BANCA D'ITALIA
EUROSYSTEM

USER MANUAL FOR "ARUBA SIGN" SIGNATURE AND ENCRYPTION SOFTWARE

PKI BI PUBLIC KEY INFRASTRUCTURE

February 2025 v1.6

Contents

Glossary

PKCS#12	*PKCS #12* defines an archive file format for storing many cryptography objects as a single file.
LDAP	Lightweight Directory Access Protocol: RFC4511. It is a protocol that provides access to distributed directory services that operate in accordance with the X.500 data and service models.
CA	Certification Authority: a component of a PKI responsible for managing the lifecycle of certificates.
PKCS#7	(CMS - Cryptographic Message Syntax) is a standard syntax for storing signed and/or encrypted data.
CAdES	Signature format introduced by the Implementing Decision (EU) 2015/1506. All files can be signed with this format; an encrypted envelope is created that contains the file to be signed.
PAdES	Signature format introduced by the Implementing Decision (EU) 2015/1506. Type of signature applicable only to the PDF format.
XAdES	Signature format introduced by the Implementing Decision (EU) 2015/1506. Type of signature applicable only to the XML format.
FEA	Advanced Electronic Signature (FEA in italian) is a specific type of electronic signature. «*It is defined as a set of electronic data attached to or logically associated with an electronic document, which allows the identification of the signatory and ensures a unique connection to the signatory. This signature is created using means that the signatory can maintain under their exclusive control and is linked to the data to which it refers in such a way that any subsequent changes to the data can be detected*.»
OCSP/CRL	OCSP (Online Certificate Status Protocol) is an alternative to the Certificate Revocation List (CRL) and is used to check whether a digital certificate is valid or has been revoked.
HASH Function	Hash functions (SHA1, SHA2, etc.) are functions that, given an input (e.g., file, string of arbitrary length), produce a fixed-length sequence of bits (or a string).

1. Introduction

The ArubaSign tool is a software that allows you to use electronic certificates to:

- Sign a document;
- Encrypt a document;
- Verify the validity of the signature with which a document was signed;
- Decrypt and encrypt document;
- Affixing timestamps;
- Verify the users for whom a file has been encrypted.

Some operations, such as signing a document or decrypting an encrypted document, require the user to have an electronic certificate located on a cryptographic device, such as a smart card or USB token, or on files in PKCS#12 format. In order to carry out other operations, such as the encryption of a document, electronic certificates must be available with the **public key** of the recipients.
These certificates may reside on a local archive of the PC on which the *tool* is installed, or be available on an LDAP server that can be accessed through the network (the recipient will be able to decrypt the document using their own electronic certificate, i.e., **private key** – see below).

2. Basic concepts

The certificate

The certificate is a small file containing essential information for verification of the signature:

- The name and tax code of the holder (e.g. Mario Rossi, RSSMRA30A01H501I);
- The name of the company which the holder is a member of, if applicable;
- The name of the certifying entity (e.g. Banca d'Italia);
- The start date and end date of validity;
- The public key of the holder;
- Other service information.

The certificate is issued to the user by a trusted third party entity, called the certifier (*Certification Authority*, CA).

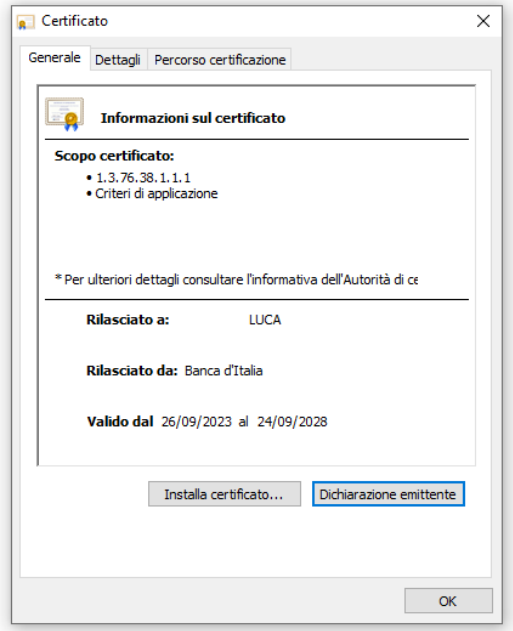


Figure 1 - Digital certificate

Encryption (cryptography)

The encryption (also known as cryptography) of a document is an operation with which that document is completely unreadable for anyone, except for those who have the key that allows it to be deciphered, i.e. bring it back "readable". Encryption therefore ensures confidentiality of confidential information.

Asymmetric encryption

Asymmetric encryption, also known as public key encryption, is a type of encryption in which each actor involved in the communication is associated with a pair of keys: The public key, which must be distributed; The private key, which is personal and secret. The fundamental property is that if one key is used to encrypt, the only way to decrypt is with the other. In this example, we will show how, with the help of the keys, the property of confidentiality can be implemented.

Objective: Marco wants to send an encrypted document to Roberto.

Initial conditions: Roberto has a pair of keys, a private one known only to the owner and a public one known to everyone.

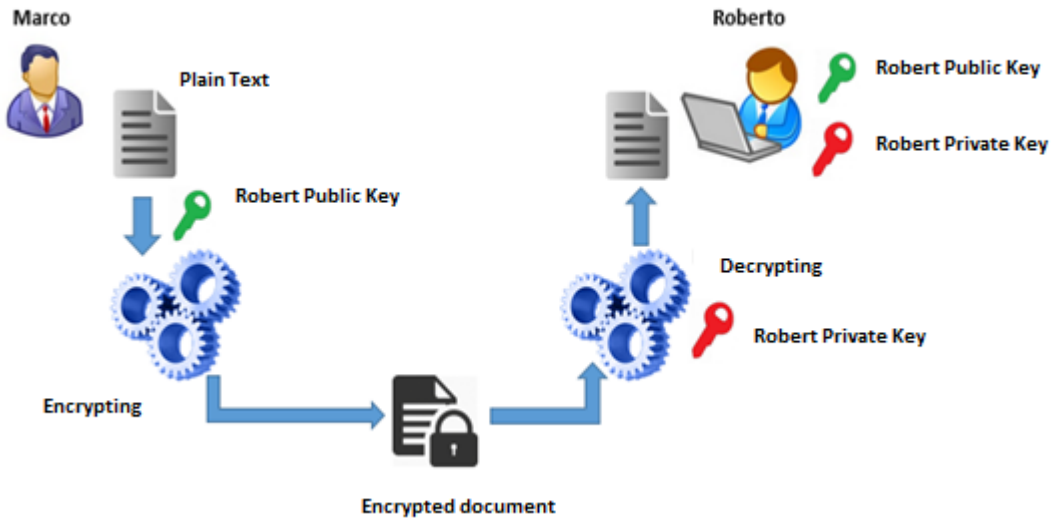


Figura 2 Asymmetric encryption

Flow:

1. Marco encrypts the document with Roberto's public key.
2. The encrypted document is sent to Roberto.
3. Roberto decrypts the document with his private key.

Digital signature

The digital signature is an operation with which a cryptographic code is generated that demonstrates the identity and integrity of a document. In other words, the digital signature makes it possible to verify that the document:

- Was signed by an identifiable person;
- Has not been altered since signature.

The digital signature is based on cryptographic algorithms that require the possession, by the user, of a private key and a corresponding certificate.

Figure 3- Smart card

Figure 4 - USB Token

The private key and certificate are normally stored on a credit card-like electronic device, called a smart card, or on a USB token (both cases are microchips with cryptographic functionality).

When generating your signature, you must type the PIN of your smartcard or USB device.

After generating a digital signature, it is usually saved in a file called a cryptographic envelope; the envelope normally also contains the document of departure and the certificate of the signee, so as to keep together all the information necessary for verification.

There are several crypto envelope formats[1]; the most widespread is the one known as PKCS# 7 (if so, the file has the extension P7M).

In order for the digital signature to have full legal value (here we are talking about qualified signature), different statutory rules that set requirements relating to the keys, certificate, smartcard, certifier, cryptographic envelope format, etcetera, must be complied with.

Digital signature with asymmetric encryption

To easily understand what a digital signature is, we will try to explain it with the help of the image.

Objective: Alice wants to be sure that the document she received (in clear) comes from Bob and has not been modified.

Initial conditions: Bob has a pair of keys, a private one known only to the owner and a public one known to everyone.

[1] According to Implementing Decision (EU) 2015/1506, CAdES (*. p7m), PAdES (*. pdf), XAdES (* .xml) formats are accepted.

