



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
HAROKOPIO UNIVERSITY

Security Project

Πληροφοριακά Συστήματα και
Καινοτομία II (ΕΦΠ08)

Μαρίνος Κουβαράς, ap23011

6 Ιουλίου 2024

Πίνακας Περιεχομένων

Πίνακας Περιεχομένων	2
Report a	3
a.1.)	3
a.2.)	4
a.3.)	5
DSA	5
RSA	6
Ed25519	6
Report b	6
b.1.)	6
1.I The issuer of the certificate	7
1.II The signature algorithm and key size	8
1.III The public key	10
1.IV The key owner	12
1.V The certificate serial number	13
1.VI . The issuing and expiration date of the certificate	14
1.VII The signing date and time	16
b.2.)	16

Report a

Η εκτέλεση του σχετικού κώδικα πραγματοποιήθηκε μέσω colab και ο κώδικας επισυνάπτεται σαν αρχείο με ονομασία *ap23011_security.ipynb*. Λεπτομέρειες και σχετική τεκμηρίωση βρίσκεται στο αρχείο του κώδικα.

a.1.)

Μετά την εκτέλεση του κώδικα μπορούμε να παρατηρήσουμε τα εξής αποτελέσματα:

```
myData length: 35
myEncryptedFernet length: 140
myEncryptedChacha length: 51
myEncryptedAes length: 51
-----
Memory Size of myData: 68
Memory Size of myEncryptedFernet: 173
Memory Size of myEncryptedChacha: 84
Memory Size of myEncryptedAes: 84
-----
Fernet time: 0.0025849342346191406
ChaCha20Poly1305 time: 0.0002942085266113281
AES-GCM time: 0.0005552768707275391
```

Απο τη σύγκριση των μεγεθών των κωδικοποιημένων πληροφοριών παρατηρούμε ότι, η επιστρεφόμενη πληροφορία που κωδικοποιείται με τη χρήση Fernet είναι μεγαλύτερη σε μέγεθος λόγω της προσθήκης μεταδεδομένων (metadata) ενώ της ChaCha20Poly1305 και της AES-GCM είναι ίδια αφού και οι δύο προσθέτουν authentication tag μεγέθους 16-byte. Παράλληλα απο τη σύγκριση των χρόνων εκτέλεσης παρατηρούμε πως το μεγαλύτερο χρόνο απαιτεί η Fernet καθώς πέρα από το χρόνο που απαιτεί για τη σχετική κρυπτογράφηση απαιτείται και η προσθήκη των μεταδεδομένων.

Σε γενικά πλαίσια η χρήση της Fernet προτιμάται κυρίως για την ευκολία χρήσης της, ενώ για λειτουργίες που απαιτούν υψηλότερες επιδόσεις και μικρότερη μνήμη επιλέγονται οι ChaCha20Poly1305 και AES-GCM

Στιγμιότυπο εκτέλεσης του κώδικα φαίνεται παρακάτω:

```
✓ [165] print("myData length: ",len(myData))
print("myEncryptedFernet length: ",len(myEncryptedFernet))
print("myEncryptedChacha length: ",len(myEncryptedChacha))
print("myEncryptedAes length: ",len(myEncryptedAes))
print("-----")
print("Memory Size of myData: ", sys.getsizeof(myData))
print("Memory Size of myEncryptedFernet: ",sys.getsizeof(myEncryptedFernet))
print("Memory Size of myEncryptedChacha: ",sys.getsizeof(myEncryptedChacha))
print("Memory Size of myEncryptedAes: ",sys.getsizeof(myEncryptedAes))
print("-----")
print("Fernet time: ",fernet_time)
print("ChaCha20Poly1305 time: ",chacha_time)
print("AES-GCM time: ",aesgcm_time)
```

```
➞ myData length: 35
myEncryptedFernet length: 140
myEncryptedChacha length: 51
myEncryptedAes length: 51
-----
Memory Size of myData: 68
Memory Size of myEncryptedFernet: 173
Memory Size of myEncryptedChacha: 84
Memory Size of myEncryptedAes: 84
-----
Fernet time: 0.0024933815002441406
ChaCha20Poly1305 time: 0.0003407001495361328
AES-GCM time: 0.001474618911743164
```

a.2.)

Μετά την εκτέλεση του κώδικα μπορούμε να παρατηρήσουμε τα εξής αποτελέσματα:

CMAC hex: 44d0df3793c5462430e3d72926a10de8

HMAC hex: d02427a8c73d0fc8ca93f44a70077c10c060f6982fd67e3f3099dc242607b94e

Poly1305 hex: 796e9a70be5d460968cf3634cabbd82c

CMAC length: 16

HMAC length: 32

Poly1305 length: 16

CMAC memory size: 49

HMAC memory size: 65

Poly1305 memory size: 49

Σε αυτό το σημείο μας ενδιαφέρει το tag που δημιουργείται και με το οποίο ελέγχουμε το integrity των δεδομένων μας. Η χρήση του αλγορίθμου AES στο CMAC δημιουργεί ένα tag μήκους 16 byte, το ίδιο και ο Poly1305. Αντίθετα ο αλγόριθμος HMAC χρησιμοποιεί το hash function SHA-256 επομένως δημιουργεί 32 bytes tag.

Στιγμιότυπο εκτέλεσης του κώδικα φαίνεται παρακάτω:

```
▶ print("CMAC hex: ", myCmac_tag1.hex())
  print("HMAC hex: ", signature1.hex())
  print("Poly1305 hex: ", p1_tag.hex())
  print("-----")
  print("CMAC length: ", len(myCmac_tag1))
  print("HMAC length: ", len(signature1))
  print("Poly1305 length: ", len(p1_tag))
  print("-----")
  print("CMAC memory size: ", sys.getsizeof(myCmac_tag1))
  print("HMAC memory size: ", sys.getsizeof(signature1))
  print("Poly1305 memory size: ", sys.getsizeof(p1_tag))

↵ CMAC hex: c0e9a1aae4be91b34387884ecfa0a387
   HMAC hex: 6732bbc84e3ff5045de84e83e51738647fb971ba26455502feafa05ffa2c0b9c
   Poly1305 hex: df63b7520622d234679c978144383dea
   -----
   CMAC length: 16
   HMAC length: 32
   Poly1305 length: 16
   -----
   CMAC memory size: 49
   HMAC memory size: 65
   Poly1305 memory size: 49
```

a.3.)

