DATE OF DOCUMENT CREATION: 8/03/2024

PEM-HEART SIGNATURE USER MANUAL

CENCERT

PUBLIC DOCUMENT

CREATED BY ENIGMA INFORMATION PROTECTION SYSTEMS SP. Z O.O. 02-230 WARSZAWA

UL. JUTRZENKI 116 | PHONE: +48 22 570 57 10 | FAX: +48 22 570 57 15

WWW.ENIGMA.COM.PL

DATE OF DOCUMENT CREATION: 8/03/2024
TYPE OF DOCUMENT: PUBLIC

ENIGMA SYSTEMY OCHRONY INFORMACJI SP. Z O.O.
JUTRZENKI 116
02-230 WARSAW
POLAND

TELEPHONE: +48 22 570 57 10
FAX: +48 22 570 57 15
WEBSITE: WWW.ENIGMA.COM.PL

TABLE OF CONTENTS

1 INTRODUCTION

PEM-HEART Signature software is used to:

- creation of qualified signatures or electronic seals based on certificates issued by Cencert,
- verification of qualified electronic signatures (also signatures based on certificates issued in other EU countries), within the validity period of the certificate.

In addition:

- verification of electronic signatures after the expiry of the validity period of the certificate, if the signature is in archive form (see description of archive form in chapter **4.3.1 Verification Panel, p. 33**),
- verification of signatures based on ordinary (non-qualified) certificates issued by Cencert.

PEM-HEART Signature performs electronic signatures in formats:

- XAdES compliant with the ETSI Technical Specification TS 101 903 XML Advanced Electronic Signatures (XadES),
- CAdES CMS compliant with the ETSI Technical Specification TS 101 733 Electronic Signature Format (CAdES stands for CMS Advanced Electronic Signatures),
- PAdES (ETSI standard TS 102 778) - PDF Advanced Electronic Signatures,
- ASiC (ETSI TS 102 918 standard) - the program executes the signature in ASX basic form, creating a file with the extension .asics . (the file contains the basic ASiC XadES container surrounding it).

These formats determine the structure of the file containing the signature. The choice of a particular format entails a requirement for software that will be able to verify the correctness of such signature.

The manufacturer of solutions for Cencert is ENIGMA Systemy Ochrony Informacji Sp. z o.o.
ENIGMA's core business is development, production and implementation of innovative information protection systems. Using its own hardware and software solutions, it provides the best data protection in state and local administration, financial institutions

and enterprises. All ENIGMA products provide full cryptographic protection of collected, processed and transmitted information. The solutions offered are certified in terms of security by specialised units of the State Protection Services.

Cencert is a registered trademark of ENIGMA Systemy Ochrony Informacji. Cencert is a qualified provider of qualified and non-qualified trust services since 2009 - for the issuance of certificates, qualified timestamps and certificate validation service (OCSP). The legal basis for Cencert's services is in particular eIDAS (Regulation (EU) No 910/2014 of the European Parliament and of the Council), as well as the Act on Trust Services and Electronic Identification (Journal of Laws 2016, item 1579).

2 PRODUCT SAFETY

The program should be used on a computer that is under the control of the certificate owner. The computer should be protected against accidental access, have up-to-date anti-virus software installed and current operating system updates.

Electronic signatures must not be made on computers whose security is not known (e.g. computers accessible to the public or to a wide range of people, computers of random people, etc.).

The program should be used in an environment where the program code is protected from modification by the operating system. This can be achieved by using operating systems that offer access control (Windows, Linux and MacOSX) or by setting access rights to directories with executable files so that the user does not have the right to modify the executable files contained therein.

The program should be used in an environment where the operating system protects against the possibility of interception by hostile systems of data sent through the computer's ports, as well as data entered from the computer's keyboard into the program's windows. This can be achieved by using operating systems that offer access control (Windows, Linux and MacOSX) and by ensuring an adequate level of protection of the computer against authorised users (protection by setting appropriate access rights and keeping the operating system up-to-date), unauthorised users and attacks from the computer network (protection by keeping the operating system up-to-date and, if necessary, using firewalls).

The software, working as "secure devices for the creation and verification of secure electronic signatures", cannot be used in a "public environment" - that is, in an environment where the software can be accessed by any natural person under normal operating conditions.

The technical component or the drivers supplied with it, which are part of the 'secure electronic signature creation and verification device' alongside the software, have the function of destroying the signature creation data (i.e. the private key) at the user's request. The destruction is performed to such an extent as to make it impossible to reconstruct the data on the basis of an analysis of the records in the devices where they were created, stored or used.

3 INSTALLATION

Installation packages are available on the Cencert website:

https://www.cencert.pl/do-pobrania/oprogramowanie-do-podpisu/

3.1 INSTALLATION FOR WINDOWS

3.1.1 INSTALLATION

The installation should be carried out from an account with administrator privileges. It is recommended that all applications other than those necessary for the operation of the operating system are terminated before the installation begins.

The following installation procedure is presented using Windows 11 as an example:

1. Run the *pemheart-signature.exe* installer, this will bring up the start-up installation window.

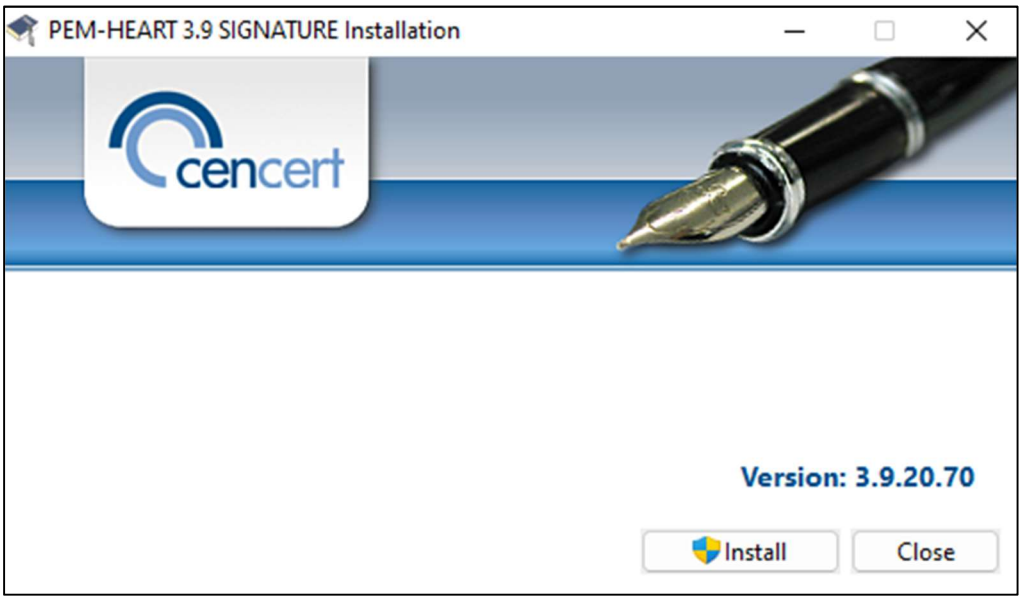


Figure 1 Start-up window of the installation wizard

2. Clicking the *Install* button will initiate *the PEM-HEART 3.9 Signature product installation wizard*.

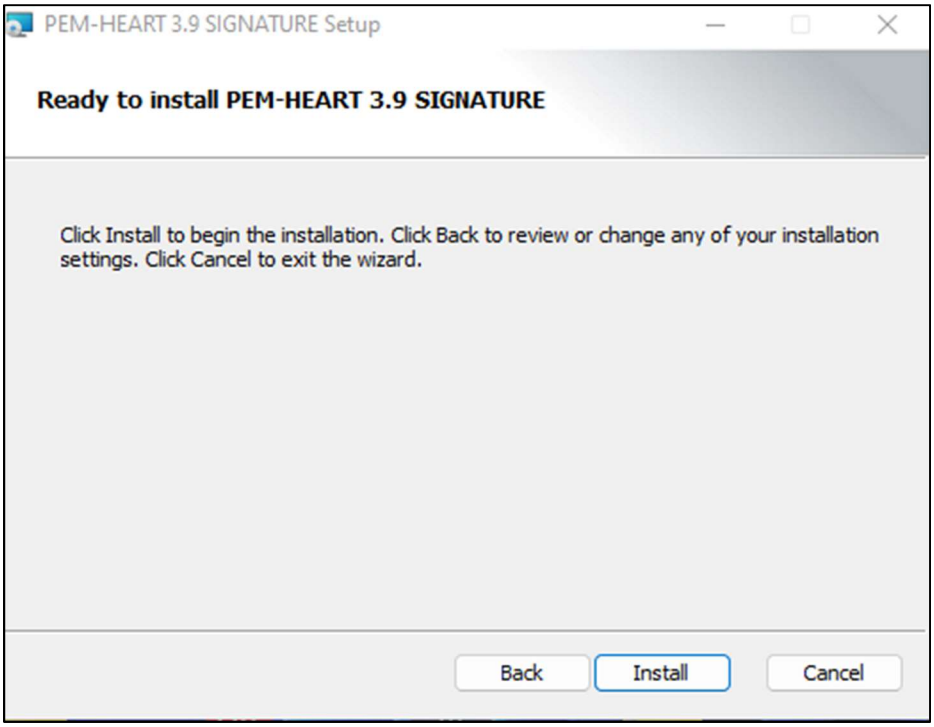


Figure 2 Installation wizard window

3. Click *Install*, the software installation will begin.
4. Click *Finish* - this completes the wizard. This will also start the installation process for the Thales SafeNet software.

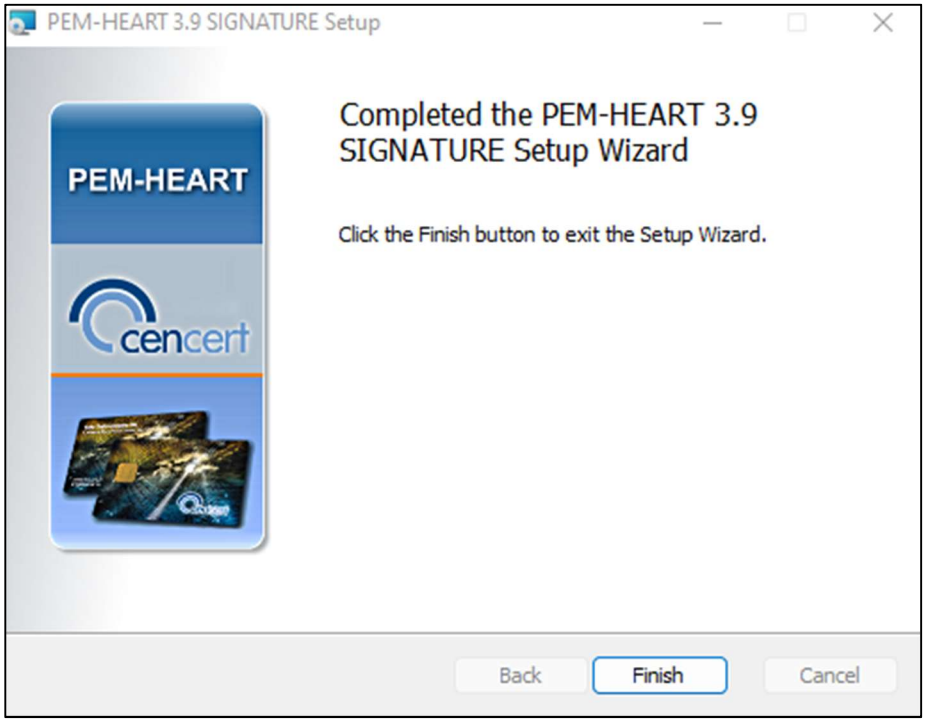


Figure 3 Window finishing the installation wizard

5. The Thales SafeNet software installation wizard is displayed.

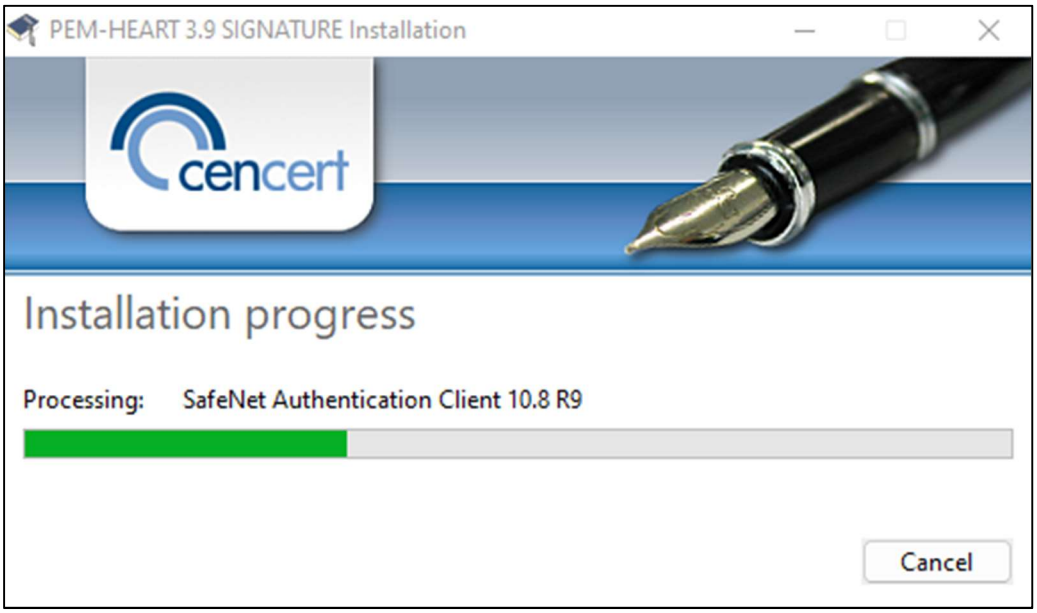


Figure 4 Installation of Thales SafeNet software

6. Click *Restart* - this is required to start the software.

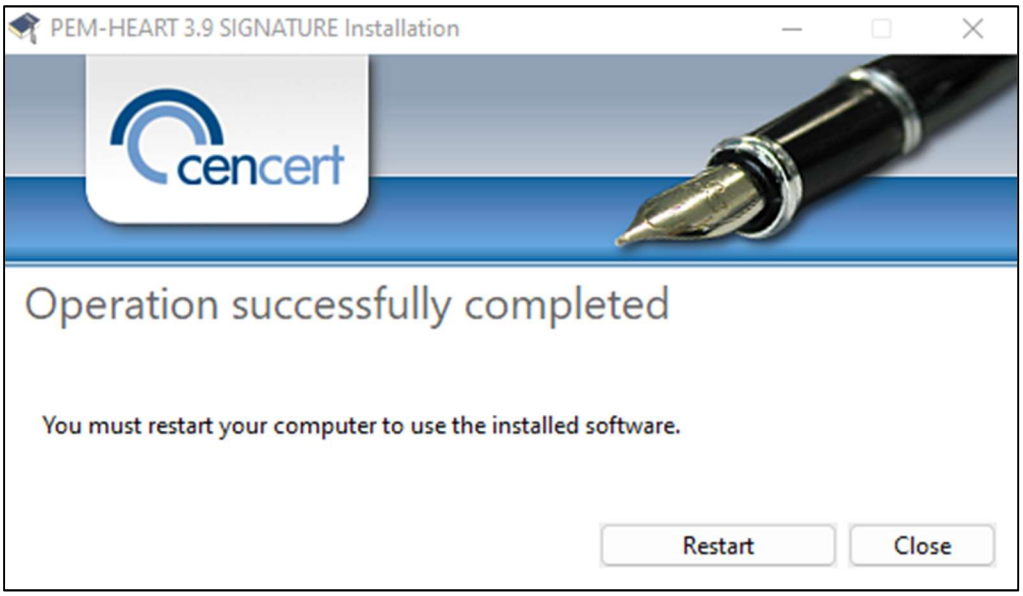


Figure 5 Window for completing the installation process

3.1.2 PROGRAM DELETION

The program is removed by selecting the "PEM-HEART SIGNATURE" package from the Windows Control Panel: Control PanelPrograms and Features.

1. The installation wizard will start. Click on the *Uninstall* button - the program will start the process of removing the program from the resources of the operating system.

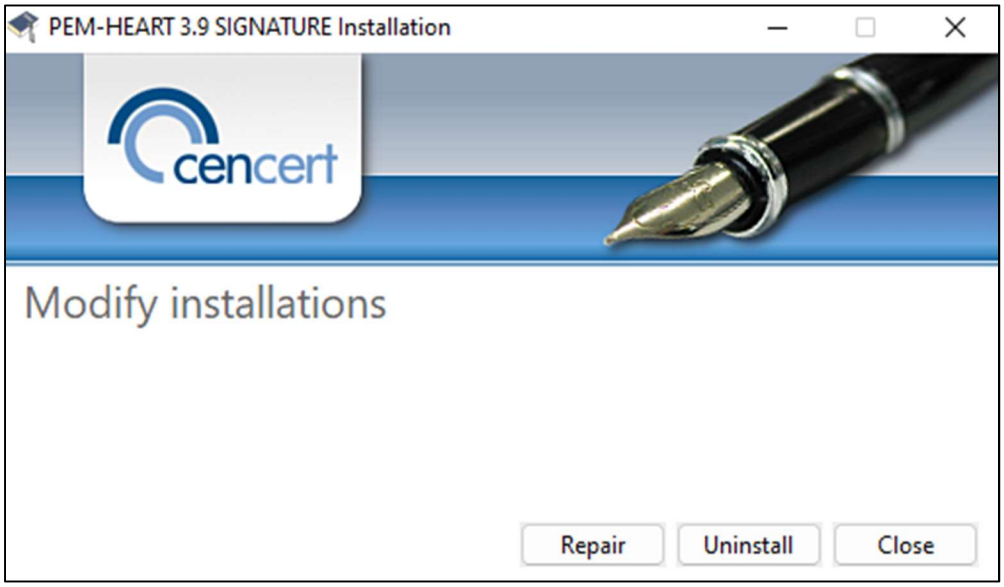


Figure 6 Installation modification options

2. During the process, a message will be displayed asking, if you want to delete or keep the configuration for SafeNet for Thales card support.

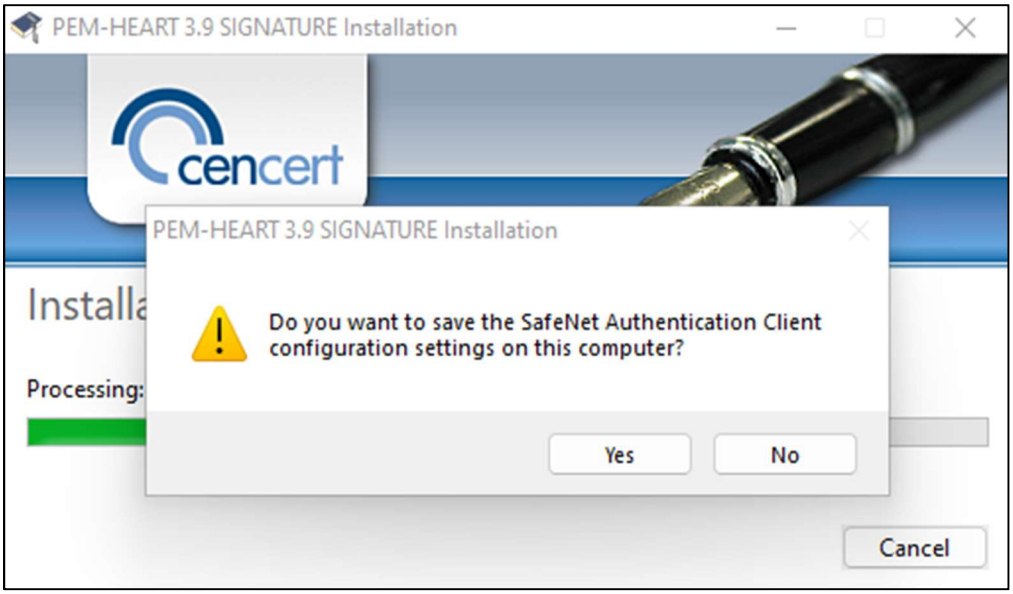


Figure 7 Deinstallation of the program

3. The installation wizard will inform you that the software removal process is complete. A reboot of the computer is required by clicking on *Restart*.

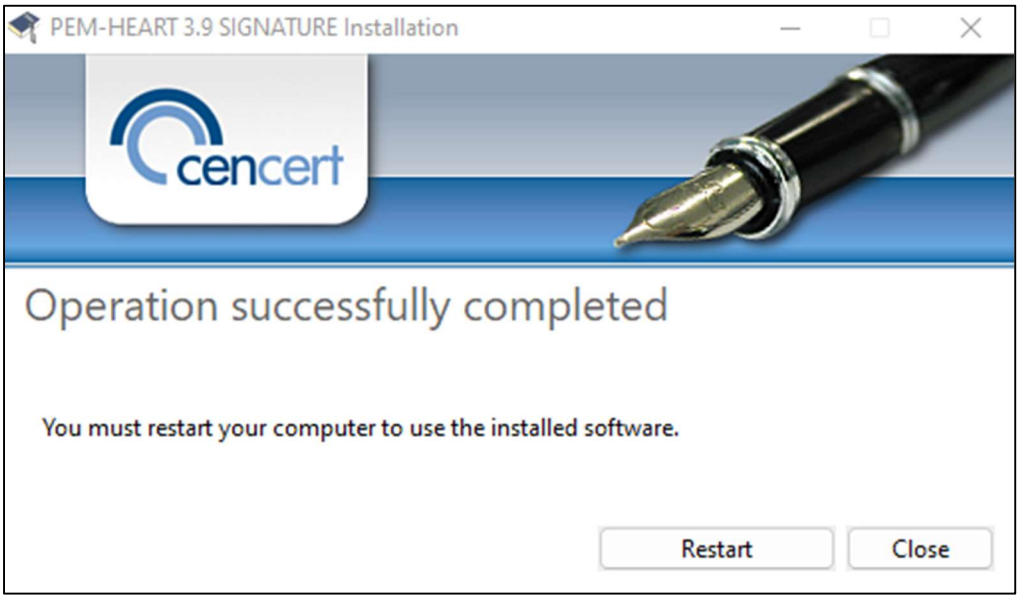


Figure 8 Confirmation of the uninstallation of the program

3.2 INSTALLATION FOR MACOS

The Pem-Heart package for macOS is distributed via the .dmg format - this contains the installation and uninstaller files.

Pem-Heart has support for macOS versions: 13 (Ventura) and 14 (Sonoma)

The following manual is based on macOS Ventura.

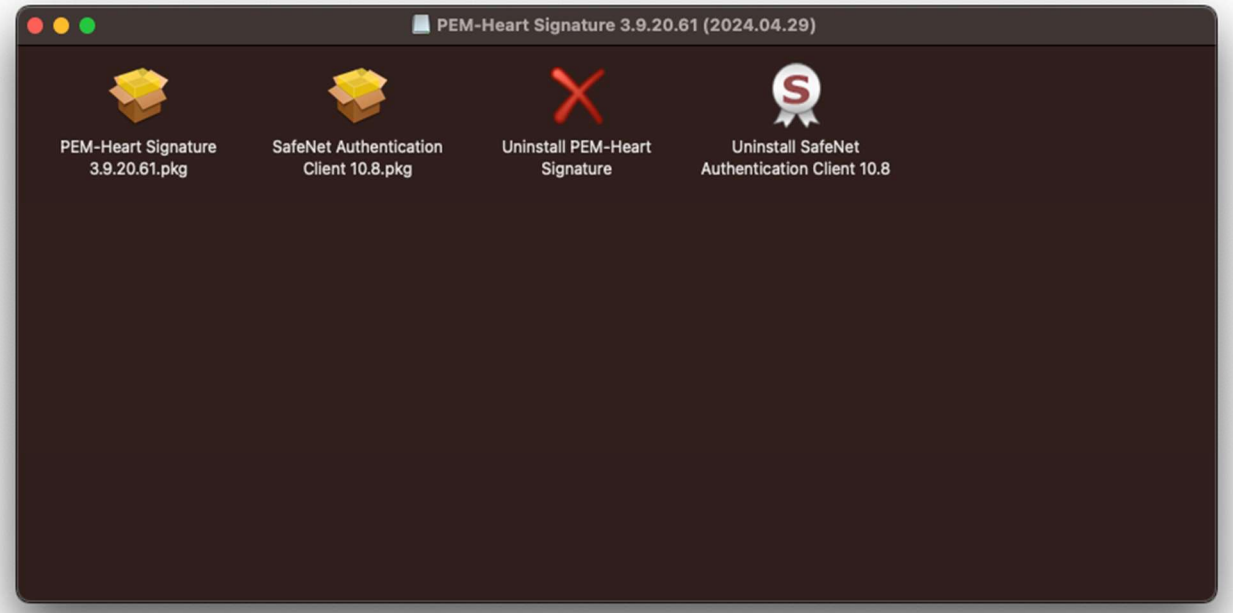


Figure 9 Pem-Heart package for macOS

3.2.1 INSTALLATION VIA THE FILE MANAGER

The installation should be carried out from an account with administrator privileges. It is recommended that all applications other than those necessary for the operation of the operating system are terminated before the installation begins.