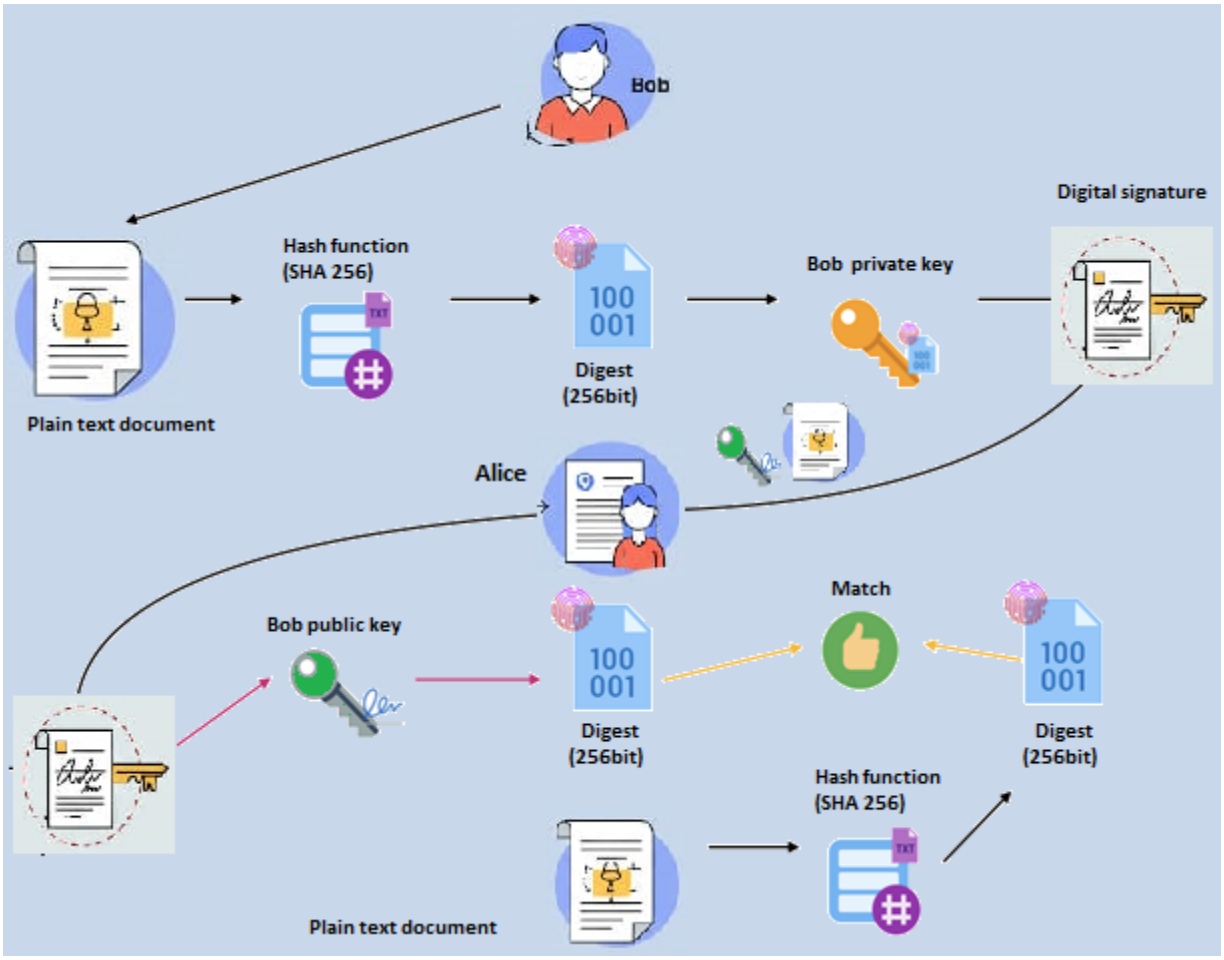


Figura 5 Digital signature in detail

1. Bob applies the hash algorithm (e.g., SHA256) to the clear message, thus obtaining the **digest**, which represents the so-called 'digital fingerprint' of the document, that is, a unique and compact representation of the original information contained in the document.
2. He uses his private key to encrypt the **digest**, thus obtaining the digital signature and authentication.
3. He sends Alice the clear message, the signature(i.e., the encrypted digest), and his public key.
4. Alice applies Bob's public key to the signature, thus obtaining the digest.
5. Alice applies the same hash algorithm (SHA256) to the plaintext message, obtaining the message digest in turn.
6. Alice compares the newly obtained digest with the one received from Bob. If the digests are the same, the signature has been verified, and we are sure that the message has not been altered.

NB. Remember that if the message changes, the digest is no longer the same.

3. Link and contextual menu

To access the procedure, you can use the desktop icon created during the tool installation or alternatively through the operating system's search functionality.

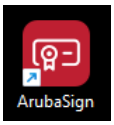


Figura 6 Desktop icon

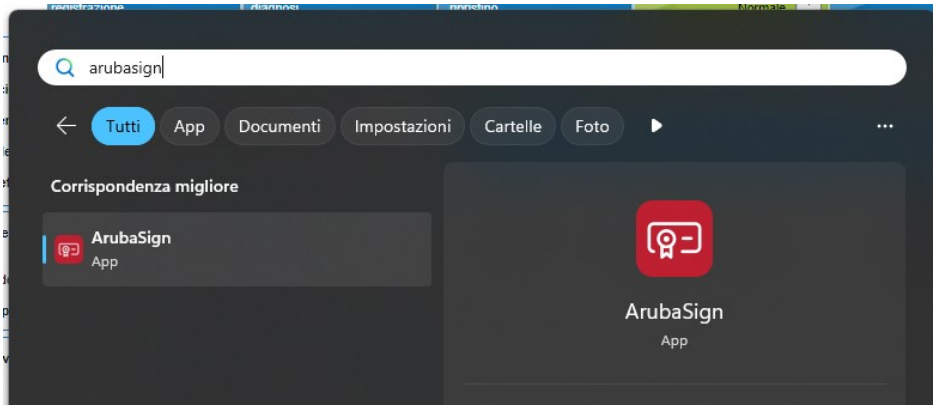


Figura 7 Search for the tool using Windows search

The functionalities of ArubaSign, once the software is installed, can also be accessed from the context menu that appears by right-clicking on the name of a file.

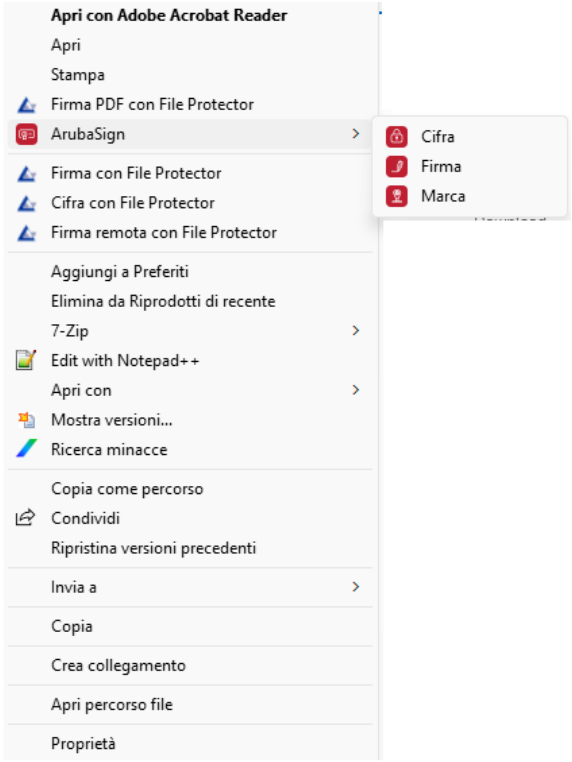


Figura 8 Aruba Sign contextual menu

4. First Startup and Basic Layout

The ArubaSign tool presents itself at the first start with a small *wizard*, where it features a renewed *layout*, a navigation menu with TAB, and a redesigned graphic signature affixing mode.