Στο κομμάτι αυτό της εργασίας ελέγχουμε τη σωστή εκτέλεση των διαδικασιών υπογραφής. Στιγμιότυπα εκτέλεσης του κώδικα φαίνονται παρακάτω:

DSA

```
✓ [188] # Verify sign with the public key
0s public_key_dsa.verify(
    signature_dsa,
    myData,
    hashes.SHA256()
)
```

RSA

```
✓ [193] # Verify sign with the public key
0s public_key_rsa.verify(
    signature_rsa,
    myData,
    padding.PSS(
        mgf=padding.MGF1(hashes.SHA256()),
        salt_length=padding.PSS.MAX_LENGTH
    ),
    hashes.SHA256()
)
```

Ed25519

```
✓ [198] # Verify sign with the public key
0s public_key_ed.verify(signature_ed, myData)
```

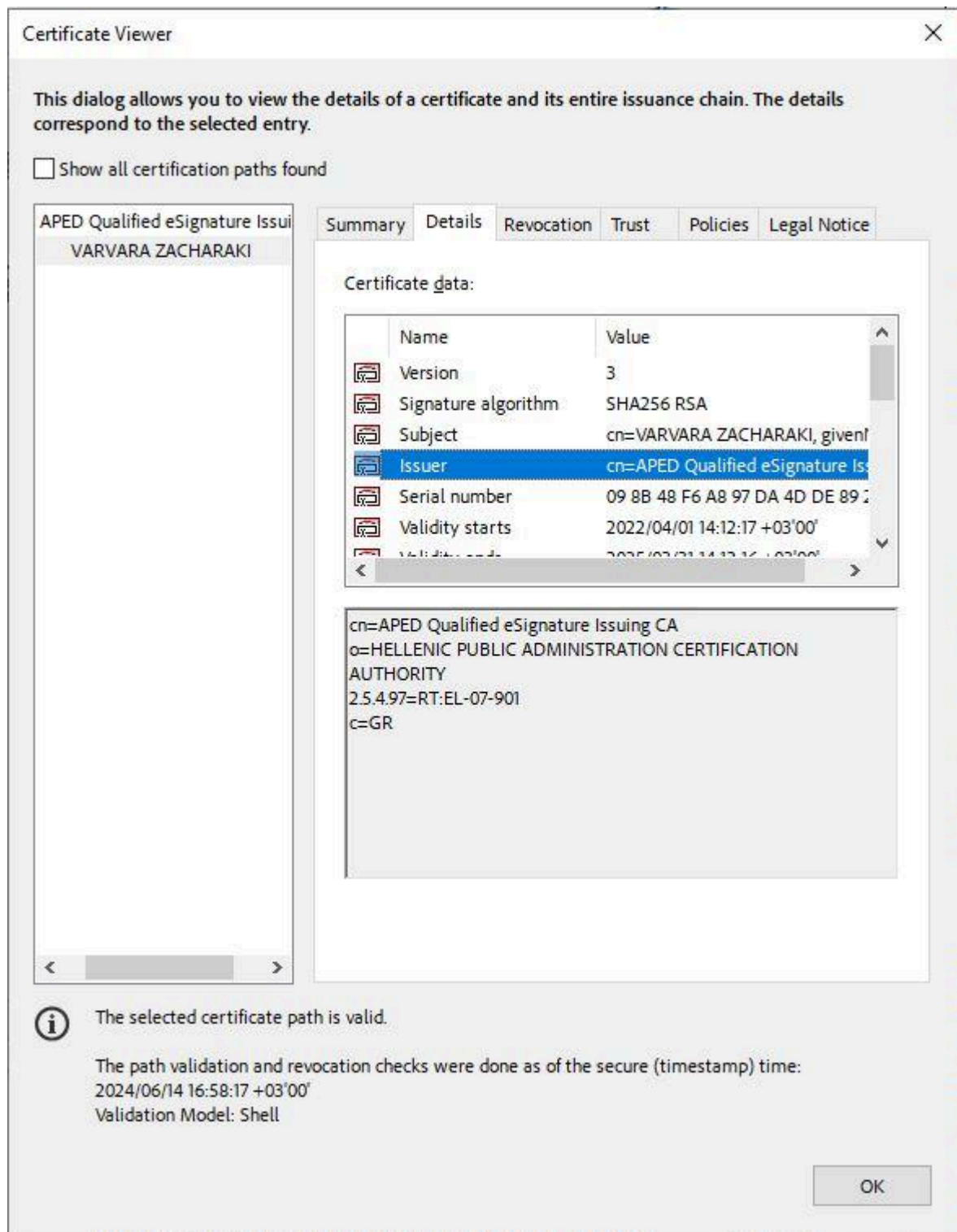
Report b

b.1.)

Το αρχείο που χρησιμοποιήθηκε είναι ΦΕΚ και μπορεί να ανακτηθεί από τον [σύνδεσμο](#).

Όσον αφορά τα στοιχεία του πιστοποιητικού που ζητούνται.