INTEL and ARM processor architecture versions are available.

In the Finder file manager, locate the place in the file structure with the PEM-HEART Signature installation file. Run the file, this will initiate the installer.

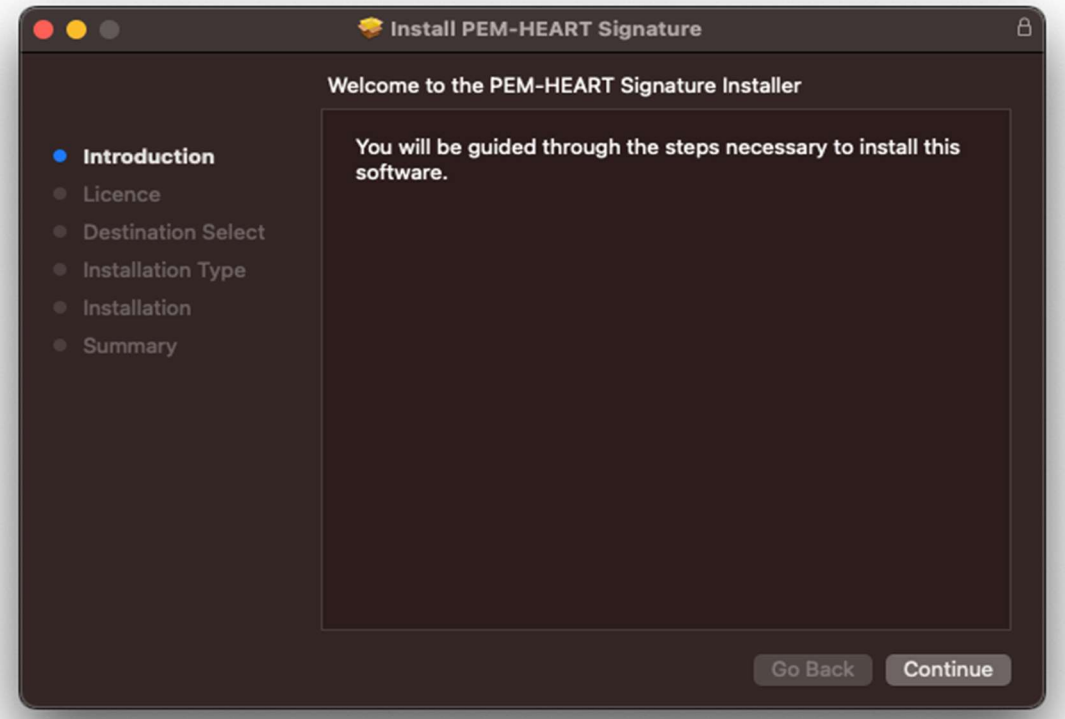


Figure 10 Pem-Heart package installer - Startup window

1. In the first installation step, the user must accept the licence agreement.

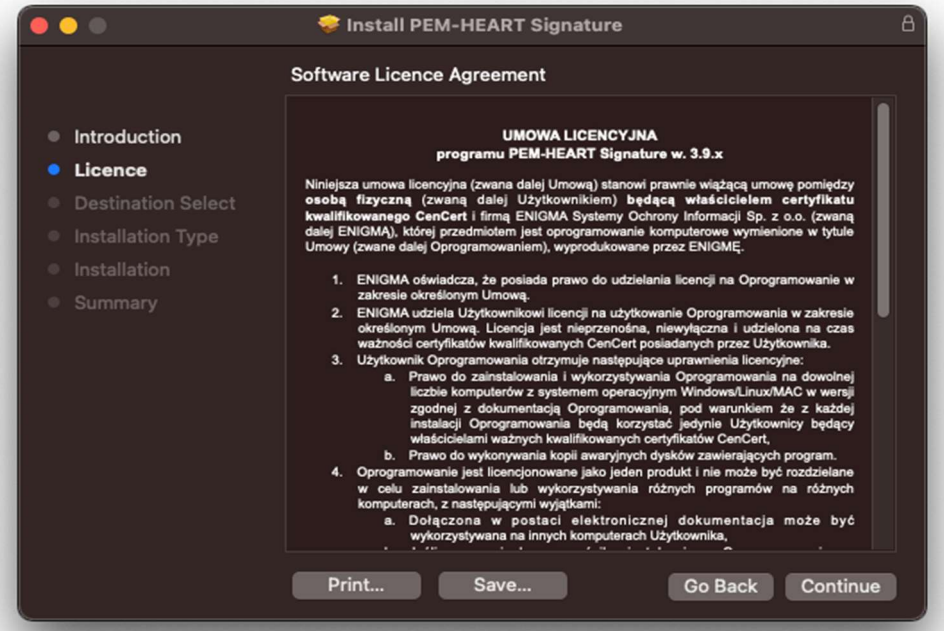


Figure 11 Pem-Heart package installer - licence agreement

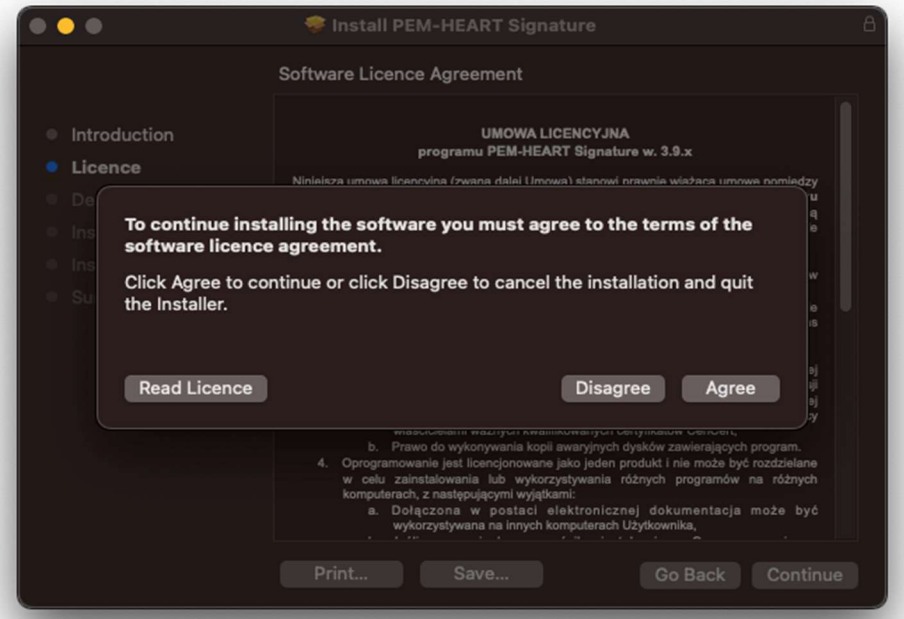


Figure 12 Pem-Heart package installer - acceptance of licence agreement

2. Then confirm your intention to install by clicking on the *Install* button and entering your user account password - the installation process will begin. At this point, it is also possible to change the installation destination by clicking on "Change Install Location...".

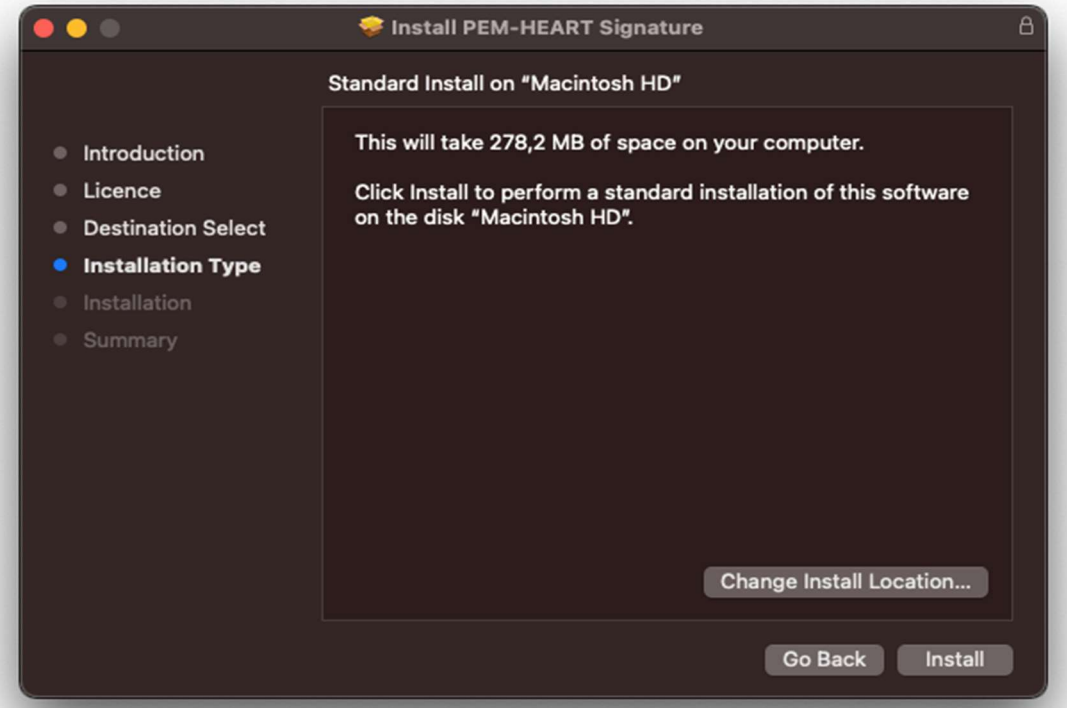


Figure 13 Pem-Heart package installer - installation information

3. Once the installation process is complete, a summary screen will be displayed.

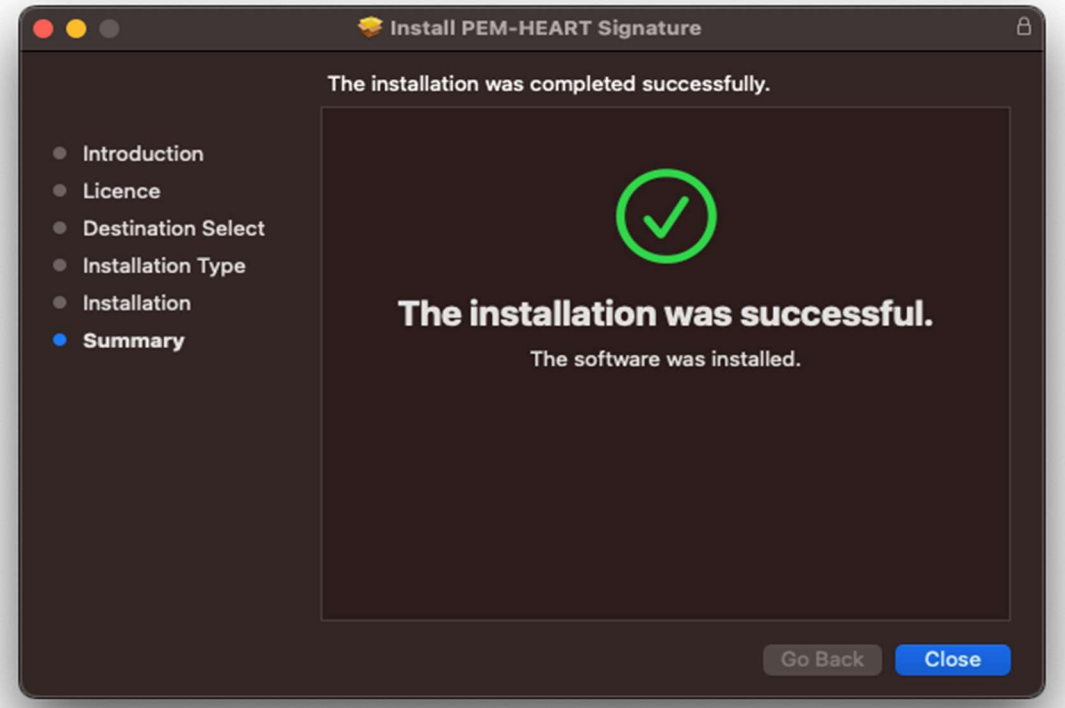


Figure 14 Pem-Heart package installer - installation summary

3.2.2 INSTALLATION OF THE SAFENET CLIENT

Thales' SafeNet program supports IDPrime cards.

The installation should be carried out from an account with administrator privileges. It is recommended that all applications other than those necessary for the operation of the operating system are terminated before the installation begins.

In the Finder file manager, locate the place in the file structure with the SafeNet Authentication Client installation file. Running the file will initiate the installer.

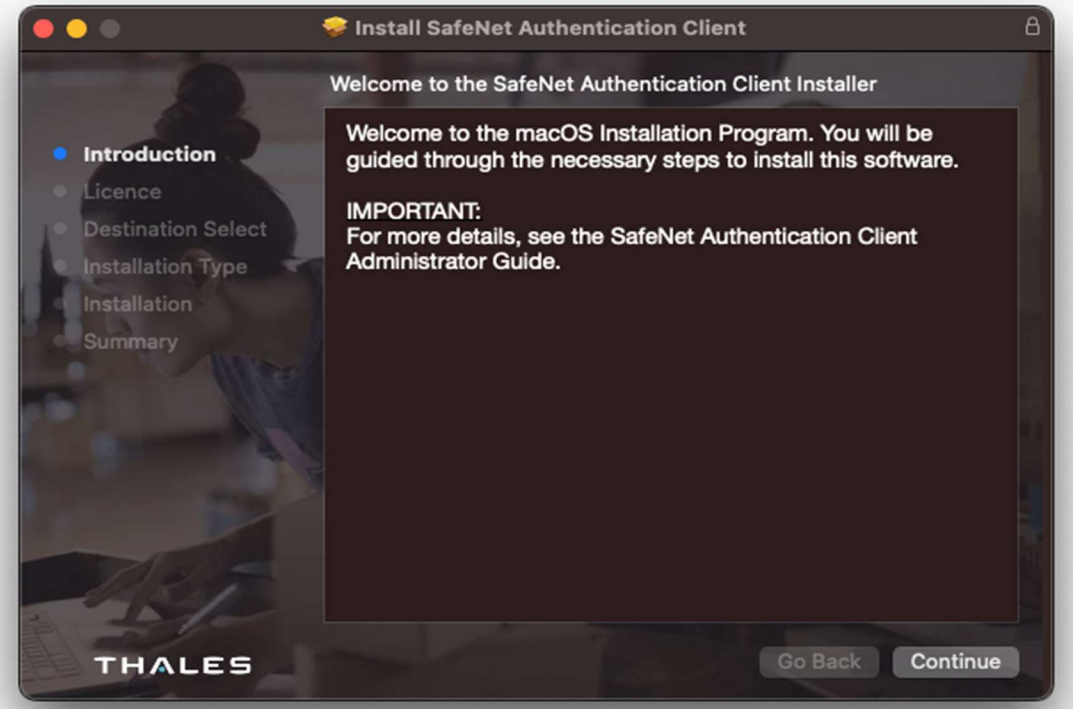


Figure 15 SafeNet Authentication Client package installer - Startup window

1. In the first installation step, the user must accept the licence agreement.

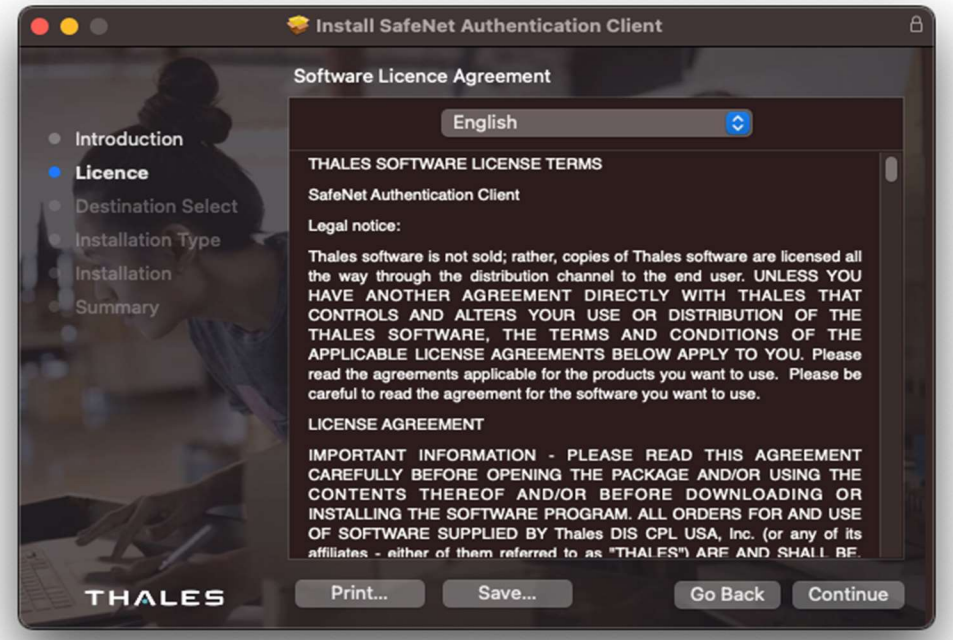


Figure 16 SafeNet Authentication Client package installer - licence agreement

Figure 17 SafeNet Authentication Client package installer - acceptance of licence agreement

2. Then confirm your intention to install by clicking on the *Install* button and entering your user account password - the installation process will begin. At this point, it is also possible to change the installation destination by clicking on "Change Install Location...".

Figure 18 SafeNet Authentication Client package installer - installation information

3. Once the installation process is complete, a summary screen will be displayed.

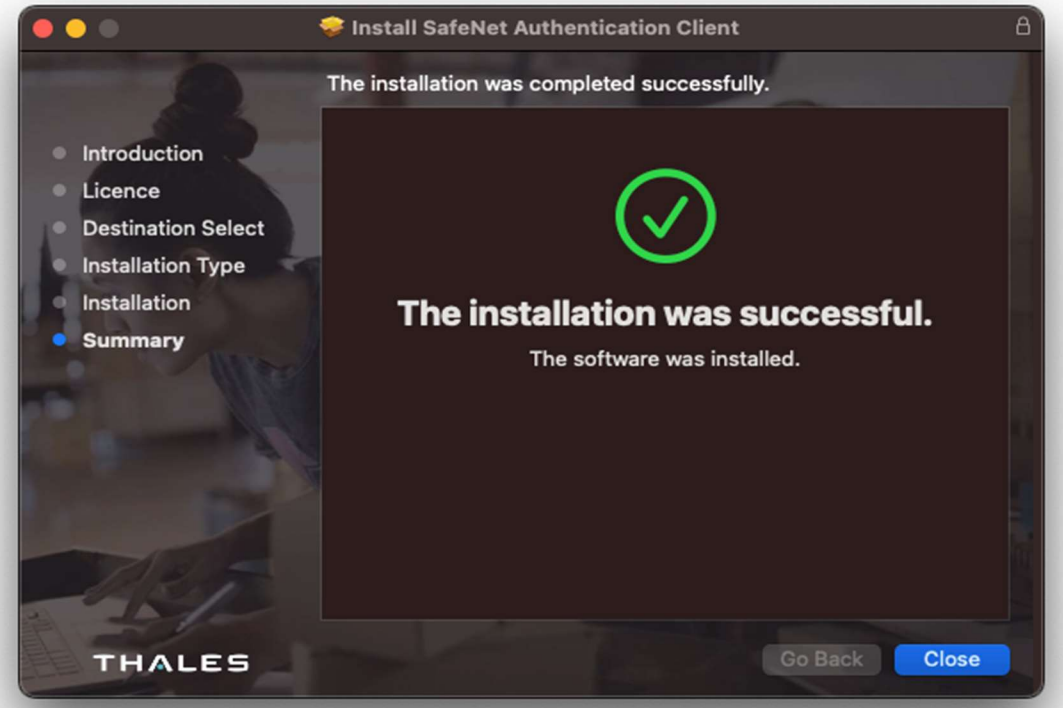


Figure 19 SafeNet Authentication Client package installer - installation summary

3.2.3 PROGRAM DELETION

The uninstallation is performed by running the *Uninstall PEM-Heart Signature* program. Dialog boxes will be displayed asking you to accept the removal:

- PEM-HEART application with individual software components,

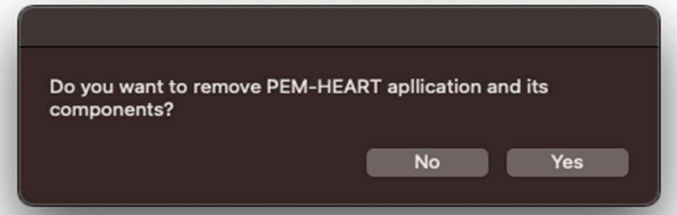


Figure 20 Uninstallation message to remove Pem-Heart application

- PEM-HEART configuration from the *opt*, *etc.* and home directories,

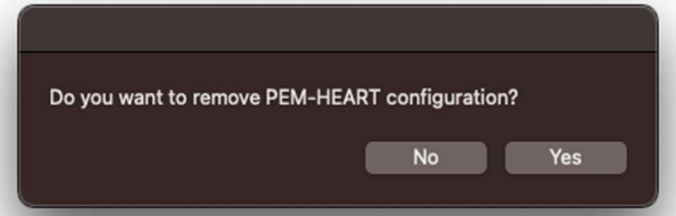


Figure 21 Deinstallation message to remove Pem-Heart configuration

- rSign configuration files (enigmaCloud.ini).

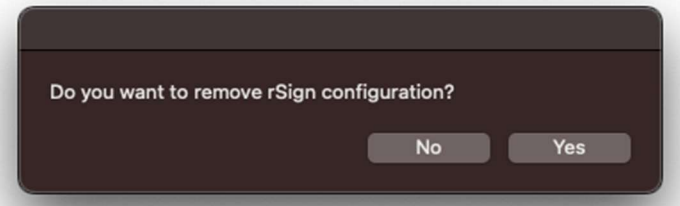


Figure 22 Uninstallation message to remove rSign configuration

At the end of the process, a window confirming the uninstallation will be displayed:

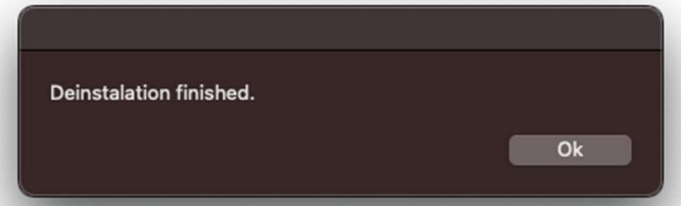


Figure 23 Confirmation of the uninstallation of the program

3.2.4 REMOVAL OF SAFENET CLIENT

Removal of the Thales card handling software is done from the Control Panel, under *Programs and Features,* where the program must be indicated from the available list and the *Uninstall* or *Uninstall/Change option selected*.