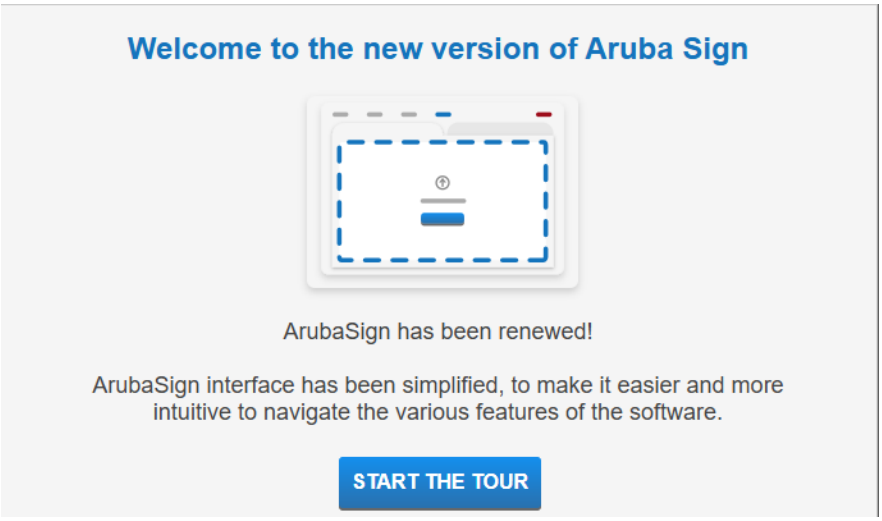


Figure Welcome 9 screen at first start

4.1. Functionality

This menu enables TABs to be enabled and disabled. You can enable the following panels:

- Encryption
- Decryption
- Tag (temporal)
- Encrypted file recipients

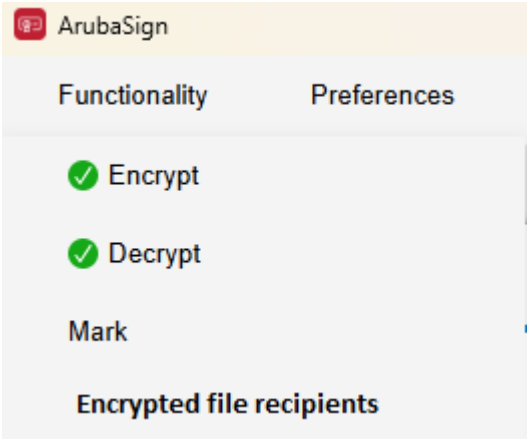


Figure 10 List of enablable features

The individual functionalities will be analysed in detail below.

4.2. Preferences

The preferences panel has the following categories:

- General
- Functionality
- Signature
- Graphic Signature PAdES
- Verification
- Card management
- Certificate database
- Advanced

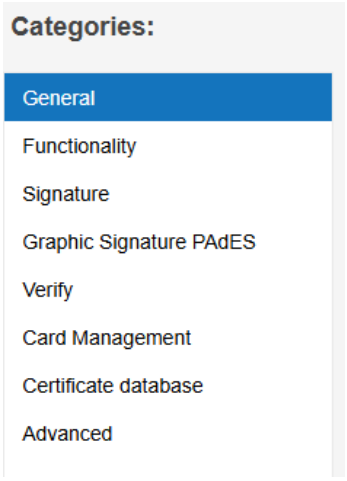


Figure 11 Personalization categories of tool

4.2.1. General

In this panel you can choose the language *layout*, select the destination folder of the signed files, and eventually open it automatically after you have affixed the signature.

The time zone of the date used during the signature can also be configured. Normally it should be UTC, but setting the *timezone* in the figure aligns with the one used in Italy.

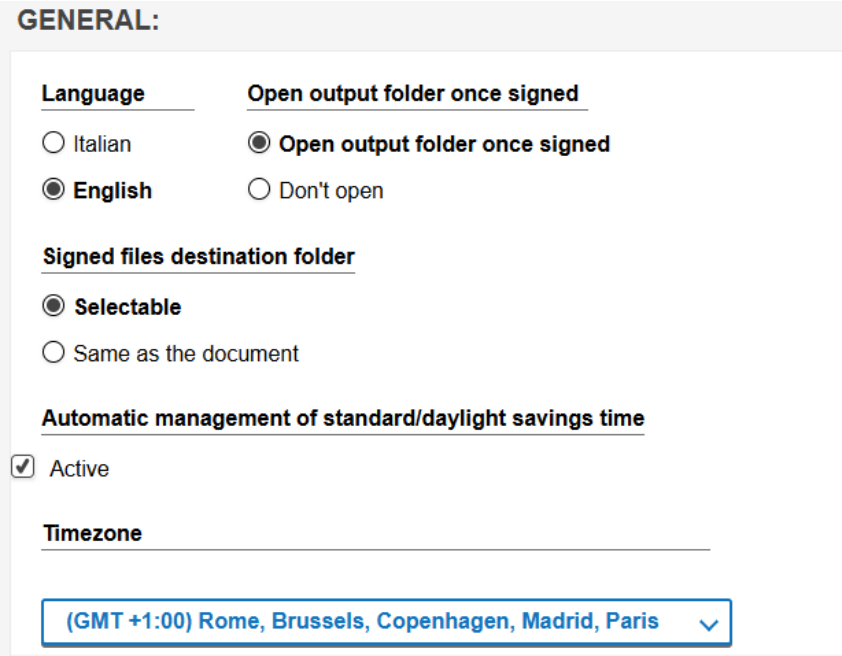


Figure 12 Customizations of a general kind

4.2.2. Functionality

This panel enables the TABs of Encrypt, Decrypt and time mark, to be set to enabled automatically upon start-up.

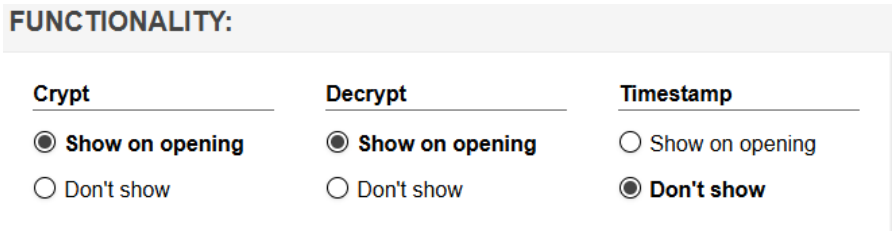


Figure 13 Customizing Tabs/Features to show at startup

4.2.3. Signature

This panel sets the signature preferences, understood as the signature format according to the file typology, be it PDF or XML. At the bottom, the option of enabling the time-mark for each signature affix.

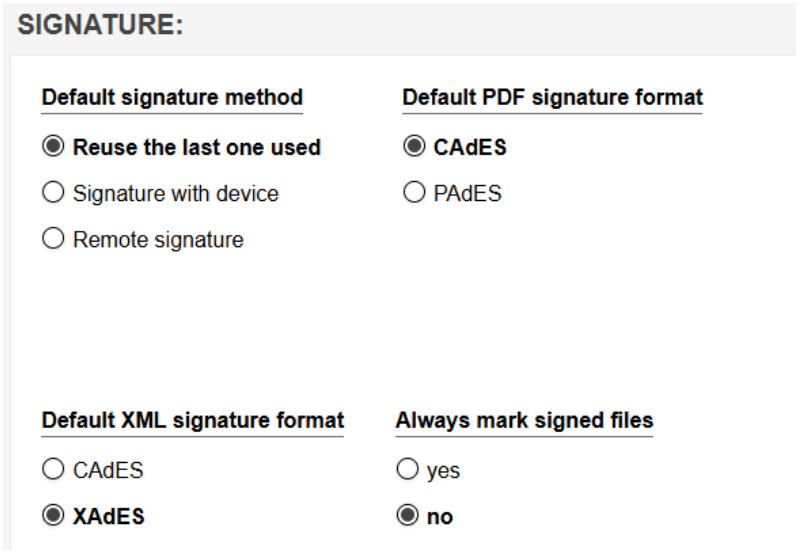


Figure 14 Customisations for digital signature

4.2.4. PAdES Graphic Signature

The graphic PAdES signature is configurable in detail, enabling an image that can also be customized, adding to the glyph also attributes such as date, location and motif.

Finally, for compatibility aspects, it is possible to maintain the PDF/A documental format.

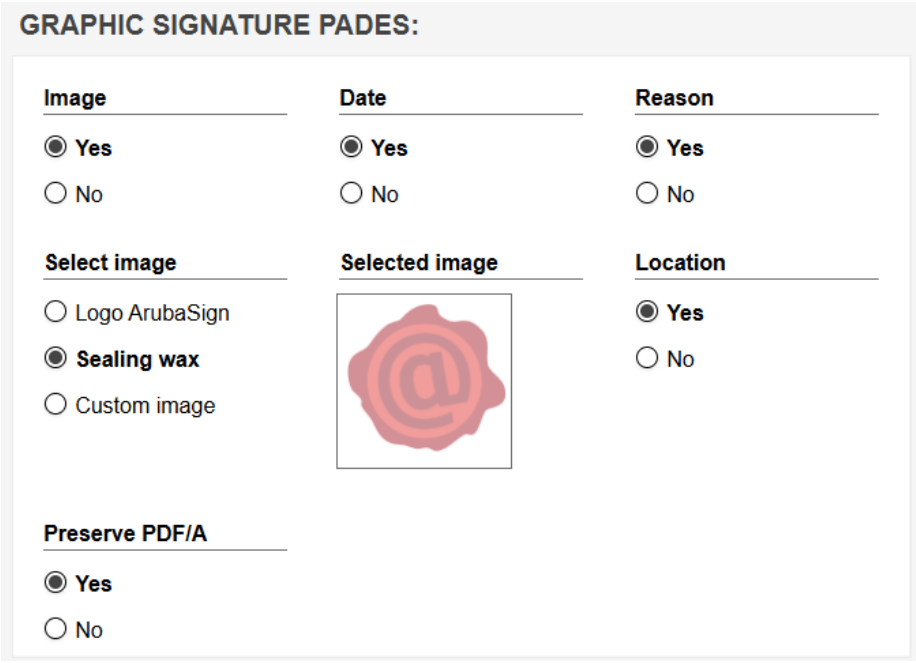


Figure 15 Customizations of PAdES signatures

4.2.5. Verification

In this panel you can customize the user experience when verifying a signed file. Note the ability to also enable the FEA verification (Firma Elettronica Avanzata (advanced electronic signature), non-skilled) and the status verification mechanism of a certificate (OCSP/CRL).

