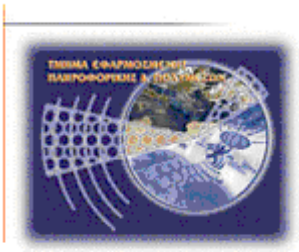
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Πτυχιακή εργασία

Χρήση ασφαλούς hardware (έξυπνων καρτών - Smartcards, eTokens) σε μία Υποδομή Δημόσιου Κλειδιού

Ρήγας Κώνσταντίνος (798)

Ηράκλειο - Ημερομηνία

17 / 10/ 2007

Επόπτης Καθηγητής: Δρ. Μανιφάβας Χαράλαμπος

Περιεχόμενα

1. Εισαγωγή

Η παρακάτω πτυχιακή εργασία έχει σαν σκοπό να μελετήσει την χρήση του ασφαλούς hardware σε μια υποδομή δημοσίου κλειδιού. Επίσης θα ασχοληθούμε και με κάποια παραδείγματα χρήσης των ασφαλών συσκευών, όπως η πρόσβαση σε Web Server μέσω πιστοποίησης SSL, η αποστολή email με ψηφιακή υπογραφή και η διασφάλιση αρχείων pdf μέσω ψηφιακή υπογραφή.

Για να γίνει αυτή η μελέτη θα χρειαστούμε ασφαλές hardware (e-Tokens, smartcards) και κάποια ψηφιακά πιστοποιητικά. Τα ψηφιακά πιστοποιητικά θα τα εξάγουμε από μια δική μας αρχή πιστοποίησης. Με αυτό τον τρόπο θα μπορέσουμε να δούμε όλη την διαδικασία δημιουργίας ενός ψηφιακού πιστοποιητικού από την αρχική αίτηση του χρήστη μέχρι την δημιουργία του και την αποθήκευση του σε μία ασφαλής συσκευή.

1.1 Περίληψη

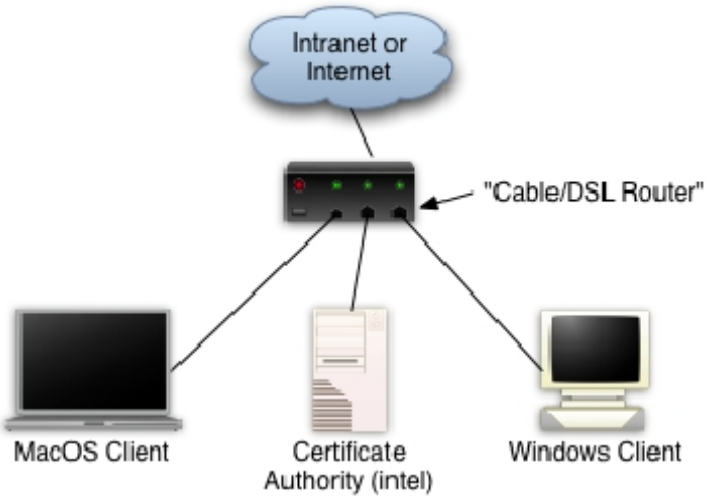
Τον Ιούνιο του 2004 το εργαστήριο PKI του πανεπιστήμιο του Dartmouth δημιούργησε ένα Live cd το οποίο περιλαμβάνει μια διανομή του OpenCA, επεξεργασμένη έτσι ώστε να είναι εύκολη στην εγκατάσταση και στην χρήση. Χρησιμοποιήθηκε η τεχνολογία από το Knoppix version 3.4 (Live cd) το οποίο επιτρέπει να εγκατασταθεί μια αρχή πιστοποίησης (CA) εύκολα και γρήγορα με το να γίνει boot του Livecd σε έναν υπολογιστή intel-based. Δεν χρειάζεται να έχουμε κάποιο προεγκατεστημένο λειτουργικό σύστημα, επίσης δεν χρειάζεται να έχουμε κάν σκλήρο δίσκο. Το live cd μπορεί να χρησιμοποιηθεί σε συνεργασία με ένα usb flash disk έτσι ώστε το flash disk να παρέχει αρκετό μόνιμο χώρο στο OpenCa για να αποθηκεύει την κατάσταση του κάθε φορά (πιστοποιητικά, αιτήσεις πιστοποιητικών). Με αυτό τον τρόπο το Openca Live Cd μπορεί να χρησιμοποιηθεί επαρκώς σαν πλήρης και λειτουργικός πάροχος πιστοποίησης.

Ενώ η διανόμη του Openca Live cd μέσω Knoppix χρησιμεύει πάρα πολύ σε όσους θέλουν να τεστάρουν ή να εγκαταστήσουν ένα πάροχο πιστοποίησης για ευρενητικούς σκοπούς, μπορούμε να το χρησιμοποιήσουμε επαρκώς για να δημιουργήσουμε SSL πιστοποιητικά για χρήστες και για Servers καθώς και να χρησιμοποιήσουμε Hardware Tokens μέσα στα οποία θα εγκατασταθεί το πιστοποιητικό του χρήστη.

