Inhaltsverzeichnis

Key Management

From Version 4 onwards, the SKDS script files are protected by a RSA encryption algorithm against unauthorized usage. That gives a probability of $1:2^{1024}$, that nobody else can ever use that script.

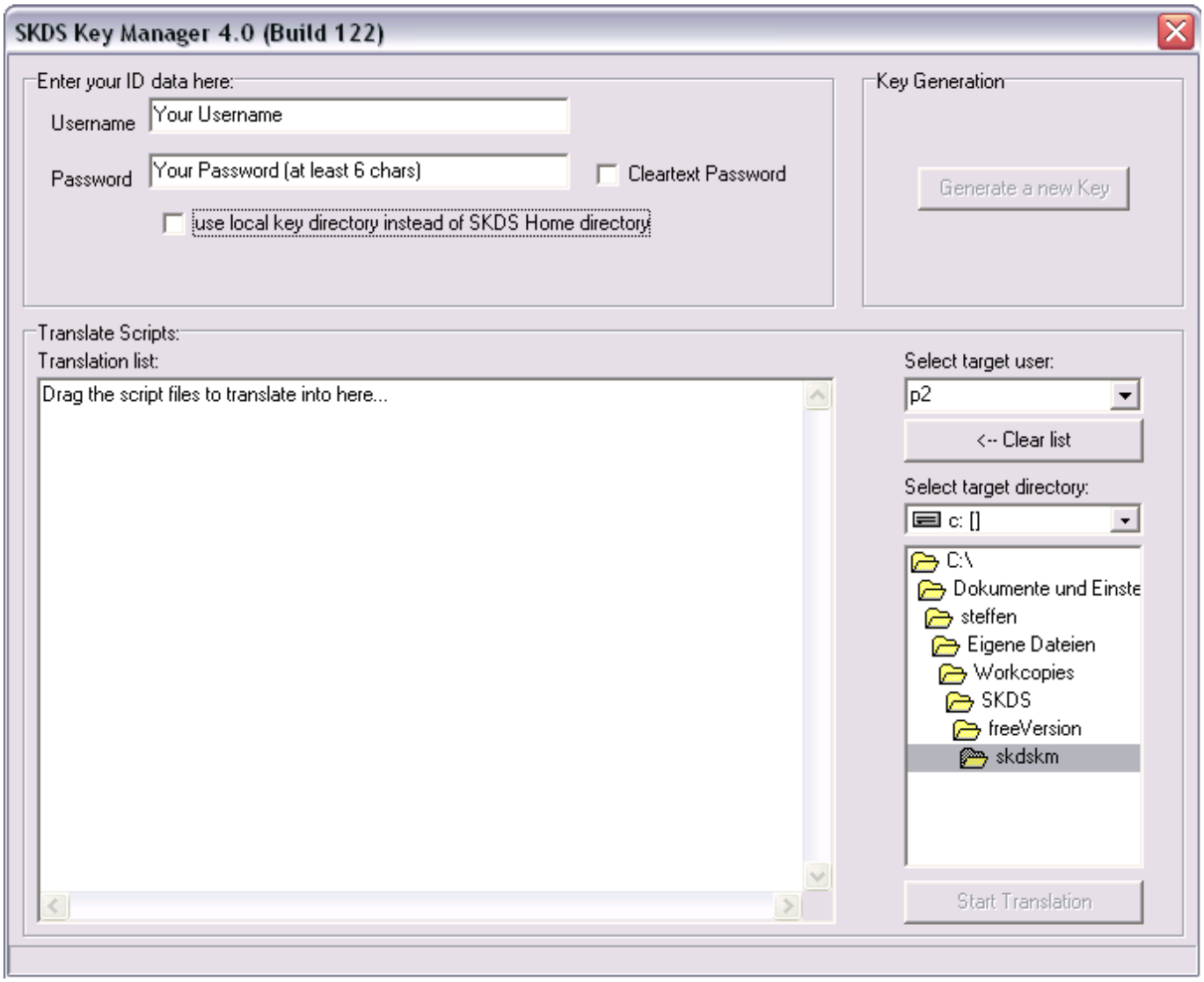
The used algorithm is a so called „public key encryption“. This means there are two so called key files, the so called public key and the so called secret key.

