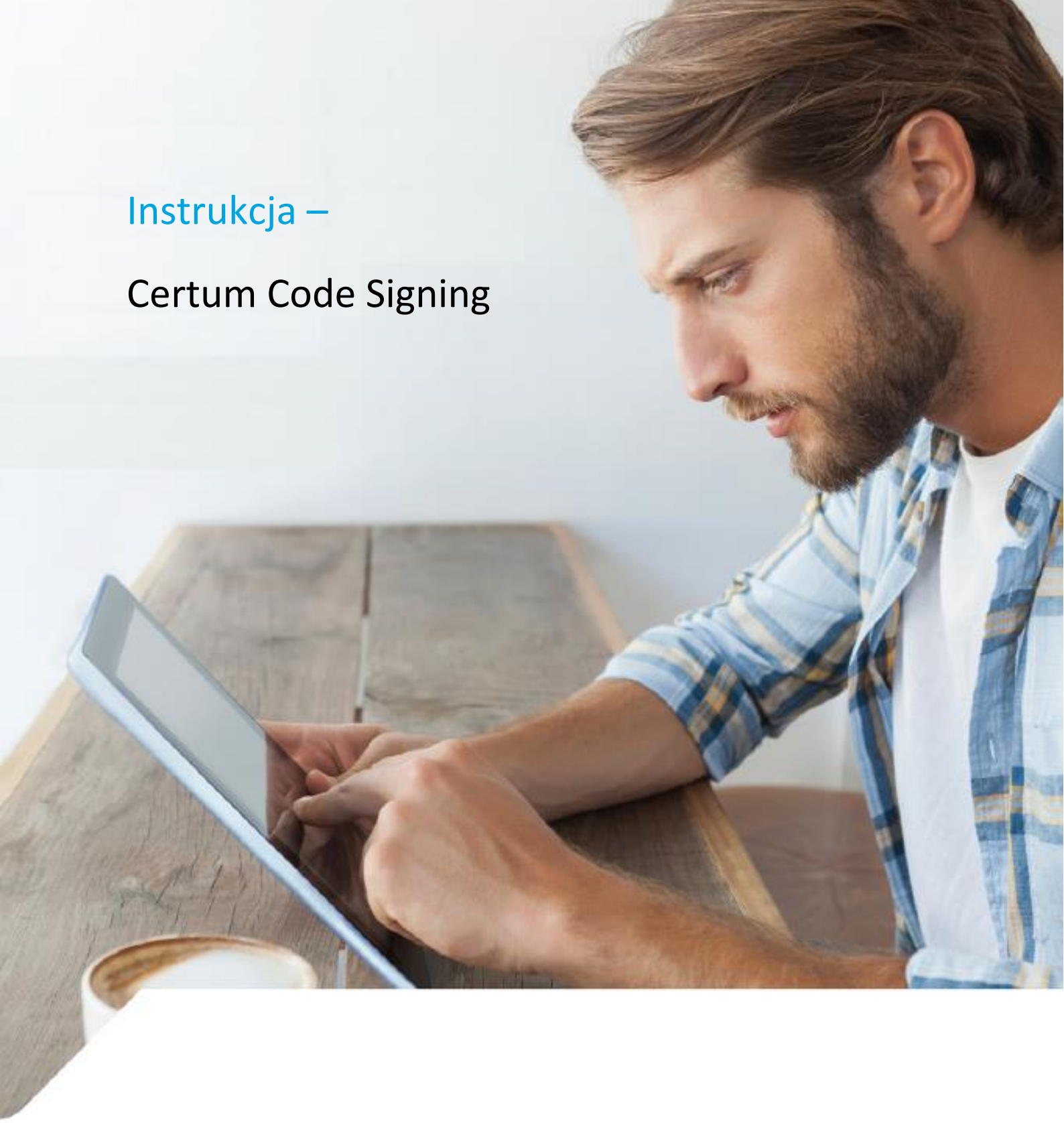


Instrukcja –

Certum Code Signing

Narzędzia do podpisywania certyfikatem Certum Code Signing

wersja **2.2**

Certum by asseco

Spis treści

1. Opis produktu

Certyfikaty Certum Code Signing zabezpieczają oprogramowanie przed nieautoryzowaną zmianą lub naruszeniem przez osoby trzecie. Często może się zdarzyć, że software pobrany z sieci web będzie traktowany przez komputer jako złośliwe oprogramowanie. Wynika to z faktu, iż nie posiada on certyfikatu wydanego przez autoryzowany urząd certyfikacji jakim jest Certum.
Zabezpieczając oprogramowanie certyfikatami Certum Code Signing można ochronić swoje oprogramowanie przed nieuprawnionym dostępem i kradzieżą oraz zminimalizować ryzyko występowania komunikatów ostrzegawczych z SmartScreen ® Application Reputation.

Oprogramowanie zabezpieczone certyfikatami Certum Code Signing spowoduje wzrost bezpieczeństwa oraz zaufania klientów a co za tym idzie większą liczbę pobrań oprogramowania. Code Signing, służy do podpisywania kodu oraz gotowych już zbudowanych używając znanych narzędzi takich jak signtool.exe oraz jarsigner.

2. Signtool

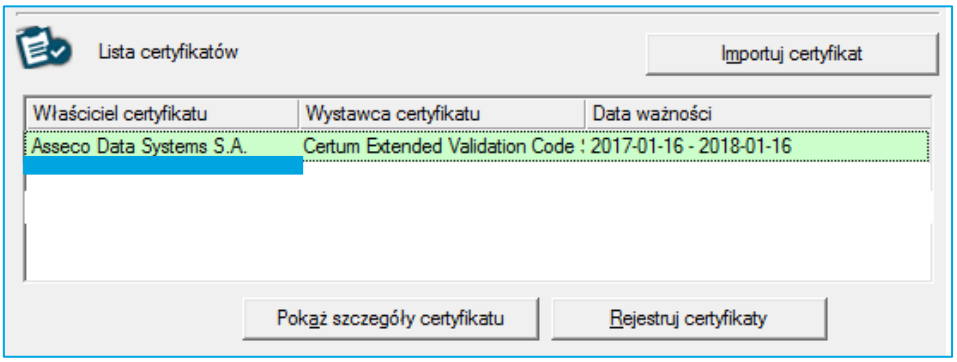
2.1 Opis narzędzia

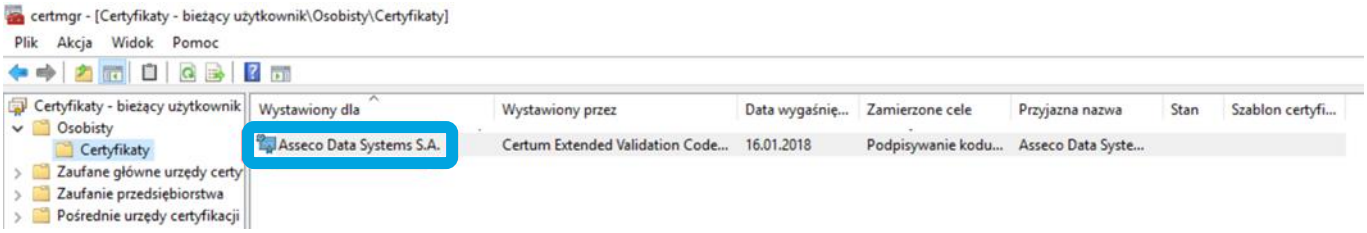
Signtool to narzędzie wiersza polecenia, które cyfrowo podpisuje pliki, weryfikuje podpisy w plikach i oznacza pliki znacznikami czasu. Narzędzie to znaleźć można w paczce deweloperskiej Windows (Windows SDK[Software Development Kit]). Wszystkie operacje wykonywane z Code Signing wymagają podłączonego czytnika wraz z kartą na której jest certyfikat Code Signing.