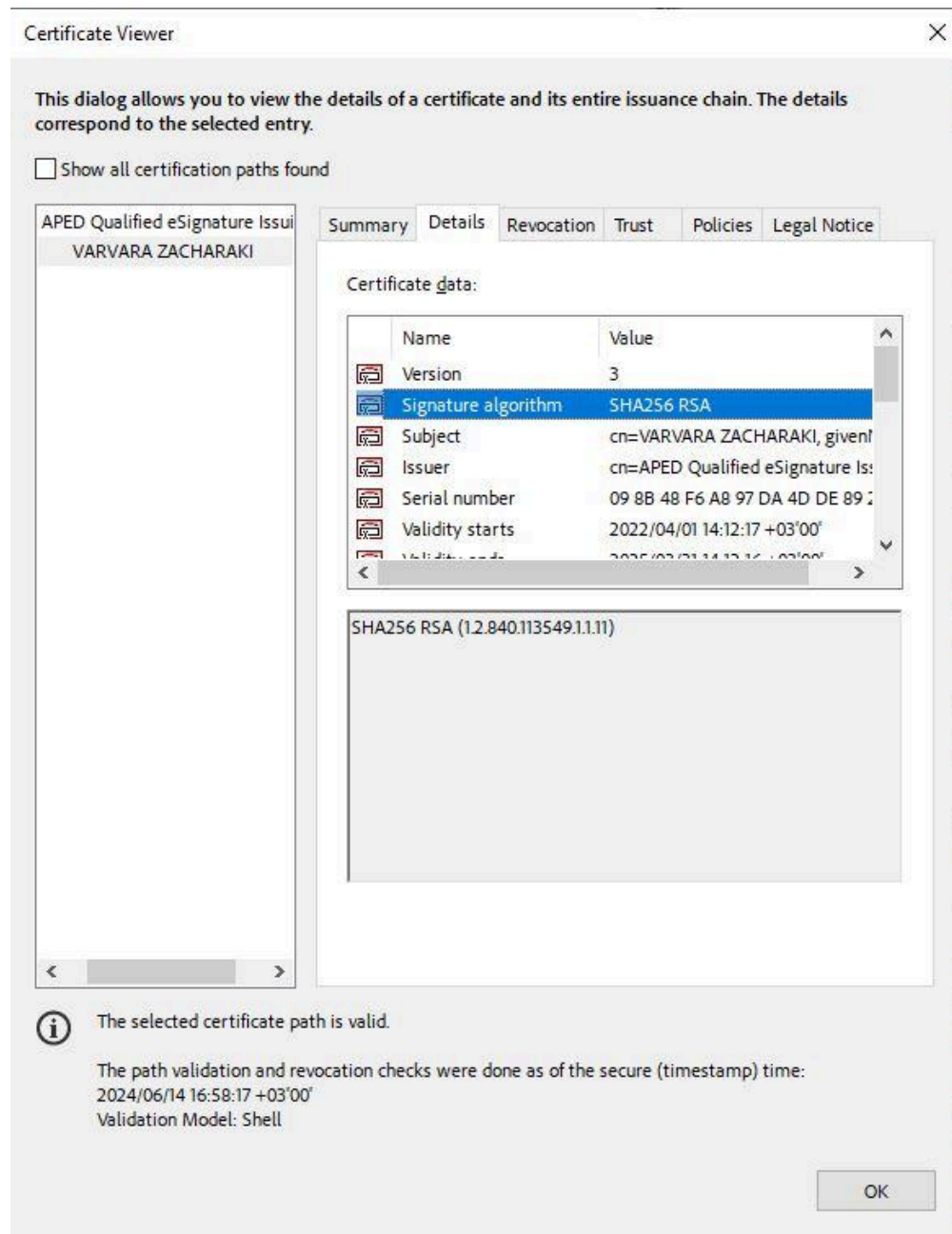
1.1 The issuer of the certificate



ISSUER	
Common Name	cn=APED Qualified eSignature Issuing CA
Organization	o=HELLENIC PUBLIC ADMINISTRATION

	CERTIFICATION AUTHORITY
Extension value	2.5.4.97=RT:EL-07-901
Country	c=GR

1.II The signature algorithm and key size

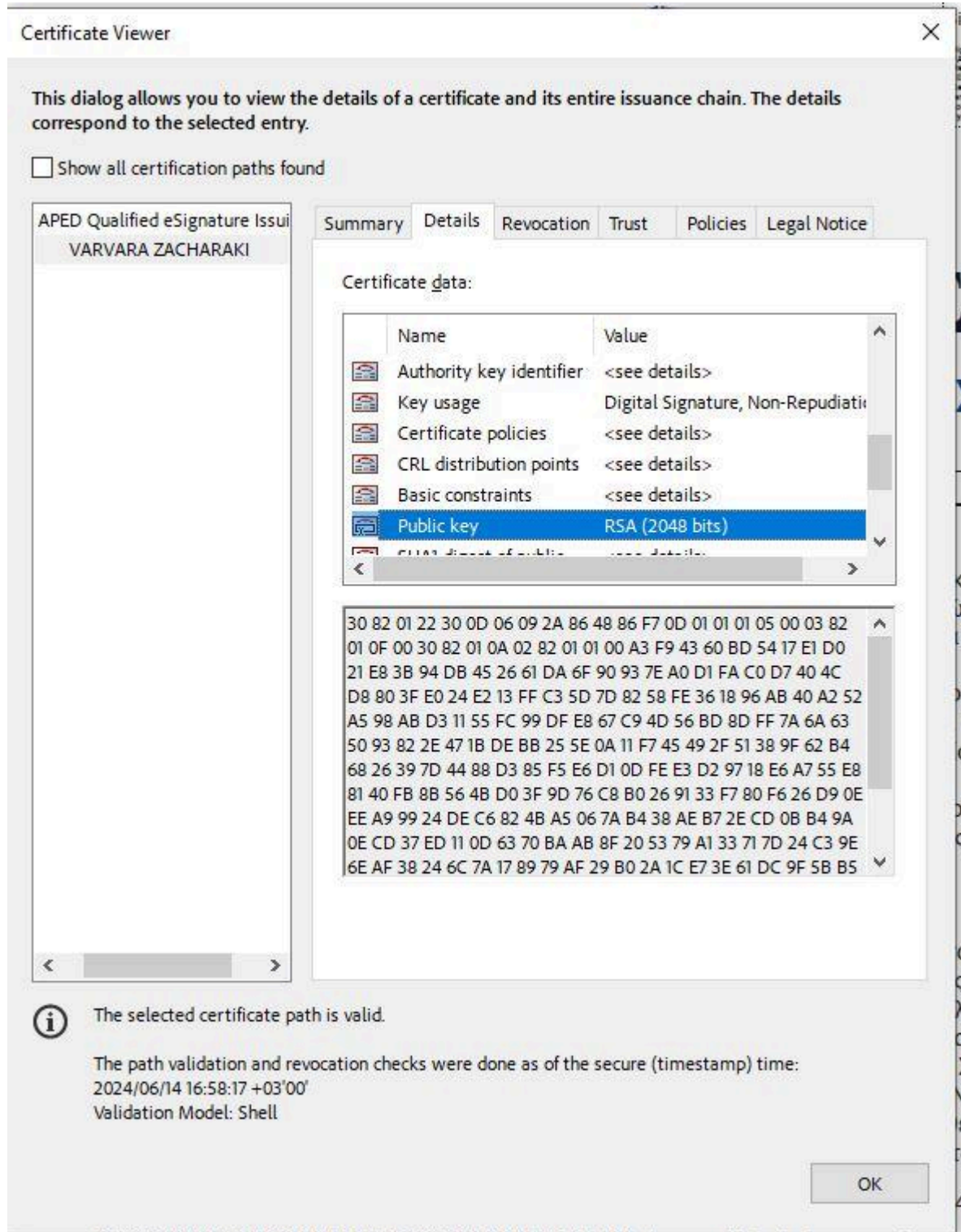


Signature Algorithm	SHA256 RSA (1.2.840.113549.1.1.11)
----------------------------	------------------------------------

Key Size

RSA (2048 bits) (256 bytes)