Πρέπει να επισυμάνουμε ότι ο πάροχος πιστοποίησης ο οποίος θα εγκατασταθεί παρακάτω είναι μονάχα για σκοπούς τεσταρίσματος και αξιολόγησης. Για να δημιουργήσουμε ένα πραγματικό πάροχο πιστοποίησης χρειάζονται νομικά δικαιώματα, παρόλα αυτά ο πάροχος πιστοποίησης δεν υστερεί καθόλου σε σχέση με ένα ενεργό και επίσημο πάροχο πιστοποίησης.

Στην πραγματικότητα ένα πάροχος πιστοποίησης (CA) συνήθως είναι εκτός δικτύου (offline) διότι υπάρχει ο φόβος της επιθέσης και της παραβίασης του συστήματος μέσω απομακρυσμένων χρηστών. Σε αυτή την περίπτωση ένας πάροχος πιστοποίησης μένει στο δίκτυο μονάχα όση ώρα χρειάζεται έτσι ώστε να διαχειριστεί τις αιτήσεις που του έχουν γίνει και να δημιουργήσει τα πιστοποιητικά. Επίσης υπάρχει και μια άλλη προσέγγιση, τα πιστοποιητικά να δημιουργούνται μονάχα από ένα πάροχο ο οποίος θα είναι συνεχώς εκτός δικτύου και να μεταφέρονται μέσω αποθηκευτικών μέσων σε ένα εναλλακτικο πάροχο ο οποίος θα είναι συνεχώς ενεργός στο δίκτυο αλλά θα έχει περιορισμένα δικαιώματα.

Στην περίπτωση που ο πάροχος πιστοποίησης θα χρησιμοποιηθεί για σκοπούς τεσταρίσματος και αξιολόγησης δεν χρειάζεται να είναι έκτος δικτύου συνεχώς. Μια εύκολη μέθοδος η οποία επιτρέπει να δημιουργηθεί ένας πάροχος πιστοποίησης με ένα ή δύο "πελάτες" (clients) οι οποίοι θα χρησιμοποιούν συσκευές Token για την αποθήκευση και χρησιμοποίηση πιστοποιητικών, μπορεί να δημιουργηθεί μέσω ενός dsl/router στον οποίο θα συνδεθούν ο πάροχος καθώς και όλοι οι πελάτες (ανεξαρτήτος λειτουργικού συστήματος), μέσα σε ένα απομονωμένο δίκτυο.



Χρησιμοποιόντας τον dsl/router μπορούμε να έχουμε ένα DHCP Server στο δίκτυο μας ο οποίος θα μπορεί να δώσει ip διευθύνσεις στους "πελάτες" (clients). Το Knoppix/LiveCd όταν κάνει boot χρησιμοποιεί το DHCP για να συνδεθεί στο δίκτυο, αν χρειάζεται ο router να "βλεπει" internet δεν έχει σχέση με την άρχη πιστοποίησης αλλά με τις ανάγκες των χρηστών, πάντως η αρχή πιστοποίησης δεν χρειάζεται την πρόσβαση στο Internet για να λειτουργήσει, μπορεί να λειτουργήσει άνετα σε ένα τοπικό δίκτου.

θα ήταν τεχνικά πιο εφικτό να δώσουμε πρόσβαση σε όλους τους υπολογιστές κατευθείαν στο internet, χωρίς να δημιουργήσουμε αυτό το τοπικό "ψεύτικο" για τον έξω κόσμο (internet) δίκτυο με την χρήση του router. Όμως δημιουργόντας το

προσεγγίζουμε την λογική και αποκτούμε εμπειρία στη χρήση μίας αρχής πιστοποίησης η οποία είναι καλά κρυμμένη απο τον "έξω κόσμο".