2.2 Podpisywanie

Aby podpisać plik, w wierszu poleceń (cmd.exe) należy użyć następującego polecenia:
signtool sign /n "[1] " / t [2] /fd [3] /v [4]

[1] – Nazwa lub fragment nazwy właściciela certyfikatu, którą sprawdzić można w aplikacji proCertum CardManager lub narzędziu systemowym certmgr.msc:





[2] – Adres znacznika czasu. Dla Certum http://time.certum.pl,
[3] – Nazwa algorytmu podpisu. Dostępne sha1 i sha256,
[4] – Ścieżka do podpisywanego pliku.

Przykładowe, poprawne polecenia:
signtool sign /n "Asseco Data Systems S.A." /t http://time.certum.pl/ /fd sha1 /v aplikacja.exe

W rezultacie konsola cmd.exe powinna zwrócić komunikat o poprawności podpisu pliku

```
The following certificate was selected:
    Issued to: Asseco Data Systems S.A.
    Issued by: Certum Code Signing CA SHA2
    Expires:   Fri Jul 06 10:16:38 2018
    SHA1 hash: E0828DF9D71C4CD87A349460027F0D9CB802BF31

Done Adding Additional Store
Successfully signed: aplikacja.exe

Number of files successfully Signed: 1
Number of warnings: 0
Number of errors: 0
```

signtool sign /n "Asseco Data Systems S.A." /t http://time.certum.pl/ /fd sha256 /v aplikacja.exe

W rezultacie konsola cmd.exe powinna zwrócić komunikat o poprawności podpisu pliku:

```
The following certificate was selected:
    Issued to: Asseco Data Systems S.A.
    Issued by: Certum Code Signing CA SHA2
    Expires:   Fri Jul 06 10:16:38 2018
    SHA1 hash: E0828DF9D71C4CD87A349460027F0D9CB802BF31

Done Adding Additional Store
Successfully signed and timestamped: aplikacja.exe

Number of files successfully Signed: 1
Number of warnings: 0
Number of errors: 0
```

2.3 Weryfikacja

Aby zweryfikować plik, w wierszu poleceń (cmd.exe) należy użyć następującego polecenia:

signtool verify /pa [1]

[1] – Nazwa podpisanego pliku

Przykładowe, poprawne polecenie:

signtool verify /pa aplikacja.exe

W rezultacie konsola cmd.exe zwraca komunikat o poprawności podpisu pliku, przykładowo:

```
File: aplikacja.exe
Index Algorithm Timestamp
========================================
0     sha1      Authenticode

Successfully verified: aplikacja.exe
```

```
File: aplikacja.exe
Index Algorithm Timestamp
========================================
0     sha256    Authenticode

Successfully verified: aplikacja.exe
```

Lub brak podpisu:

```
File: aplikacja.exe
Index Algorithm Timestamp
========================================
SignTool Error: No signature found.

Number of errors: 1
```

2.4 Podpisywanie wsadowe

W celu wsadowego podpisania wielu plików podczas jednej sesji należy je podać jako kolejne parametry polecenia. Działanie takie eliminuje konieczność każdorazowego wywoływania komendy w konsoli oraz wpisywania kodu PIN przy podpisie kolejnych plików.

Przykładowe polecenie:

signtool sign /n "Asseco Data Systems S.A." /t http://time.certum.pl/ /fd sha1 /v aplikacja1.exe aplikacja2.exe aplikacja3.exe

W rezultacie konsola cmd.exe zwraca komunikat o poprawności podpisu plików:

```
Done Adding Additional Store
Successfully signed and timestamped: aplikacja1.exe
Successfully signed and timestamped: aplikacja2.exe
Successfully signed and timestamped: aplikacja3.exe

Number of files successfully Signed: 3
Number of warnings: 0
Number of errors: 0
```

2.5 Podpis dualny

W celu złożenia podpisu dualnego (wykorzystującego oba algorytmy: SHA-1 oraz SHA-2 należy przeprowadzić następującą procedurę:

1. Wykonać podpis aplikacji z wykorzystaniem algorytmu SHA-1 przykładowym poleceniem:

signtool sign /n "Asseco Data Systems S.A." /t http://time.certum.pl/ /fd sha1 /v aplikacja.exe

2. Następnie wykonać podpis tej samej aplikacji wykorzystując algorytm SHA-2 oraz przełącznik /as:

signtool sign /n "Asseco Data Systems S.A." /tr http://time.certum.pl/ /fd sha256 /td sha256 /as /v aplikacja.exe

Wynikiem weryfikacji pliku podpisanego dualnie powinien być następujący komunikat z konsoli:

```
File: aplikacja.exe
Index Algorithm Timestamp
========================================
0     sha1      Authenticode
1     sha256    RFC3161

Successfully verified: aplikacja.exe
```

Do wykonania i weryfikacji podpisu dualnego wymagany jest Windows 8 lub wyższy. W celu wykonania lub weryfikacji podpisu dualnego na systemach Windows 7 należy zapoznać się z artykułem opublikowanym przez Microsoft: https://technet.microsoft.com/en-us/library/security/2949927.

3 Jarsigner

3.1 Opis narzędzia

Jarsigner to narzędzie wiersza poleceń, które cyfrowo podpisuje pliki oraz weryfikuje podpisy. Narzędzie to znaleźć można w paczce deweloperskiej Oracle(JDK [Java Development Kit]). Wszystkie operacje wykonywane z Code Signing wymagają podłączonego czytnika wraz z kartą na której jest certyfikat.

3.2 Konfiguracja

3.2.1 Utworzenie pliku konfiguracyjnego provider.cfg

Przed rozpoczęciem używania jarsigner potrzebna jest dodatkowa konfiguracja. W pierwszym kroku należy utworzyć plik konfiguracyjny providera dla PKCS#11. W tym celu tworzymy nowy plik o rozszerzeniu *.cfg (przykład: provider.cfg). Jego zawartość wygląda następująco:

name=[1]
library=[2]
slot=[3]

[1] – Nazwa providera. Najlepiej Crypto3PKCS.
[2] – Ścieżka do biblioteki PKCS. Jeżeli posiadamy zainstalowanego proCertum CardManagera ścieżka domyślna to: C:\Windows\System32\crypto3PKCS.dll
[3] –Numer slota w którym znajduje się karta. Domyślna wartość to -1 która powoduje automatyczne wykrycie pierwszego dostępnego slotu.

