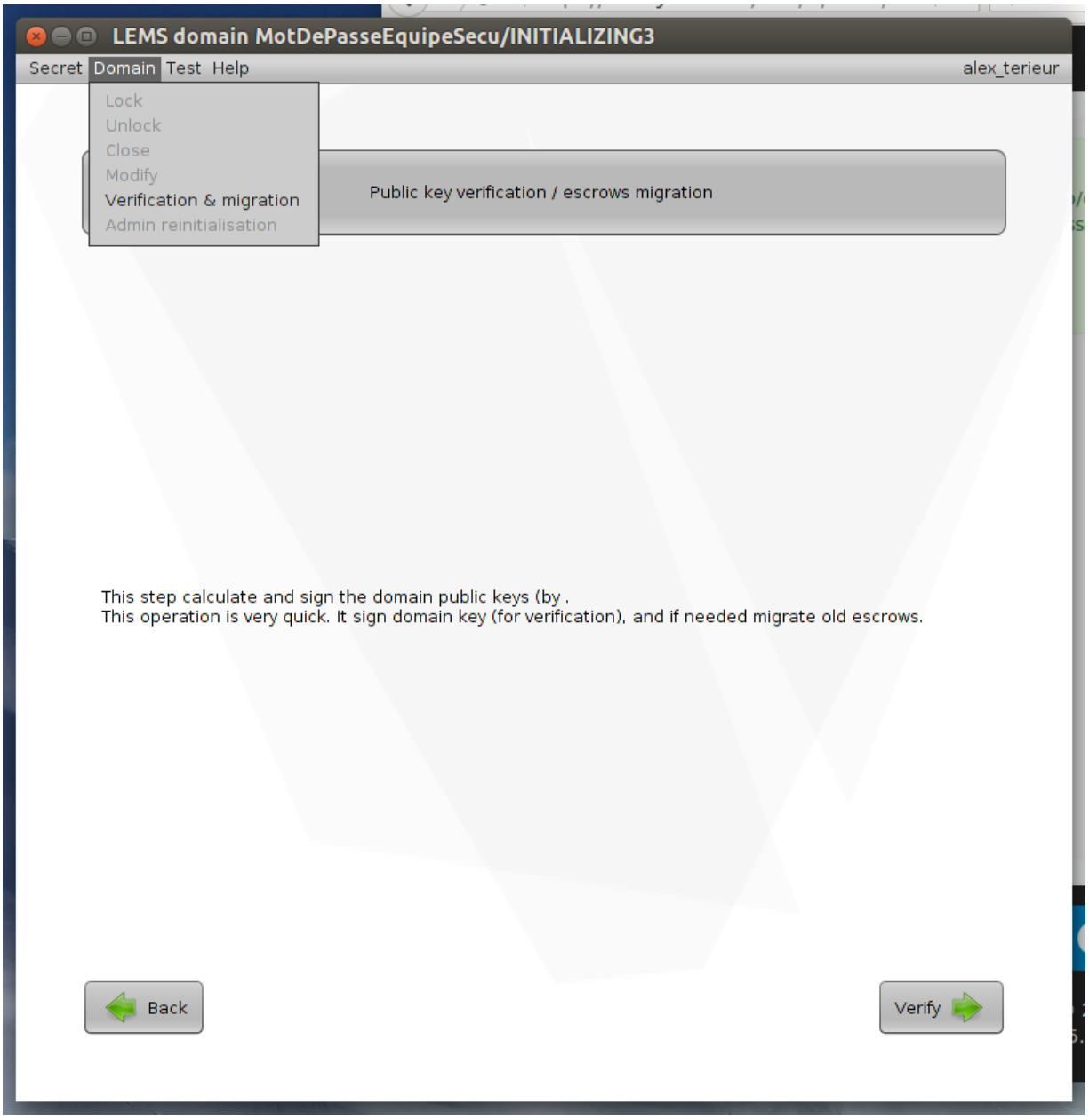


Step 3: Verification & Migration

For each admin given in the administrator list when creating your domain, this step will sign the domain public keys and migrate if needed old escrows.

To do that, go to the Domain menu and click on Verification & Migration.

Again, each and all admins must do this step before being able to go to the next step.



Congratulations! Your domain is now configured and ready to use.