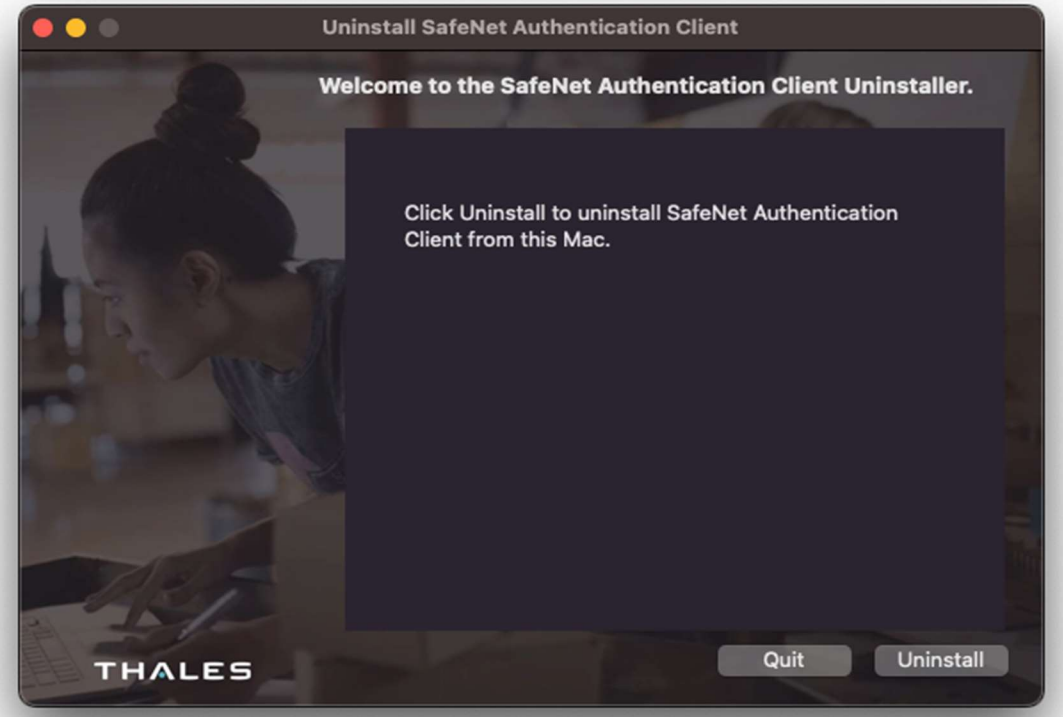


Figure 24 SafeNet Authentication Client uninstaller - Startup window

In the uninstall wizard window, click *Uninstall* and enter the password. A confirmation of the process will be displayed.

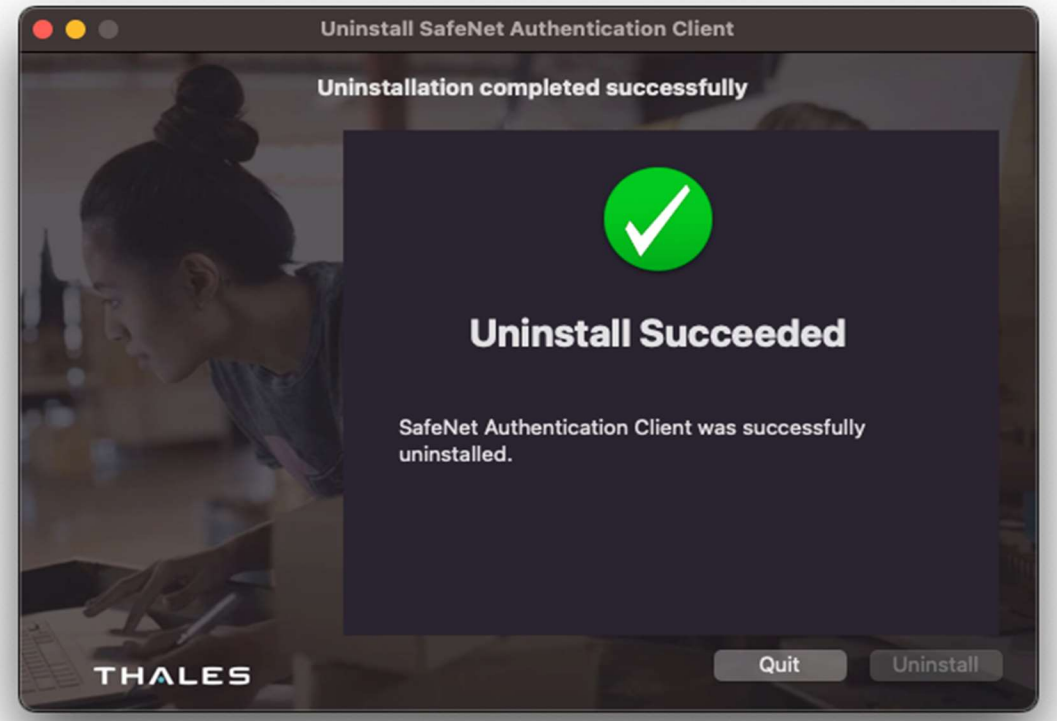


Figure 25 The SafeNet Authentication Client uninstaller - a summary of the uninstallation process

3.3 INSTALLATION FOR LINUX

The installation should be carried out from an account with administrator privileges. It is recommended that all applications other than those necessary for the operation of the operating system are terminated before the installation begins.

The following installation procedure is shown using Ubuntu 20.04 LTS as an example, in the system package manager (e.g. Ubuntu Software) and via the command line terminal.

The package installation operation must be preceded by installing the required packages: pcscd and libncurses5. This can be done using the commands:

sudo apt-get install pcscd

sudo apt-get install libncurses5

3.3.1 INSTALLATION VIA THE FILE MANAGER

In the manager, locate where the PEM-HEART Signature installation files are located in the file structure. The installation is performed by running (double-click) the respective file. The default associated package manager will then be opened. When the *Install* button is pressed, PEM-HEART Signature is installed on the system.

Once the installation is complete, a corresponding message will appear and the installer window can be closed. Finally, the Safenet Authentication Client must be installed to enable the use of all available cards. The installation file can be downloaded from the cencert.co.uk website.

During installation, the Safenet Authentication Client requires the libgdk-pixbuf2.0-0 package to be installed on the system. Once installed, the program can be accessed via the file context menu or the system's program menu.

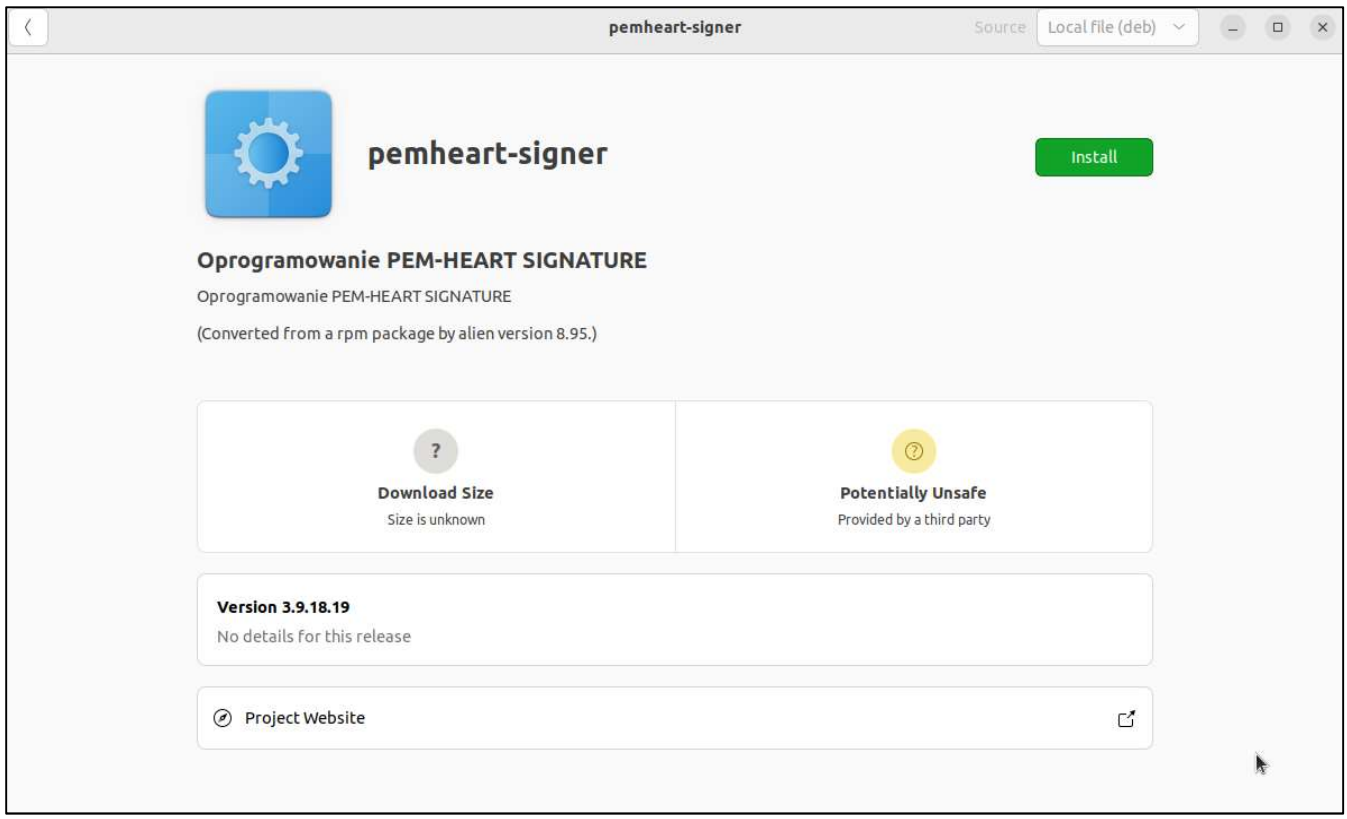


Figure 26 Installation of a Linux program via the file manager

To install, click on the green *Install* button in the top right corner.

3.3.2 INSTALLATION VIA THE COMMAND LINE

The option to install PEM-HEART Signature via the command line is shown below.

After launching the terminal window, locate where the installation files are located in the file structure. The software is distributed as an installation package (.deb file). To install the "PEM-HEART Signature" package on an Ubuntu system (here version 20.4 LTS), call the command:

sudo dpkg -i PH-3.9.X.X_amd64.deb

where X.X is the number of the released software version.

3.3.3 SOFTWARE REMOVAL

A command-line terminal can be used to remove the software from the system, or the package manager can be run.

- Deletion via the command line

The uninstallation of PEM-HEART Signature is performed using two console programs:

sudo apt-get purge pemheart-signer or

sudo apt remove pemheart-signer

- Deletion via the file manager

After launching the package manager (the example uses Ubuntu Software), find pemheart-signer in the "Installed" tab and then click the red bin (***Figure 27 Uninstallation of a Linux program via the file manager***).

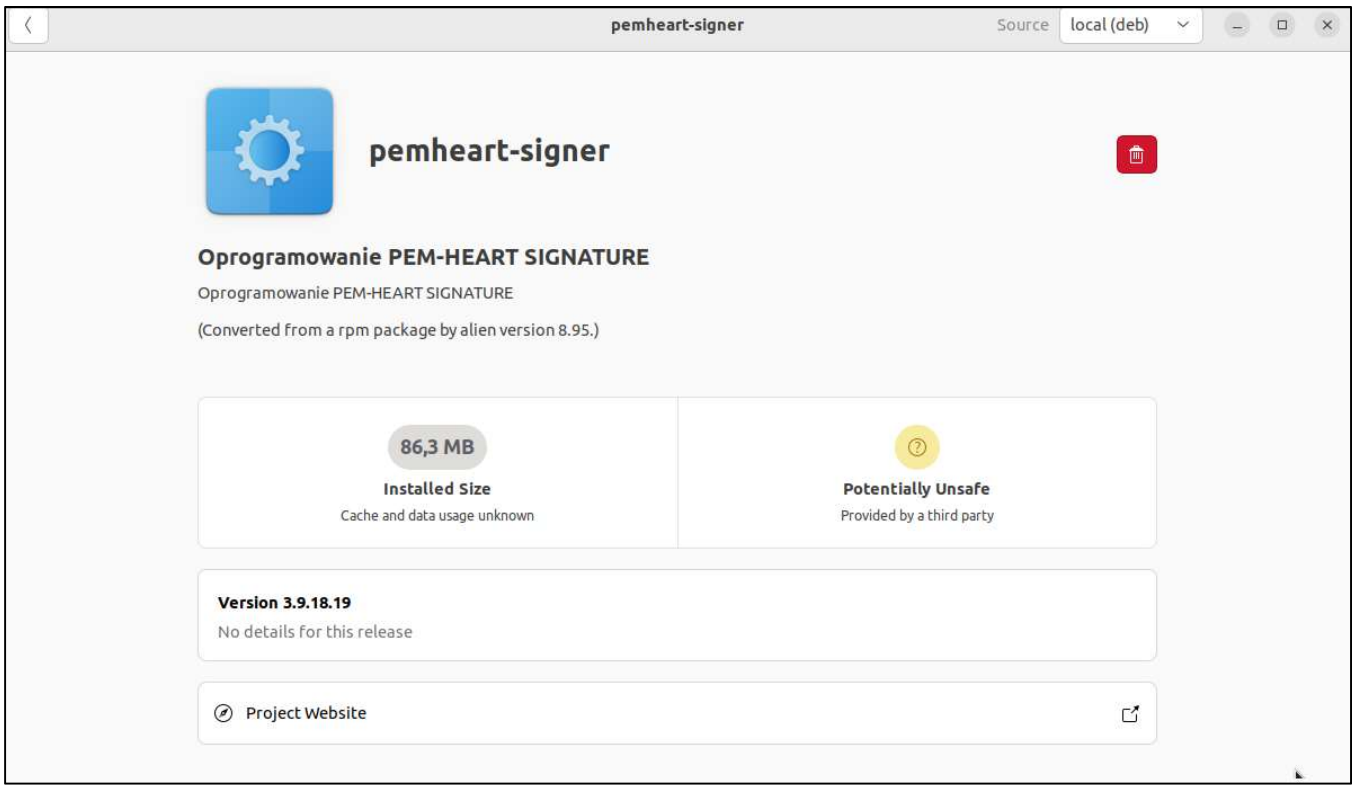


Figure 27 Uninstallation of a Linux program via the file manager

4 FILE OPERATIONS

The user can perform certain operations without directly launching the program - by right-clicking (PPM) on the file. Various functions are available from the context menu, including signature creation and signature verification. Depending on the type of file, the choice of options may vary.

Figure 28 Examples of PPM functions for a PDF file

4.1 SIGNING - SIGNATURE ON A CARD OR USB TOKEN

To sign, insert your Cencert card (into the usb token reader into the USB port), then right-click (PPM) on the file to be signed to expand the context menu - select *PEM-HEART Signature -> Sign* (for many applications, the option may be under *Show more options*). The operation is also carried out from within the running PEM-HEART Signature application (for a description of the steps see **5.2.1 Signing - signature on a card or USB token, pp. 37**)

The program will automatically select the recommended signature format and then ask for the PIN for the card.

Notes:

- Advanced options, such as changing the signature format, signature in a separate file, timestamping and other settings, are available under the *Options* button. Settings changed in this way refer to a specific signature and are not saved for later use. See also chapter **8.1 Changing the signing parameters, p. 59**.
- Depending on the format of the signature, it will be saved in the same file with no name change or in a new file with a changed extension.

- If 'signature in separate file' is selected, the signature will be saved in a separate file. Selecting this option requires the subsequent transfer of two files to the recipient: the original file and the signature.
- If the signature is to include a timestamp and/or OCSP response, an Internet connection is required at the time of signing. You may also need to subscribe to a timestamping service.

4.2 SIGNING - RSIGN (CLOUD SIGNATURE)

To sign, right-click (PPM) on the file to be signed to expand the context menu - select *PEM-HEART Signature -> Sign* (for many applications, the option may be under *Show more options*). The operation is also carried out from within the running PEM-HEART Signature application (for a description of the steps see **5.2.2 Signature creation - rSign (cloud signature), pp.40**)

The program will automatically select the recommended signature format. Then click *Next* and enter the signature PIN (the user will be prompted to enter it). In the next step, you are required to launch the *rSign by Cencert* app on your mobile device, from which you need to read the code "ACTIVE *SIGN* PIN" from the screen - this must be entered into the app on your computer, then click OK. In the next step, you need to confirm your intention to sign in the rSign app, after which the program will execute the signature.

Note: We recommend that you set the *Settings -> PIN -> Remember PIN* option in the application *to a specific time*, with the time set to 3 minutes. This will allow you to make a time-stamped signature or even multiple signatures (if you have indicated multiple files to be signed in the application) without having to validate each signature operation on your phone. If the signature is set to "Ask for PIN every time", performing a time-stamped signature will require double approval of the signature on the phone (document signature, timestamping request signature).

Notes:

- Advanced options, such as changing the signature format, signature in a separate file, timestamping and other settings, are available under the *Options* button. Settings changed in this way relate to a specific signature and are not saved for later use. See also chapter **8.1 Changing the signing parameters, p. 59**.
- Depending on the format of the signature, the signature will be saved in the same file with no name change or in a new file with a changed extension.

- If 'signature in separate file' is selected, the signature will be saved in a separate file. Selecting this option requires the subsequent transmission of two files to the recipient: the original file and the signature.
- Internet connection is required at the time of signing.

4.3 VERIFICATION OF SIGNATURE

To verify the signature, right-click (PPM) on the file to be signed and then select *PEM-HEART Signature -> Verify* (for many applications, the option may be under *Show more options*). The signature verification window will then be displayed. Then click the Verify button. The program will verify the signatures stored in the document and display the result of the verification. The operation is also carried out from within the running PEM-HEART Signature application (for a description of the operation see **5.3 Verification of the signature in the program, pp. 44**)

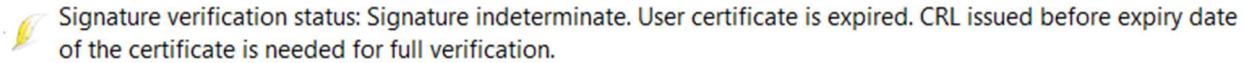
If the signature has been timestamped - the moment for which the signature is verified is taken from the timestamp (possible subsequent revocation of the certificate will not affect the result of the verification of such signature).

If the signature does not have a timestamp - the signature is verified for the current moment or for another moment entered manually into the program ("verify for the given time: ..."). In the case of manual entry of the moment for which the signature is verified, the responsibility for the correctness of the time (and possibly provable) lies entirely with the user.

The verification result is marked with coloured symbols to clearly distinguish it:

- Green indicates correct verification of the signature.

- Yellow indicates incomplete verification - the signature is mathematically correct, but it is not yet possible to confirm whether the certificate was valid at the time of signing. In this case, verification should be repeated later - e.g. in a few hours or the next day.



- Red indicates a signature verification failure (e.g. a mathematical inconsistency, i.e. the integrity of the document is compromised, or the certificate is found to be invalid).

4.3.1 VERIFICATION PANEL

Once the signature has been verified, various additional actions are available in the top menu, including:

- *Present document* - display the original (signed) document if a program for displaying the type of document is installed in the system.
- *Open directory* - opens a view of the directory on the drive where the document is saved.
- *Signature attributes* - showing additional data attached to the signature.
- *Show certificate* - displaying the certificate with the data of the person who signed the document. It is also possible to export the certificate in *.crt* format via the *Export* button.
- *Show xml report* - show the xml report in the program window.
- *Create PDF report* - save to disk a readable report (in PDF format) confirming signature verification.
- *Show help - the* user manual pdf will be displayed.

Once the signature has been correctly verified, advanced forms of signature can be created:

- *Create an archive form* - securing the ability to correctly verify the signature for the validity period of the timestamp (practically about 7-10 years). Creating an archive form requires access to the Internet and downloading of, among other things, two timestamps. It may be necessary to purchase a timestamp package.

The validity of the archival form of the signature can be extended any number of times (by adding another timestamp), each time for a further 7-10 years.

- *Create a long form* - securing the ability to correctly verify the signature for the validity period of the OCSP and timestamp (practically approx. 5-10 years). Creating a *long form* requires access to the Internet and downloading a timestamp, among other things. It may be necessary to purchase a timestamp package.