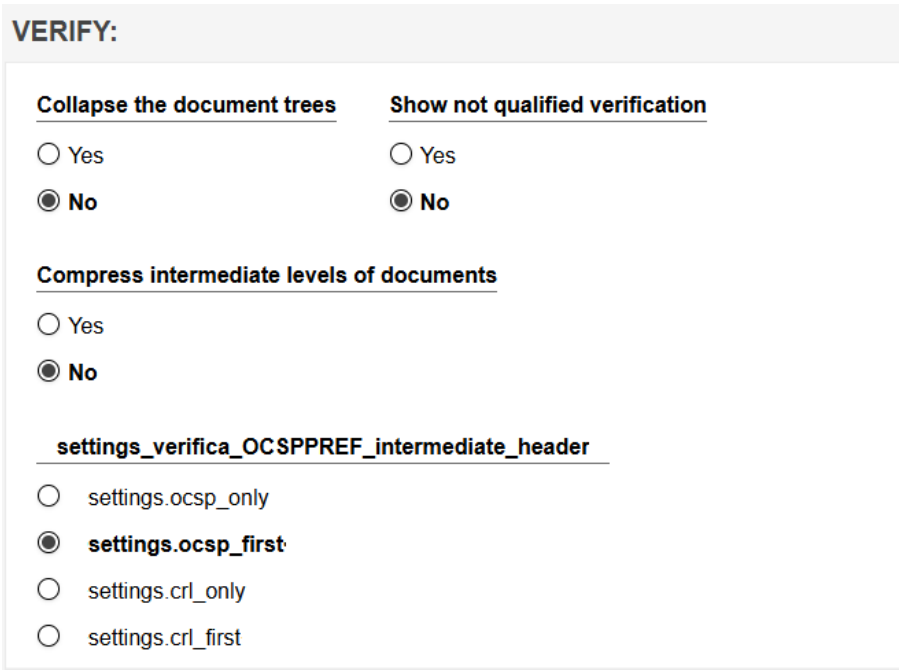


Figure 16 Options for verifying a signed file

4.2.6. Card management

This panel is of great utility in the management of the cryptographic device (smart card), in cases of PIN change, unblocking and PUK change.

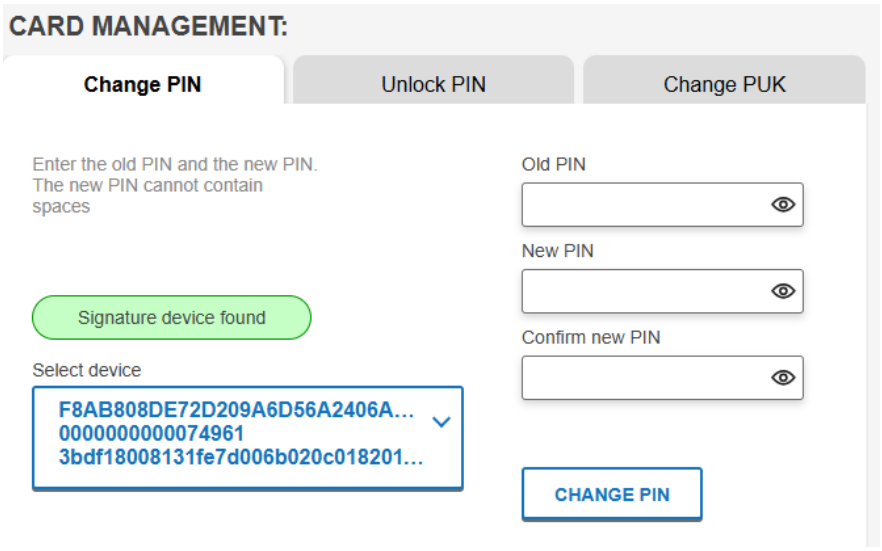


Figure 17 Form for modifying PIN

4.2.7. Certificate database

This panel shows the list of certificates contained in its internal Database. Certification Authorities (CA) certificates, including those contained in the *trusted list* eIDAS, can be distinguished, but personal or third-party certificates that can be used in file encryption may also be uploaded. You can import from both files and from Bank of Italy LDAP servers.

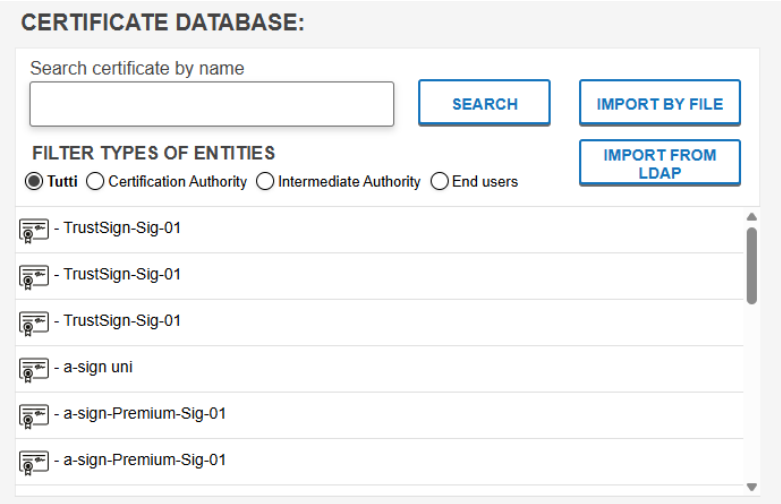


Figure 18 Repository of the reliable certificates numbered in the tool

4.2.8. Advanced

In this section you can enable the tool log, possibly with an increasing verbosity level and the ability to export it.

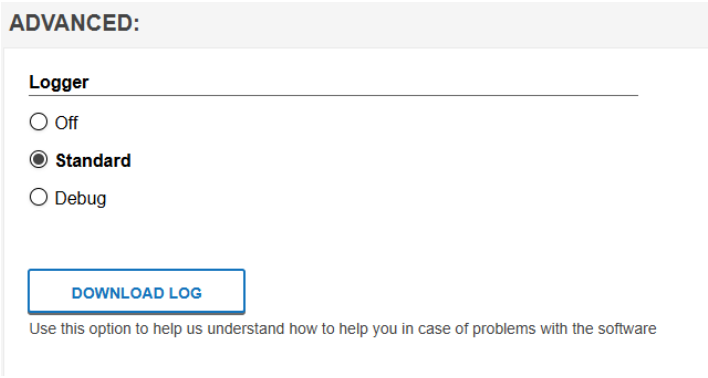


Figure 19 Enable Logger

4.3. Support

In this section you can navigate between the support options for using the *tool*. Through the Videoguide button the manufacturer's site will be opened with useful videoguides to support the various operations available; the "OnLine Guide" option is a link to this manual, while the final button refers to the small *wizard* that was presented at the first start of the *tool*.

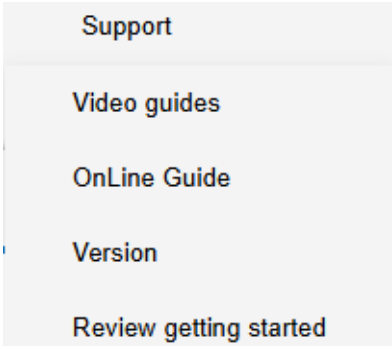


Figure 20 List of ways to support the use of the product

5. Sign and verify a file

To sign a file, you must have at least one valid certificate of signature on your smart card.

If the smart card contains more than one certificate, the desired certificate must be selected at the signature stage.

You can start the digital signature of a file in three different ways, described below:

- From outside the application, through the Windows contextual menu;
- From within *ArubaSign*, by means of the "*Drag & Drop*" functionality;
- From within *ArubaSign*, selecting the file from a folder.

Any of the methods highlighted will initialize the signature TAB panel, as shown in the figure.

Depending on the type of the file, the drop-down menu "Select signature format" will be enhanced with all possible compatible types (e.g. [2] P7m for all file types, PAdES for PDF and XAdES files for XML files).