Przykładowa konfiguracja dla profilu zwykłego karty cryptoCertum:

name=Crypto3PKCS
library=C:\Windows\System32\crypto3PKCS.dll
slot=-1

Przykładowa konfiguracja dla karty wirtualnej Certum:

name=SimplySignPKCS.dll
library=C:\Windows\System32\SimplySignPKCS.dll
slot=-1

3.2.2 Utworzenie pliku ścieżki certyfikatu bundle.pem

Kolejnym krokiem jest utworzenie pliku ścieżki certyfikatu o rozszerzeniu*.pem (przykład: bundle.pem). Jego zawartość wygląda następująco:

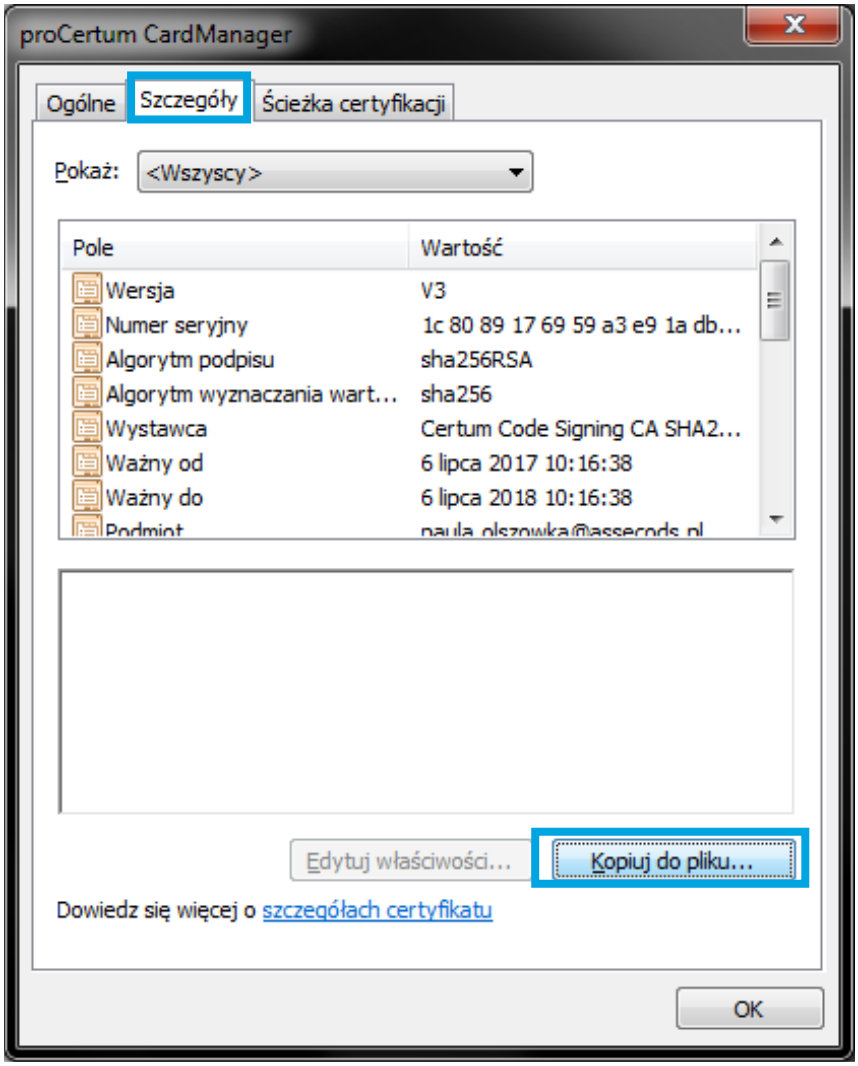
1. „Na górze”: Certyfikat użytkownika
2. „Poniżej”: certyfikat pośredni dla certyfikatu użytkownika

Uwaga: Zawartość pliku bundle.pem musi być koniecznie we wspomnianej wyżej kolejności.

Uzyskiwanie certyfikatu użytkownika

Certyfikat użytkownika może zostać uzyskany poprzez uruchomienie programu proCertum CardManager, kliknięcie przycisku czytaj kartę i przejście do zakładki Profil zwykły.

Następnie należy wybrać z listy certyfikat, który chcemy zapisać i użyć przycisku „Pokaż szczegóły certyfikatu”. Wyświetlony zostanie certyfikat, używając przycisku „Kopiuj do pliku” znajdującego się na karcie „Szczegóły” istnieje możliwość zapisania certyfikatu:

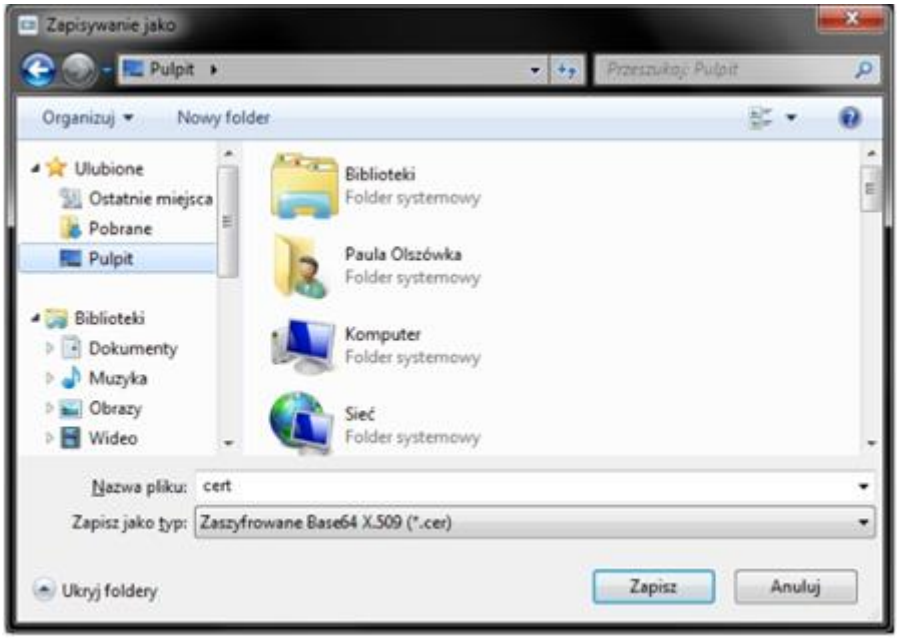


Warto w tym kroku zapisać sobie zawartość pola Wystawca. Pomoże to w późniejszym doborze certyfikatu pośredniego

Po kliknięciu „Kopiuj do pliku” zostanie uruchomiony kreator zapisu:

Należy kliknąć Dalej. W kolejnym kroku należy wybrać opcję **„X.509 szyfrowany algorytmem Base-64 (.CER)”** i kliknąć Dalej:

Następnie należy wybrać gdzie ma zostać zapisany plik oraz nadać mu nazwę. W tym celu należy kliknąć Przeglądaj, wybrać lokalizację i wpisać nazwę pliku, następnie kliknąć Zapisz, a w oknie kreatora przejść do kolejnego kroku przyciskiem Dalej oraz następnie Zakończ. Kreator potwierdzi eksport pliku.