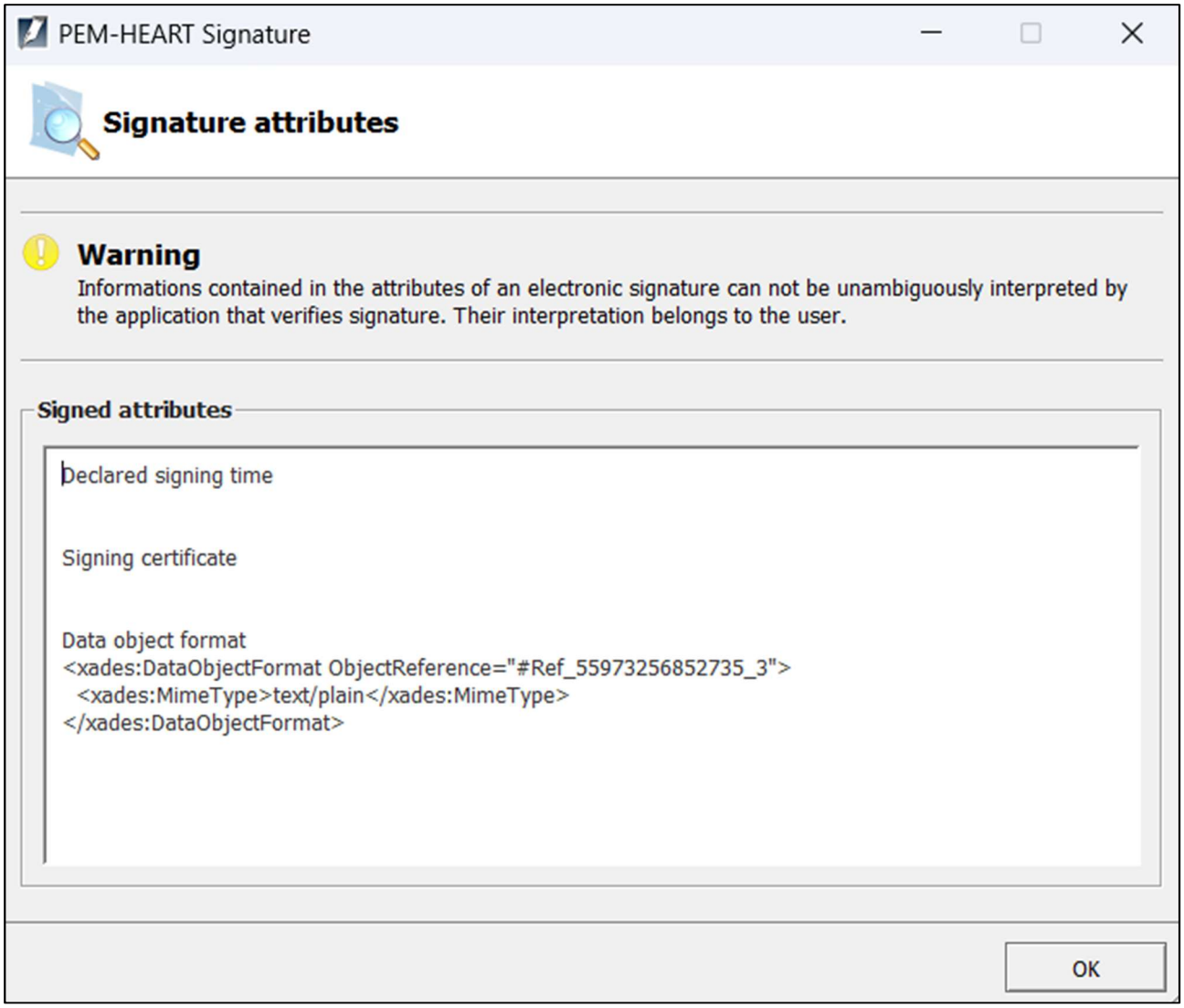


Figure 29 Window showing signature attributes

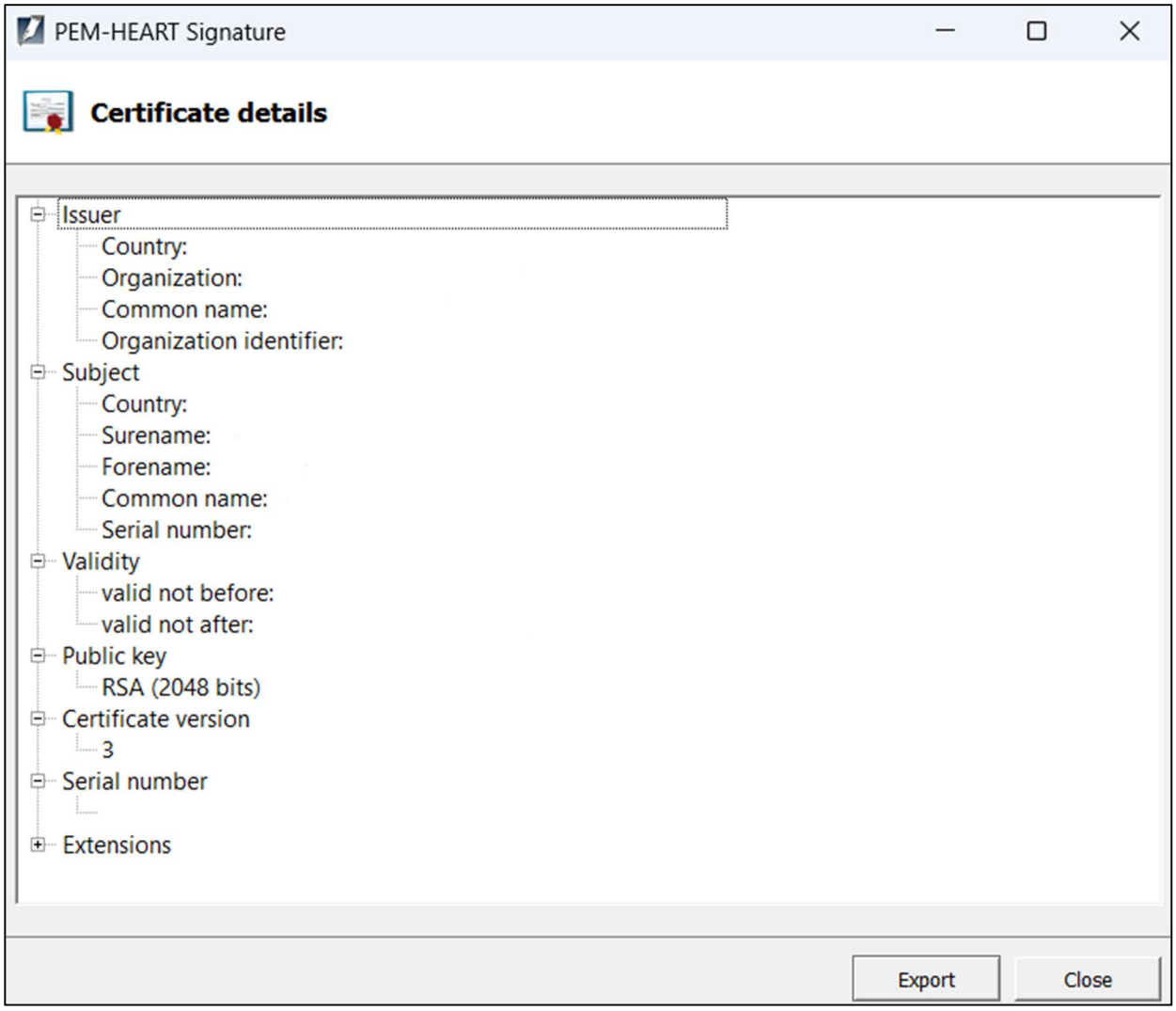


Figure 30 Window showing certificate details

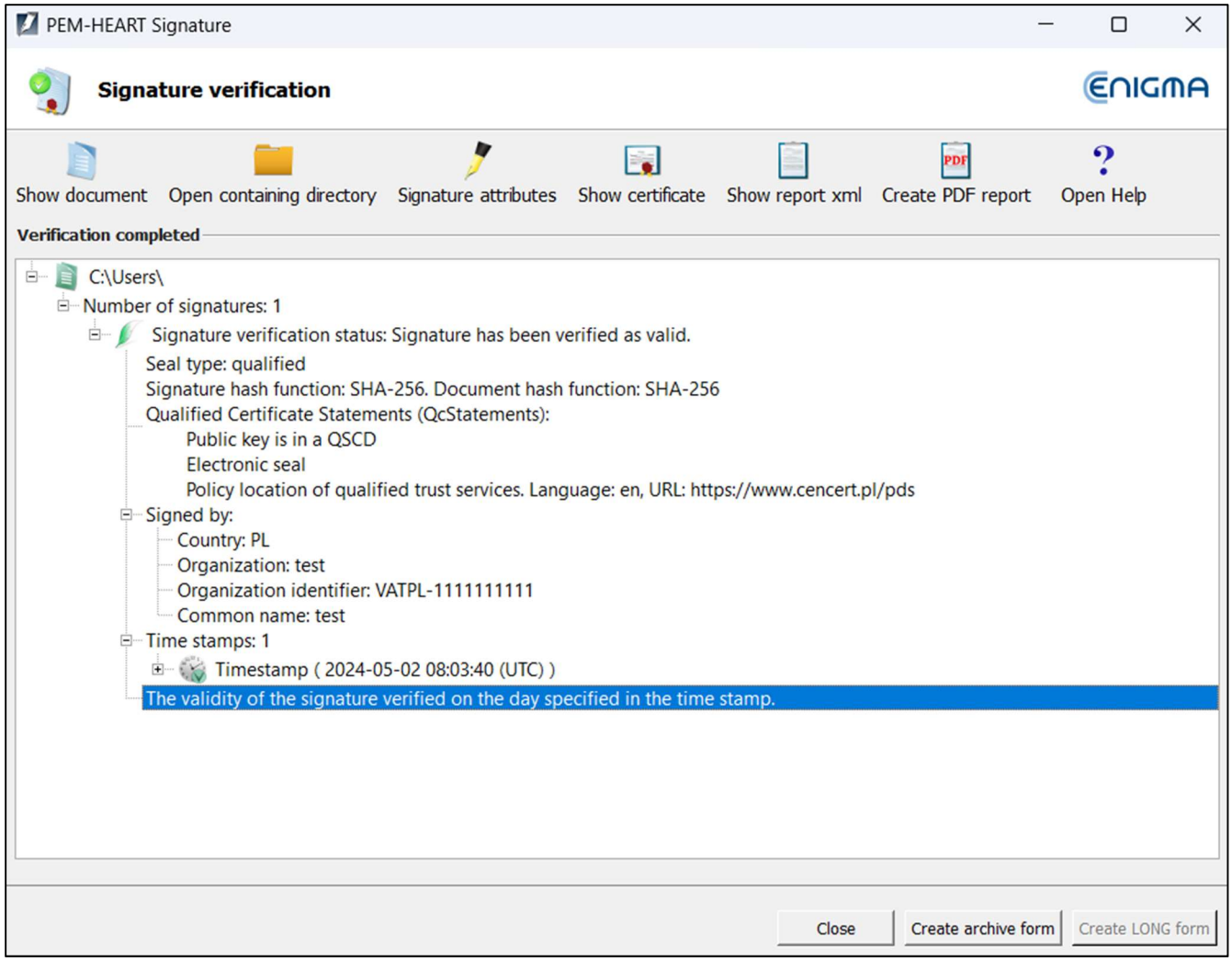


Figure 31 Signature status after verification

5 BASIC FUNCTIONS

5.1 PROGRAM START-UP

All functions of the program are available by launching *PEM-HEART Signature* from the *Start* menu (Windows) or from the icon on the desktop. On other operating systems, the program must be started in the manner appropriate to the system. The appearance of the program is the same as in Windows.

5.2 SIGNING IN THE PROGRAM

5.2.1 SIGNING - SIGNATURE ON A CARD OR USB TOKEN

To sign after starting the program, click the *Sign* icon (on the left-hand side of the window, in the *Basic Functions* panel). This will display a window allowing the selection of files to be signed. Here, you have the option of adding the file or files to be signed (*Add File* button) or dragging and dropping the file into the file list window. If an entire directory is indicated (*Add Directory button)*, all files from that directory and its subdirectories will be inserted in the list of *files to be signed*. Once all files have been added to the signature, click the *Next* button. If there is one certificate reader connected to the operating system, the program will ask for the PIN for the card. If there are more readers with certificates - the program will show a window with the selection of the token. After a correct operation, the signature will be executed.

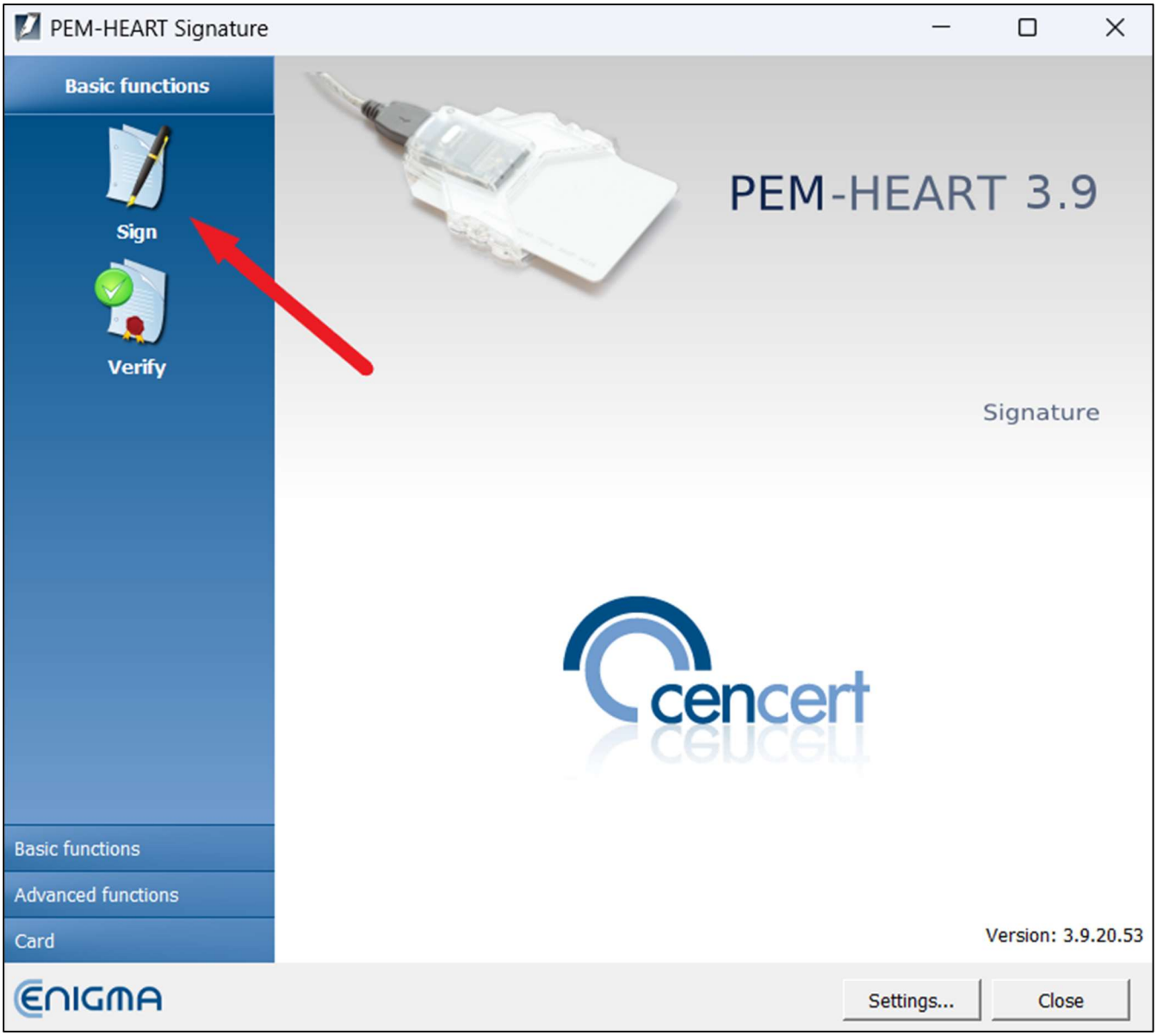


Figure 32 Main menu of the PEM-HEART Signature application - selection of the "Sign" option

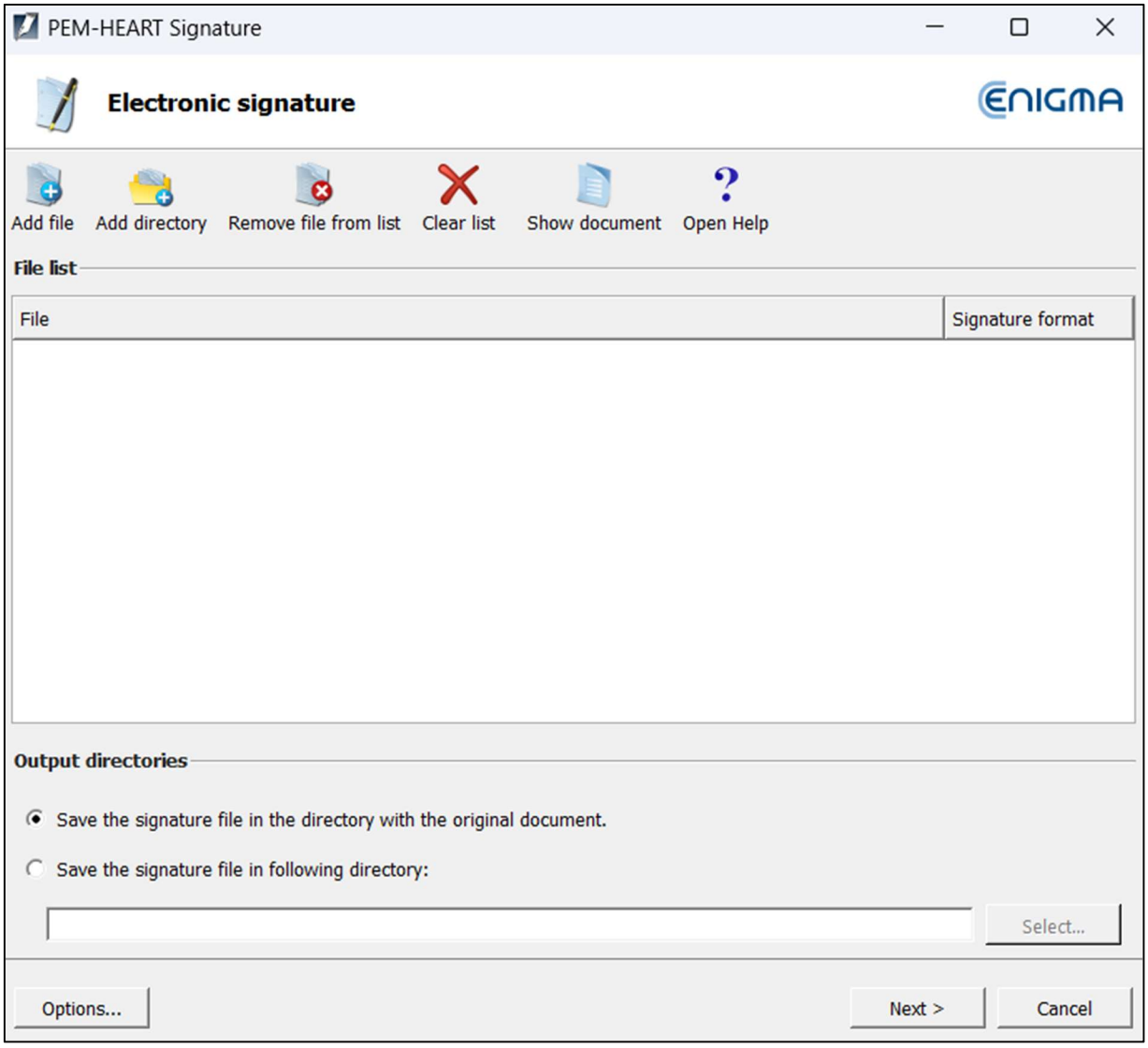


Figure 33 Window for affixing the electronic signature

Notes:

1) Advanced options, such as changing the signature format, signature in a separate file, timestamping and other settings, are available under the *Options* button. Settings changed in this way relate to a specific signature and are not saved for later use. See also chapter **8.1 Changing the signing parameters, p. 59**.

2) Depending on the format of the signature, the signature will be saved in the same file with no name change or in a new file with a changed extension.

3) If 'signature in separate file' is selected, the signature will be saved in a separate file. In this case, the recipient must be provided with two files: the original file and the signature file.

4) If the signature is to include a timestamp and/or OCSP response, an Internet connection is required at the time of signing. You may also need to subscribe to a timestamping service.

5.2.2 SIGNATURE CREATION - RSIGN (CLOUD SIGNATURE)

To sign rSign, after starting the program, click the *Sign* icon (on the left-hand side of the window, in the *Basic Functions* panel). This will display a window allowing you to indicate the files to be signed. Here you have the option of adding the file or files to be signed (*Add File* button) or dragging the file into the file list window. If an entire directory is indicated (*Add Directory button)*, all files in the directory and its subdirectories will be inserted in the list of *files to be signed*. Once all files have been added to the signature, click the *Next* button. Once all files have been added to the signature, press the Next button. If there is one certificate reader connected to the operating system, the program will ask for the signature PIN. If there are more readers with certificates - the program will show a token selection window. After a correct operation the signature will be executed.

Figure 34 Message regarding the use of the rSign application on the phone

Now start the *rSign by Cencert* mobile app, then read *the Active Signature PIN and* transcribe it into the program on the computer and confirm *OK*. Confirm your intention to sign in the rSign app.

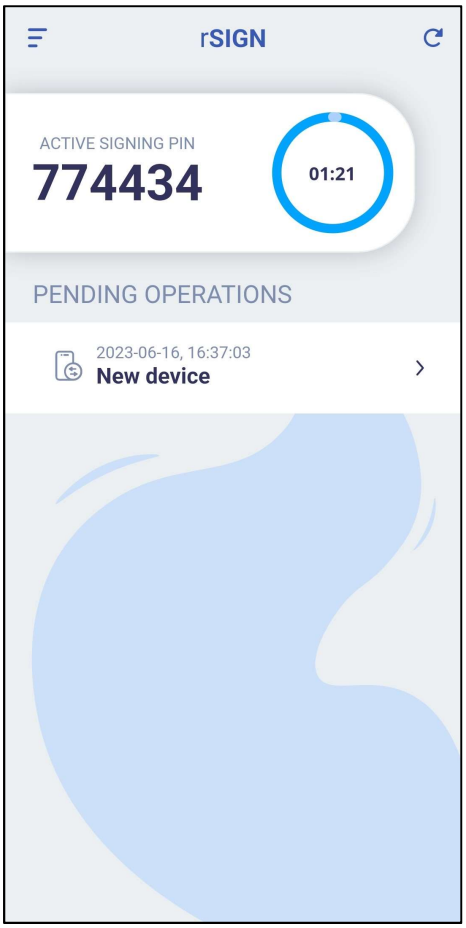


Figure 35 Screen of the rSign application with Active Signature Pin

In the application, you must validate your desire to sign by clicking on the Signature *Validation* button in the *LOOKING FOR OPERATIONS* section just below the displayed Active Signature Pin (Figure 35 Screen of the rSign application with Active Signature Pin). If the data are correct, the next step is to confirm the operation by clicking *CONFIRM* (**Figure 36 Approval of the execution of an electronic signature operation**)

Figure 36 Approval of the execution of an electronic signature operation

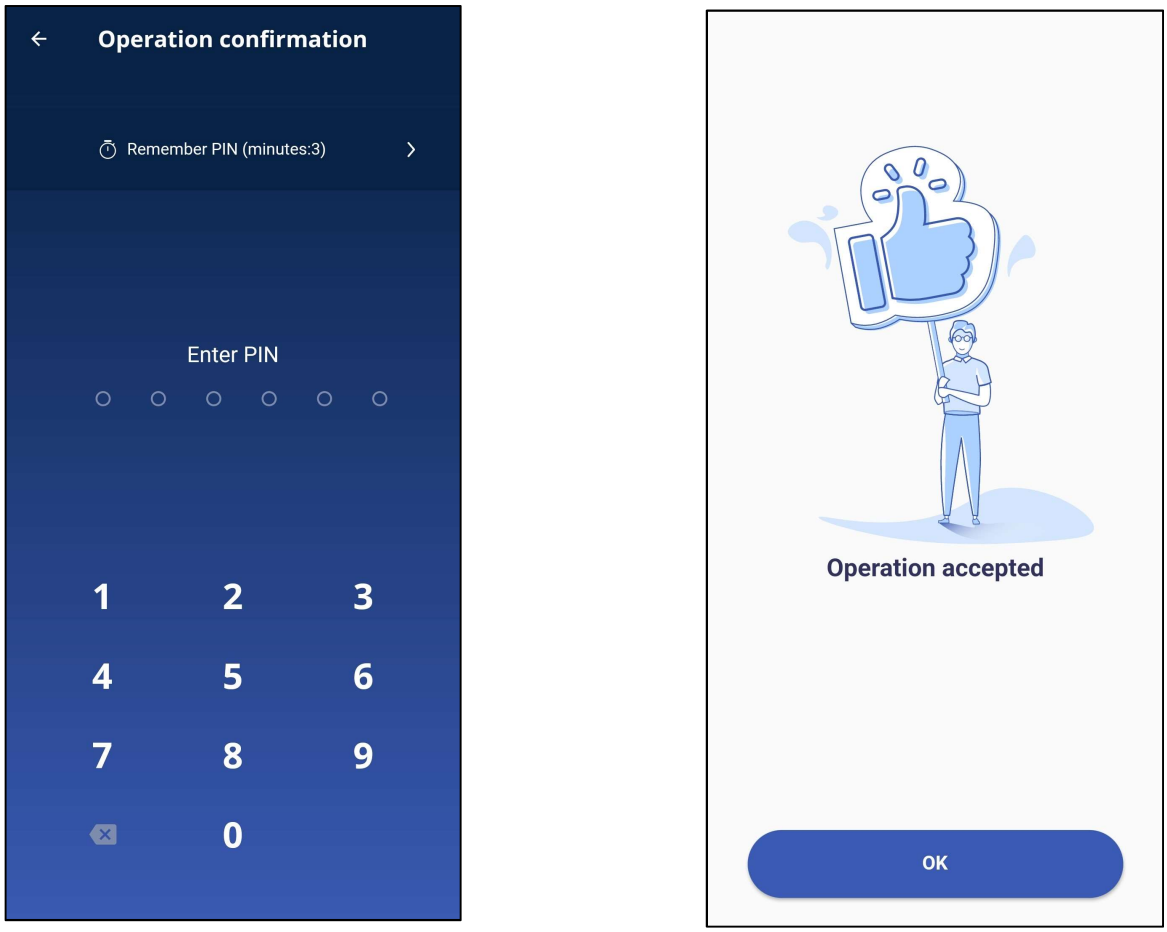


Figure 37 PIN code entry window and confirmation of the signature operation (telephone)

A screen will be displayed for entering the PIN code, after which you will need to wait for confirmation of the operation - if the code has been entered correctly, the procedure will be accepted and a window will appear as follows **Figure 37 PIN code entry window and confirmation of the signature operation**.

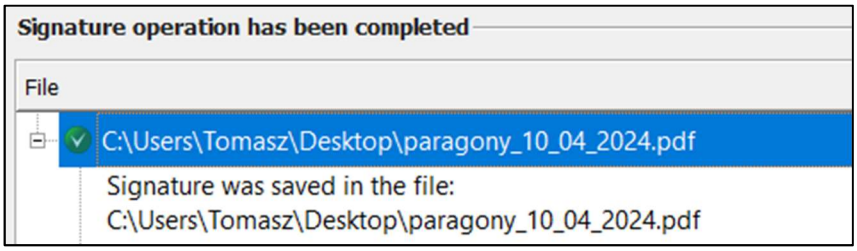


Figure 38 Confirmation of signature operation (computer)

The program on the computer will display information that the document has been correctly signed (**Figure 38 Confirmation of signature operation (computer)**).

5.3 VERIFICATION OF THE SIGNATURE IN THE PROGRAM

To verify the signature, after starting the program, click the Verify icon (on the left-hand side of the window, in the *Basic Functions* panel).