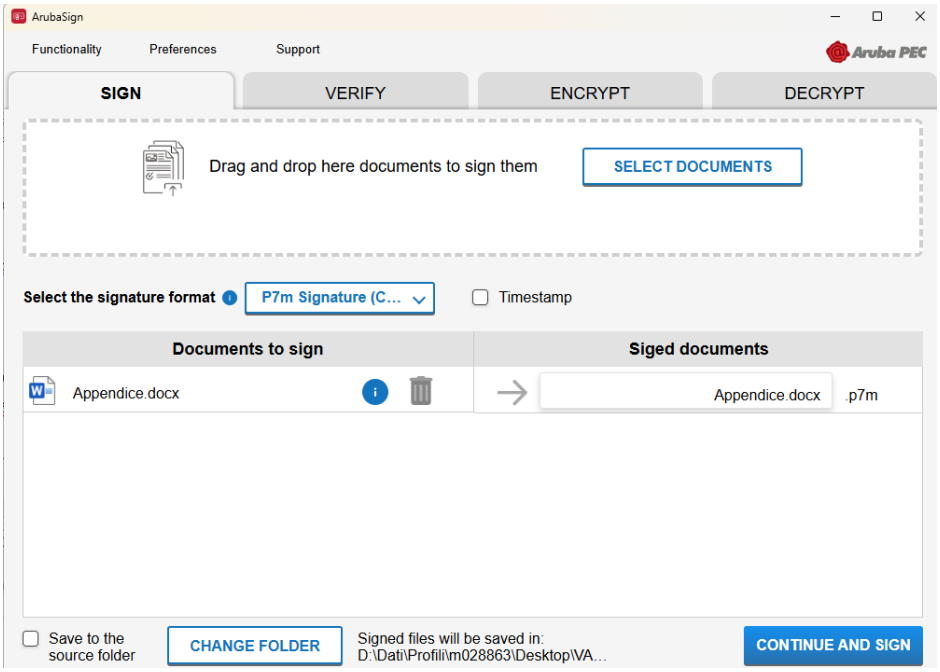


Figure 21 Signature affix panel

It shows that:

- It is possible edit contextually also the name of the signed file and the customization of the path where the file will be saved;
- Arbitrarily add other files to perform massive/multiple signatures in a single block as well.

[2] In some scenarios the signature PAdES is also proposed, suggesting an implicit conversion to PDF before signing.

5.1. Graphic PAdES signature

In cases where a graphic signature can be affixed, the *tool* automatically proposes it.

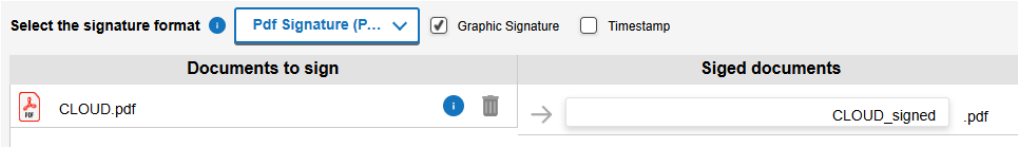


Figure 22 Evidence of the possibility of affixing a graphic signature

Continuing in the next step, a preview of the document will be proposed using the "continue" button, in which the area to be used to affix the digital signature glyph will be selected with the mouse.

Clicking on the Options menu, to the right of the panel, will show the customization features of that glyph.

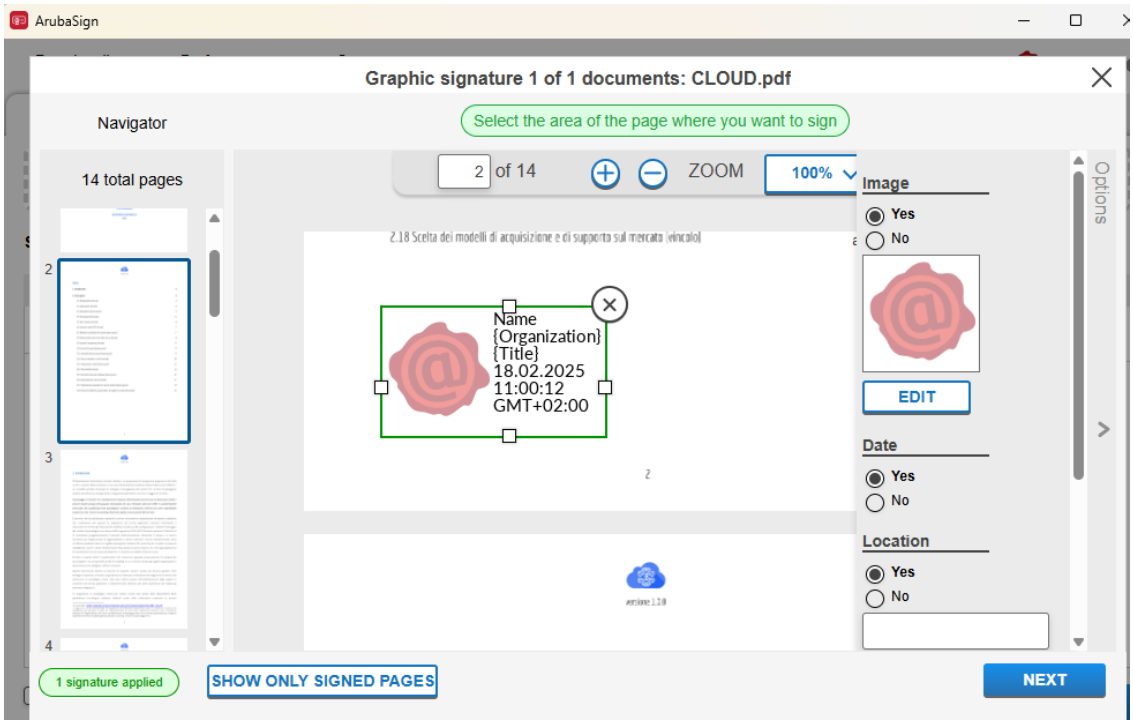


Figure 23 Selection of the area where the digital signature glyph will be affixed

To complete this step, and after clicking on "Continue and Sign", the panel in the figure will appear to select the signature mode: Remote Signature or Signature with Device (smart card/token).

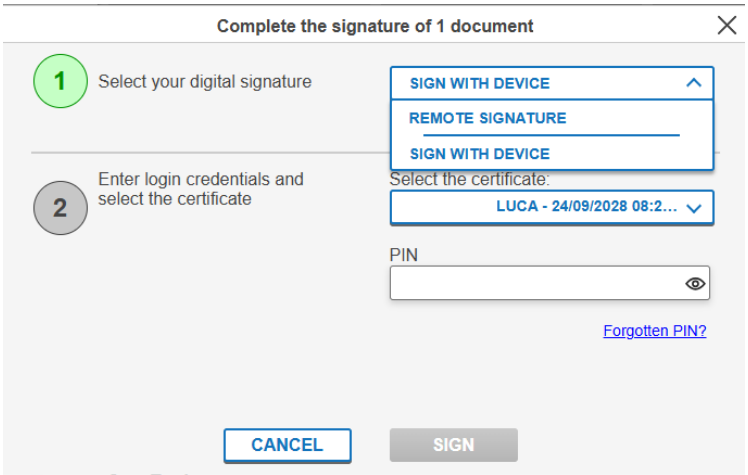


Figure 24 Form for selecting digital signature mode

When signing with the device, the drop-down menu is automatically populated with the signature certificates available and once you have completed the PIN field, you can proceed with the Signature.

Once the signature is successful, the pop-up appears in the figure. If the option has been selected, the pop-up will be opened after the signature. The pop-up appears in the figure.

Status 25 popup Figure for correct signature affixing

5.2. Multiple signatures

Several digital signatures may be applied to the same document, referred to as "multiple signatures". This allows to demonstrate that more people have assumed authorship and/or responsibility for the document, possibly at different times, as well as often happens in the case of the traditional autograph signature (consider contracts, budgets, etc.).

There are three types of multiple signatures:

- "matrioska" type signatures;
- Parallel signatures (also known as independent);
- counter-signatures (also called nested).

5.2.1. "Matrioska" signatures

The first type is obtained by simply signing a P7M cryptographic envelope (which contains an already signed document). This digital operation equates, in the world of paper, to signing the envelope that contains a signed document, which in reality sometimes done (i.e. envelopes containing bids in response to calls for tenders). To carry out a "matrioska" signature simply select a signed file and resubmit the signature. Each signature will add a layer to the file, as highlighted in the following paragraph.

5.2.1.1. Verification of "matrioska" signatures

In the example we will see how an additional level of signing is added to a file, signed multiple times.

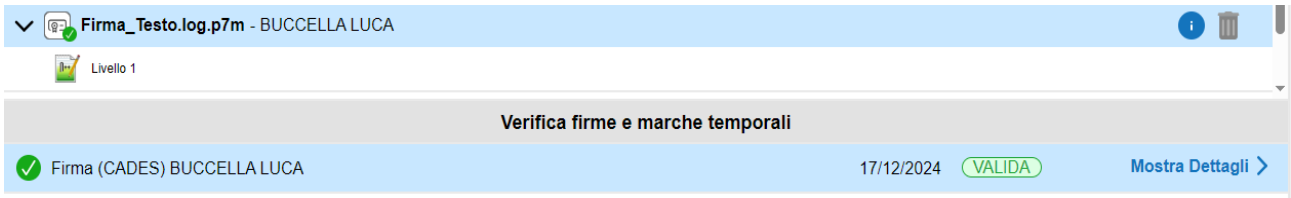


Figure 26 File signed digitally only once

In this example the file is signed 2 times (2 levels), with the evidence of the signee at each node in the tree structure representing the signatories.