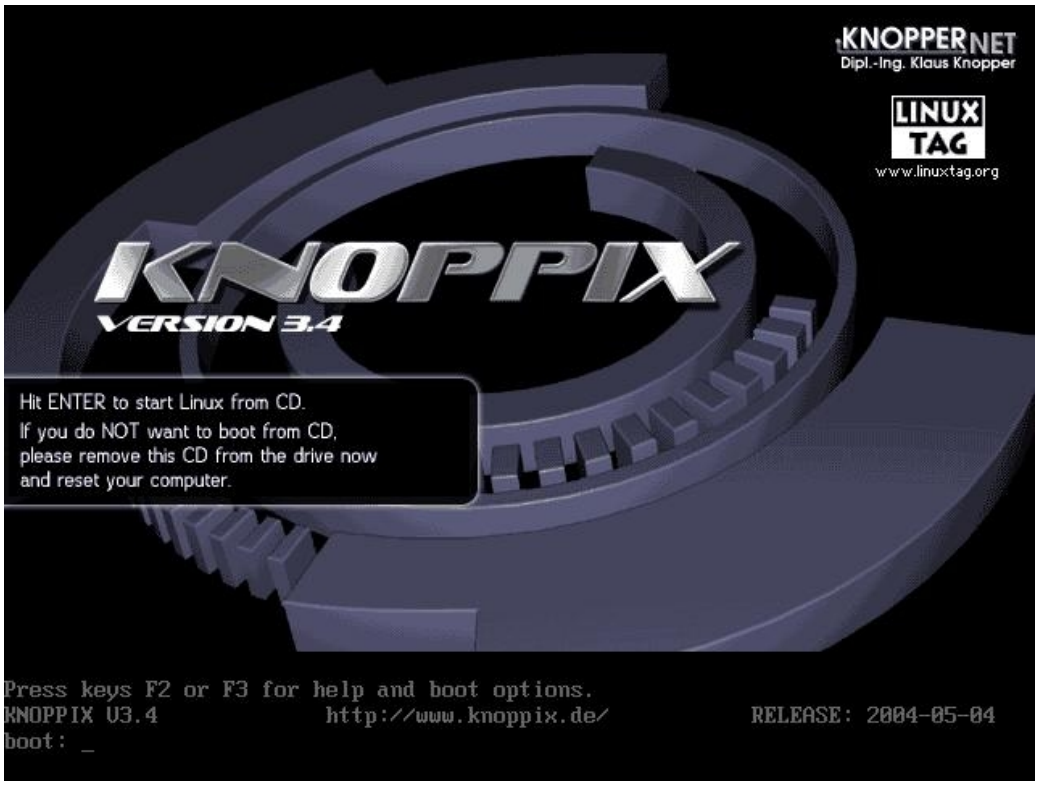
Απο την στιγμή που το OpenCa έχει εγκατασταθεί και είναι έτοιμο για λειτουργία μπορούμε να εγκαταστήσουμε κάποιο software σε Windows, MacOs X και intel-based Linux έτσι ώστε να μπορέσουμε να διαχειριστούμε συσκεύες Token. Στις συσκευές Token αποθηκεύουμε πιστοποιητικά τα οποία μπορούμε να χρησιμοποιήσουμε σε εφαρμογές στις οποίες χρειάζεται πιστοποίηση, όπως για παράδειγμα σε ένα browser. Το software μπορούμε να το προμηθευτούμε απο site όπως το http://www.aladdin.com, ανάλογα με το λειτουργικό σύστημα που διαθέτουμε επιλέγουμε και το αντίστοιχο software (για Windows, MacOs X και intel-based Linux).