Uzyskiwanie certyfikatu pośredniego

Certyfikaty pośrednie należy pobierać ze strony Certum:

https://www.certum.pl/pl/wsparcie/cert_wiedza_zaswiadczenia_klucze_certum/

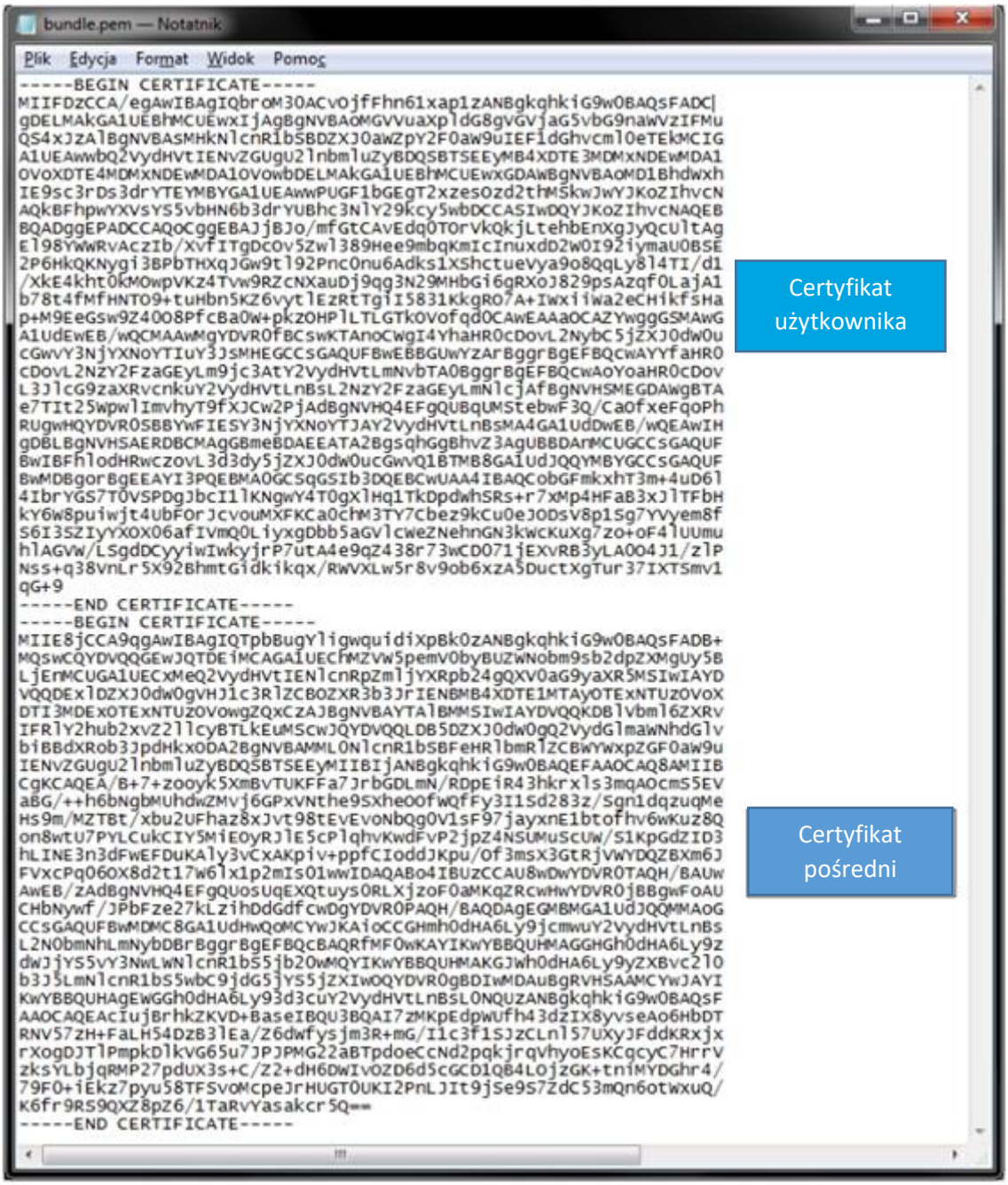
W doborze odpowiedniego certyfikatu (certyfikatów) pośrednich pomoże zapisana wcześniej nazwa Wystawcy z pola „Wystawca” certyfikatu użytkownika. Należy odszukać na stronie Certum wystawcę swojego certyfikatu i zapisać jego certyfikat w formacie tekstowym PEM.

Następnie mając dwa pliki z certyfikatami, należy utworzyć nowy plik tekstowy. Zawartość obu uzyskanych wcześniej plików (**Certyfikat użytkownika** oraz **certyfikat pośredni**) należy wkleić do jednego pliku tekstowego we wspomnianej wyżej kolejności:

1. „Na górze”: Certyfikat użytkownika
2. „Poniżej”: certyfikat pośredni dla certyfikatu użytkownika

Plik należy zapisać i zmienić jego rozszerzenie na *.pem.

Poniżej przedstawiono przykładowy plik bundle.pem:



3.2.3 Zmiana aliasu użytkownika (labela) na karcie (tylko dla użytkowników posiadających znaki diakrytyczne w polu Common Name (CN) w certyfikacie)

Do wskazywania certyfikatu, który ma zostać użyty do podpisu aplikacji z wykorzystaniem narzędzia Jarsigner używany jest alias certyfikatu użytkownika. Standardowo alias certyfikatu tworzony jest na podstawie zawartości pola Common Name z pola Podmiot z certyfikatu. Jeśli w polu Common Name umieszczony został ciąg znaków zawierający znaki diakrytyczne, następuje konieczność zastąpienia tego ciągu ciągiem znaków nie zawierającym znaków diakrytycznych.