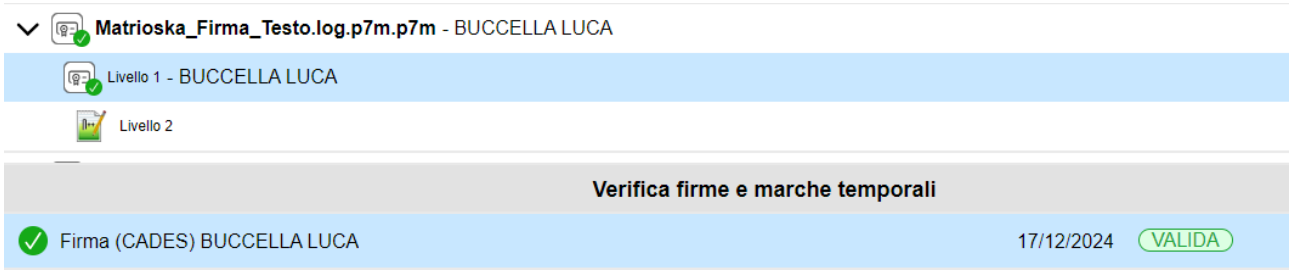


Figure 27 Digitally signed file twice

With this approach you can add N signature levels.

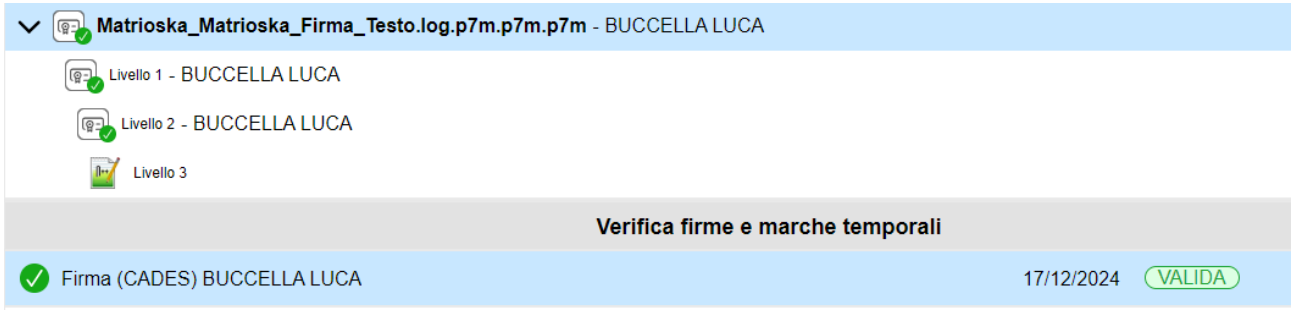


Figure 28 Digitally signed file three times

5.2.2. Parallel Signatures

The second type of multiple signature (known as parallel or independent) consists of adding additional "side" signatures to the first one, where each signature maintains its independence (each signer signs the same data as the others sign). This digital operation equates to, in the world of paper, to affixing multiple signatures by different persons at the end of the same document. To add an independent signature, click on the "Add Signature" button in the window shown during verification.

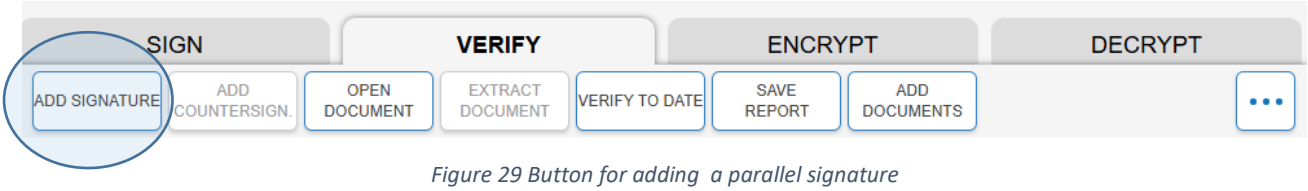


Figure 29 Button for adding a parallel signature

5.2.2.1. Verifying Parallel Signatures

In the example we can notice how a multi-signed file has only one signature level, but two distinct signatures. The screen shows that the two signatures are at the same level; therefore, both signees have signed the original document independently.

30 Figure Verifying parallel signatures

5.2.3. Nested signatures

The third type of multiple signature (called counter-signing or nested) is obtained by signing an existing signature and preserving the result (called counter-signature) within the same envelope. Doing so, the second signee in practice approves or "validates" the first signature. In turn, the second signature can be signed by a third person, and so on.

To add a counter-signature, select a signature in the detail section of the specific signature and select the "**Add counter-signature**" button. During verification, it can be seen that the document contains the counter-signature (note the tree representation in the next paragraph).

5.2.3.1. Nested Signatures Check

Also in this example, as in the case of parallel signatures, we can note that there is only one level of signature, but differently from the previous one there is a relationship between the two signatures: the second depends on the first going to highlight that the counter-signature expresses the will to sign the main document and the first digital signature affixed.

Figure 31 Nested Signature Check (note evidence "against signature")

6. Cipher a file

If you need to encrypt a file, simply enable the specific TAB, drag the file using the *Drag & Drop* technique or select it using the "Select documents" button.

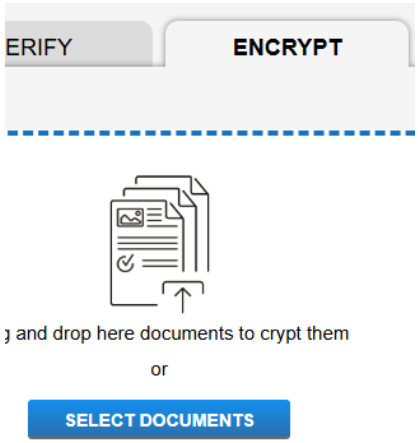


Figure 32 Panel for ciphering files

Once selected, the files will be collected in the table, with the option to add others if necessary, via the highlighted button.

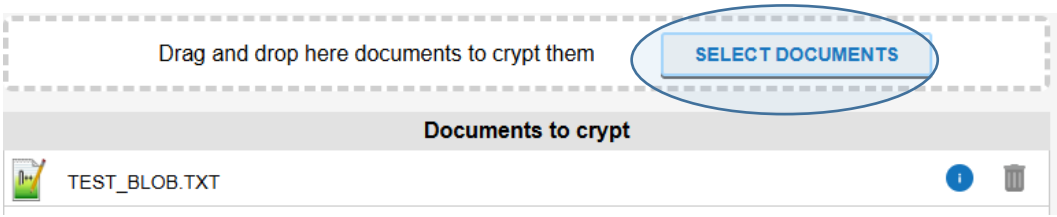


Figure 33 Inserting additional files to be encrypted

Finally clicking on the "**Continue and Crypt**" button will appear the panel for the selection of certificates to be used for the encryption of the file.

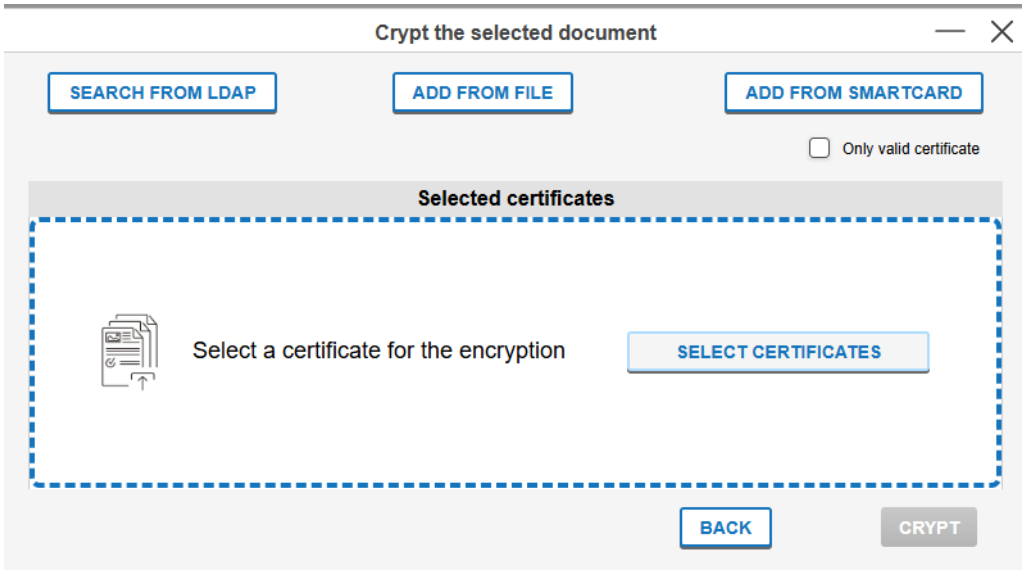


Figure Selection 34 options of the cipher certificate to be used

You can select the certificate in the following modes:

- Searching Bank of Italy's LDAP servers such as the *domain controller* (*Active Directory*) or PKI *servers*;
- Add a file by selecting it manually, either via the "Add from file" button or via "Select certificates";
- Adding it from your badge/smart card.