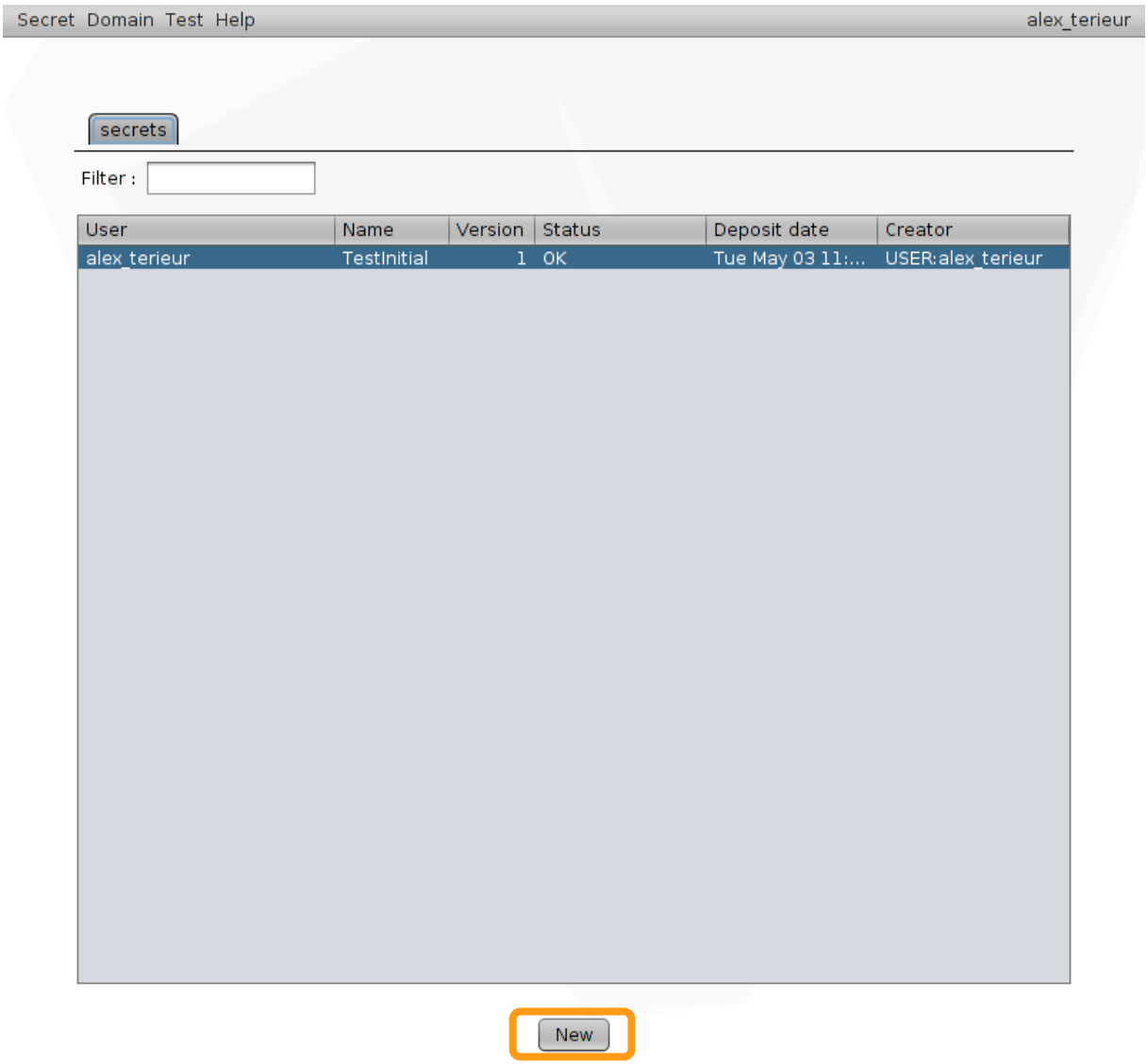
Previous: Creating a domain

Next: Add a secret

Add a secret

Secret is the real deal. It represents for LEMS the file, the string, the exact thing you want to keep from other.

To add a secret, make sure you are on the secrets tab and click on the **new** button, located on the bottom of the screen.