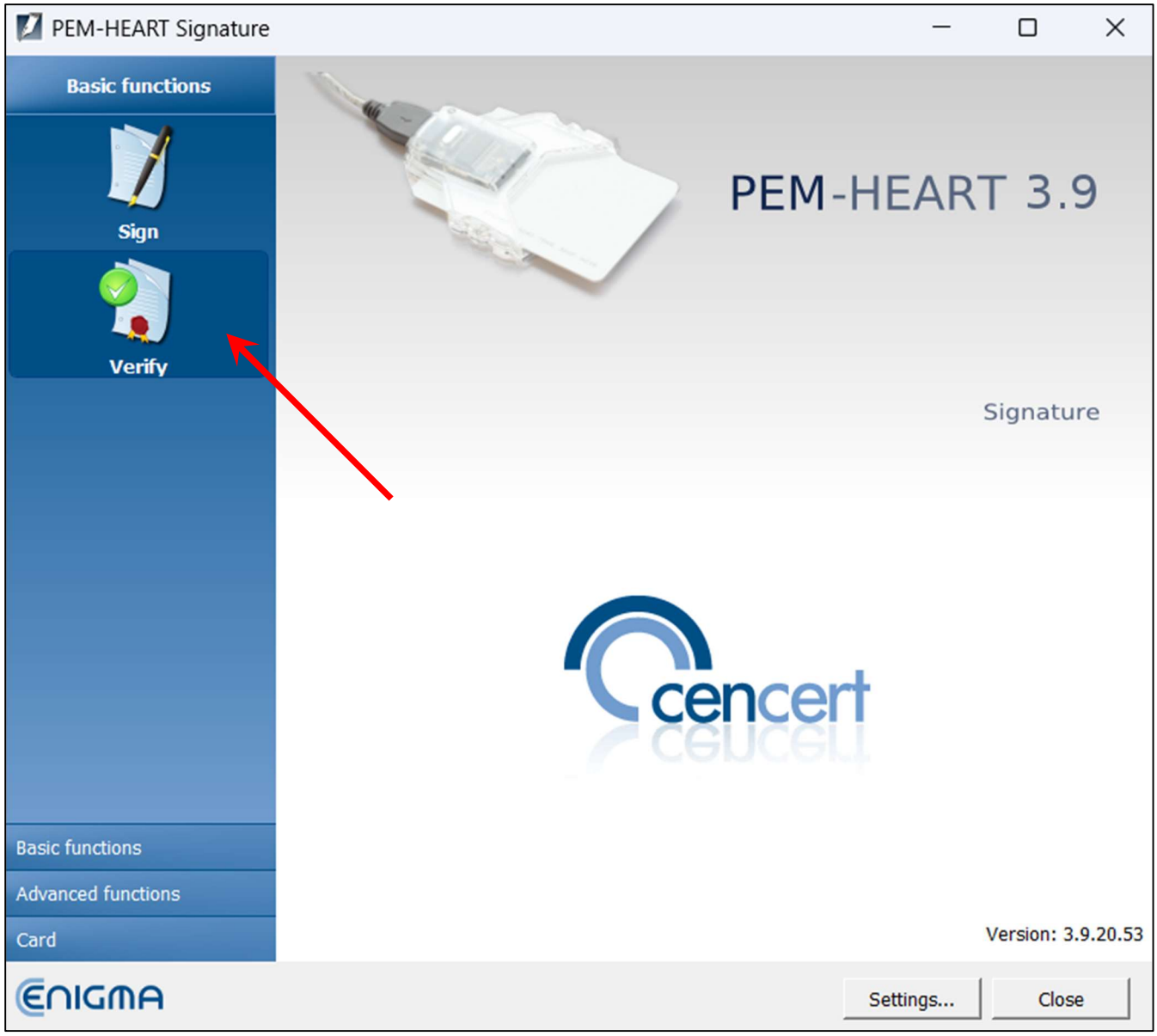


Figure 39 Main menu of PEM-HEART Signature application - selection of "Verify" option

This will bring up a window allowing you to indicate the files to be verified. It is possible to add a single file or multiple files to be checked (*Add File* button) or drag a file into the file list window. If an entire directory is indicated (*Add Directory button)*, the program will insert all files from that directory and its subdirectories into the list of *files* to be checked. When all files have been added, click the *Verify* button.

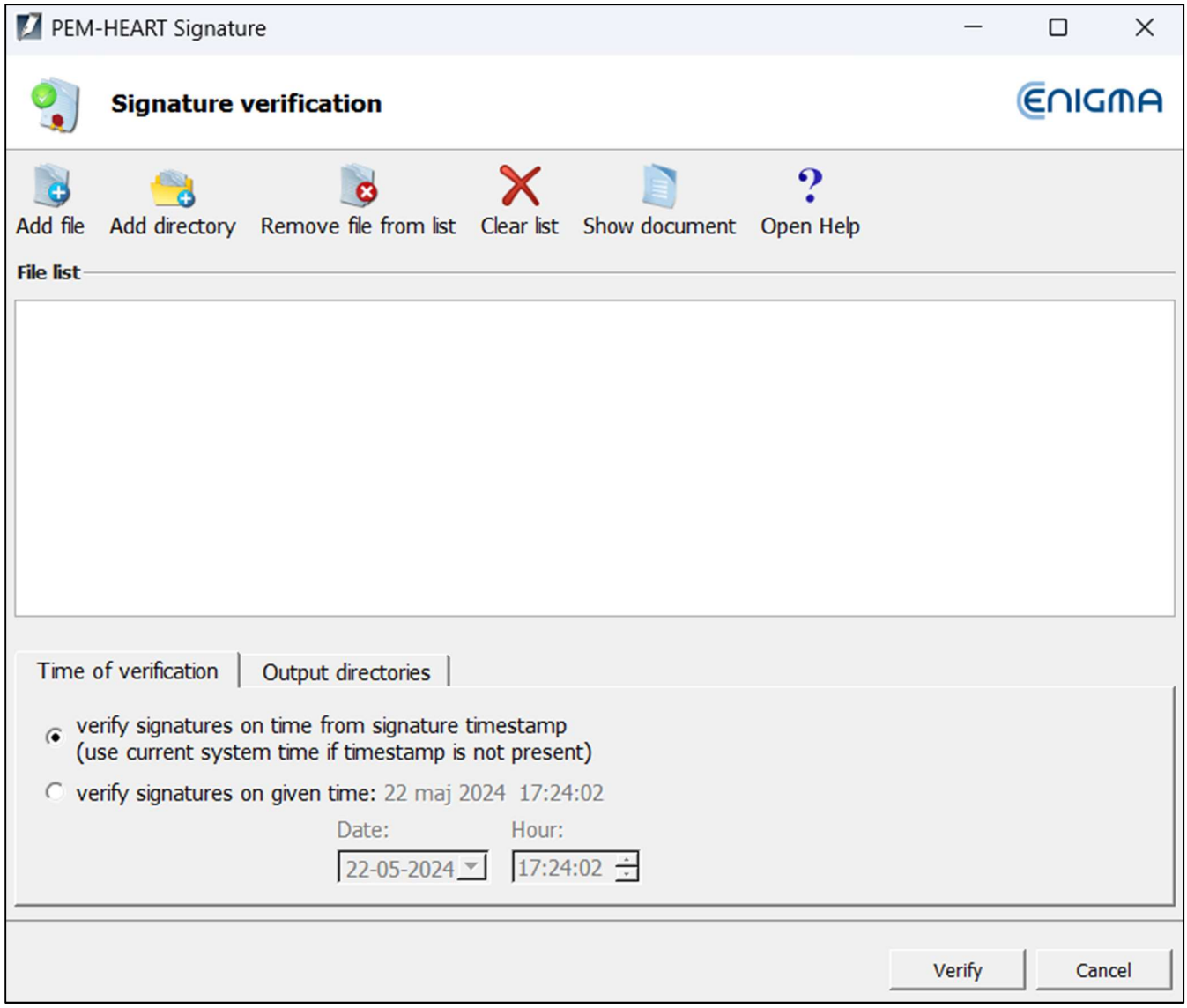


Figure 40 Electronic signature verification window

The program will verify the signatures recorded in the document and display the result of the verification.

For more information on verification, see **4.3 Verification of signature, pp. 31**.

6 ADVANCED FUNCTIONS

Figure 41 Main menu of PEM-HEART Signature application - advanced functions

6.1 COUNTER-SIGNATURE

A countersignature is referred to as a special method of signature execution whereby the signature is technically executed not under the document itself, but under previous signatures (the document is signed indirectly). This implementation of the signature prevents previous signatures from being removed from the document. In the case of standard multiple signatures, it may be technically possible to remove one of the previous signatures from the document, while preserving the validity of the signatures of the others ('countersignature' makes this impossible).

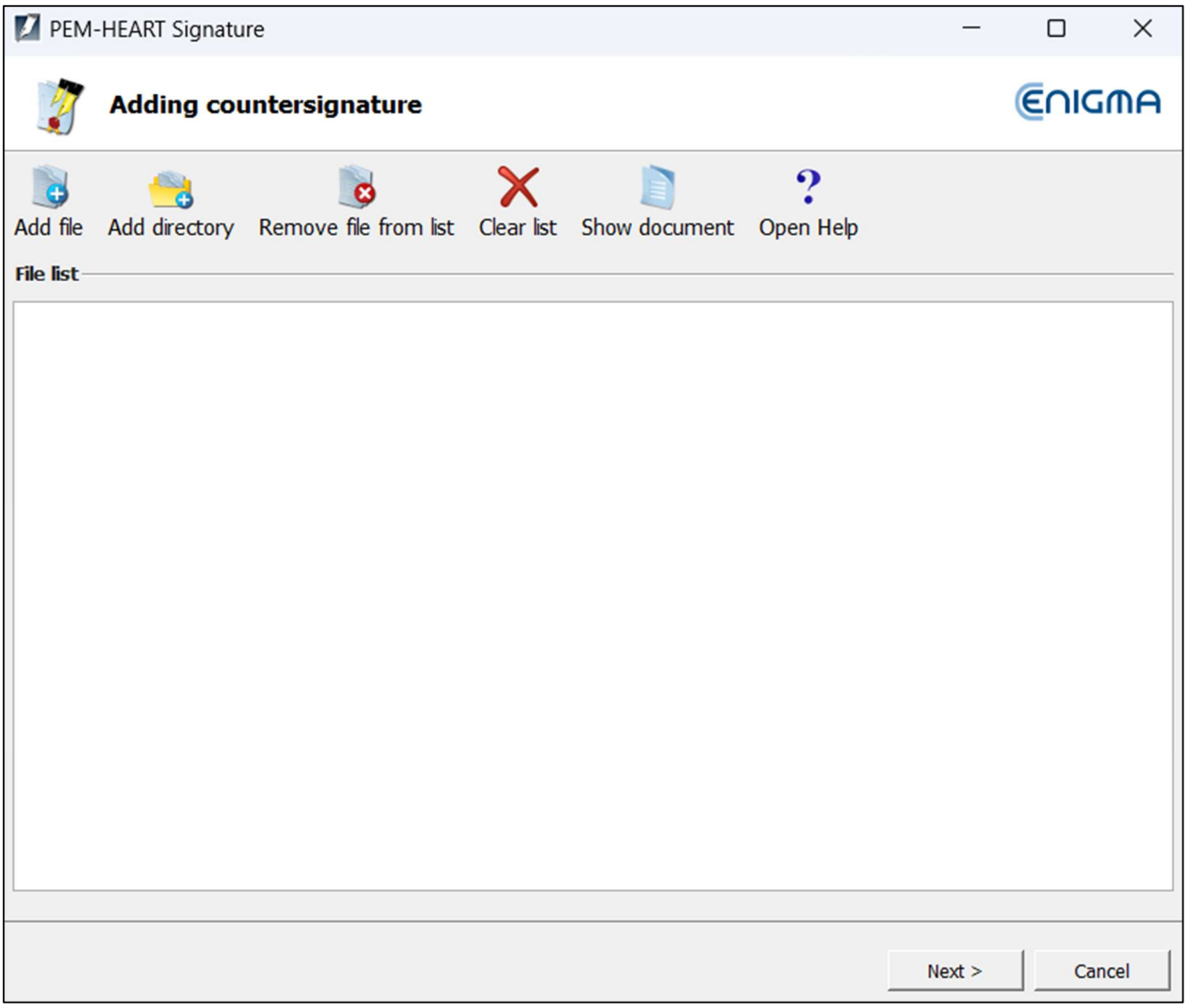


Figure 42 Countersignature window

The term 'countersignature' in the sense described above should not be confused with the same term operating in legal circulation. The creation of an electronic signature as a "countersignature" (in the sense described above) is not mandated by the legal provisions on electronic signatures. The general provisions on electronic signature apply. In the legal sense, the 'countersignature' described herein operates under the same rules as any other electronic signature.

6.2 TIMESTAMPING

A qualified timestamp is evidence of the existence of a document at a particular point in time. In Polish law, a legal act bearing a qualified timestamp has a 'certified date'. Across the EU (under the *eIDAS* Regulation), a qualified electronic timestamp enjoys a

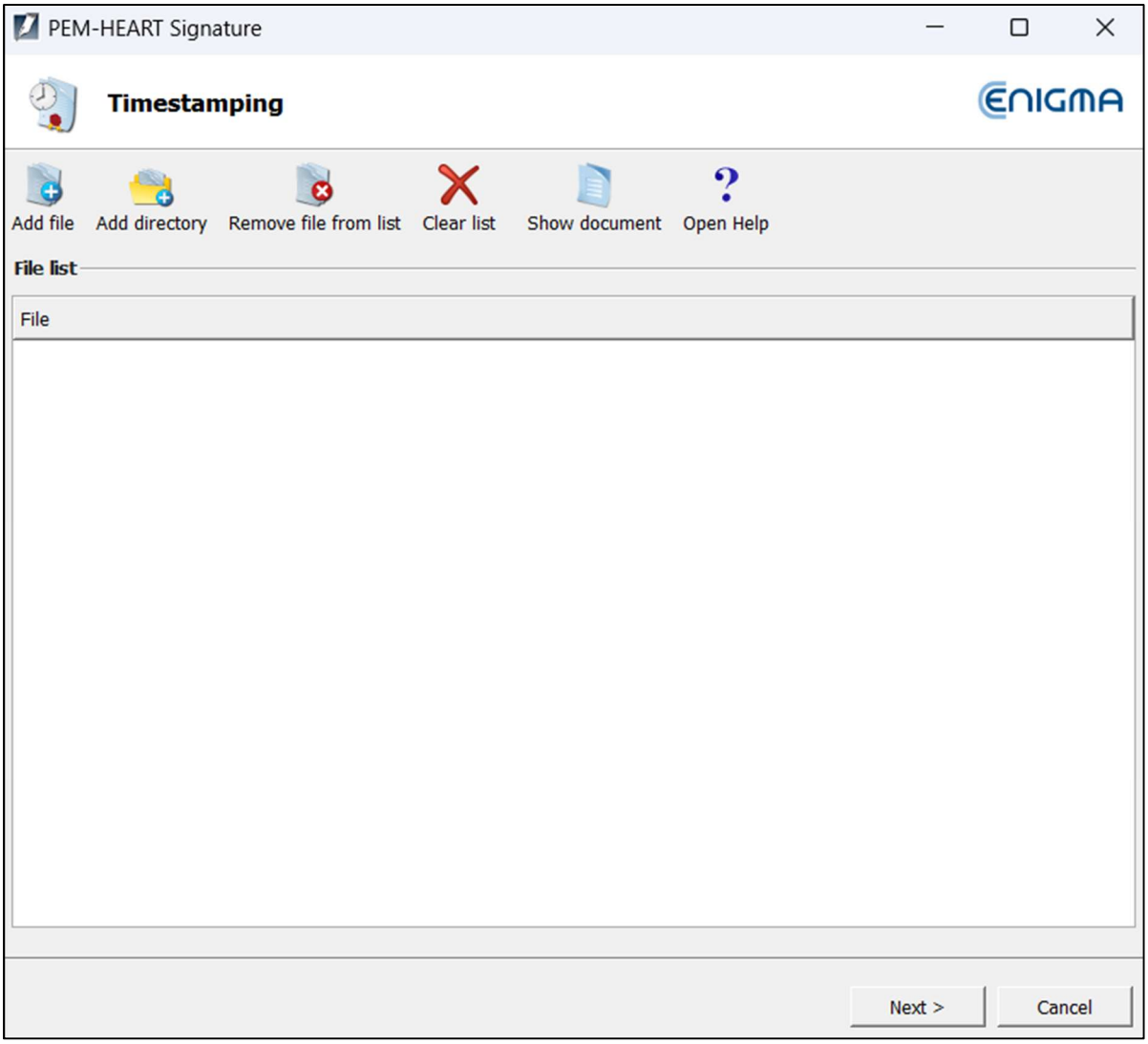


Figure 43 Timestamp application window for electronic signature

presumption of the accuracy of the date and time it indicates and of the integrity of the data with which the indicated date and time are linked.

When a timestamp is applied to a signature, it certifies not only the existence of the signed document, but also the signature itself, which protects against the legal consequences of the subsequent invalidation of the certificate used for the signature.

The timestamp can also be attached to the signature later, even by the recipient of the document (in fact, the recipient of the document is often more interested in being able to verify the signature correctly in the long term). More advanced forms of signature are also worth considering - that is, *long* and *archival* (see section **4.3.1 Verification Panel, pp. 33**). These forms can also use timestamps, but supplement it with other data needed for verification.

Select the *Advanced functions* menu (the bar on the left of the main window) and press the *Mark time* icon.

This brings up a window allowing you to indicate files and/or directories, as in signing and verifying the signature. Once the files are indicated and the *Next* key is pressed, the program asks for the card PIN (to sign the timestamping request) or the rSign PIN, then adds a timestamp to each signature contained in that file.

Note: Downloading timestamps may require the purchase of a timestamping package.

6.3 SIGNING AN XML DOCUMENT WITH ATTACHMENTS

By default, when a program signs an XML document with an enveloped signature (*XAdES enveloped*), it places the signature at the end of the document structure. In the vast majority of cases, this behaviour of the program is sufficient and meets the requirements of systems using signatures. However, if there was a need for a different position of the signature inside the document, the option Sign *XML document with attachments* should be used. The use of this option is dedicated to advanced users and requires knowledge of XML file construction, in particular, knowledge of the XML *Pointer Language* (*XPointer*) documentation.

Select the *Advanced functions* tab in the menu (the bar on the left of the main window) and press the *Sign XML document with attachments* icon. When the program displays the window for adding files to be signed, indicate the XML file (the file may possibly indicate attachments). If the signature is to be placed in a different place than the end of the file, it is required to indicate the appropriate place in the structure of the XML

document. In such case, the user must select the *Add New* option from the Signature *Location* section, which will bring up the Signature Location Configuration window.

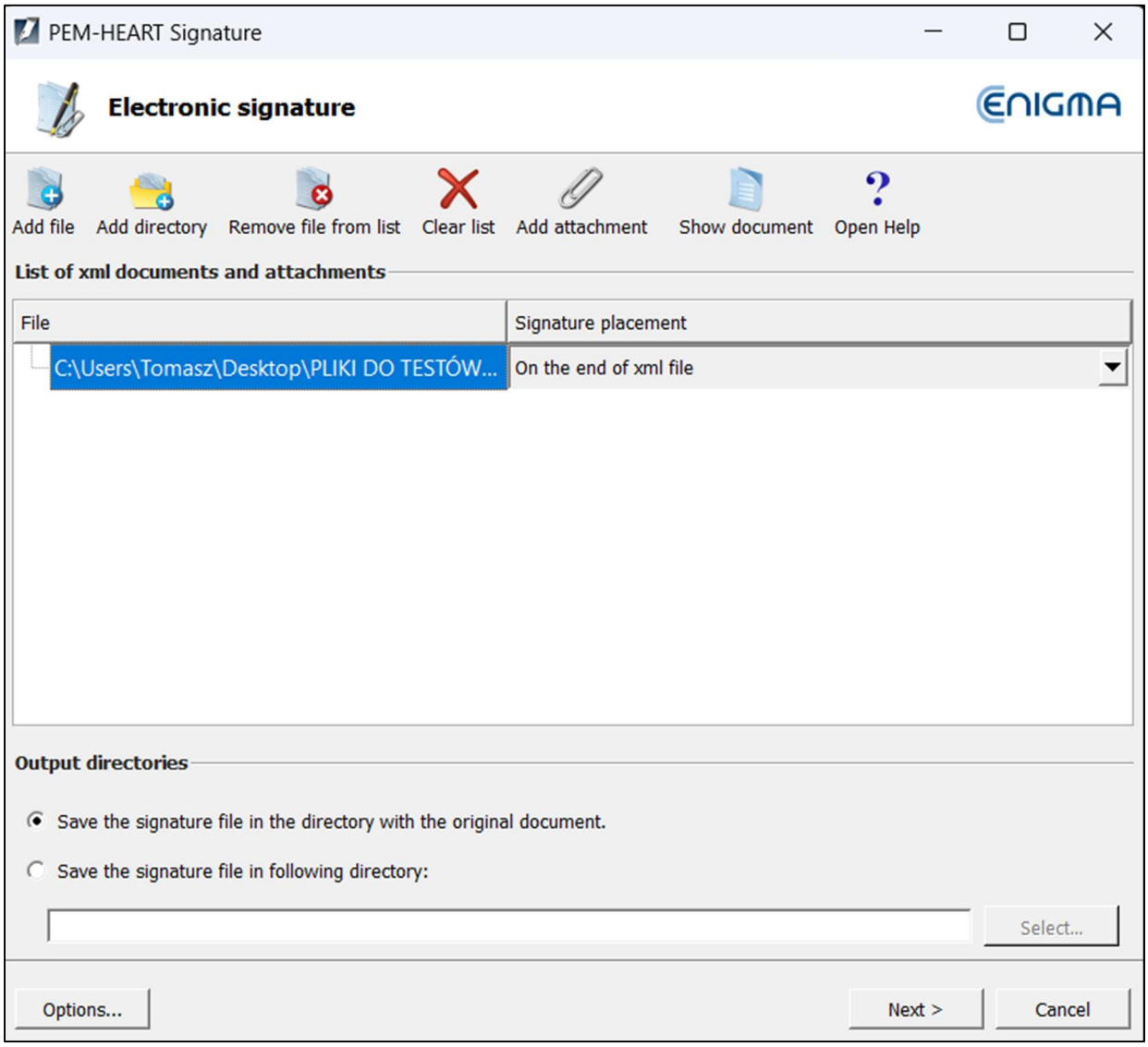


Figure 44 Signing an XML document with attachments - transition to signature placement configuration

Then, in the next window, click , enter your configuration name, the *xpointer* structure and possibly a description of the configuration to be defined. The *xpointer structure* is defined in the form: xpointer([*point to XML node*]). The available forms of specifying this location are described in the documentation of the *XML Pointer Language* (*XPointer*) available, among other places, at http://www.w3.org/TR/WD-xptr.

When the signing is complete, a summary window will be displayed. Signing in *XAdES enveloped* format does not change the extension of the XML file or its structure.

7 CRYPTOGRAPHIC CARD HANDLING IN THE PROGRAM

7.1 CHANGE OF PIN

(Function not available for rSign) To change the PIN for your card, select the *Card* tab from the main menu (bar on the left of the main window) and click on the *Change PIN* icon.