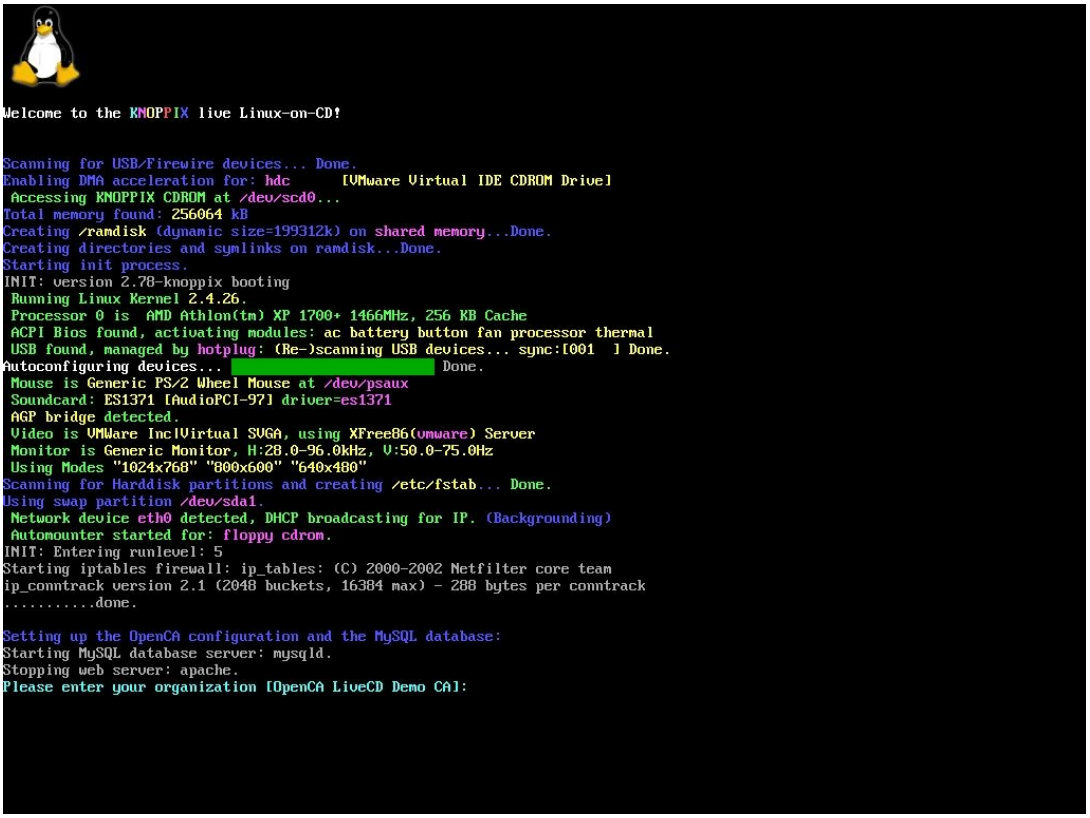
Σε περιβάλλον Windows Xp Pro το software είναι μια πλήρης σουίτα εφαρμογών η οποία σου επιτρέπει να εξετάσεις το περιεχόμενο των πιστοποιητικών που έχουν αποθηκευτεί στη usb συσκευή όπως επίσης και να τα χρησιμοποιήσεις για να τα προωθήσεις σε εφαρμογές όπως browsers και email clients. Σε περιβάλλον MacOs X και intel-based Linux τα πράγματα είναι λίγο διαφορετικά. Οι χρήστες MacOs X χρειάζεται να αρχικοποιήσουν ένα software το οποίο λέγεται PCSC/Lite το οποίο περιέχεται στο λειτουργικό MacOs X έτσι θα μπορούν να χρησιμοποιήσουν την εφαρμογή OpenSc η οποία σχετίζεται με smartcards. Επίσης τα πιστοποιητικά τα οποία αποθηκεύονται στο token έχουν διαφορετικό format στο MacOs X απο ότι σε Windows, γι αυτό το λόγο τα πιστοποιητικά δεν μπορούν να χρησιμοποιηθούν εναλλακτικά και στα δύο λειτουργικά συστήματα. Οι χρήστες Linux έχουν την επιλογή να ακολουθήσουν την ίδια διαδικασία με αυτή των χρηστών του MacOs X (δημιουργόντας συμβατά πιστοποιητικά MacOs X και intel-based Linux) ή μπορούν να χρησιμοποιήσουν επιπρόσθετο software το οποίο παρέχεται απο το http://www.aladdin.com μέσω του οποίου θα αρχικοποιήσουν την συσκευή Token και θα της περάσουν πιστοποίητικά σε format που αναγνωρίζεται και απο τα Windows (δημιουργόντας συμβατά πιστοποιητικά μεταξύ Linux και Windows λειτουργικών).

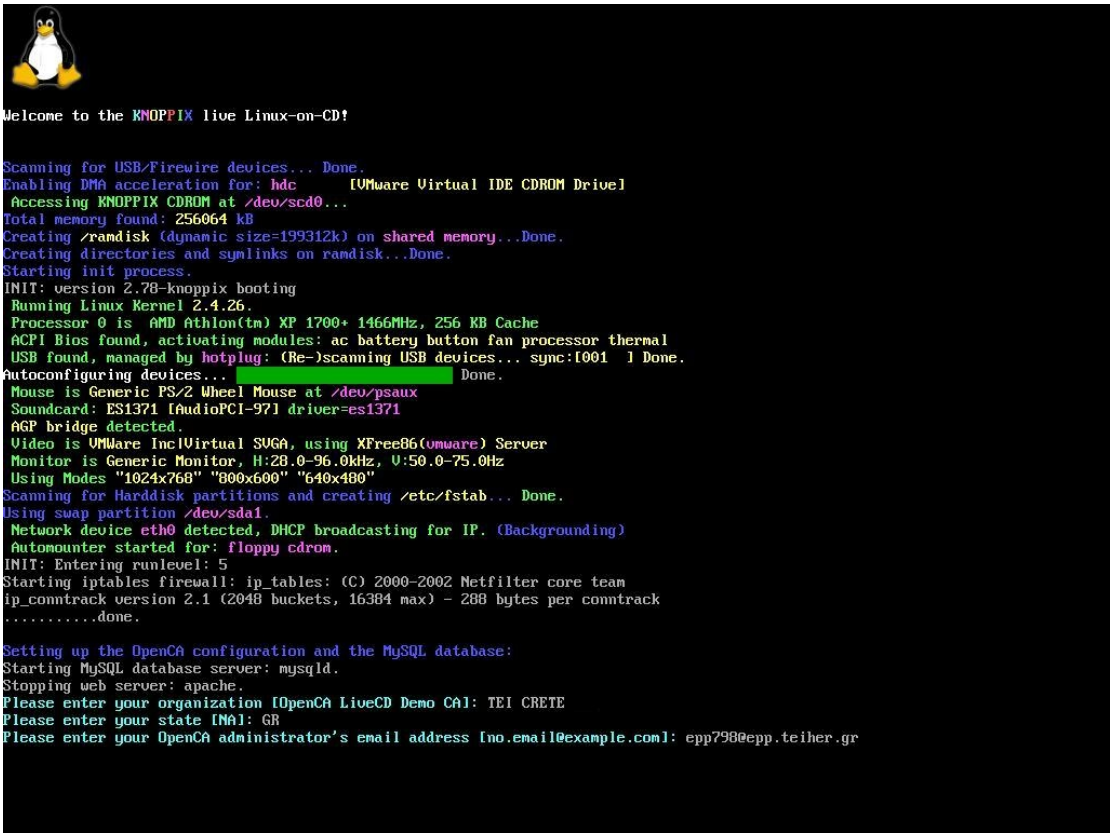
Όταν καταφέρουμε και περάσουμε πιστοποιητικά στη συσκευή token θα μπορούμε πλέον να την χρησιμοποιούμε οπουδήποτε χρειάζεται πιστοποίηση (αποστολή encrypted email, υπογραφή σε αρχεία pdf, απόκτηση πρόσβασης σε ένα ασφαλή web server κλπ) εύκολα απλά και με ασφάλεια χωρίς να μας απασχολεί η αλλοίωση των πιστοποιητικών μας απο κάποιον κακόβουλο χρήστη.