NB: By selecting the "Valid certificates only" option, only those that comply will be highlighted and used.

Figure 35 Filter of valid certificates only among those selected

6.1. Encrypt a file for a colleague

In case you want to encrypt a file for a colleague, simply select the "Search from LDAP" button from the panel in the figure.

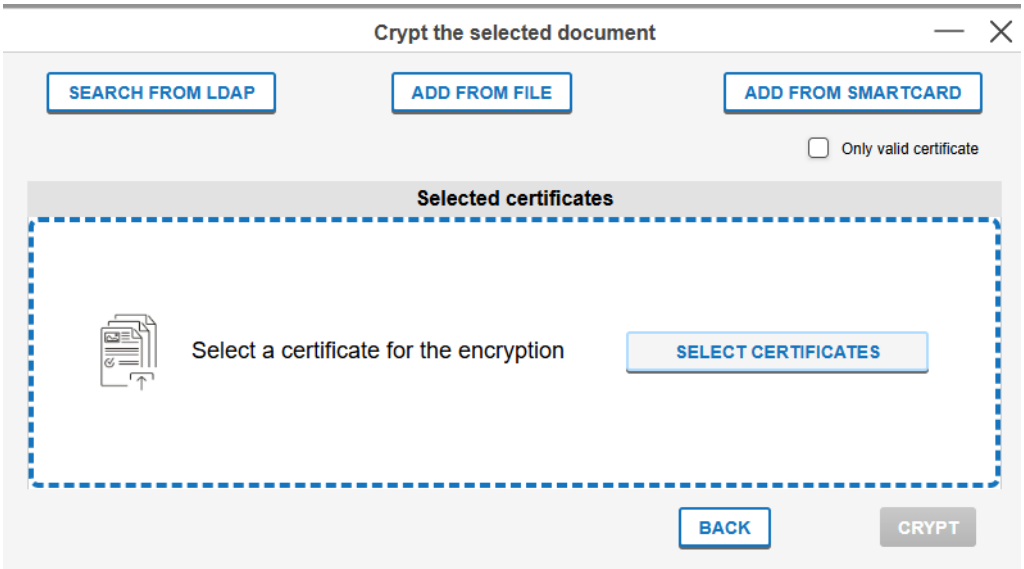


Figure Source preselection 36panel of the encryption certificate to use

Next, from the search window, enter the search parameters and click on "Search". The list of available certificates that comply with the parameters entered will appear in the table below.

Use the relative button to select the certificate to be used.

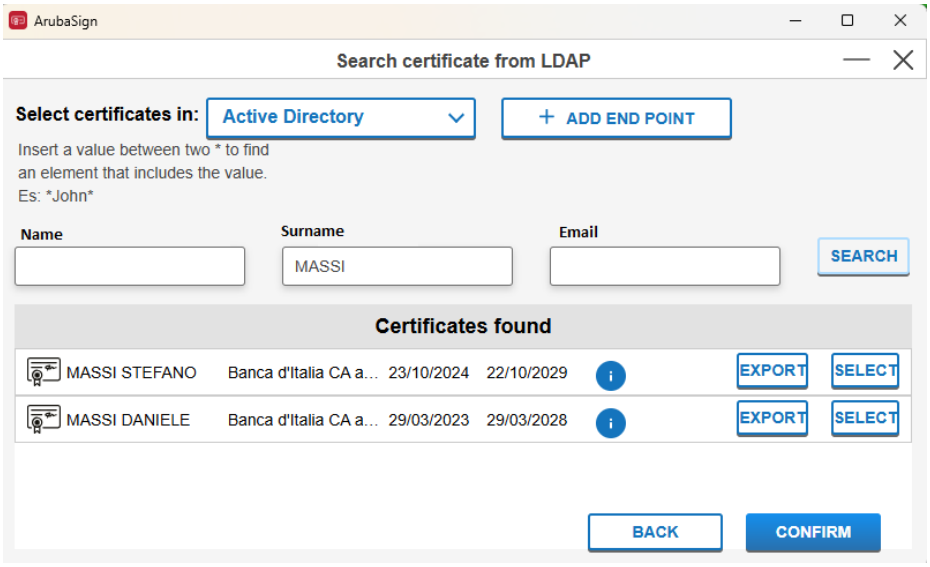


Figure 37 Select certificate after LDAP search

You can perform subsequent searches; the *tool* will store the list of "selected" certificates and will then summarize them in the table at the end.

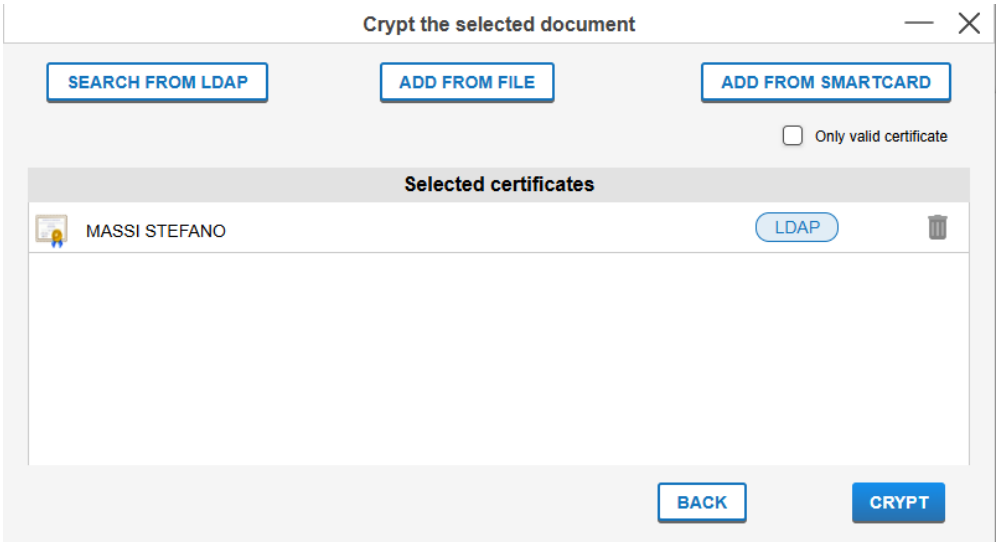


Figure 38 Summary of certificates for which encryption will be carried out

The process will end after pressing the "Crypt" button with the pop-up in the figure.

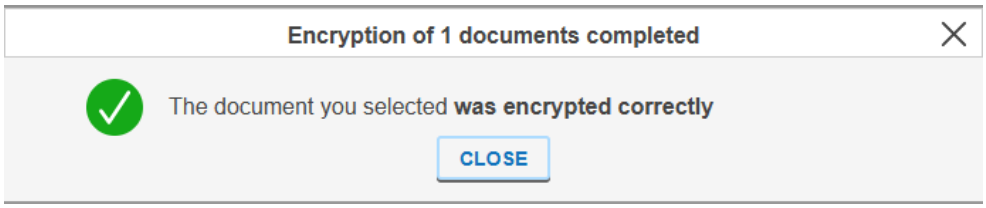


Figure 39 File encryption popup

7. Decryption of a file

If you need to decrypt a file, simply enable the specific TAB, drag the file using the *Drag & Drop* technique or select it using the "Select documents" button.

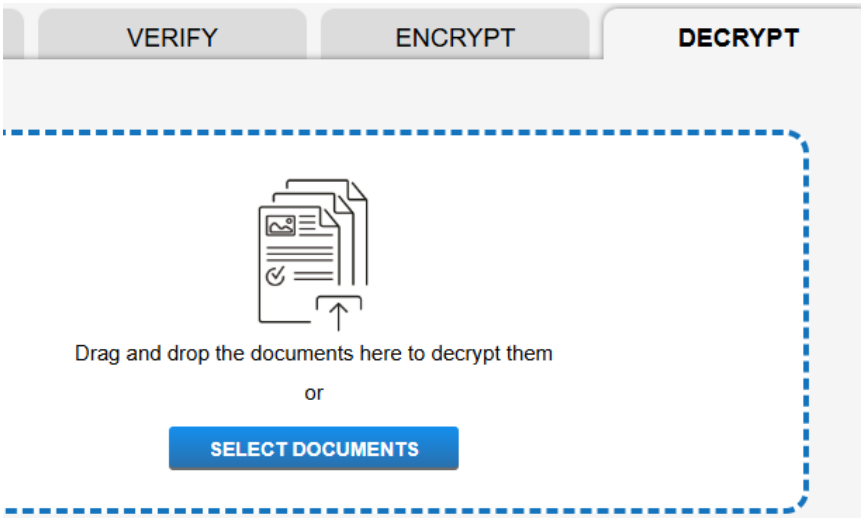


Figure 40 Selection Tab "Decrypt"

After selecting the files/files a summary window will be proposed with the selected files to which you can add or remove other files.

