GINIT

User manual

May 2011

SOFTIDEA

Contents

Pictures

1 Overview

Application GINIT is dedicated to management of the cryptography devices GNT USB Token [1] by *administrator*. It allows to query information about *Token* actual state, change passwords *APW, PIN1* and *PIN2* and to realize the following transitions between *Token* life cycle states*:*

- to configure and initialize *Token* for use by the *user*,
- to deinitialize *Token*, i. e. delete all user configurations and put the Token into the production state.

GINIT allows to work with just one Token present in the system. If more Tokens are present error window popups to notify the user.

2 User interface

The main view of the GINIT user interface is the dialog window displayed in the Figure 1. Meaning of all controls is explained in this chapter.

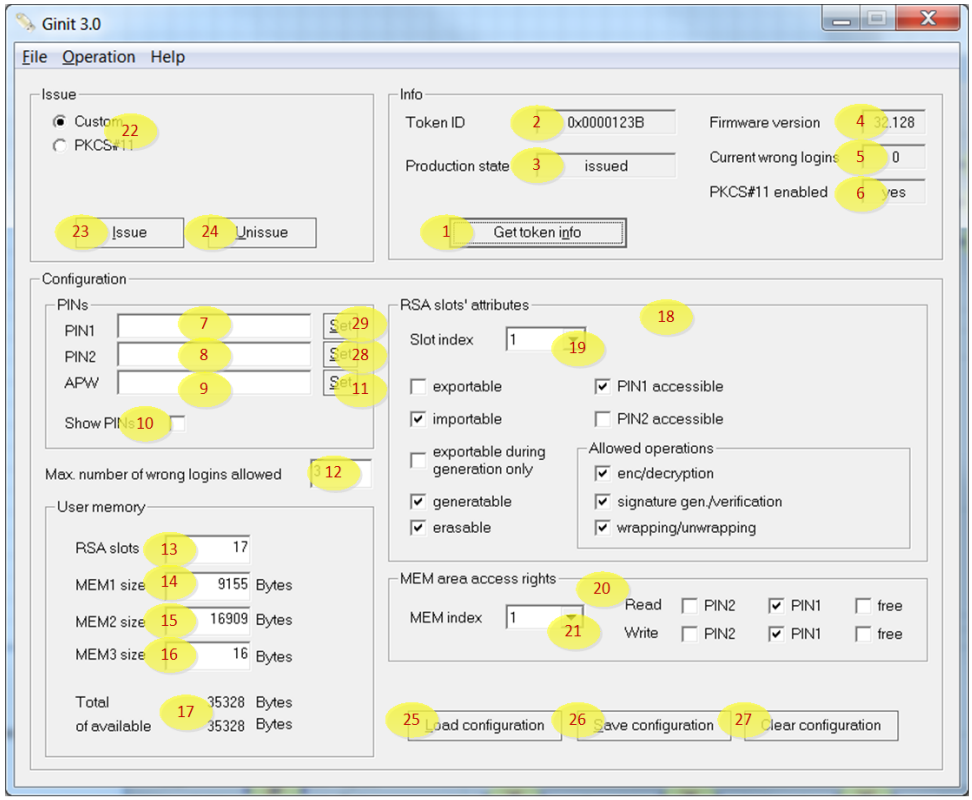

Figure 1: Main GINIT view

1. Button *Get token info*: Reads and displays public available *Token* configuration data.
2. Text box *Token ID*: Shows unique *Token* identifier.

3. Text box *Production state*: Shows actual *Token* life cycle state.

4. Text box *Firmware version*: Shows firmware version.

5. Text box *Curent wrong logins*: Shows actual value of *Wrong Login Counter* feature.

6. Text box *PKCS#11 enabled*: Indicates if *Token* is configured for SIPKCS [2].

7. Text box *PIN1*: Allows to set password PIN1.

8. Text box *PIN2*: Allows to set password PIN2.

9. Text box *APW*: Allows to set password APW.

10. Check box *Show PINs*: Allows to show or hide display of passwords.

11. Button *Set*: Set or change password *APW* to the value in *Text box APW* (9).

12. Text box *Max number of wrong logins allowed*: Configure / displays the configuration of *Wrong Login Counter* feature.