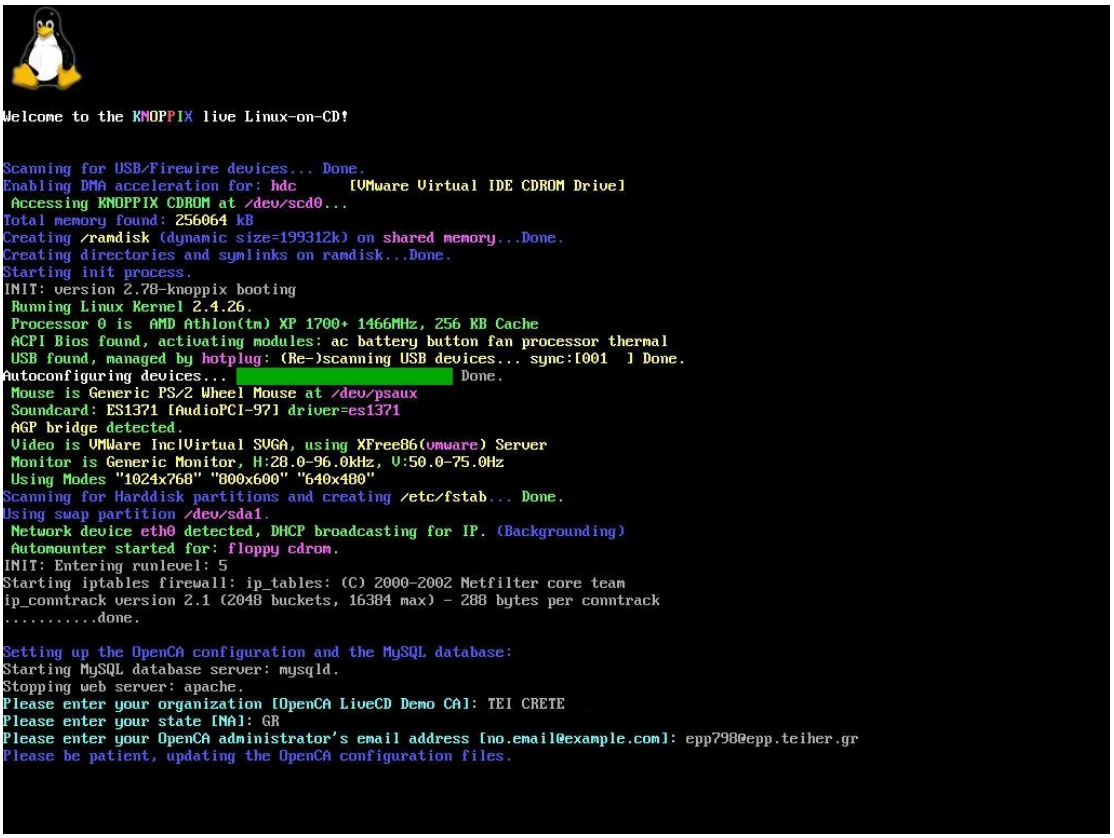
2. Εγκατάσταση/Αρχικοποίηση OpenCa Live Cd