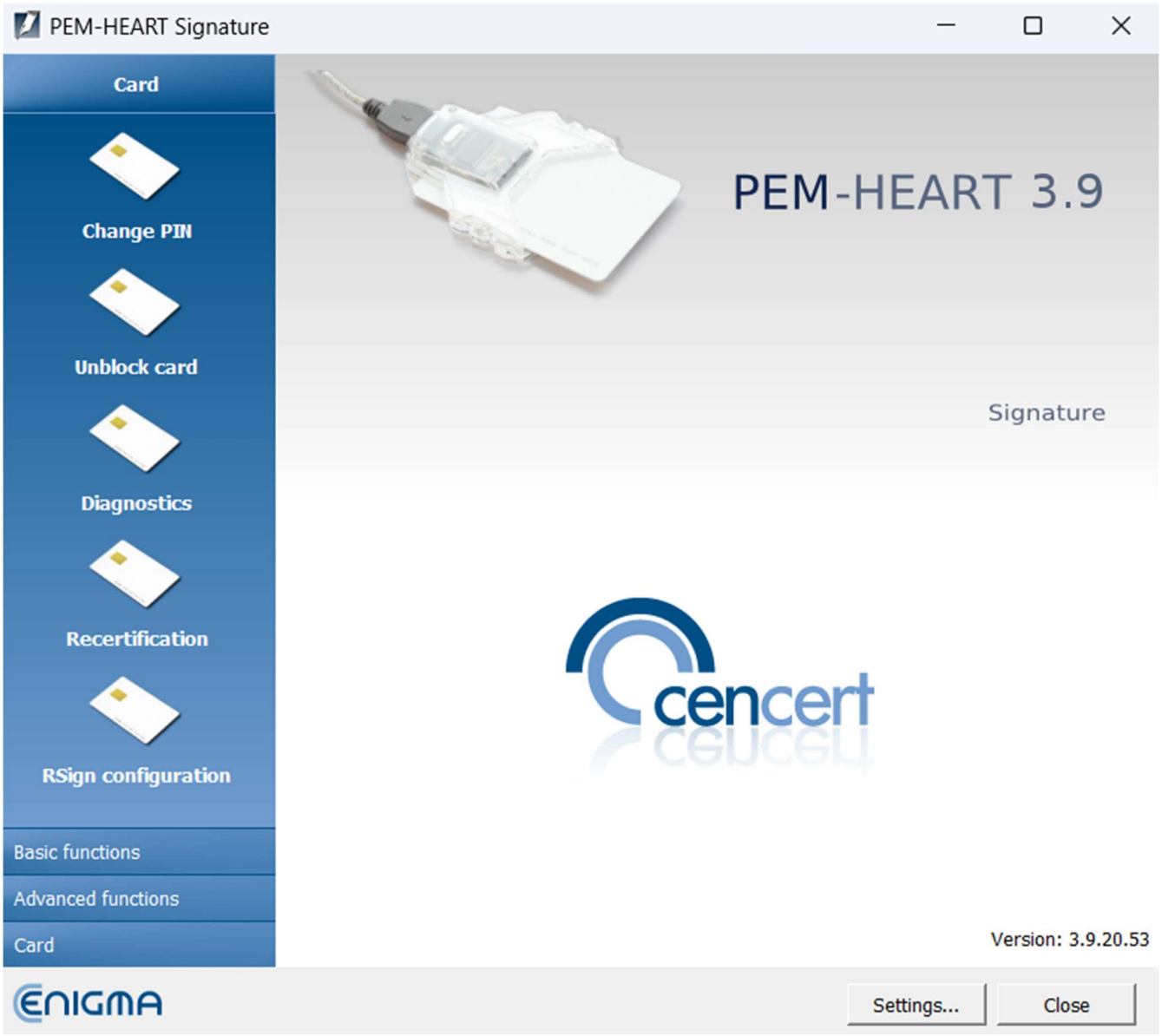


Figure 45 Main menu of PEM-HEART Signature application - selection of "Card" tab

It is then required to indicate the token for which the PIN should be changed.

Attention:

- For IDEMIA cards: qualified signature objects are always placed in the first token from the top; the other tokens can be used for other purposes, e.g. for the electronic seal.
- For IDPrime cards: the objects associated with the qualified signature are always placed on the second token from the top - this has the name 'Digital Signature PIN'.

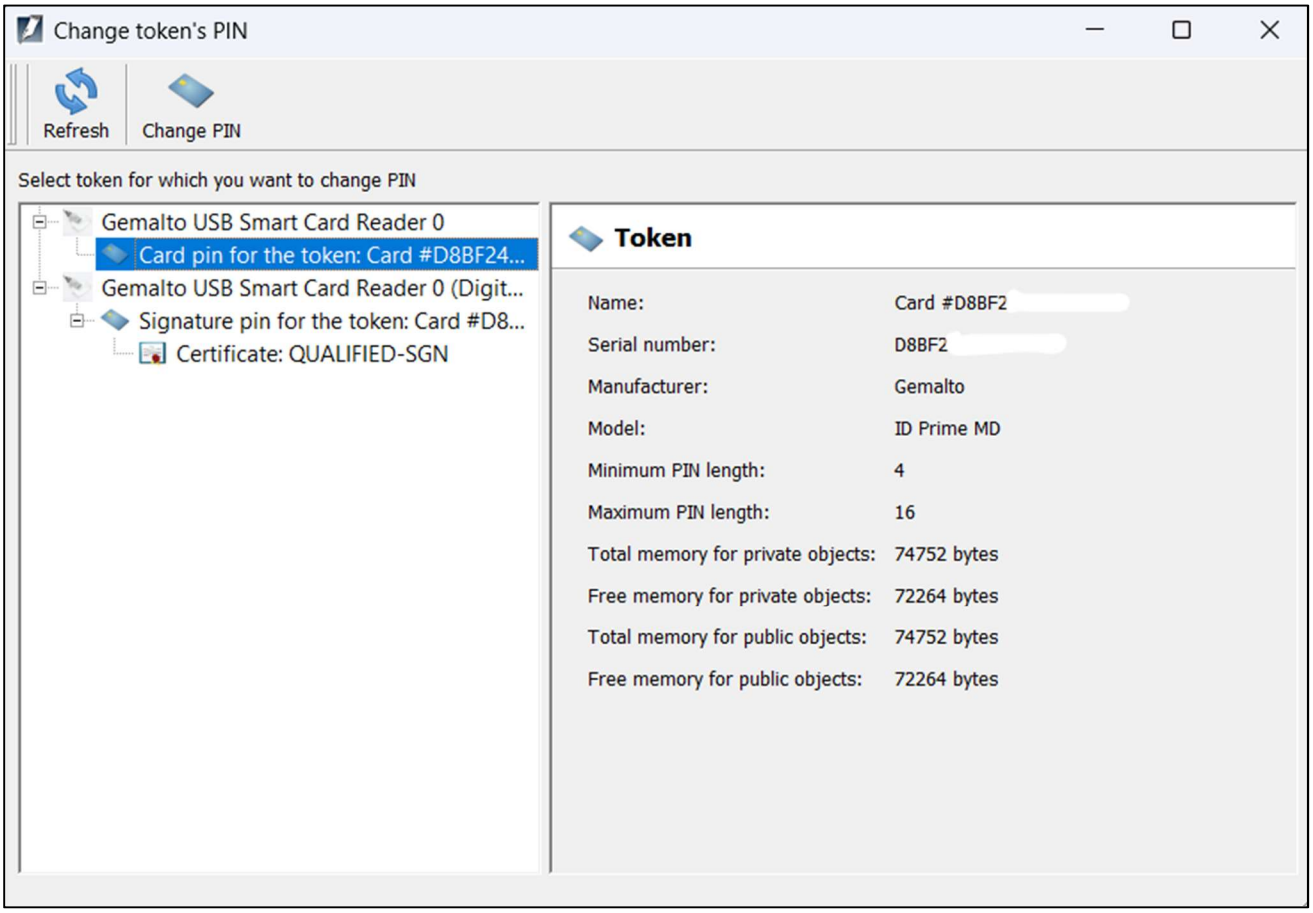


Figure 46 Example of program screen for Thales card type A

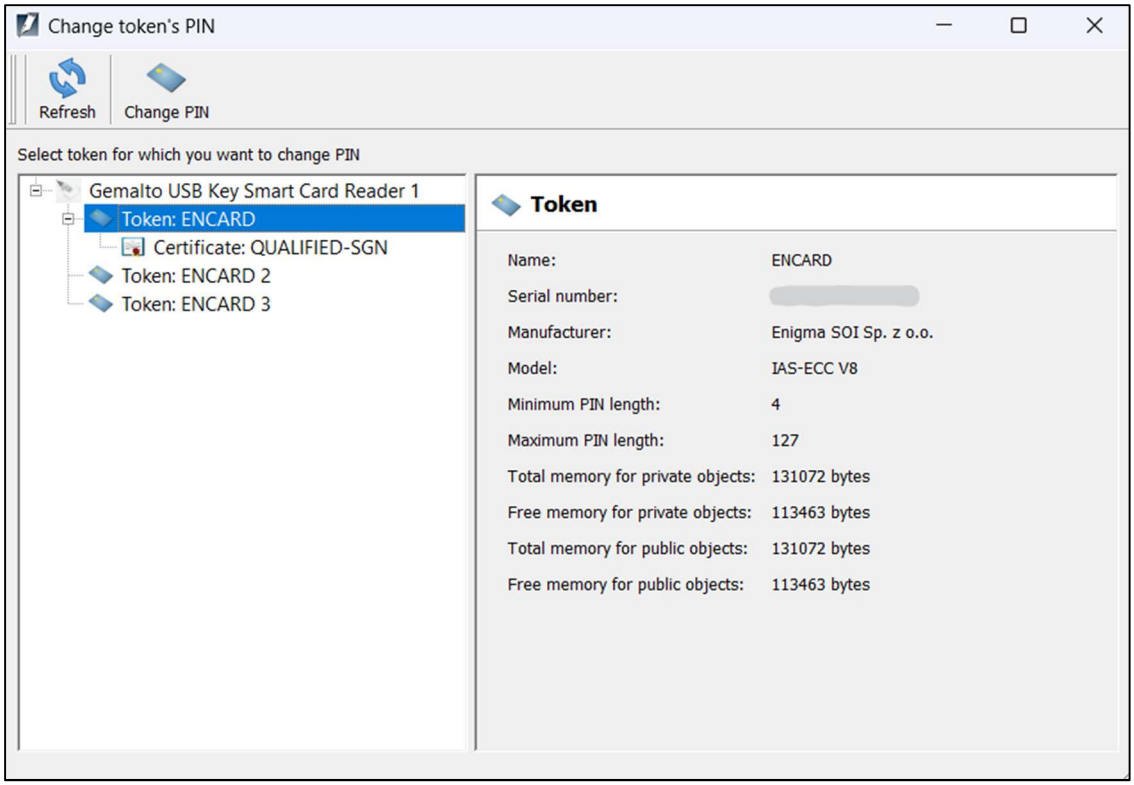


Figure 47 Example program screen for IDEMIA Encard

To make a change, the user selects the "Change PIN" option above the list of tokens.

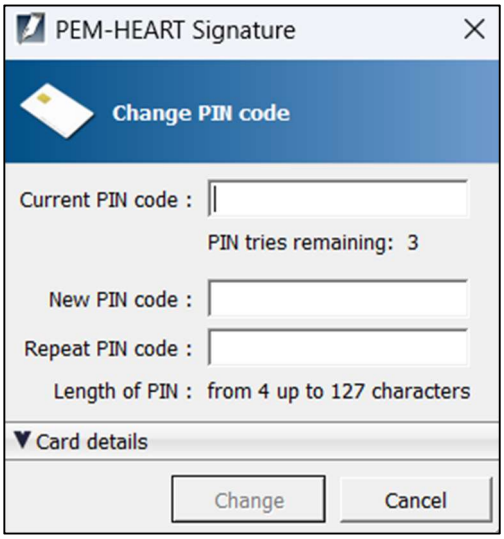


Figure 48 PIN change screen IDEMIA card

To change the code, you must enter the correct current PIN code and enter the new code twice. It is not recommended to use Polish letters or other characters for the PIN, which may not be entered correctly at different language settings of your computer's keyboard (the card will lock itself after 3 attempts to enter an incorrect code). It is recommended to save the PIN in a safe place (separate from the card), the exception here is the PIN for thales cards (first token), in which case it will lock after 5 attempts.

Note: If the PIN is blocked, the card can only be unlocked with the PUK code.

PIN / PUK codes are assigned by the user when activating the card. Cencert does not have PIN / PUK codes and it is not possible to unlock the card due to an incorrect PIN / PUK code.

7.2 CARD UNLOCKING

(Function not available for rSign) If the card is blocked after too many incorrect PIN codes have been entered, it can be unblocked using the PUK code. The PUK code is assigned by the user during card activation. After using the *Unblock card* button, a window with a selection of tokens will open.

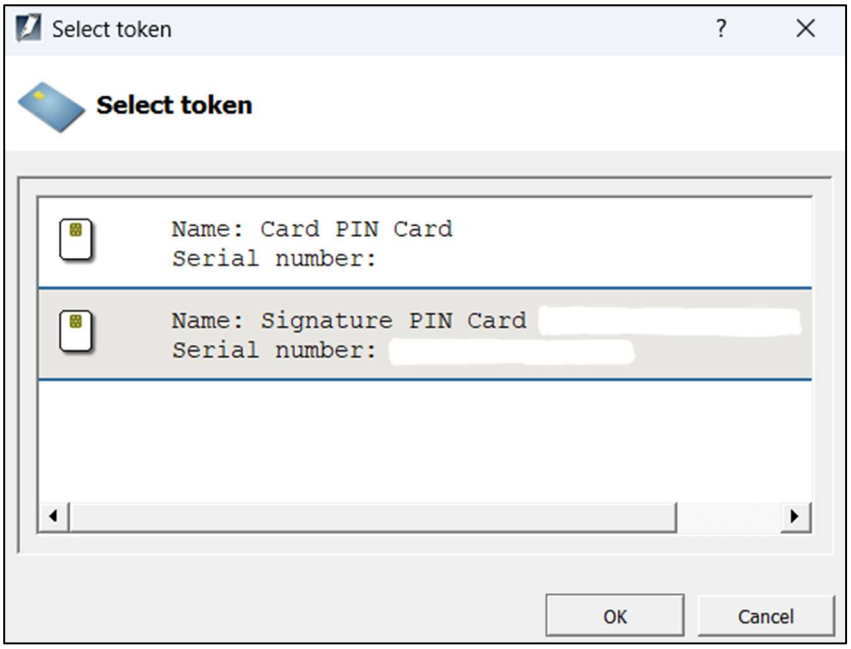


Figure 49 Card unlocking - token selection - IDPrime card

Once the PUK code has been entered correctly, it will be possible to set a new PIN code and the card will be unlocked.

Attention! A limited number of attempts are available to unlock the card using the PUK code. If you fill in the data incorrectly on each attempt, the card is permanently blocked and no further use is possible.

PIN / PUK codes are assigned by the user when activating the card. Cencert does not hold PIN / PUK codes and cannot assist you, if your card is blocked due to an incorrect PIN / PUK code.

7.3 DIAGNOSTICS

The *Diagnostics* panel shows additional information about the certificate data, allows you to save the certificate to a file, register it in Windows, download the Administrator PIN (only for IDPrime cards) and enable logging (only for IDEMIA and rSign cards - a function described in the **10.1 Card operation logs for Windows, pp. 76**).

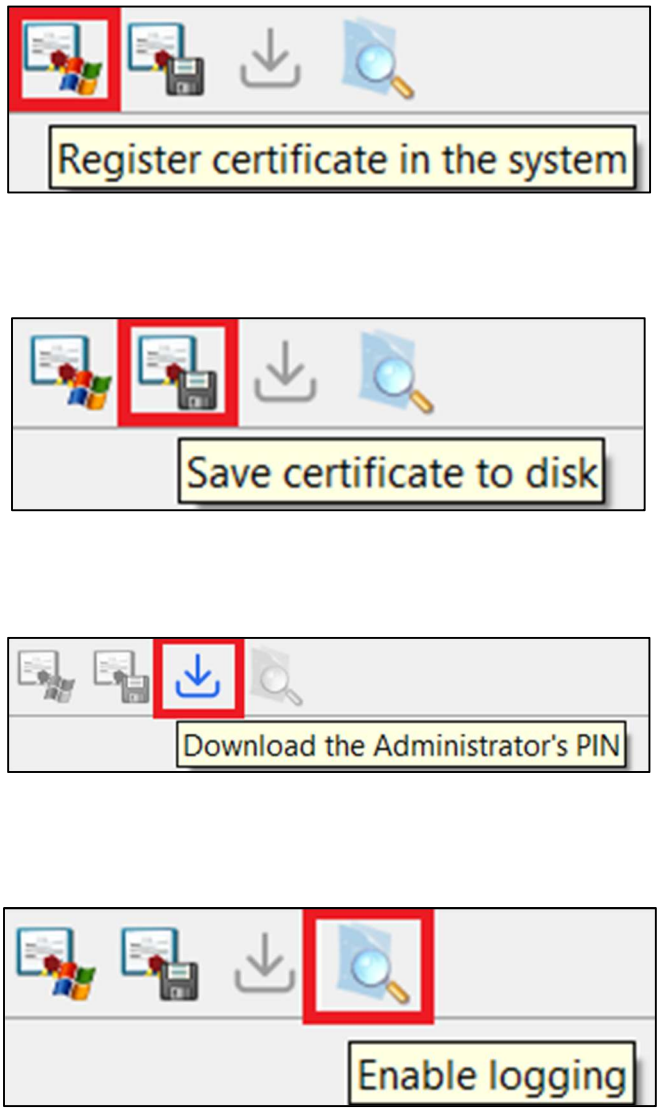


Figure 50 Additional options in the *Diagnostics* screen

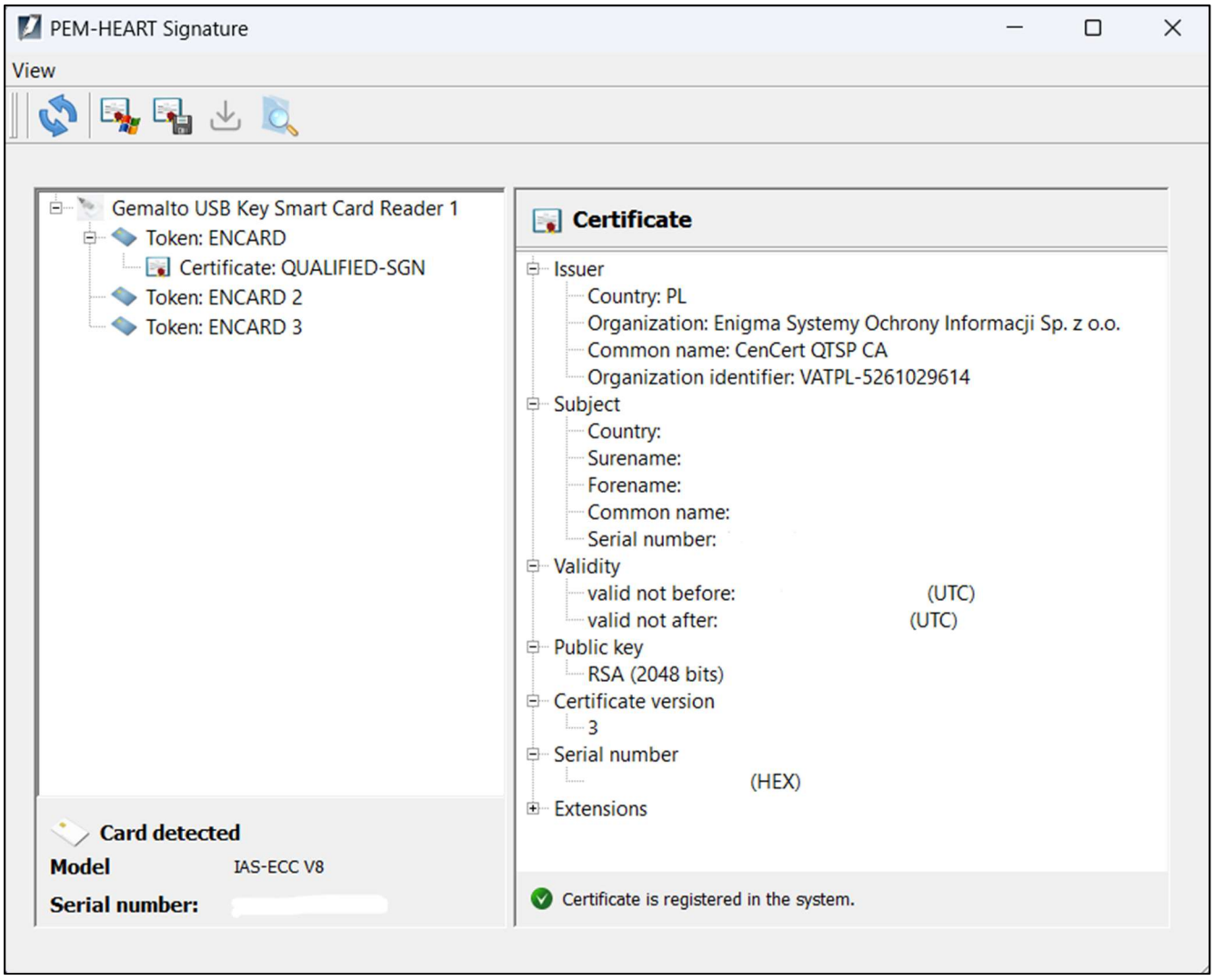


Figure 51 Open view of the *Diagnostics* panel - display of card token and rSign

7.4 ADDITIONAL OPTIONS

7.4.1 CERTIFICATE RENEWAL

The PEM-HEART renewal certificate is redirected and opened. Link to page with user guide: https://www.cencert.pl/poradnik-uzytkownika/

7.4.2 CONFIGURATION OF RSIGN

PEM-HEART is redirected and opens the PEM-HEART Configuration rSign.

8 PROGRAM SETTINGS

Attention!

All parameter change operations will be stored in the program memory when saved via the *Save* button at the bottom right of the program menu.

8.1 CHANGING THE SIGNING PARAMETERS

To change the signing options, in the main window of the application, click on the Settings button (located in the bottom right-hand corner of the application screen, right next to the Close button). A new settings window will be displayed, with the Signature tab open. All options specifying the signature format (XAdES, CAdES, PAdES, ASiC) will be applied to files with the extension currently selected in the Extension list. All files with extensions *.* (and therefore all files other than those defined in the list under this extension) will be signed in XAdES format in a separate file by default. At the same time, *.PDF and *.XML files have their own default signature formats, which will be visible when the *.PDF or *.XML line is selected in the list respectively.

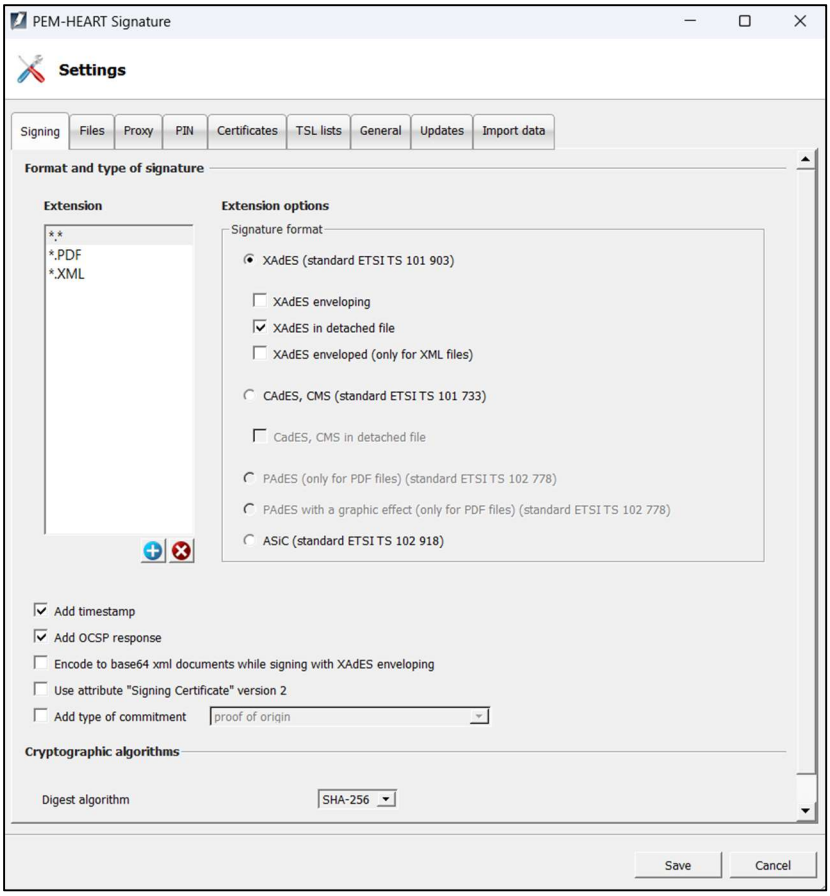


Figure 52 Signing settings change window

Here it is possible to add (or remove) file extensions (using the icons and) for which a different default signature format is to be used. For example, by adding a new item "*.docx" and defining that for these files a signature is to be performed, e.g. CAdES and CMS in a separate file, then for the signature call for each MS Word document file (*.docx), the program will propose by default a signature in CAdES format and CMS in a separate file.

In the section below selecting the extension and setting the signature format for the extension, there are additional options for signatures. These settings apply to all signatures - irrespective of the file name.

The *Add Timestamp* option means that a timestamp will be added to each signature (Note: A timestamp package may be required for correct operation).

The *Add OCSP response* option means that, in addition to the timestamp (selecting *Add timestamp* unlocks the possibility of selecting *Add OCSP response*), information about the status of the certificate used for the signature will be added to the signature (this creates a long form signature - see also section **4.3.1 Verification Panel, pp. 33**).

The option *Encode base64 xml documents when signing surrounding XAdES* is needed in specific situations, if the system verifying the signed documents has limited capacity to verify different signature formats and requires it.