1.III The public key

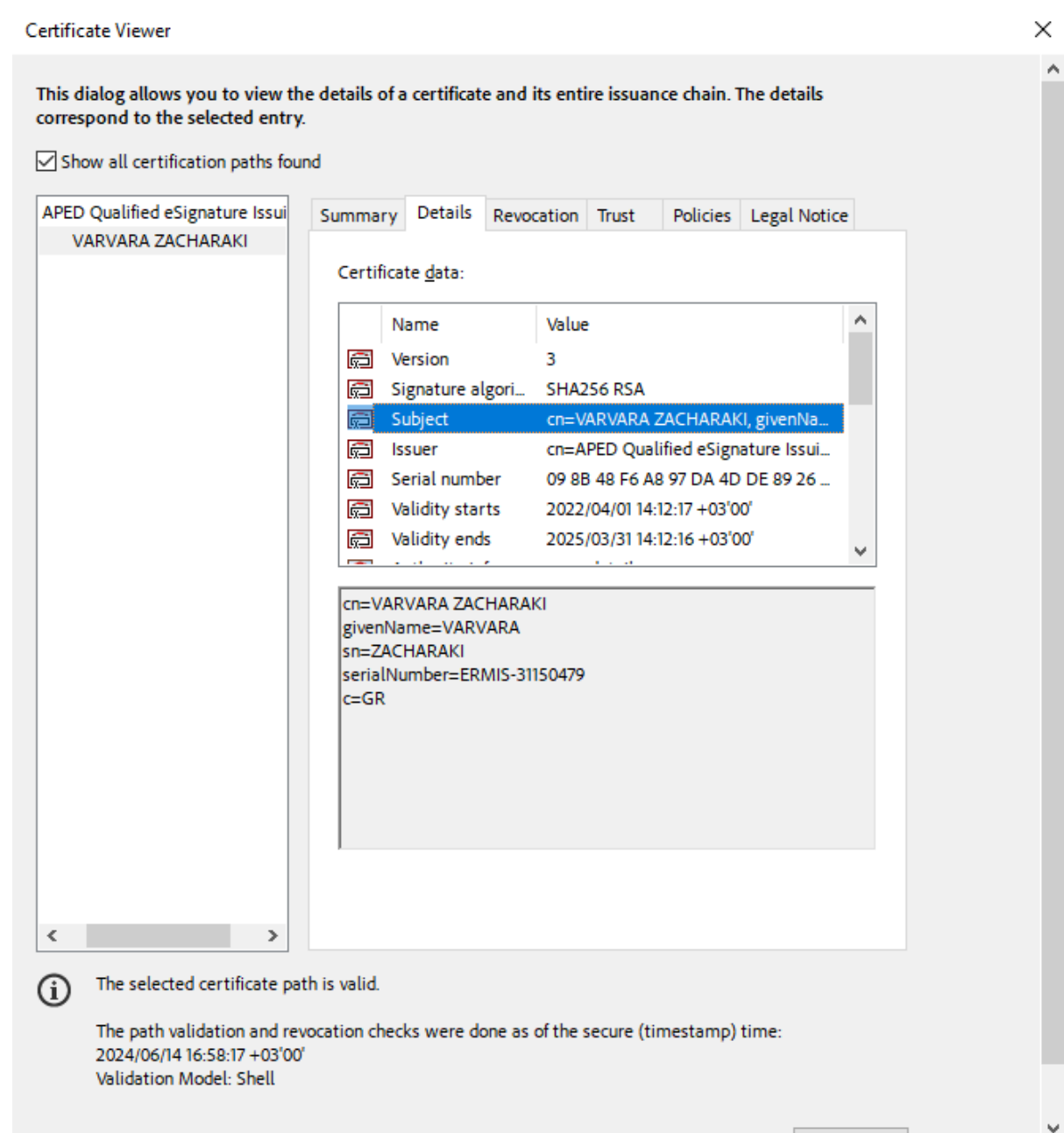


Public Key

30 82 01 22 30 0D 06 09 2A 86 48 86 F7

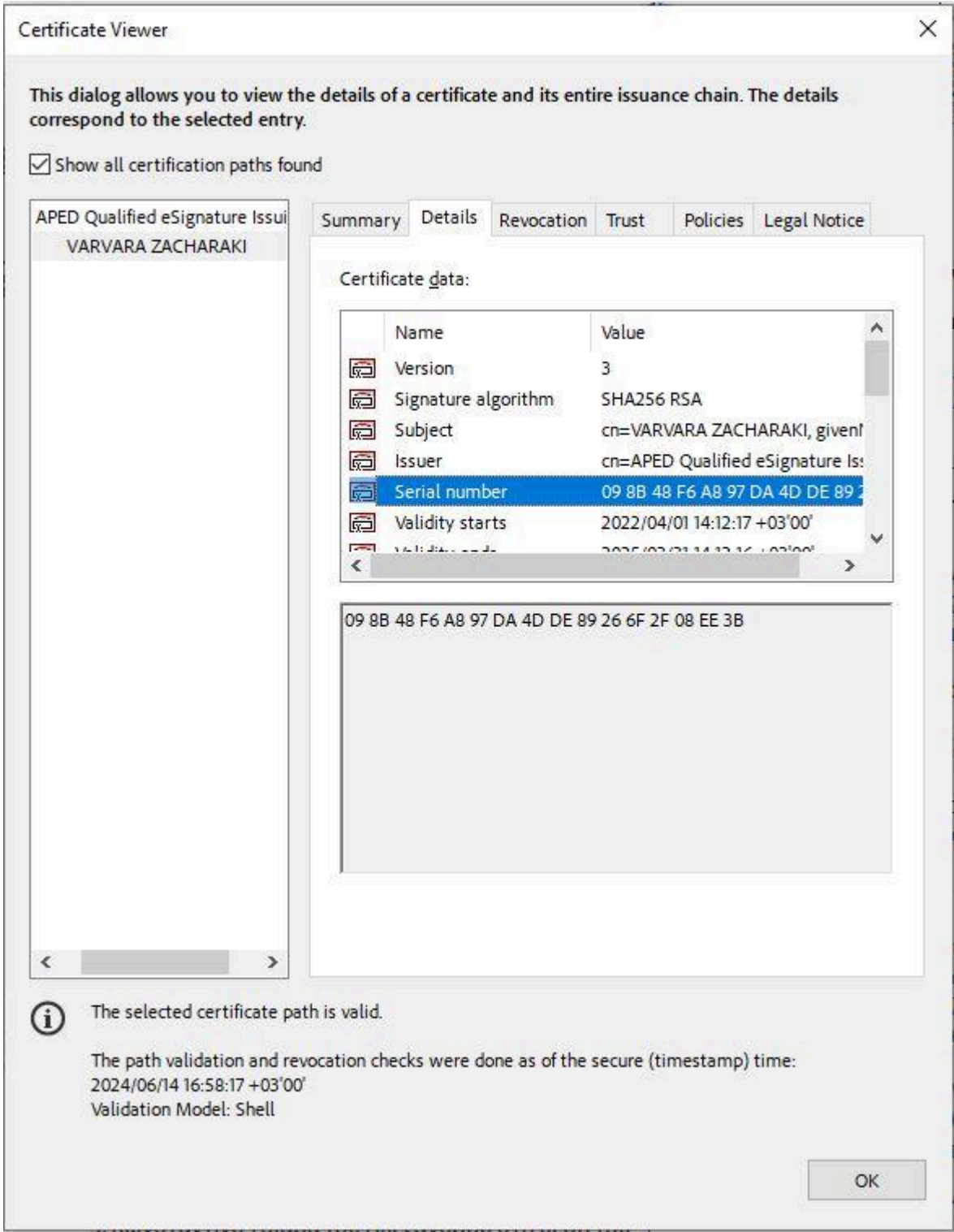
	0D 01 01 01 05 00 03 82 01 0F 00 30 82 01 0A 02 82 01 01 00 A3 F9 43 60 BD 54 17 E1 D0 21 E8 3B 94 DB 45 26 61 DA 6F 90 93 7E A0 D1 FA C0 D7 40 4C D8 80 3F E0 24 E2 13 FF C3 5D 7D 82 58 FE 36 18 96 AB 40 A2 52 A5 98 AB D3 11 55 FC 99 DF E8 67 C9 4D 56 BD 8D FF 7A 6A 63 50 93 82 2E 47 1B DE BB 25 5E 0A 11 F7 45 49 2F 51 38 9F 62 B4 68 26 39 7D 44 88 D3 85 F5 E6 D1 0D FE E3 D2 97 18 E6 A7 55 E8 81 40 FB 8B 56 4B D0 3F 9D 76 C8 B0 26 91 33 F7 80 F6 26 D9 0E EE A9 99 24 DE C6 82 4B A5 06 7A B4 38 AE B7 2E CD 0B B4 9A 0E CD 37 ED 11 0D 63 70 BA AB 8F 20 53 79 A1 33 71 7D 24 C3 9E 6E AF 38 24 6C 7A 17 89 79 AF 29 B0 2A 1C E7 3E 61 DC 9F 5B B5 C0 80 E7 E0 BA CE F7 92 69 66 E4 5B CF 18 08 60 6C 49 7D E7 8B 97 61 46 C8 C2 AF 86 47 40 A3 86 7B 62 40 7E 82 98 8F B0 74 AA 2A 03 59 AC 6A 6F 4E 31 27 76 95 F9 2F 5D 76 49 DD 90 DB 02 03 01 00 01
--	---