13. Text box *RSA Slots*: Configure / displays the configured count of RSA slots.

14. Text box *MEM1 size*: Configure / displays the configured size of MEM area.

15. Text box *MEM2 size*: Configure / displays the configured size of MEM area.

16. Text box *MEM3 size*: Configure / displays the configured size of MEM area.

17. Text box *Total X of available Y*: Displays the total available user memory (*Y*) and length of the user memory (*X*) consumed by the chosen configuration.

18. Group *RSA Slots attributes*: Configure / displays the configured attributes of *RSA slot X* where *X* is the number in the text box *Slot index* (19).

19. Text box *Slot index*: Defines RSA slot which attributes are displayed in group (18).

20. Group *MEM area acess rights*: Configure / displays the access rights of MEM *area X* where *X* is the number in the text box *MEM index* (21).

21. Text box *MEM index*: Defines MEM area which access rights are displayed in group (20).

22. Radio button *Issue*: Allows to select how is the Token to be initialized. If set to "*Custom*" all features are configurable. If set to "*PKCS #11*" some features are configured automatically by GINIT and Token will be initialized for SIPKCS.

23. Button *Issue*: Initializes Token based on the selected configuration data.

24. Button *Unissue*: Deinitializes Token.

25. Button *Load configuration*: Allows to load existing configuration from file.

26. Button *Save configuration*: Allows to store selected configuration to file.

27. Button *Clear configuration*: Allows to clear the selected configuration.

28. Button *Set*: Set or change password *PIN2* to the value in *Text box PIN2* (8).

29. Button *Set*: Set or change password *PIN1* to the value in *Text box PIN1* (8).

Certain operations require authentication using password *APW* or *PIN2*. In such case GINIT prompts user with dialog like one in the Figure 2.

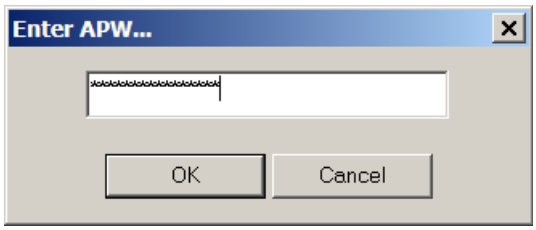

Figure 2: Request for password

3 Operations with Token

3.1 Configuration query

To query the actual Token configuration press button "*Get token info*" (control 1. on Figure 1). After successfully obtained actual Token configuration data (excluding passwords) is displayed. Extent of displayed data depends on the actual Token life cycle state.

3.2 Change of *APW*

Password *APW* can be changed using the following sequence:

- set the desired new value of password *APW* (control 9. in Figure 1),
- press button *Set* (control 11. in Figure 1),
- if prompted input old password *APW*.

User is notified about the successful operation through the dialog in the Figure 3.

Note: It is also possible to change APW as a part of Token initialization process (chapter 3.5) through setting the new value of APW into the APW text box (control 9. in Figure 1).

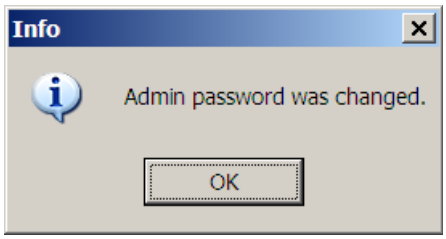

Figure 3: Notification about successful password change

3.3 Change of *PIN2*

Password *PIN2* of the configured Token can be changed using the following sequence:

- set the desired new value of password *PIN2* (control 8. in Figure 1),
- press button *Set* (control 28. in Figure 1),
- if prompted input old password *APW*.

User is notified about the successful operation through the dialog.

Note: Without the knowledge of old PIN2 it is not possible to apply the sequence described above. In such case it is possible to change PIN2 only through Token reinitialization (see 3.5 and 3.6).

3.4 Change of *PIN1*

Password *PIN1* of the configured Token can be changed in GINIT using the following sequence:

- set the desired new value of password *PIN1* (control 7. in Figure 1),
- press button *Set* (control 29. in Figure 1),
- if prompted input password *PIN2*.