The *Use Signing Certificate Attribute* (*Signing Certificate) Version 2* option places an indication of a certificate in the signature in a format that is compatible with newer versions of the ETSI standards for signature format. This option should be checked, if required by a signature verification system using only the new formats.

Selecting the *Add Commitment Type* option adds a signed attribute, which indicates for what purpose (in what role) the signatory has signed (e.g. as 'formal approval', or 'acknowledgement of receipt', etc.).

The *hash algorithm* option specifies the cryptographic hash algorithm used to issue the signature. The program only allows selection from good algorithms that guarantee adequate security (when the version of the program is current).

8.2 FILES

The tab contains options for setting output directories for the documents to be processed. By default, the program processes documents in the same directory in which the document is located. It is possible to set other directories into which signed or verified documents will be saved.

To define a directory, the box in front of the option description must be ticked; the *Point* button will then be activated, which can be used to point to a directory in the file system.

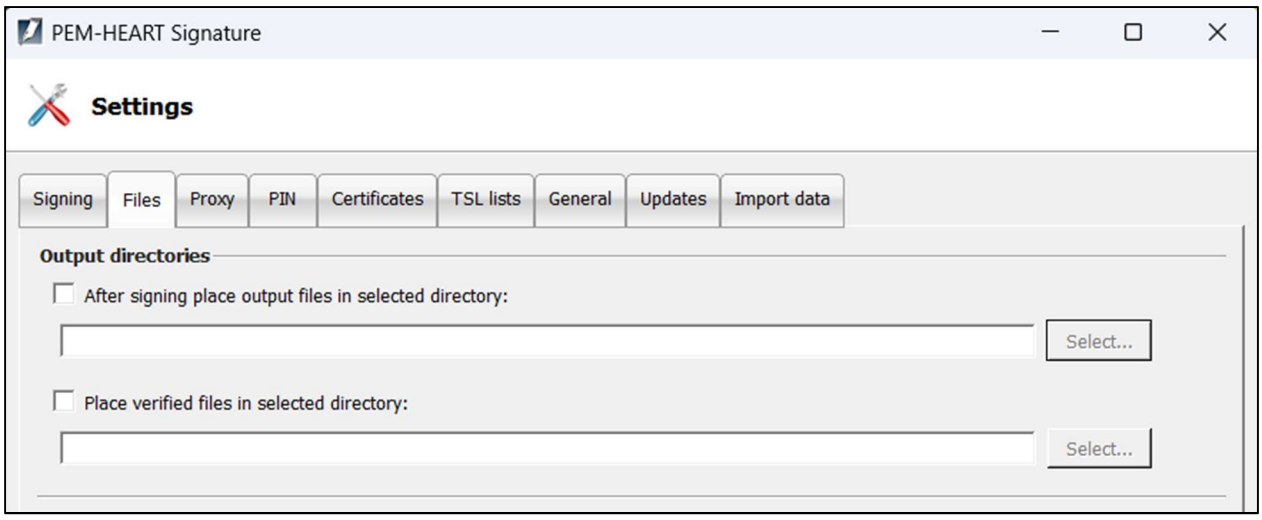


Figure 53 Defining output directories for processed documents

8.3 PROXY

The tab is used to specify the connection to the proxy server. There are two possible configurations to choose from:

- *Use system settings* (Windows only) - default option, the configuration is taken from the system settings (registry)

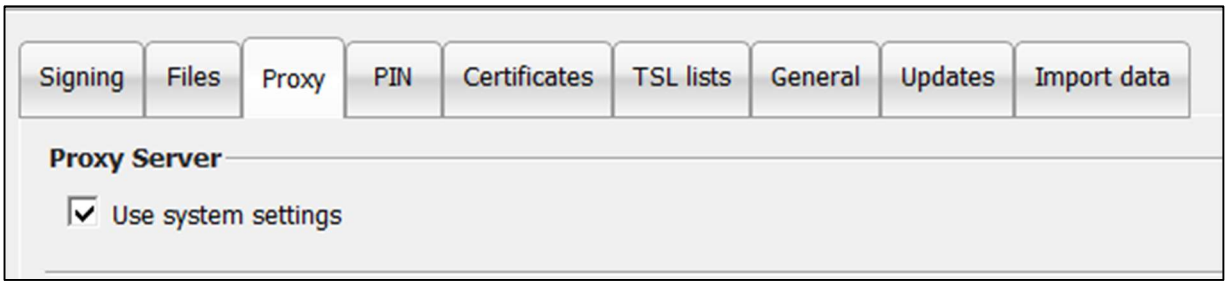


Figure 54 Proxy server - system settings

- *Configure proxy* - manually set the configuration, indicate the port address and/or authentication data. All mandatory fields for the *proxy* must be completed. Activation of the settings is confirmed via the *Save* button in the bottom right corner of the program. *Proxy authentication* is optional here and not required.

Incorrectly configuring the server results in the program not being available on the Internet (timestamp cannot be downloaded, signatures may not be able to be verified).

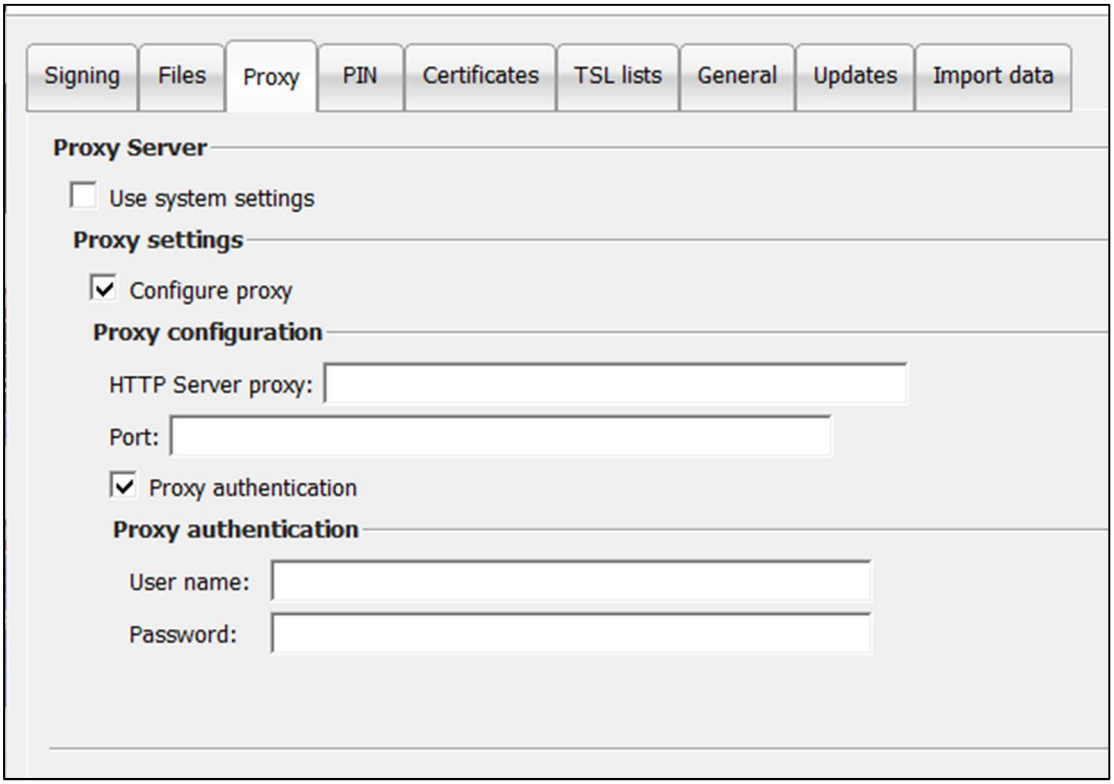


Figure 55 Proxy server - manual settings

8.4 PIN

The PIN tab is used to set the option for the program to store the PIN for the cryptographic card.

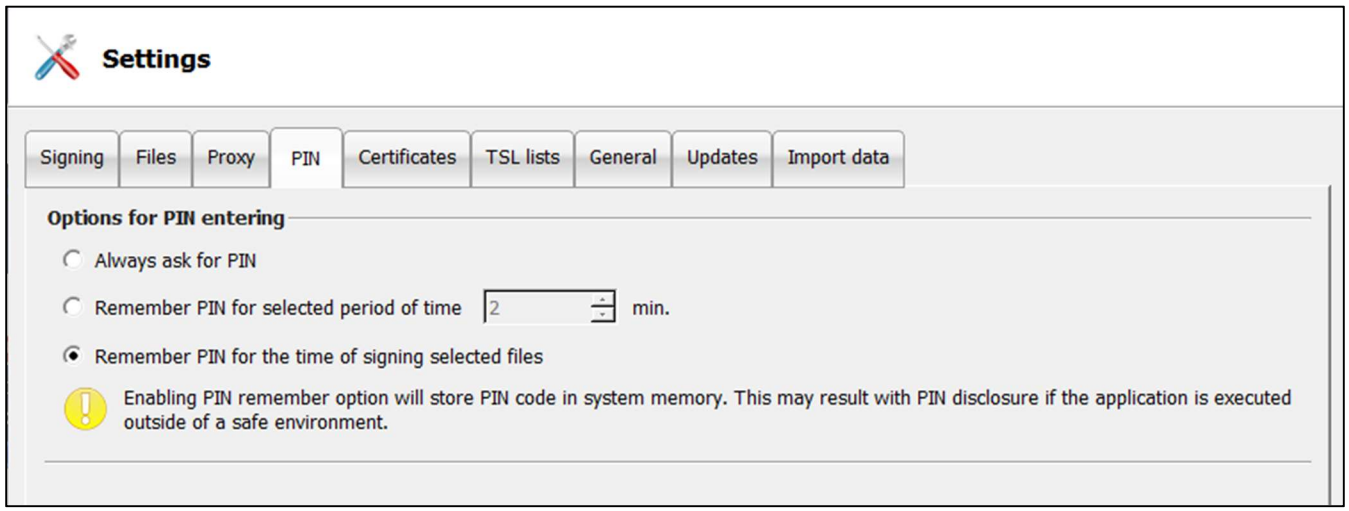


Figure 56 PIN settings

Please note that this option does not apply to rSign signatures (in the cloud). For this type of signature, the settings are configured in the mobile app.

By default, the PIN is remembered for the duration of the signatures for all documents in the signing window. In order to sign all files in the signing window, the PIN only needs to be entered once - once the signatures have been made, reselecting the files to be signed (even without closing the program) means that the PIN needs to be entered again. It is also possible to set the option: that the PIN will always be entered for each individual document, or that it will be stored in the computer's memory for a specific time range.

8.5 CERTIFICATES

The tab relates to the presentation, registration in the system and export of the user certificate. If a card is inserted in the reader, the program will automatically read the data from it and display them in a window. If the certificate is not read out, check the placement of the card and use the *Load* button. If an rSign token has been installed in the system, it will also be displayed in the list after the *Read* action.

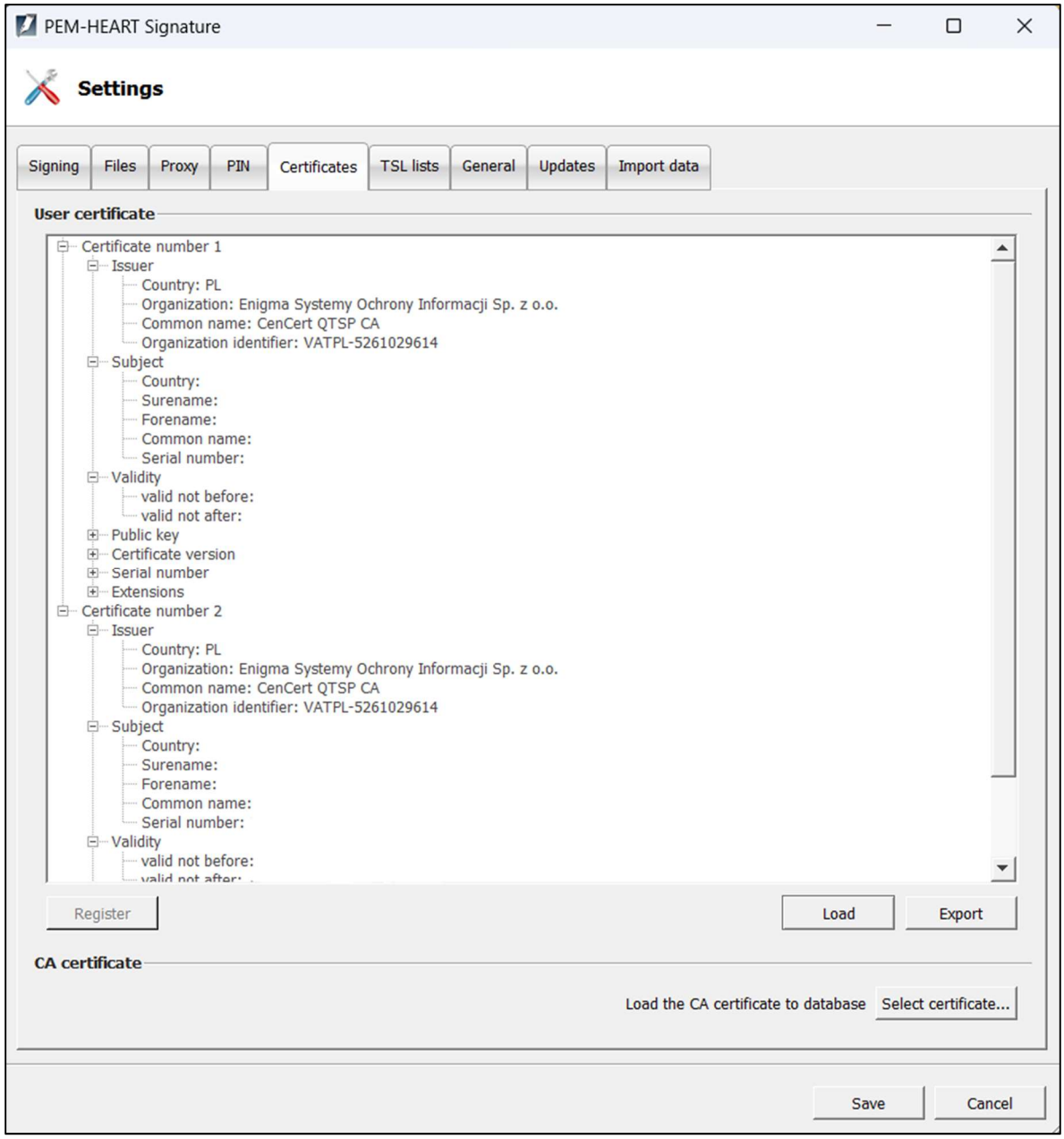


Figure 57 User certificate settings

The *Register* button is used to register the certificate read from the carrier in the system store. Exporting the certificate to a file is possible via the *Export* button. The *Certificate of Authority* section is used to indicate and load such a certificate of a trust service provider into the program database. This option is used in specific situations concerning non-qualified signatures - when the program does not have a current "intermediate office" certificate of the trust service provider in the database.

8.6 TSL LISTS

The TSL lists contain all necessary data on qualified EU (including Polish) trust service providers. They allow the verification of signatures created using qualified certificates issued by Polish and other EU trust service providers.

This tab shows the current status of the TSL lists available to the program. It also makes it possible to manually download the current TSL lists issued in individual countries (however, manual download is not necessary for normal operation, as the program automatically downloads new TSL lists if, during signature verification, it encounters a certificate that cannot be verified on the basis of the TSL lists currently held by the program). To download TSL lists, click on the *Download TSL lists* button.

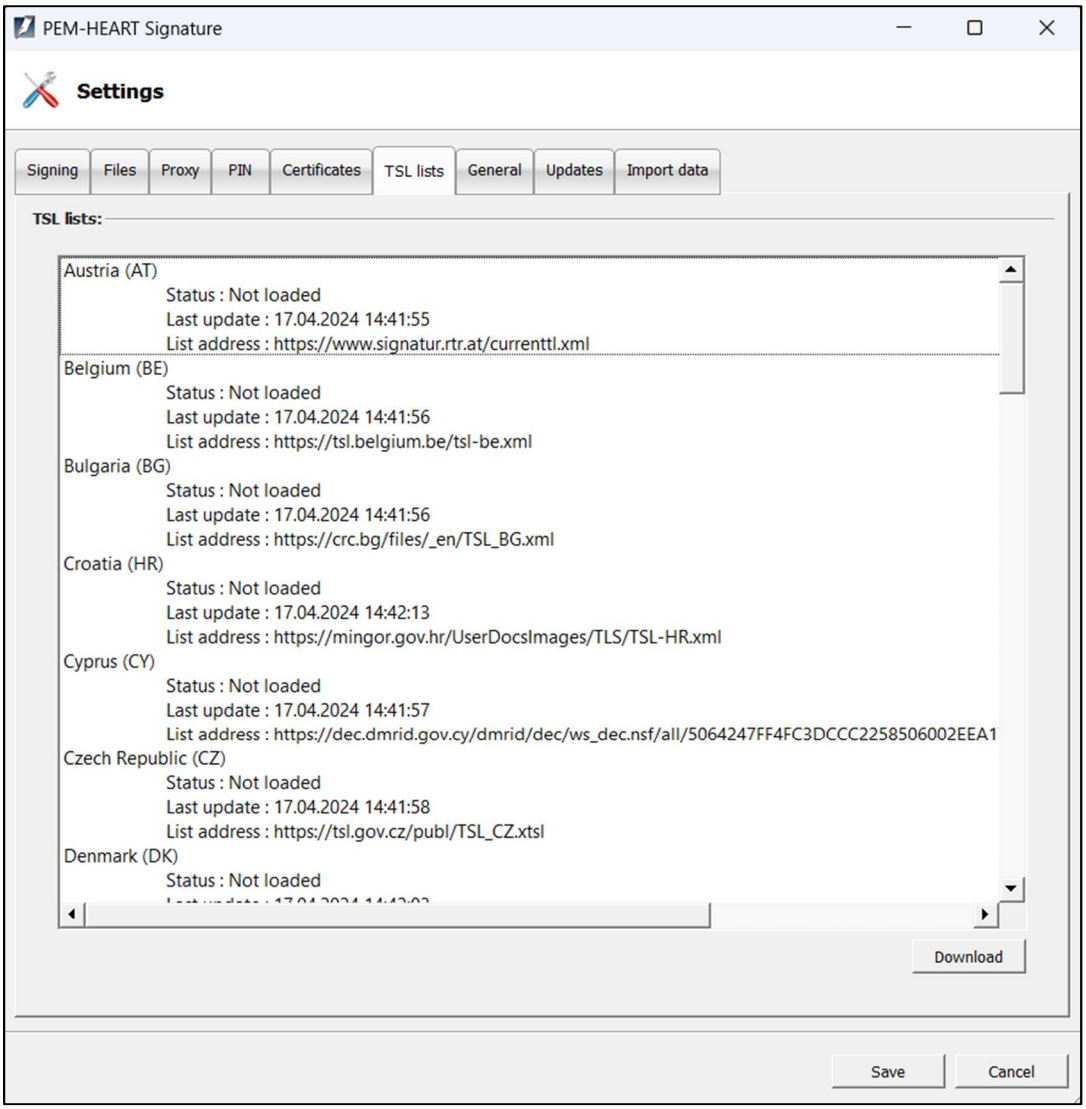


Figure 58 Settings - TSL lists

8.7 LANGUAGE SETTINGS

Changing the program language is possible via the *General* tab in the *Settings* panel. Selectable languages: *Polish, English, Ukrainian, Russian*.

The *Use built-in file selection windows* option is used to change the appearance of the windows that show up with the selection of a file for signature, for example.

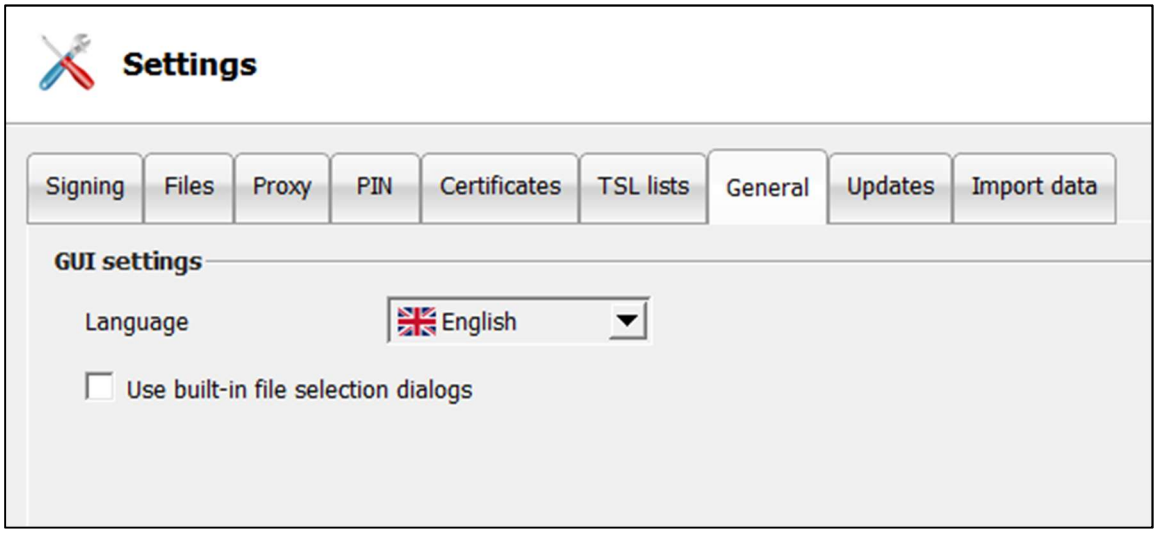


Figure 59 Selection of the language used in the program

8.8 UPDATES

In the main window (bottom right corner) of PEM-HEART Signature, the version of the program is indicated. In addition, in the *Updates* tab it is possible to check, if there is a new version. Information about an available update can be made manually by pressing the *Check for Updates* button, or you can set the option to automatically check when starting the program. If a new software version is detected, messages will be displayed.

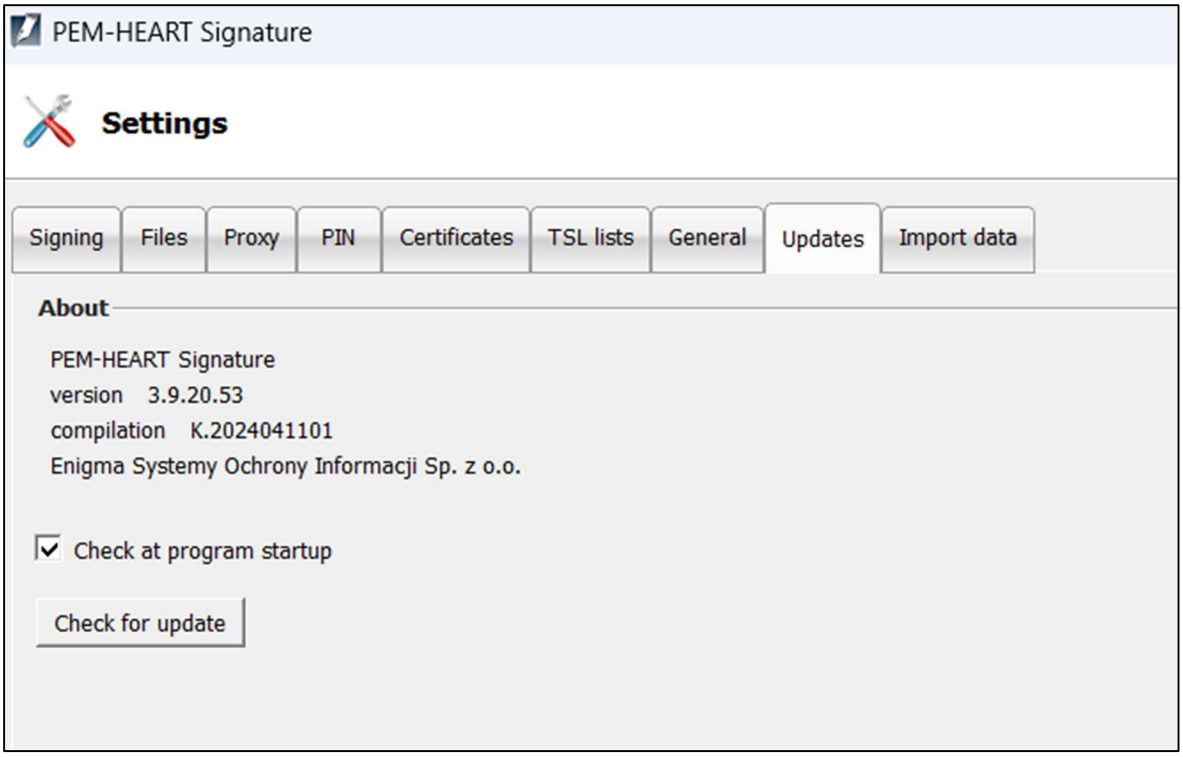


Figure 60 Window with information about the software version

8.9 DATA IMPORT

The data import options are for the operation of the program in an environment where there is no Internet access. Adding timestamps and checking certificate status based on OCSP is not possible in such a situation, but signature execution and verification are still possible, provided the program has up-to-date TSLs and CRLs - which in this case have to be transferred and loaded into the program manually.