1.IV The key owner



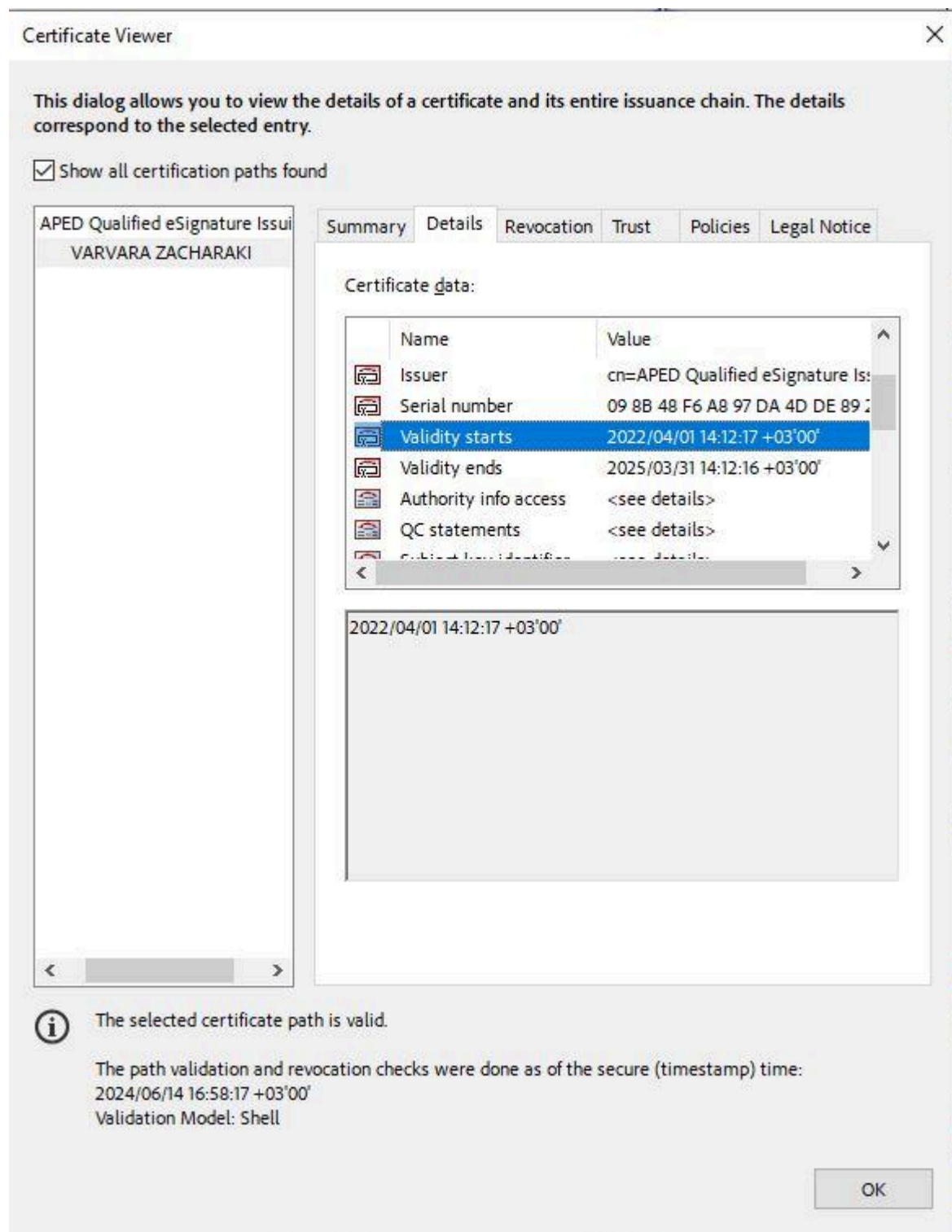
KEY OWNER	
Common Name	cn=VARVARA ZACHARAKI givenName=VARVARA
Surname	sn=ZACHARAKI
Serial Number	serialNumber=ERMIS-31150479
Country	c=GR