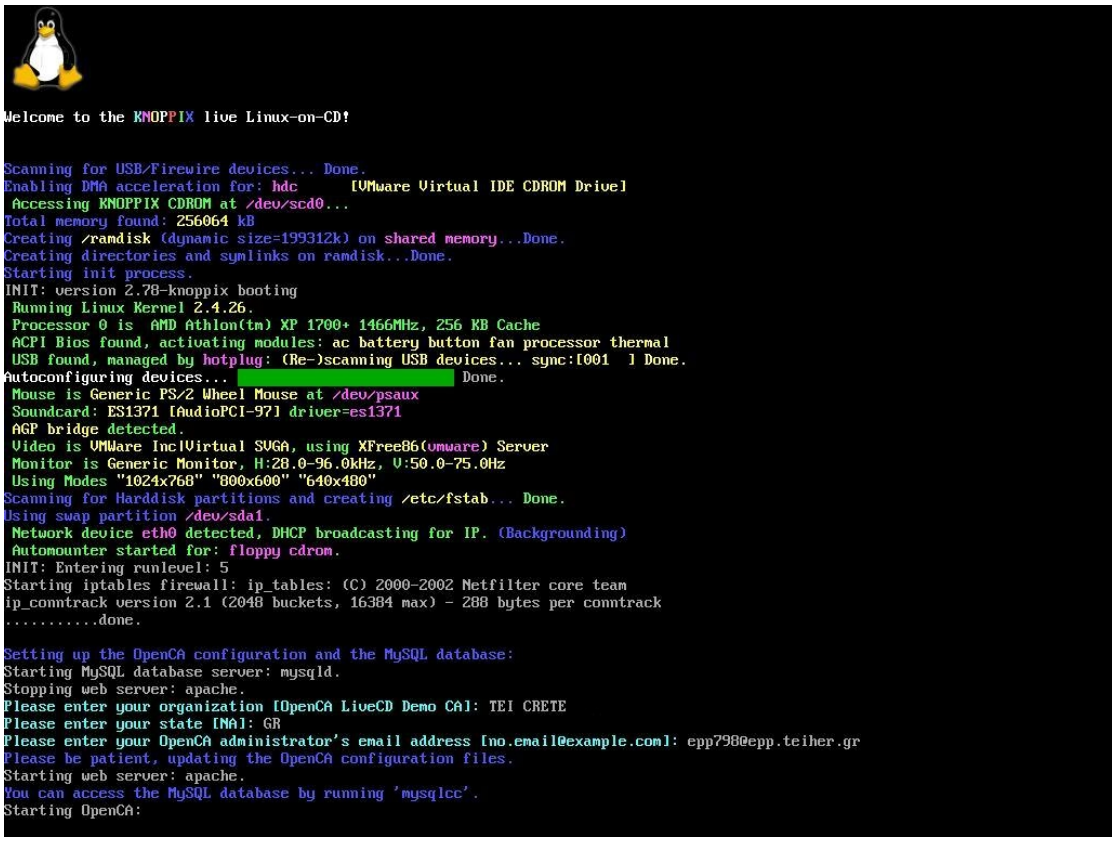
Για να σετάρουμε το **OpenCa Live Cd** δημιουργούμε ένα Boot cd και κάνουμε boot απο το cd, μας ζητείται να ορίσουμε Organization, Location(State) και email, παρακάτω παρατηρούμε τις σχετικες οθόνες.




Welcome to the KNOPPIX live Linux-on-CD!
Scanning for USB/Firewire devices... Done.
Enabling DMA acceleration for: hdc [VMware Virtual IDE CDROM Drive]
Accessing KNOPPIX CDROM at /dev/scd0...
Total memory found: 256064 kB
Creating /ramdisk (dynamic size=199312k) on shared memory...Done.
Creating directories and symlinks on ramdisk...Done.
Starting init process.
INIT: version 2.78-knoppix booting
Running Linux Kernel 2.4.26.
Processor 0 is AMD Athlon(tm) XP 1700+ 1466MHz, 256 KB Cache
ACPI Bios found, activating modules: ac battery button fan processor thermal
USB found, managed by hotplug: (Re-)scanning USB devices... sync:[001] Done.
Autoconfiguring devices... Done.
Mouse is Generic PS/2 Wheel Mouse at /dev/psaux
Soundcard: ES1371 [AudioPCI-97] driver=es1371
AGP bridge detected.
Video is VMWare Inc|Virtual SVGA, using XFree86(vmware) Server
Monitor is Generic Monitor, H:28.0-96.0kHz, V:50.0-75.0Hz
Using Modes "1024x768" "800x600" "640x480"
Scanning for Harddisk partitions and creating /etc/fstab... Done.
Using swap partition /dev/sda1.
Network device eth0 detected, DHCP broadcasting for IP. (Backgrounding)
Automounter started for: floppy cdrom.
INIT: Entering runlevel: 5
Starting iptables firewall: ip_tables: (C) 2000-2002 Netfilter core team
ip_conntrack version 2.1 (2048 buckets, 16384 max) - 288 bytes per conntrack
.....done.
Setting up the OpenCA configuration and the MySQL database:
Starting MySQL database server: mysqld.
Stopping web server: apache.
Please enter your organization [OpenCA LiveCD Demo CA]:


Welcome to the KNOPPIX live Linux-on-CD!
Scanning for USB/Firewire devices... Done.
Enabling DMA acceleration for: hdc [VMware Virtual IDE CDROM Drive]
Accessing KNOPPIX CDROM at /dev/scd0...
Total memory found: 256064 kB
Creating /ramdisk (dynamic size=199312k) on shared memory...Done.
Creating directories and symlinks on ramdisk...Done.
Starting init process.
INIT: version 2.78-knoppix booting
Running Linux Kernel 2.4.26.
Processor 0 is AMD Athlon(tm) XP 1700+ 1466MHz, 256 KB Cache
ACPI Bios found, activating modules: ac battery button fan processor thermal
USB found, managed by hotplug: (Re-)scanning USB devices... sync:[001] Done.
Autoconfiguring devices... Done.
Mouse is Generic PS/2 Wheel Mouse at /dev/psaux
Soundcard: ES1371 [AudioPCI-97] driver=es1371
AGP bridge detected.
Video is VMWare Inc|Virtual SVGA, using XFree86(vmware) Server
Monitor is Generic Monitor, H:28.0-96.0kHz, V:50.0-75.0Hz
Using Modes "1024x768" "800x600" "640x480"
Scanning for Harddisk partitions and creating /etc/fstab... Done.
Using swap partition /dev/sda1.
Network device eth0 detected, DHCP broadcasting for IP. (Backgrounding)
Automounter started for: floppy cdrom.
INIT: Entering runlevel: 5
Starting iptables firewall: ip_tables: (C) 2000-2002 Netfilter core team
ip_conntrack version 2.1 (2048 buckets, 16384 max) - 288 bytes per conntrack
.....done.
Setting up the OpenCA configuration and the MySQL database:
Starting MySQL database server: mysqld.
Stopping web server: apache.
Please enter your organization [OpenCA LiveCD Demo CA]: TEI CRETE
Please enter your state [NA]: GR
Please enter your OpenCA administrator's email address [no.email@example.com]: epp798@epp.teiher.gr



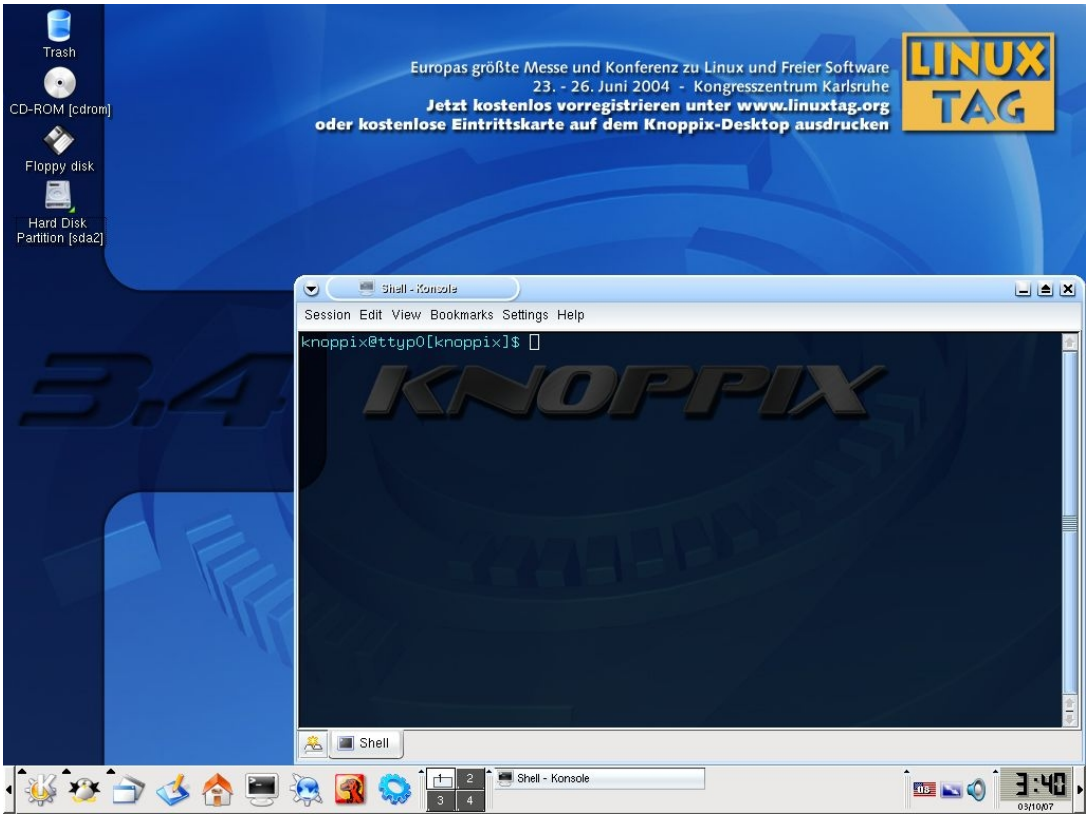


Παρακάτω παρατηρούμε την αρχική είκονα της διανομής Linux με ονομασία Knoppix (version 3.4)