Please note that rSign signatures (in the cloud) always require Internet access.

To load a file with either a CRL or TSL list, press the *Point* key next to the relevant list (after which you need to select the relevant file on disc) and then the *Add CRL* or *Add TSL* key respectively.

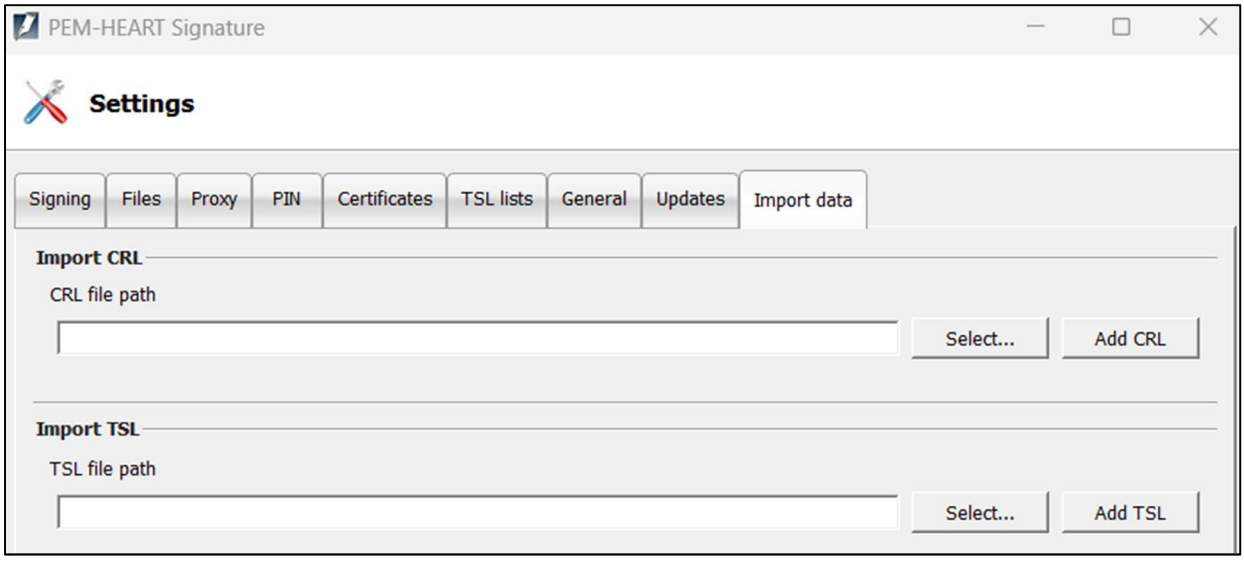


Figure 61 Settings - Import of CRL and TSL data

8.9.1 CLEANING THE CACHE

The 'Clear cache' button deletes the *PEM-HEART Signature database*. This should be tried in specific cases, e.g. when there is a database error. The database contains cached data (e.g. the current CRL), deleting it has no negative consequences as the program will automatically retrieve the missing data from network resources.

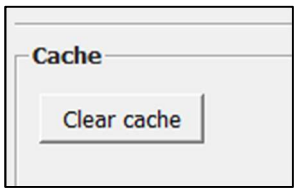


Figure 62 Option to clear the cache

9 SIGNATURE RSIGN

9.1 CONFIGURATION ON A COMPUTER

9.1.1 ADDING THE RSIGN TOKEN

To be able to use rSign on a particular computer (on a particular Windows account), the signature must be configured on each such computer (account).

The objectives of this operation are twofold - firstly, when starting the signature, the program needs to know who will be signing (with which certificate the signature will be created). Secondly, an important objective is to increase the security of your signature - an rSign signature can only be created on a computer previously recognised by you as trusted.

To configure rSign, start *PEM-HEART Signature -> Tab -> Configure rSign* or from the Windows menu *PEM-HEART Configure rSign*. Then select the *Activation* button

Figure 63 Configuration of rSign

Then transcribe the rSign Key ID from the mobile phone app.

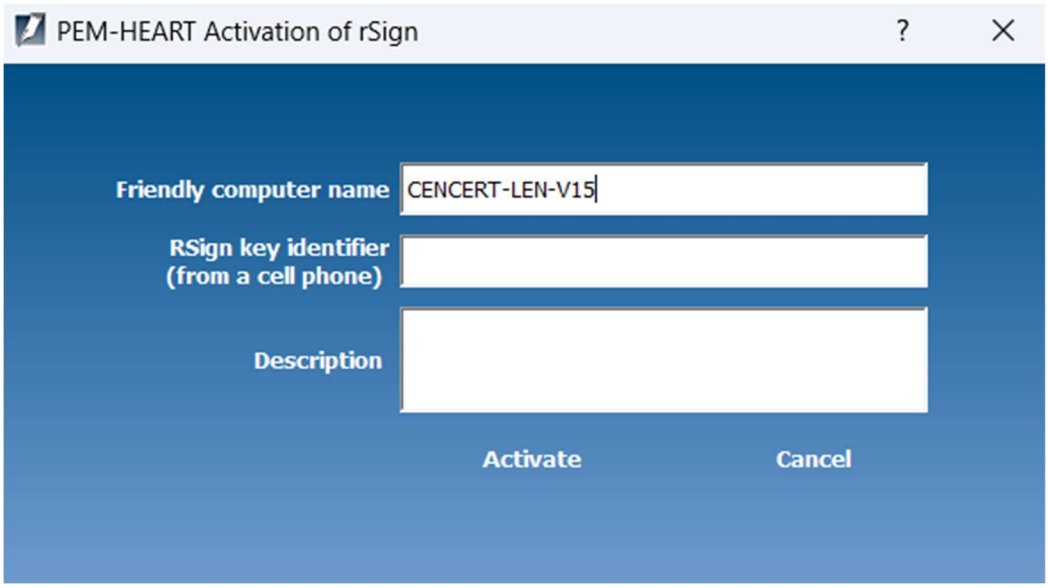


Figure 64 Activation window for rSign

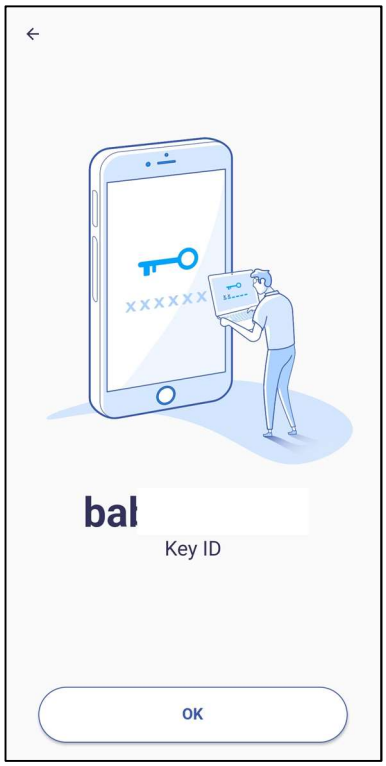


Figure 65 Mobile application window with key identifier

9.1.2 DELETING THE RSIGN TOKEN

If there was a need to remove the rSign token from the computer, such an operation can be performed via the *PEM-HEART Configuration rSign* program. After starting the program, click on the option *Delete token. The* active rSign token in the configuration and the certificate data will then be shown.

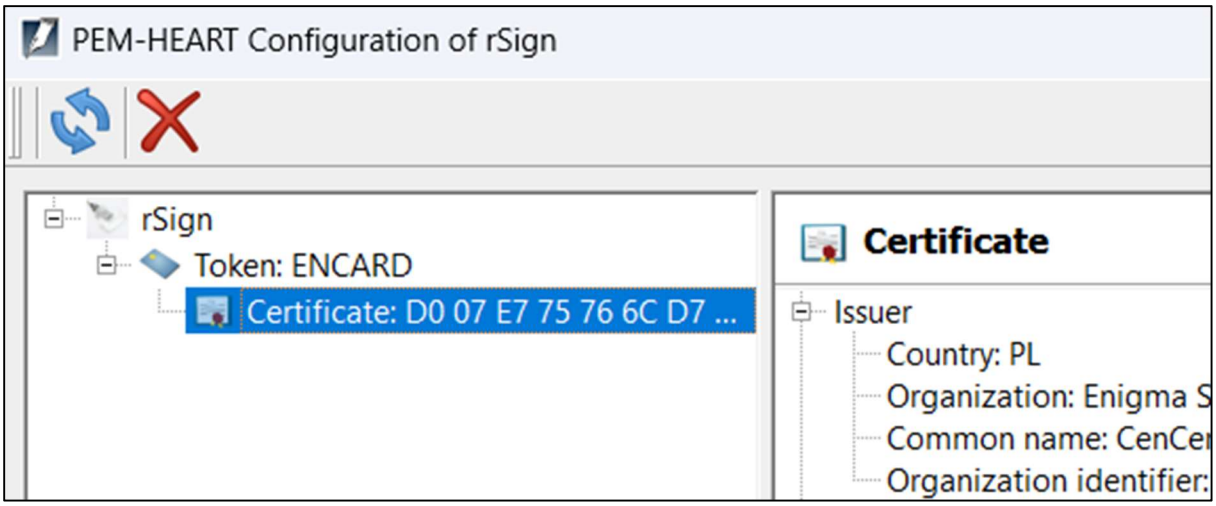


Figure 66 View of active rSign token with certificate data

Then click on *Token: ENCARD* in the left window, which will activate the button  in the top panel - selecting it will trigger the token deletion process. The user must confirm the deletion in a separate window:

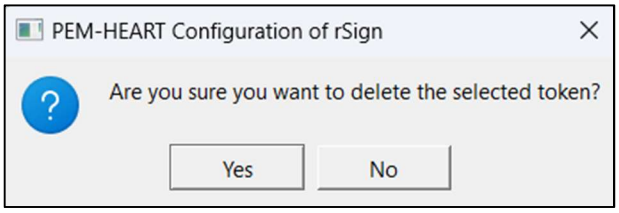


Figure 67 Confirmation of rSign token deletion

Clicking *Yes* will remove the token.

9.2 CONFIGURATION OF THE MOBILE APPLICATION

9.2.1 INSTALLATION

The app is available for download from the AppStore and Google Play.

9.2.2 HOME SCREEN

When the application is launched, a screen with an active signature PIN is displayed by default. Within the view, the user can see the PIN, a timer indicating the time for its use and a queue of pending operations. In addition to this, in the top right corner there is an icon , the use of which refreshes the notification view, and in the top left corner there is an icon , the selection of which will display options: *Operations* (the default option displayed when the application is launched), *Key ID*, *Settings*.

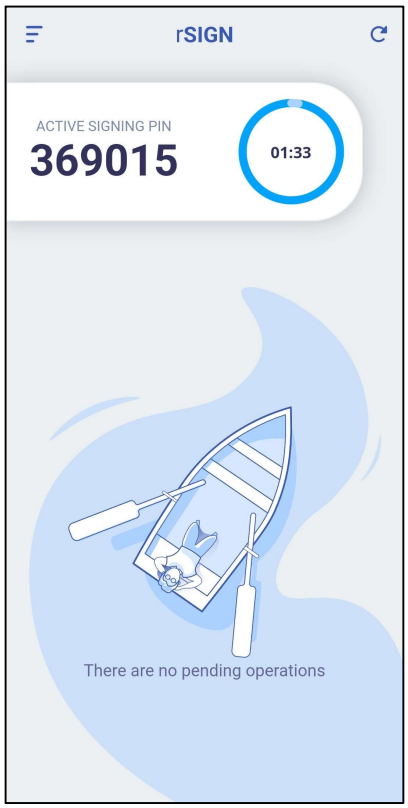


Figure 68 Home screen of the rSign mobile application

9.2.3 KEY IDENTIFIER

Selecting the *Key ID* option will display a screen with the key ID used among other things, to configure the use of the rSign signature on the computer. Before the User can see it, however, the program will first ask for a PIN - only if the PIN is entered correctly and approved will the key ID be displayed.

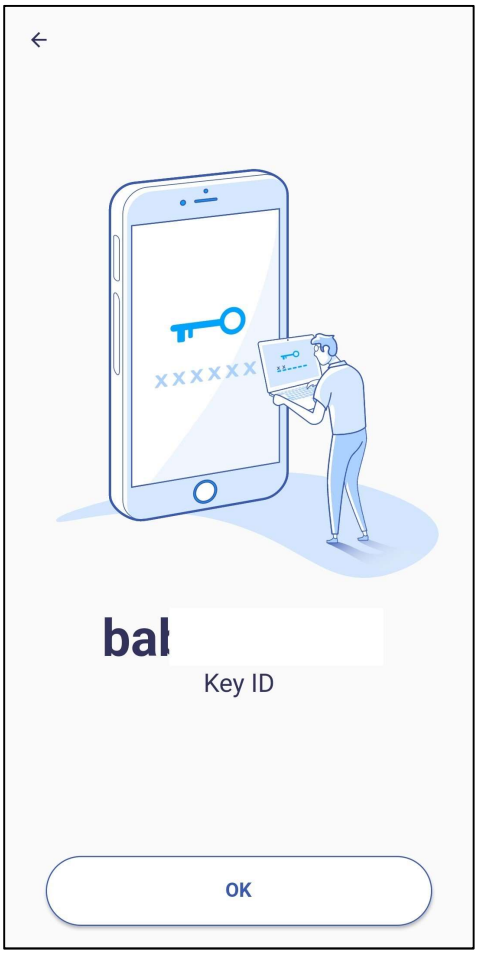


Figure 69 Application screen after selecting the *Key ID* option

9.2.4 SETTINGS

Selecting *Settings* will display a screen with a list of available application settings. These are:

- *Signature PIN memorisation* - an option to memorise the PIN code entered for a period of time specified by the User. Four options are available - three predefined ones: 3, 5 and 10 minutes, and any from 1 to 60 minutes.
- *Change PIN code* - option to change the PIN code.
- *Associated telephone number* - this is where the User enters the telephone number associated with the account.
- *Backup* - allows you to create a backup to activate rSign on any device.
- *Deactivate device* - option to delete rSign activation data from the device.
- *Language* - allows you to change the language in the application. Selectable languages: Polish, English, Russian, Ukrainian.

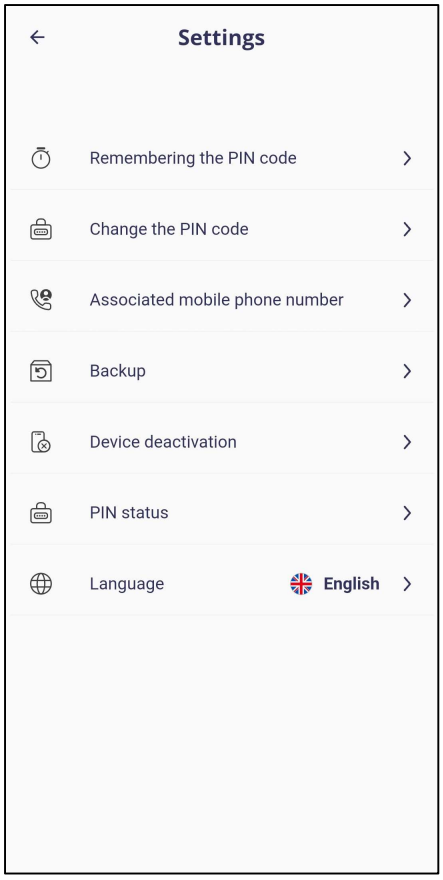


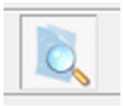
Figure 70 Application screen after selecting *Settings*

10 ADMINISTRATION OF PEM-HEART SOFTWARE

10.1 CARD OPERATION LOGS FOR WINDOWS

The software allows logging to be enabled for card operations, e.g. signing a document. There are two ways to enable such functionality:

- By clicking on the icon with the "magnifying glass" in the Diagnostics panel launched by Advanced functions.

-  - icon indicating that the function has been activated - the first time it is used, PEM-HEART Signature must be restarted for the changes to be saved,

-  - icon indicating that the function is disabled - the first time the function is activated, PEM-HEART Signature must be restarted for the changes to be saved.

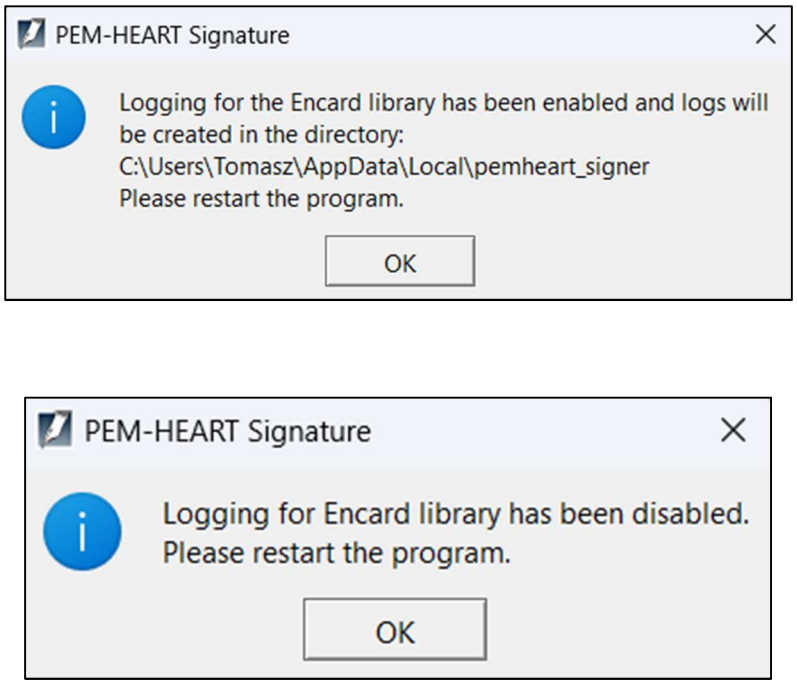


Figure 71 Message after operation to enable and disable logging to IDEMIA cards

- A more advanced configuration is invoked via the windows for PKCS#11 library configuration. Select *Start->Programs->ENCARD->Konfiguracja ENCARD PKCS#11 Menu* respectively Selecting the configuration option will invoke the window for PKCS#11 library configuration.

Selecting the *Save called functions to file* option (which is also the name of the first section) is used to configure the saving to the log file of all information, in particular the contents of private objects. The PINs given are not saved - when logging commands to the card, they are replaced by XX characters.

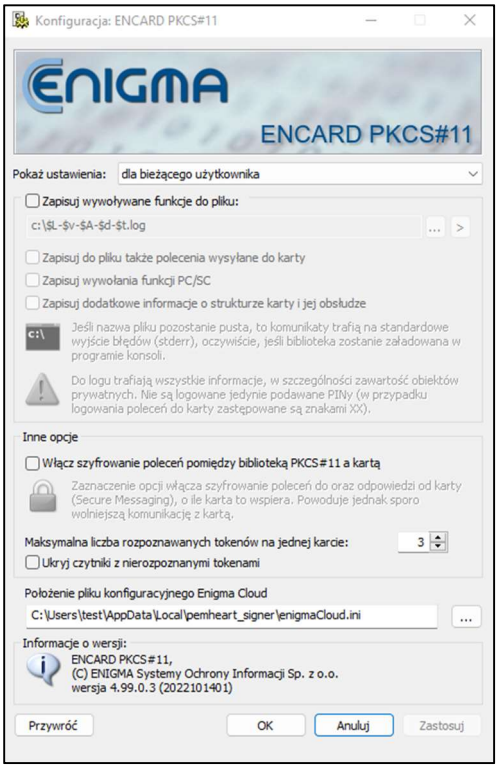


Figure 72 PKCS#11 library configuration screen

Use the ... button to determine the location and name of the log file. If the file name is left blank, the messages will go to the standard error output (stderr), if the library is loaded in the console program.

The library accepts special macros in the filename to write to different log files depending on the application that loads it, the current time and date, the library version and others. Pressing the > button next to the log file name brings up a configuration dialog box listing all the macros:

- $A - file name of the application loading the library (without path and extension).

- $L - file name of the loaded library (without path and extension).
- $l - the internal name of the library.
- $D - date of library loading in the form YYYY-MM-DD.
- $d - date the library was loaded in the form YYYYMMDD.
- $T - hh-mm-ss library loading time.
- $t - hhmmss library loading time.
- $K - library compilation number (e.g. 2008080901).
- $V - the main version of the library (e.g. 2.0).
- $v - full version of the library (e.g. 2.01.2.2).
- $$ - $ sign.

As part of the recording of the called functions to the file, the user can select other information, causing additional information to be added to the log. The additional options are:

- *Save to file also the commands sent to the card*
- *Record PC/SC function calls*
- *Record additional information on the structure of the card and its operation*

The second section of the PKCS#11 library configuration window allows you to add encryption to the connection between the PKCS#11 library and the card, which is done by ticking the *Enable command encryption between PKCS#11 library and card* option. In addition to this, the maximum number of recognised tokens per card can be indicated and readers with unrecognised tokens can be hidden.

Just below the second section is a place to indicate the path of the location of the Enigma Cloud configuration file and information about the configuration software.

Any changes are saved by selecting *Zastosuj*. Selecting *OK* also saves the changes and closes the open configuration window. Any changes to the options can be cancelled by selecting *Anuluj* (closing the configuration window without saving the changes) or *Przywróć* (restoring the settings from just after the configurator was launched, without closing the program).

11 PROBLEM SOLVING

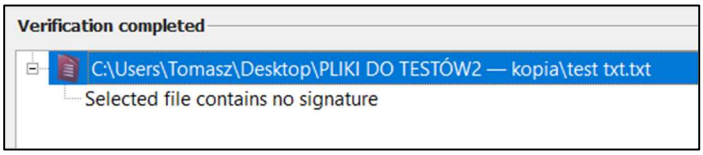


Figure 73 Message about no valid signature in the file

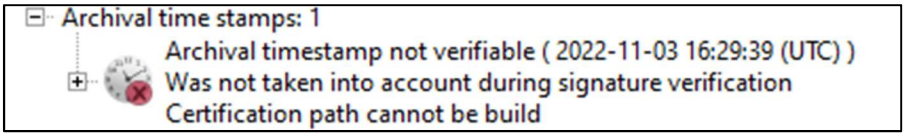


Figure 74 Message about an unverifiable archive timestamp

INSTALLATION		
Problem	Cause	Solution
Operation failed	The user cancelled the program installation process	Restart the installer

SIGNATURE		
Problem	Cause	Solution
The timestamp could not be downloaded from any of the servers	No timestamp package is associated with the certificate	- try again the next day or - purchase a timestamping package (details: https://www.cencert.pl), or - disable the timestamping option for signatures (see section **6.2 Timestamping**)
	The program has no internet access or the request has not been approved in the rSign application	- check your internet connection - check the proxy settings (if you are using a proxy server) - see section **8.3 Proxy**)

Functionality not available for current carrier !!!	No verification support function implemented for the file in question, signature made in an unsupported standard	Notification to Cencert

VERIFICATION		
Problem	Cause	Solution
Indicate the location of the documents. Not all disconnected documents were found	The program did not find the signed file in the signature directory.	Indicate the file that was signed (appropriate to the signature being verified)
Error opening input file	File is corrupted or currently open in another program	Close the other program and then try to open the file again in PEM-HEART Signature.
	The file is already open in another program	
	The program cannot access the location of the file	Verify that the location actually contains the file
None of the files contain a valid signature	The file is corrupted or has been signed in a format not supported by PEM-HEART SIgnature	Indicating another file or reporting to Cencert

12 LIST OF FIGURES