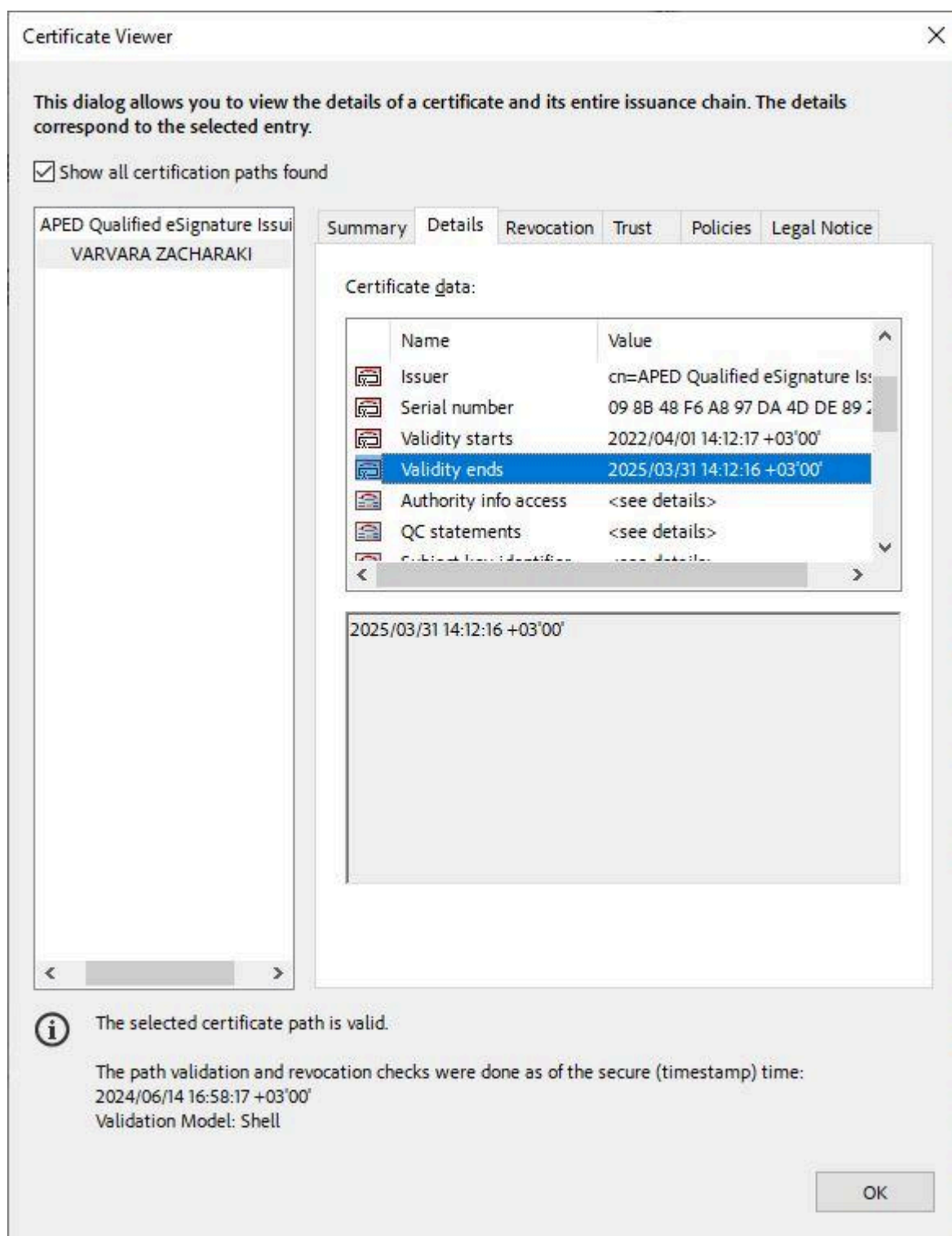
1.V The certificate serial number



Serial Number	09 8B 48 F6 A8 97 DA 4D DE 89 26 6F 2F 08 EE 3B
---------------	---

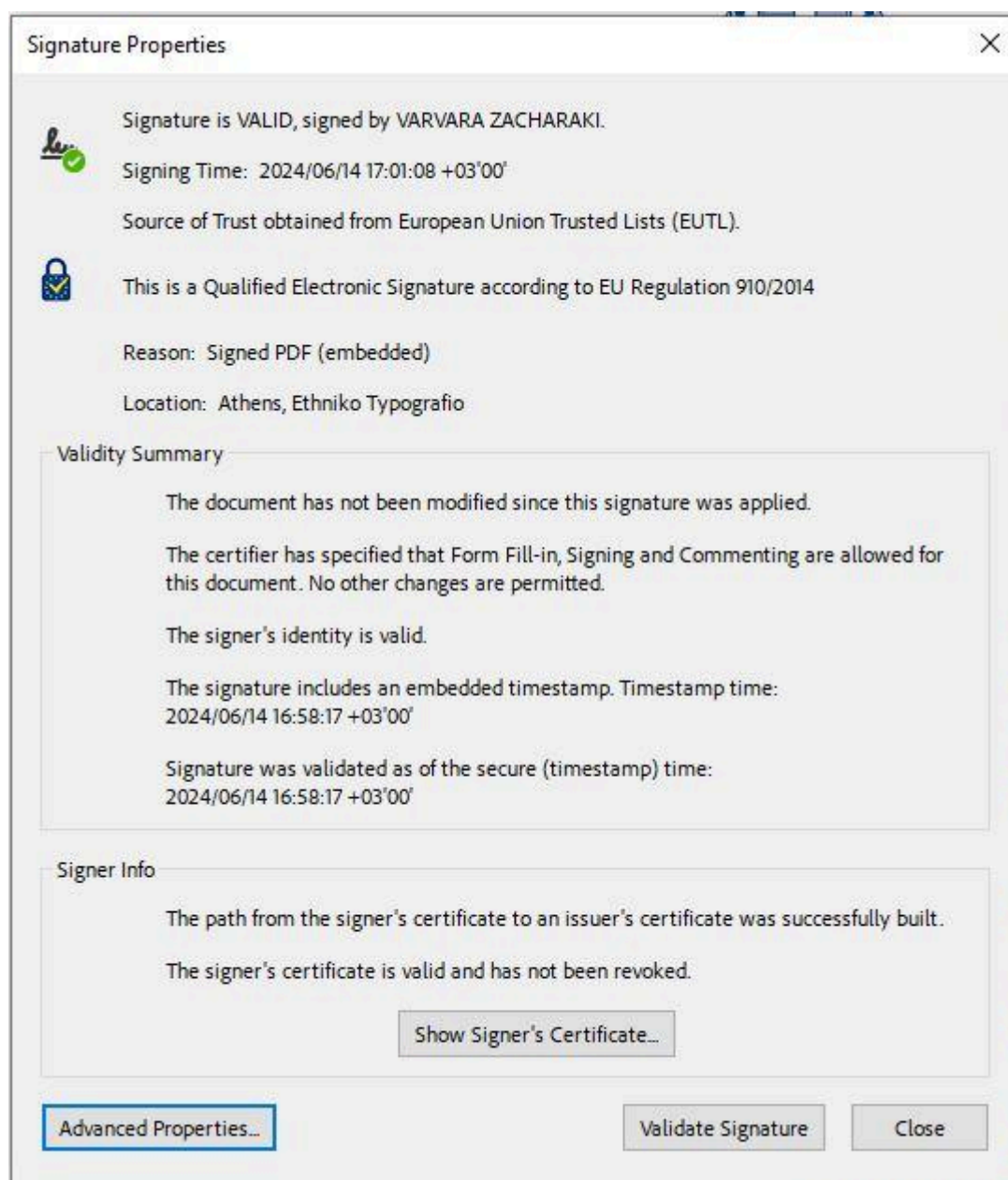
1.VI . The issuing and expiration date of the certificate





Validity Starts	2022/04/01 14:12:17 +03'00'
Validity Ends	2025/03/31 14:12:16 +03'00'

1.VII The signing date and time



Signing date	2024/06/14
Signing time	17:01:08 +03'00'

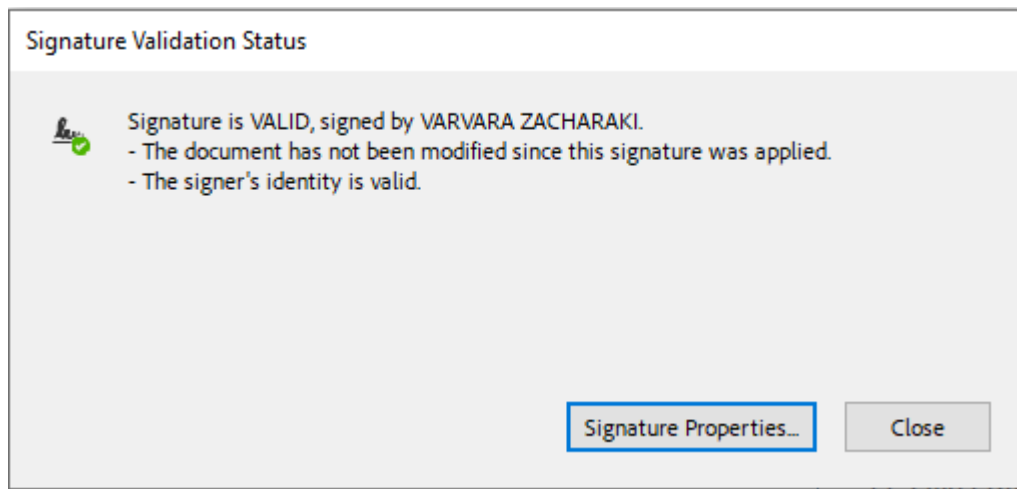
b.2.)