The public key is named yourchoosenusername.pubkey and is stored in the `pubkeys`- subdirectory, the secret key is named yourchoosenusername.seckey and is stored in the `seckeys`- subdirectory.

The secret key is protected by your chosen password and is needed to decrypt all script files, which where encrypted „for you“, means by using your public key. The public key instead can be given to anybody else, as the others would need this file to encrypt a script for you, which only you can read.

Inital Key setup - The Key Manager

This so called key pair is generated by the SKDS Key Manager.



The usage is simple:

Type in your chosen username and password (at least 6 characters long), press „Generate a new key“ and take a cup of coffee... Because of the mass of mathematics behind this key calculation takes a long time. If it takes longer than five minutes, than you either need a faster PC or something went wrong..

Key Generation & Handling Guideline

As the key system is the key point of the whole script handling, it's strongly recommended to follow a few rules:

- As the username is used also as the filename of the generated key files, you should use only characters [a-z], Digits or an understroke _, but no spaces, /:\ or other special characters, which might get in conflict with general file naming conventions. Remember also that some special characters (öääüß) might not be available on all keyboards.
- As the choosen username is your identification in the whole file exchange process, make your username self explaining, e.g. instead of using something anonymous like „flutschie1324“ use e.g. your company id „skoehle6“ to help other users to identify your public key easily.
- Always remember that your generated key files are **MANDATORY** to read your scripts! Whenever you forget your password, override the data with a new generated key or just loose the files, your script collection will become **UNUSABLE!**. So always store your password and a copy of the key files on a secure place - you have been warned..

Key file storage location & search algorithm

The way how the storage location of the key files is determined works as follow:

- **IF** there's a SKDS installation found on the PC (identified by its registry entry), **THEN** there will be two directories („pubkeys“ and „seckeys“) created in the SKDS installation directory, in which the key files are been stored (or searched for)
- **IF** there's no SKDS installation found, **THEN** the actual program directory is used instead.
- The use of the actual program directory can be forced by using the „use local key directory..“ check box. That's for the case that you run SKDS from USB-stick on a foreign PC where you want to use presumely your personal key collection but not the one from the PC owner.

Unlock your keys at program startup

To use your keys during the normal SKDS usage, they need to be loaded and „unlocked“ at startup. For this a key dialog appears at each program start

where you type in your username & password to load your keys.

How to compile and encrypt scripts with SKDSC

As we've learned above, script files are protected by encryption. But how to generate these files to work with them?

For this the already known command line compiler `skdsc` has been slightly modified.

First of all the -r option (who is able to read the scripts) has been removed, because it's not distinguished anymore between reading and executing a script [1)]

The -x option (who is allowed to execute a script) is modified in two ways:

1. because of the underlaying principle, you can (and have to) choose only **ONE** user, who is allowed to use (and able to read) the script. So it's not longer possible to define a group of people or organizations.
2. the -x option is now mandatory, as the encryption algorithm **always** needs a public key to encrypt to.

In practice that means that you start `skdsc` as follow

```
skdsc -x yourusername test.pas
```

to compile (and encrypt) a script for yourself and

```
skdsc -x theotherusername test.pas
```

to compile (and encrypt) a script for somebody else.

Please remind that you always need to have the public key file of „username“ in your public key directory, as this file is needed for the encryption process.

The Key Exchange process

This process is quite simple: Everybody, who wants to generate a script for you, needs to have your public key file (stored as yourusername.SKDSpubkey in your pubkey directory) and has to store it in his pubkey directory - so the easiest way is to just exchange these file and collect as much as possible keys of other users in your own public key folder.

[1)]

reading is not implemented anymore at all. Later, when SKDS becomes OpenSource, somebody could create its own reader, but it's not foreseen that the official releases will support this feature