Clicking on the "**Continue and Decrypt**" button will propose a summary window of the files grouped by certificate (serial number and *Certification Authority* are given for each of the certificates used for ciphers the specific file).

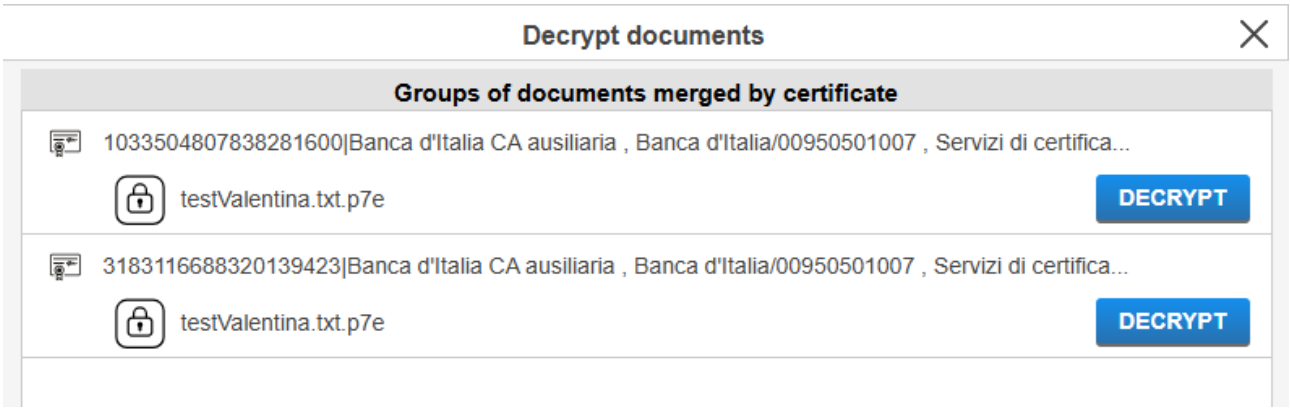


Figure Summary 41 table of certificates for which the files have been encrypted

Clicking on the "**Decrypt**" button will propose which container to use to decrypt (smart card or pkcs12 file). In case no certificate is found useful to decryption, among those made available, an error window will be shown as in the figure.

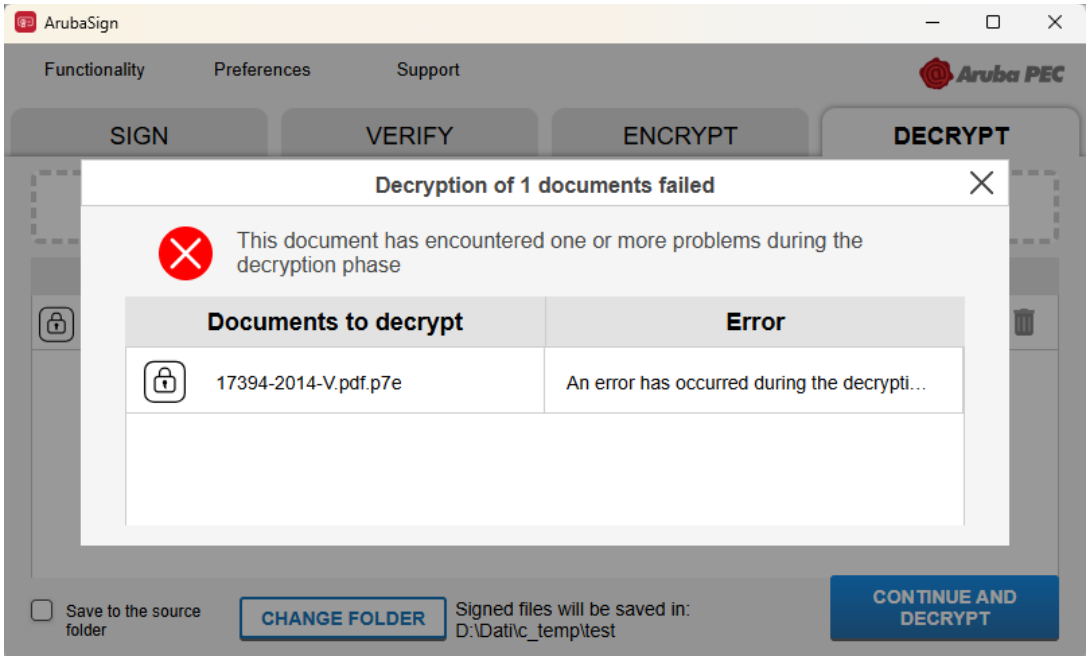
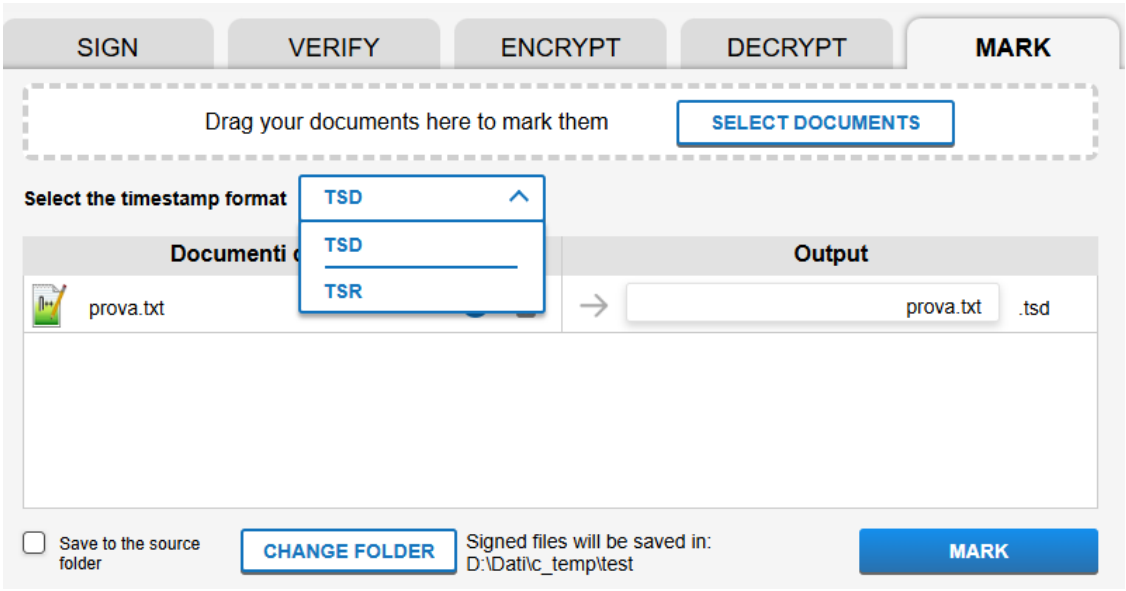


Figure 42 Deciphering error popup

8. Temporally tag a file

In case you need to temporally tag a file, simply enable the specific TAB, drag the file using the *Drag & Drop* technique or select it via the "Select Documents" button. Then, from the summary window you will be able to choose the specific format [3] and proceed with the affixing of the tag.



43 Figure Panel added temporal tag

9. Encrypted file recipients

The *tool* also makes available the recovery functionality of the user list for which a file has been encrypted, via the "**Functionality**" menu and the "Encrypted file recipients" link.

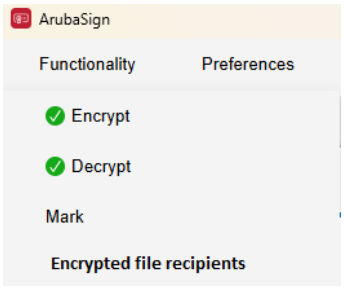


Figure 44 Function selection curtain "Encrypted file recipients"

[3] **TSR format**: it is the simplest format and contains only the imprint of the file, NOT the whole file, and the computer evidence of the marking made. To verify a TSR file you need to have the original file that you have tagged.

[3] **TSD Format**: it is a format that contains both the evidence of the time mark (the file with TSR format) and the file itself subjected to the marking, for this reason the file itself can be subjected to verification procedure as it contains all the information necessary to check.

By selecting a file, or a specific folder, the *tool* will retrieve for each file the list of names (including serial number of certificate) for which the specific file has been encrypted. The current status of the certificate (optionally also with verification via CRL) will also be shown.

Figure 45 Extract table of recipients of an encrypted file