Για να επιβεβαιώσουμε πως η υπογραφή είναι έγκυρη ακολουθούμε μια σειρά από ενέργειες που θα μας βοηθήσουν σε αυτό.

Αρχικά ανοίγουμε το έγγραφό μας με τον Acrobat Reader και ελέγχουμε για την εγκυρότητα.

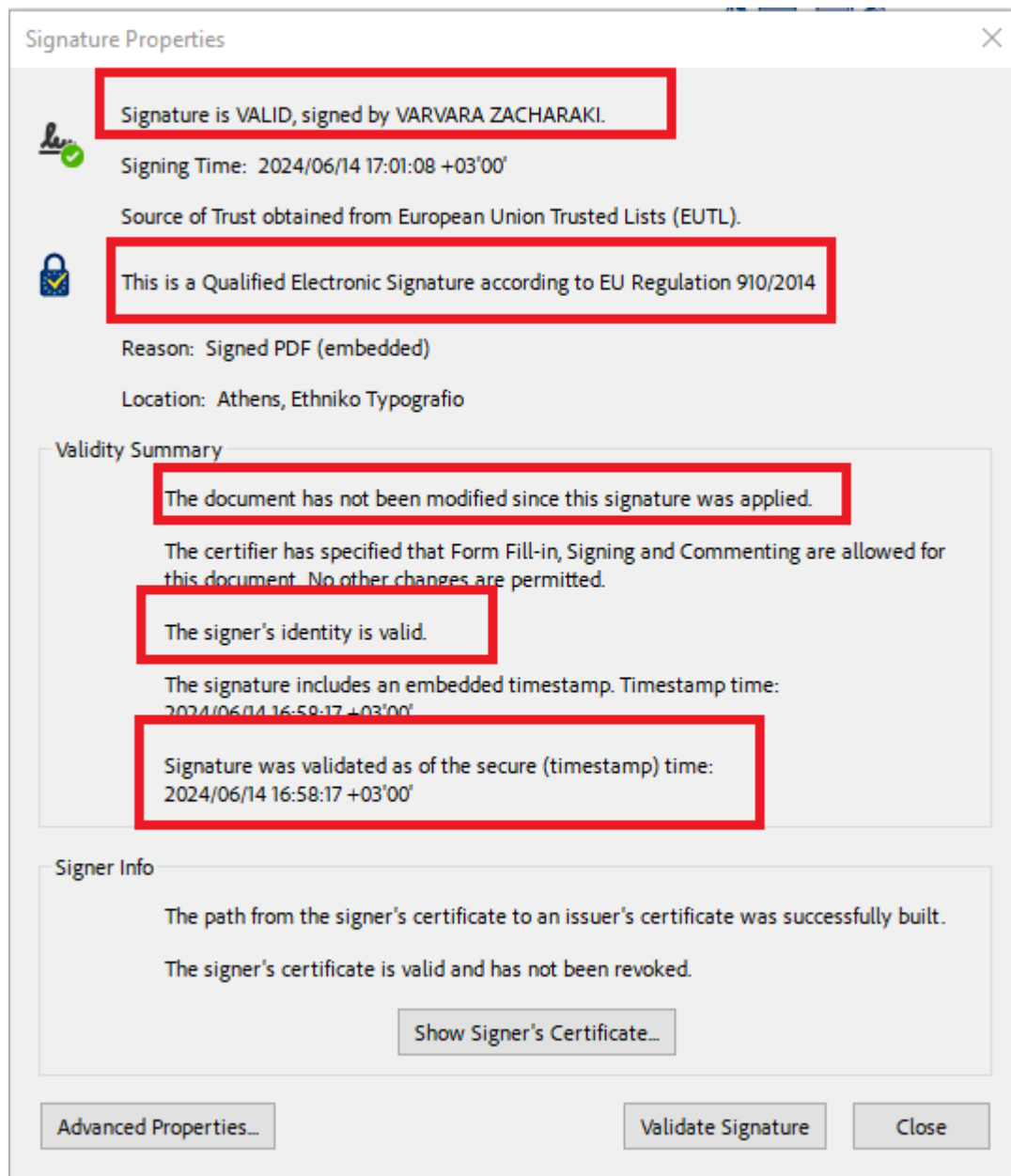


Η ένδειξη αυτή είναι ένα στοιχείο πως η υπογραφή είναι έγκυρη. Πατώντας “click” πάνω στην ένδειξη μας επιστρέφεται η εξής ένδειξη.