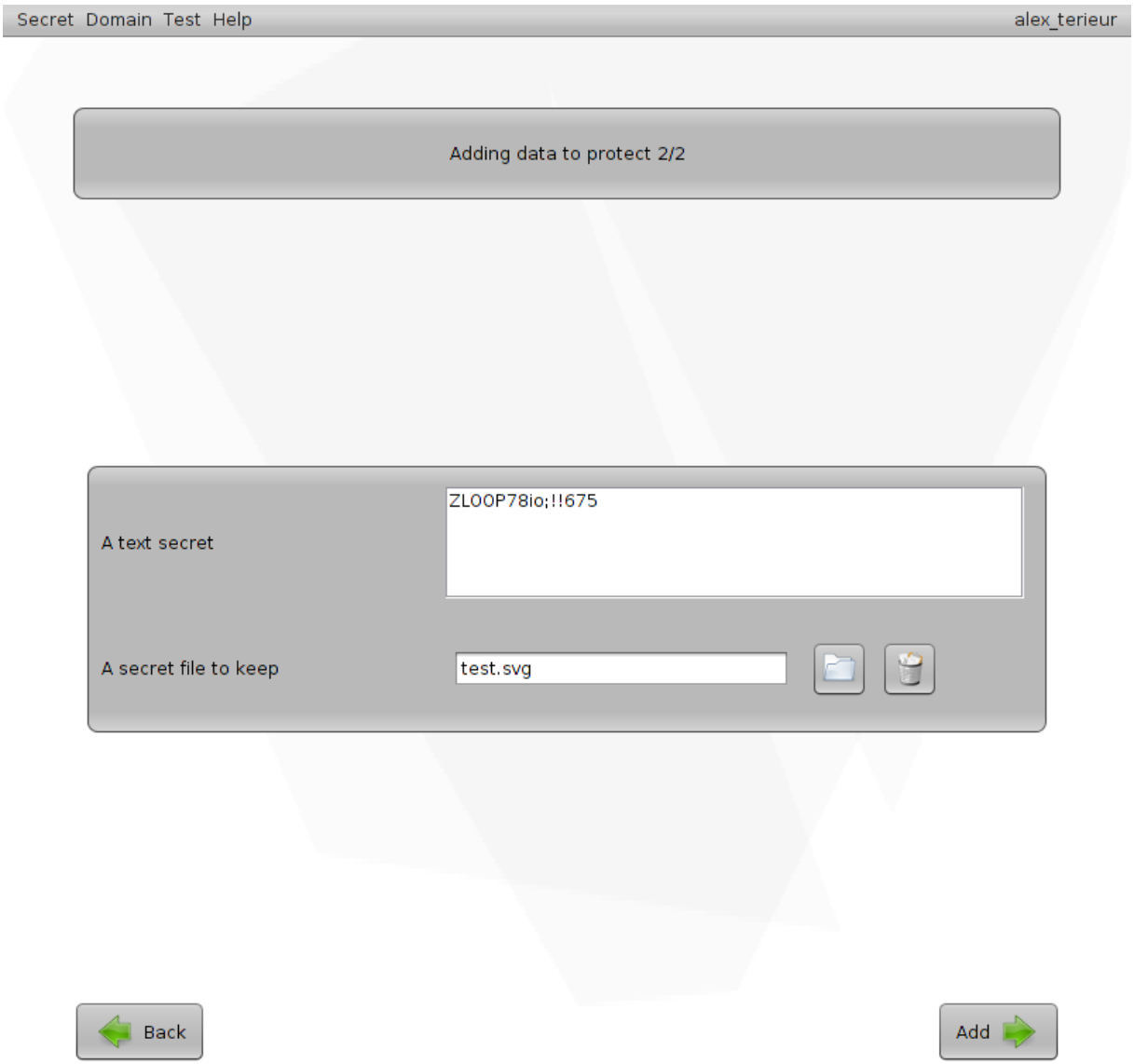


A wizard will open helping you to create an escrow which contains your secret.

Several information will be needed:

- Name for your escrow, in order to find among other your secret
- Description, to describe the escrow
- Some recover information, contains a description about who you allow to recover your secret.
- A text secret (first choice) is a field allowing you to write down a string that will be encrypted.
- A secret file to keep (second choice – can be both) is a field allowing you to import one to several files that will be encrypted.

Clicking on Add will begin the encryption of everything you entered, text and files.



Thank You For Using LEMS.

Please visit us at lybero.net to keep up to date on all the awesome product we create.