User is notified about the successful operation through the dialog.

Note: Without the knowledge of PIN2 it is possible to change PIN1 in GINIT only through Token reinitialization (see 3.5 and 3.6).

3.5 Initialization

Token initialization (also known as "*Token* issue") means definition of user configuration according to the desired security policy, setting of default user passwords *PIN1* and *PIN2* and issuing of the *Token* to user. Technically it is the transition "*issue*" of *Token* life cycle described in details in [1]. *Token* is usually initialized by the security officer after authenticated using the password *APW*. The default administrator password *APW* set by the manufacturer is*"admin"*. In

order the Token can be initialized its actual state must be "unissued".

If the radio button *Issue* (control 23. in Figure 1) is set to "*Custom"*, it is possible to set all user configuration values in the full extent. The preset configuration values can be stored in and restored from the file using buttons 25-27 in Figure 1.

If the radio button *Issue* (control 23. in Figure 1) is set to "*PKCS#11"* some of the UI controls are not enabled. More information about initialization for SIPKCS is in chapter 3.5.1.

After all configuration data is preset administrator commits the Token initialization process pressing the button *Issue.* If administrator is not currently logged in she/he is prompted for password APW. After the successful operation the notification dialog (Figure 4) popups.

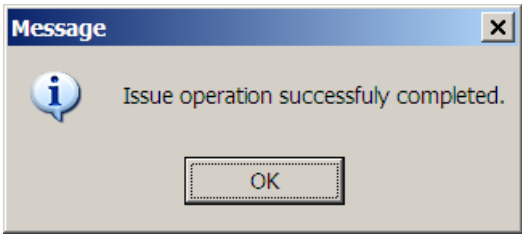

Figure 4: Notification about successful Token initialization

3.5.1 Initialization for SIPKCS

SIPKCS [2] is implementation of the PKCS #11 standard [3] abstracting access to hardware cryptography devices. Token is initialized and configured for SIPKCS using the following sequence:

- set radio button *Issue* to "*PKCS#11"*,
- input default *PIN1*,
- optionally input new password *APW*,
- press the button *Issue*
- if prompted input current password *APW*.

Note: Loading of configuration of Token initialized for SIPKCS and subsequent initialization of other Token with that configuration is not correct way to initialize for SIPKCS. Correct initialization for SIPKCS requires additional automated step (write of configuration data into the configuration memory MEM3) that takes part only if the radio button Issue is set to "PKCS#11".

After loading of actual configuration of the Token initialized for SIPKCS the SIPKCS support is indicated by value "*yes*" in the field "*PKCS#11 enabled*" (control 6. in Figure 1). For more information about SIPKCS see the document [2].

3.6 Deinitialization

Token deinitialization (also known as "*Token* unissue") means deletion of any user data and configuration and putting the Token into the same state as it was delivered from the manufacturer (production state). Technically it is the transition "*unissue*" of Token life cycle described in details in [1]. *Token* can be deinitialized by the security officer after authenticated using the password *APW*. In order the Token can be deinitialized its actual state must be "issued". Administrator performs deinitialization by pressing the button *Unissue*. If not already authenticated administrator is prompted for *APW*. During the Token deinitialization the password *APW* is automatically reset to the default value "admin".

Caution: Token deinitialization irreversibly delete all data in Token user memory, including cryptographic keys and data in MEM areas.

4 References

[1] GNT USB Token - datasheet, SoftIdea, s.r.o. , May 2011, http://www.softidea.sk/gnt_datasheet_en.pdf

[2] SIPKCS - PKCS #11 provider for GNT USB Token, SoftIdea, s.r.o. , May 2011, http://www.softidea.sk/sipkcs_specification_en.pdf

[3] PKCS #11 v2.20: Cryptographic Token Interface Standard, RSA Laboratories, June 2004, http://www.rsasecurity.com

SoftIdea s.r.o.
Sliačska 10, 831 02 Bratislava
tel.: +421 2 444 60 444
fax.: +421 2 446 40 441
http://www.softidea.sk
info@softidea.sk