Przykład:

Alias (label) przed zmianą: Urząd

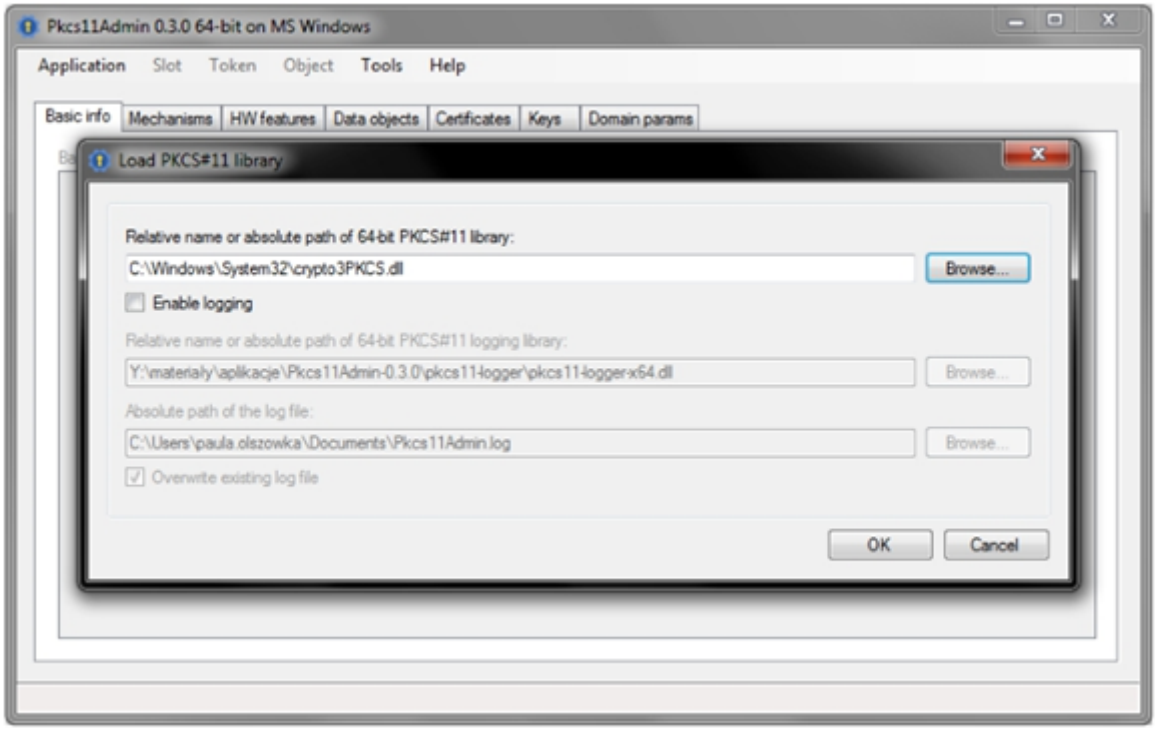
Alias (label) po zmianie: Urzad

Uwaga: Zmiana aliasu nie powoduje zmian w certyfikacie. Edycji podlega jedynie identyfikator certyfikatu na karcie. Podpisana aplikacja w polu podpisu nadal zawierać będzie dane Subskrybenta zawierające znaki diakrytyczne.

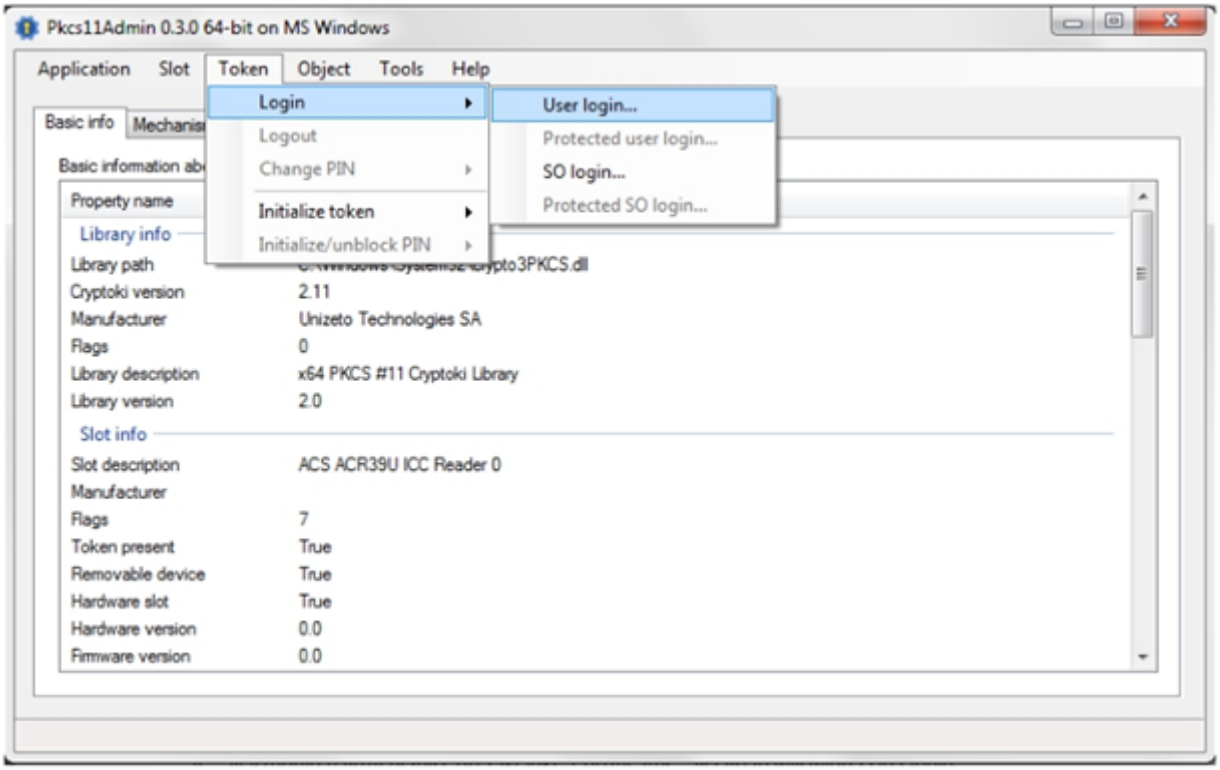
Zmiany aliasu (labela) można dokonać za pomocą oprogramowania udostępnianego nieodpłatnie PKCS11Admin, do pobrania tutaj: http://www.pkcs11admin.net/

Prezentowany sposób zmiany labela został wykonany w oprogramowaniu **PKCS11Admin** w wersji **0.3.0**. Procedura zmiany labela wygląda następująco:

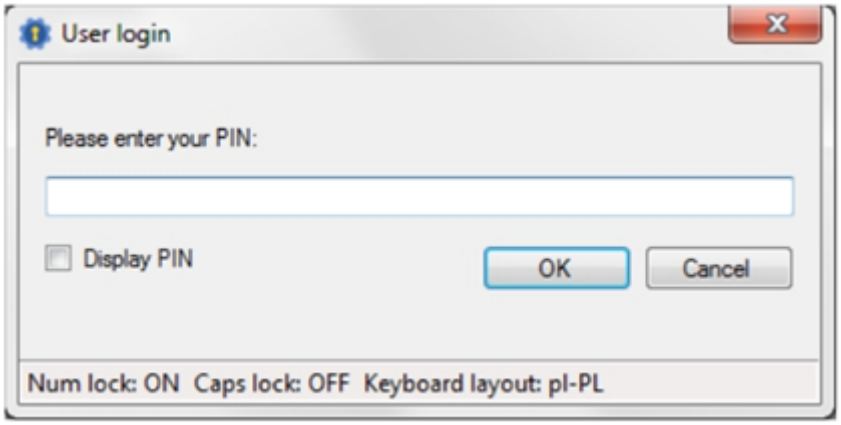
1. Należy pobrać i wypakować program PKCS11Admin do wybranego katalogu.
2. Następnie należy wejść do katalogu zawierającego wypakowane oprogramowanie PKCS11Admin i uruchomić aplikację Pkcs11Admin-x86 lub Pkcs11Admin-x64, zależnie od wersji systemu operacyjnego. Uruchomienie skutkuje otwarciem programu oraz okna wyboru biblioteki obsługującej kartę. Dla kart Certum wskazać należy bibliotekę crypto3PKCS.dll, znajdującą się w katalogu C:\Windows\System32 i zatwierdzić przyciskiem OK:



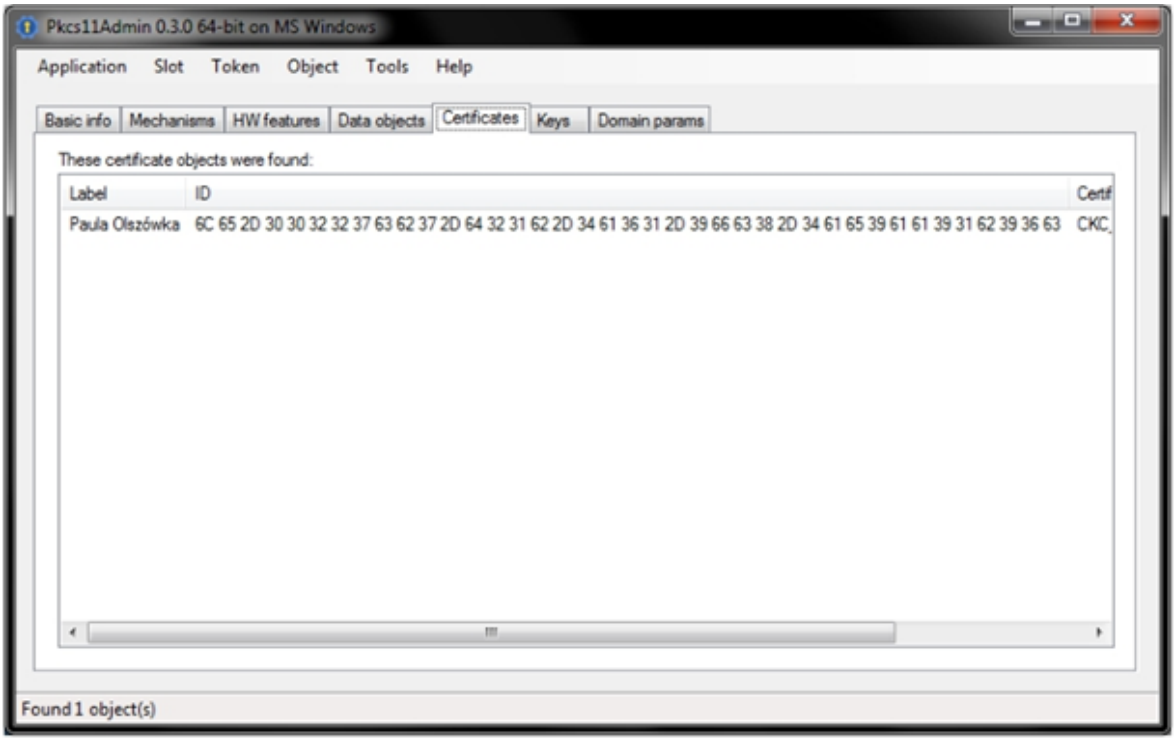
3. Gdy program wczyta zawartość karty kryptograficznej należy z paska wybrać opcję Token >Login > User login:



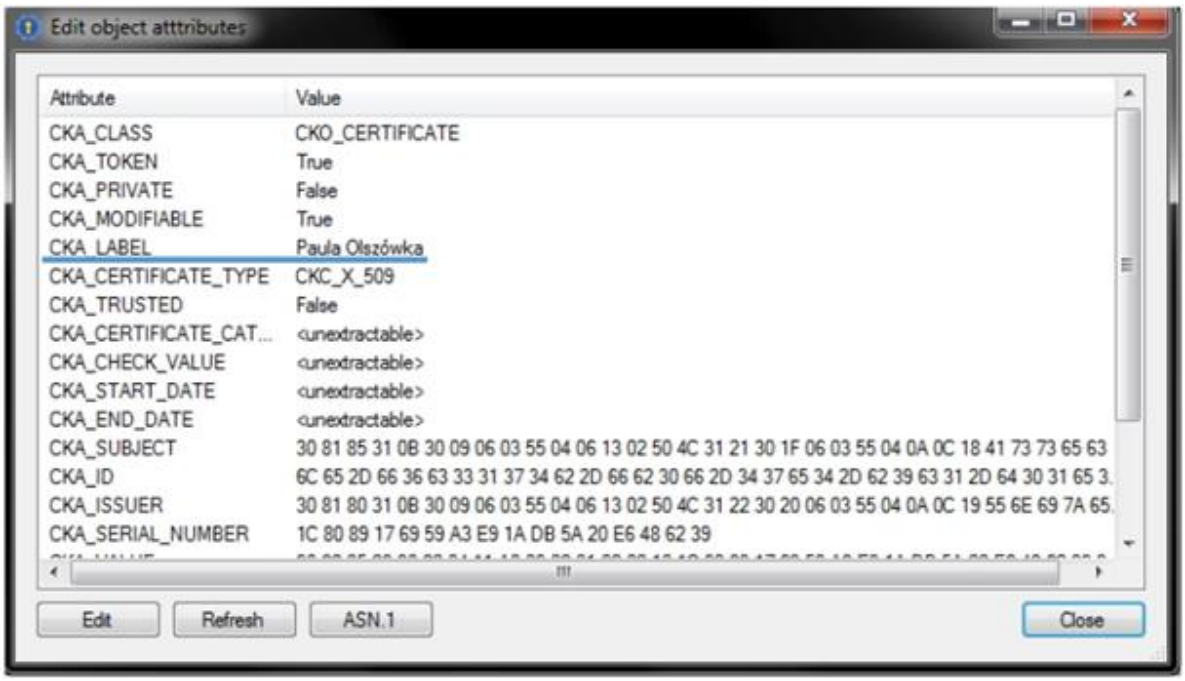
4. Program zamonituje o wpisanie **kodu PIN** do **profilu zwykłego** karty. Należy go wprowadzić i zatwierdzić przyciskiem OK:

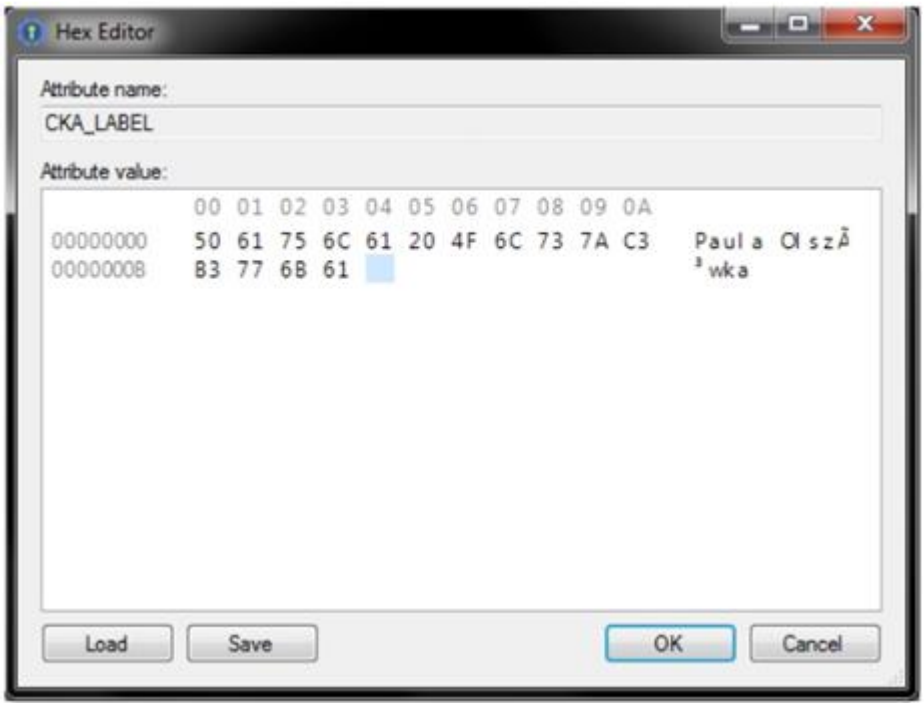


5. Następnie należy przejść do zakładki Certificates. Na liście wyświetlą się labele certyfikatów, dla których zostanie wykonana zmiana (tu dla przykładu label użytkownika zawierający znak diakrytyczny „ó”):

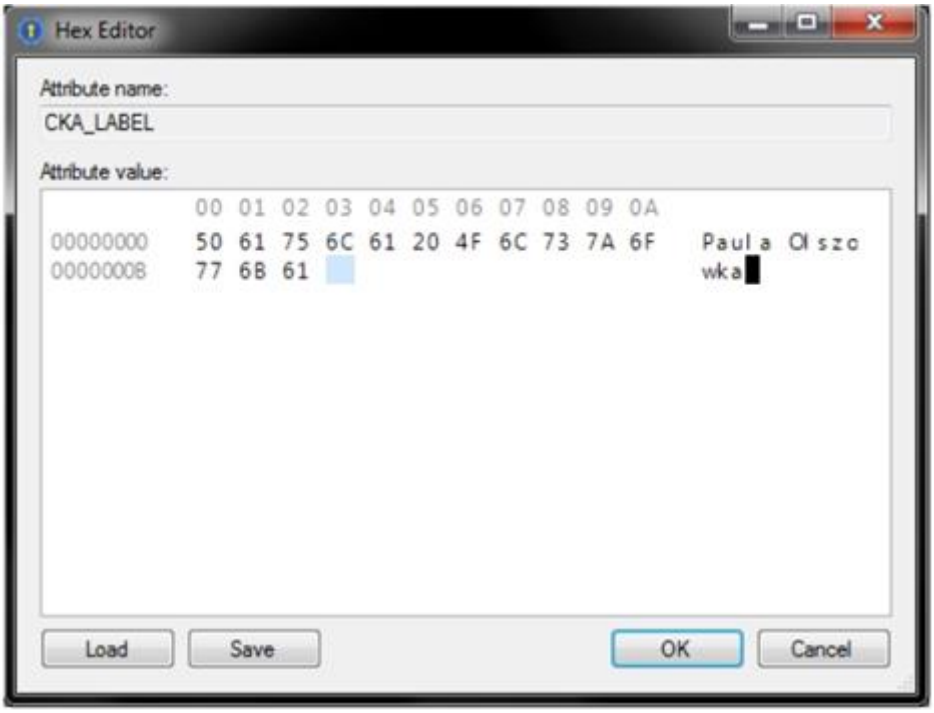


6. Kliknięcie prawym przyciskiem myszy na labelu powoduje wywołanie menu, z którego należy wybrać opcję Edit attributes… .
7. W wyświetlonym oknie, prezentującym atrybuty wpisu, należy odnaleźć wpis CKA_LABEL. Następnie należy zaznaczyć wpis CKA_LABEL i użyć przycisku Edit do zmiany błędnej zawartości wpisu:





8. Pole labela jest teraz edytowalne. Wystarczy skorygować label usuwając znaki diakrytyczne i potwierdzić przyciskiem OK:



9. Proces zmiany został ukończony. Dzięki temu przy wyborze certyfikatu do podpisu będzie możliwość podania aliasu nie zawierającego znaków diakrytycznych i co za tym idzie poprawne użycie certyfikatu.

Przed przystąpieniem do podpisywania rezultat zmiany aliasu użytkownika sprawdzić można poleceniem:

keytool -list -keystore NONE -storetype PKCS11 –providerclass sun.security.pkcs11.SunPKCS11 -providerArg provider.cfg

W rezultacie instrukcja zwraca zawartość magazynu kluczy:

```
Picked up _JAVA_OPTIONS: -Xms256m -Xmx1024m
Enter keystore password:

Keystore type: PKCS11
Keystore provider: SunPKCS11-Crypto3CSP

Your keystore contains 1 entry

Paula Olszowka, PrivateKeyEntry,
Certificate                    fingerprint                    (SHA1):
E0:82:8D:F9:D7:1C:4C:D8:7A:34:94:60:02:7F:0D:9C:B8:02:BF:31
```

3.3 Podpisywanie

Aby podpisać plik, w wierszu poleceń (cmd.exe) należy użyć następującego polecenia:

Jarsigner -keystore NONE -tsa "[1]" -certchain "[2]" -storetype PKCS11 -providerClass sun.security.pkcs11.SunPKCS11 -providerArg "[3]" -storepass "[4]" "[5]" "[6]"

[1] – Adres znacznika czasu. Dla Certum http://time.certum.pl, [2] – Ścieżka do pliku ścieżki certyfikatu (Sekcja „Konfiguracja”),
[3] – Ścieżka do pliku konfiguracyjnego providera (Sekcja „Konfiguracja”),
[4] – Hasło do profilu zwykłego karty, [5] – Ścieżka do pliku podpisywanego,
[6] – Nazwa właściciela certyfikatu którą sprawdzić można w proCertum CardManagerze lub używając narzędzia **Keytool**

Przykładowe, poprawne polecenie:

jarsigner -keystore NONE -certchain "bundle.pem" -tsa "http://time.certum.pl" - storetype PKCS11 -providerClass sun.security.pkcs11.SunPKCS11 -providerArg "provider.cfg" -storepass "123456" "aplikacja.jar" "Asseco Data Systems S.A."