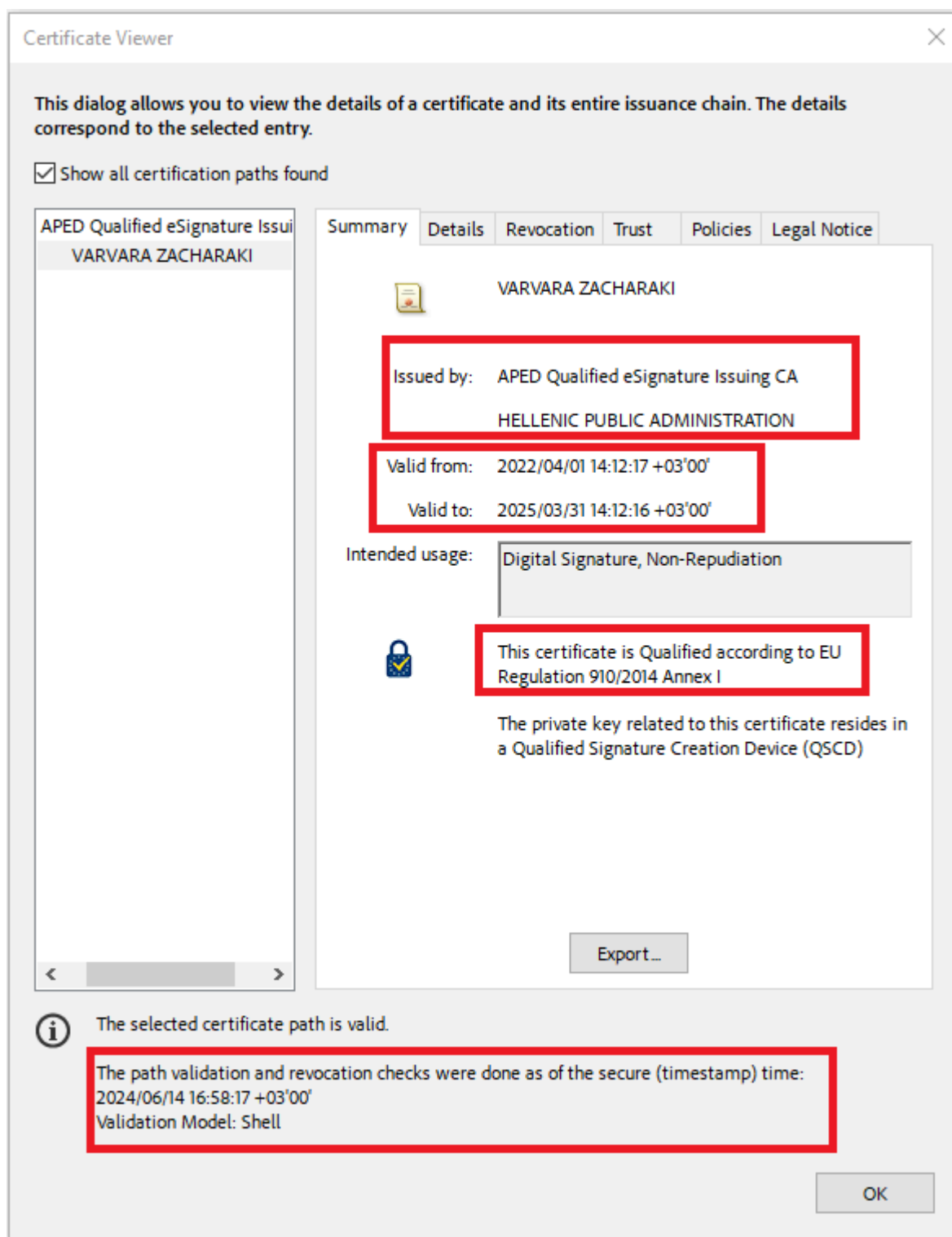


Που είναι θετική ένδειξη για την εγκυρότητα.

Η επιλογή Signature Properties μας ενημερώνει γενικά για τις ιδιότητες της υπογραφής.



Στη συνέχεια επιλέγοντας “Show Signer’s Certificate”





Μέσω [διασύνδεσης](#) του εγγράφου (καρτέλα “Legal Notice”) ελέγχουμε και τα στοιχεία της ΑΠΕΔ

ΑΠΕΑ

ΑΡΧΗ ΠΙΣΤΟΠΟΙΗΣΗΣ
ΕΛΛΗΝΙΚΟΥ
ΔΗΜΟΣΙΟΥ

ΑΠΟΘΕΤΗΡΙΟ





ΑΡΧΙΚΗ ΣΕΛΙΔΑΠΟΛΙΤΙΚΕΣΠΙΣΤΟΠΟΙΗΤΙΚΑΚΑΤΑΣΤΑΣΗ ΠΙΣΤΟΠΟΙΗΤΙΚΟΥΣΥΜΜΟΡΦΩΣΗΛΗΨΕΙΣ

Δήλωση Πρακτική Πιστοποίησης


Δήλωση Πρακτική Πιστοποίησης για ΕΥ Εγκεκριμένα Πιστοποιητικά σύμφωνα με το κανονισμό λειτουργίας (ΕΥ) Νο 910/2014

Τελευταία Έκδοση:
Version 1.1 (GR), PDF

Προηγούμενες Εκδόσεις:
Version 1.0 (GR), PDF
Version 1.0 (EN), PDF

Η καρτέλα details περιέχει πληροφορίες που εξετάσαμε στο προηγούμενο ερώτημα, επομένως μεταβαίνουμε στην καρτέλα Revocation ελέγχοντας πως δεν έχει υπάρξει κάποια ανάκληση.

SummaryDetailsRevocationTrustPoliciesLegal Notice

 The selected certificate is valid

Details

The selected certificate is considered valid because it has not been revoked as verified using the Online Certificate Status Protocol (OCSP) response that is contained in the local cache.

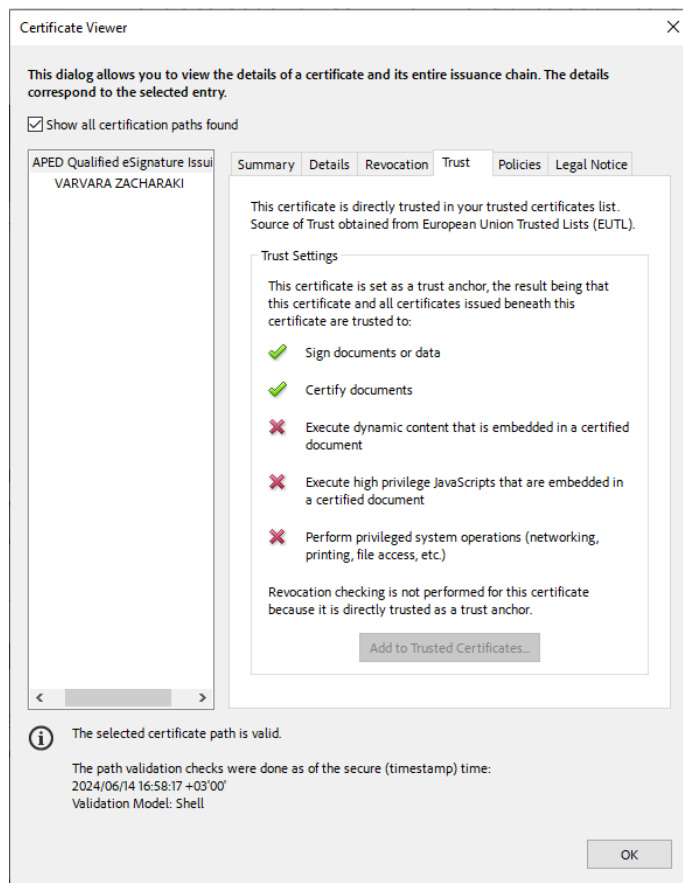
The OCSP Response was signed by "APED Qualified eSignature OCSP Responder" on 2024/06/24 20:54:52 +03'00'

Signer Details...Problems encountered...
Check revocation

Η καρτέλα "Trust" μας δείχνει τι δυνατότητες έχει το certification.

Security Project - Πληροφοριακά Συστήματα και Καινοτομία II (ΕΦΠ08)

19



Επομένως και σύμφωνα με τα βήματα που παρουσιάστηκαν στην παρουσίαση “Digital Signature, PKI, TLS, eIDAS” με αριθμό 22 επιβεβαιώνουμε τα κάτωθι:

CERTIFICATE VALIDATION

- **Certificate Integrity:** Signature is VALID, signed by VARVARA ZACHARAKI
- **Signed by a trusted CA:** YES
- **The certificate has not expired:** CORRECT
- **The certificate has not been Revoked:** CORRECT
- **Check the other fields:** OK