Jeśli operacja podpisu przebiegła prawidłowo, konsola wyświetli następujący wynik:

```
Picked up _JAVA_OPTIONS: -Xms256m -Xmx1024m
jar signed.
```

3.4 Weryfikacja

Aby zweryfikować plik, w wierszu poleceń (cmd.exe) należy użyć następującego polecenia:

jarsigner -verify "[1]"

[1] – Ścieżka do pliku podpisywanego,

Przykładowe, poprawne polecenie:

jarsigner -verify "aplikacja.jar"

W przypadku poprawnej weryfikacji pliku konsola wyświetli:

```
Picked up _JAVA_OPTIONS: -Xms256m -Xmx1024m
jar verified.
```

W przypadku braku podpisu wynik jest następujący:

```
Picked up _JAVA_OPTIONS: -Xms256m -Xmx1024m
jar is unsigned.
```

3.5 Podpisywanie wsadowe

W celu wsadowego podpisania wielu plików podczas jednej sesji należy utworzyć plik *.bat, zawierający tyle wpisów, ile plików ma zostać podpisane podczas jednego procesu podpisu. Działanie takie eliminuje konieczność każdorazowego wywoływania komendy w konsoli oraz wpisywania kodu PIN przy podpisie kolejnych plików.

W celu utworzenia pliku, należy utworzyć nowy plik tekstowy *.txt, wkleić wpisy do podpisywania plików, zapisać plik oraz zmienić jego rozszerzenie z *.txt na *.bat.

Poniższy przykład prezentuje zawartość pliku *.bat dla podpisu trzech aplikacji jednocześnie:

jarsigner -keystore NONE -certchain "bundle.pem" -tsa "http://time.certum.pl" - storetype PKCS11 -providerClass sun.security.pkcs11.SunPKCS11 -providerArg "provider.cfg" -storepass "123456" "aplikacja1.jar" "Asseco Data Systems S.A."

jarsigner -keystore NONE -certchain "bundle.pem" -tsa "http://time.certum.pl" - storetype PKCS11 -providerClass sun.security.pkcs11.SunPKCS11 -providerArg "provider.cfg" -storepass "123456" "aplikacja2.jar" "Asseco Data Systems S.A."

jarsigner -keystore NONE -certchain "bundle.pem" -tsa "http://time.certum.pl" - storetype PKCS11 -providerClass sun.security.pkcs11.SunPKCS11 -providerArg "provider.cfg" -storepass "123456" "aplikacja3.jar" "Asseco Data Systems S.A."

Tak zapisany plik można uruchomić w konsoli cmd.exe lub dwuklikiem, a rezultatem będzie rozpoczęcie podpisywania kolejnych plików, zawartych w pliku *.bat.

Rezultatem uruchomienia pliku *.bat w konsoli będzie informacja o kolejnym wywołaniu komend i podpisie plików:

```
C:\Users\user\Desktop\jarsigner>jarsigner -keystore NONE -certchain
"bundle.pem" -tsa http://time.certum.pl -storetype PKCS11 -
providerClass sun.security.pkcs11.SunPKCS11 -providerArg
"provider.cfg" -storepass "123456" "aplikacja1.jar" "Asseco Data
Systems S.A"
Picked up _JAVA_OPTIONS: -Xms256m -Xmx1024m
jar signed.

C:\Users\user\Desktop\jarsigner>jarsigner -keystore NONE -certchain
"bundle.pem" -tsa http://time.certum.pl -storetype PKCS11 -
providerClass sun.security.pkcs11.SunPKCS11 -providerArg
"provider.cfg" -storepass "123456" "aplikacja2.jar" "Asseco Data
Systems S.A"

Picked up _JAVA_OPTIONS: -Xms256m -Xmx1024m
jar signed.

C:\Users\user\Desktop\jarsigner>jarsigner -keystore NONE -certchain
"bundle.pem" -tsa http://time.certum.pl -storetype PKCS11 -
providerClass sun.security.pkcs11.SunPKCS11 -providerArg
"provider.cfg" -storepass "123456" "aplikacja3.jar" "Asseco Data
Systems S.A"
Picked up _JAVA_OPTIONS: -Xms256m -Xmx1024m
jar signed.
```

4. Najczęstsze problemy

1. Podczas podpisu narzędziem **Signtool** przy wykorzystaniu algorytmu SHA-2 występuje problem z podpisaniem:

```
Done Adding Additional Store
SignTool Error: An unexpected internal error has occurred.
Error information: "Error: SignerSign() failed." (-
2146893784/0x80090028)
```

Rozwiązanie: W oprogramowaniu proCertum CardManager należy wybrać przycisk Opcje > zaznaczyć opcję EV Code Signing – zastąp CSP biblioteką minidriver. Następnie zrestartować system i podjąć próbę podpisu ponownie.

2. Podczas podpisu narzędziem **Jarsigner** pojawia się komunikat:

```
Picked up _JAVA_OPTIONS: -Xms256m -Xmx1024m
jar signed.

Warning:
The signer's certificate chain is not validated.
```

Rozwiązanie: Należy zweryfikować zawartość pliku bundle.pem. Plik zawiera prawdopodobnie nieprawidłowe certyfikaty bądź certyfikaty w nieprawidłowej kolejności.

Plik bundle.pem powinien zawierać certyfikaty:
1. Certyfikat Subskrybenta,
2. Odpowiedni certyfikat pośredni.

Więcej o pliku bundle.pem w **punkcie 3.2.2.**

3. Podczas weryfikacji podpisu narzędziem **Jarsigner** pojawia się komunikat:

```
Picked up _JAVA_OPTIONS: -Xms256m -Xmx1024m
jarsigner: java.lang.SecurityException: cannot verify signature
block file META-INF/PAULA_OL
```

PAULA_OL to przykładowa sygnatura, zależna od aliasu użytkownika. Więcej o aliasach **w punkcie 3.2.3.**

Rozwiązanie: Należy zweryfikować zawartość pliku bundle.pem. Plik zawiera prawdopodobnie nieprawidłowe certyfikaty bądź certyfikaty w nieprawidłowej kolejności.
Plik bundle.pem powinien zawierać certyfikaty:
1. Certyfikat Subskrybenta,
2. Odpowiedni certyfikat pośredni.

4. Podczas podpisu narzędziem **Jarsigner** pojawia się komunikat:

```
Picked up _JAVA_OPTIONS: -Xms256m -Xmx1024m
jar signed.

Warning:
The signer certificate's KeyUsage extension doesn't allow code
signing.
```

Rozwiązanie: Należy zweryfikować zawartość pliku bundle.pem. Plik zawiera prawdopodobnie nieprawidłowe certyfikaty bądź certyfikaty w nieprawidłowej kolejności.
Plik bundle.pem powinien zawierać certyfikaty:
1. Certyfikat Subskrybenta,
2. Odpowiedni certyfikat pośredni.

Więcej o pliku bundle.pem w **punkcie 3.2.2.**