2. Διαμορφώνοντας το OpenCA δίνοντας του μόνιμο αποθηκευτικό χώρο

Σε αυτό το σημείο το OpenCa είναι έτοιμο να ξεκινήσει όμως το σύστημα μας λογώ του ότι "φορτώνεται" απο Livecd αδυνατεί να αποθηκεύσει δεδομένα σε κάποιο μόνιμο χώρο. Γι αυτό το λόγο θα χρησιμοποιήσουμε ένα flash disk 1GB (απο 128 MB και πάνω είναι ικανοποιητικό μέγεθος), στο οποίο θα αποθηκεύουμε πιστοποιητικά και ότι άλλες πληροφορίες δίνει ο πάροχος πιστοποίησεις οι οποίες χρειάζονται αποθήκευση.

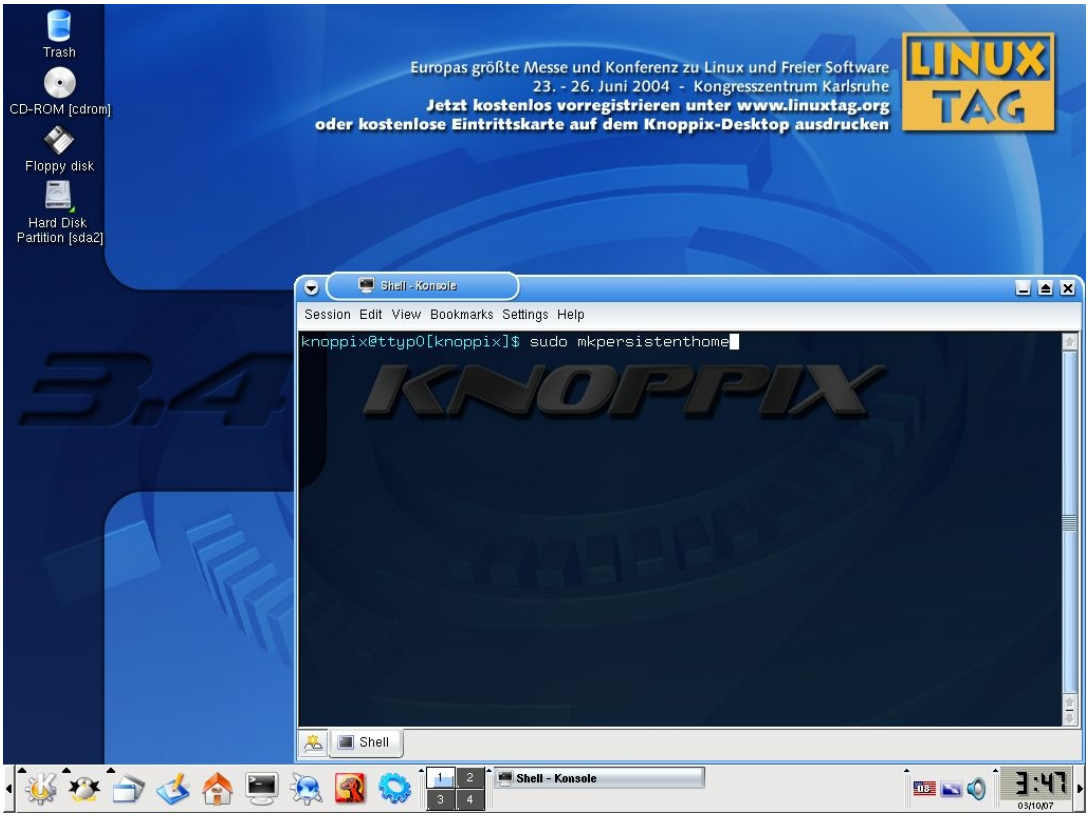
Για να χρησιμοποιήσουμε το flash disk θα πρέπει αρχικά να το "δεί" το σύστημα και στην συνέχεια να το δεσμεύσει. Όποτε ανοίγουμε την γραμμή εντολών του Knoppix απο το εικονίδιο που βλέπουμε στην μπάρα στο κάτω μέρος του Desktop, θα πρέπει να εμφανιστεί μια οθόνη όπως η παρακάτω



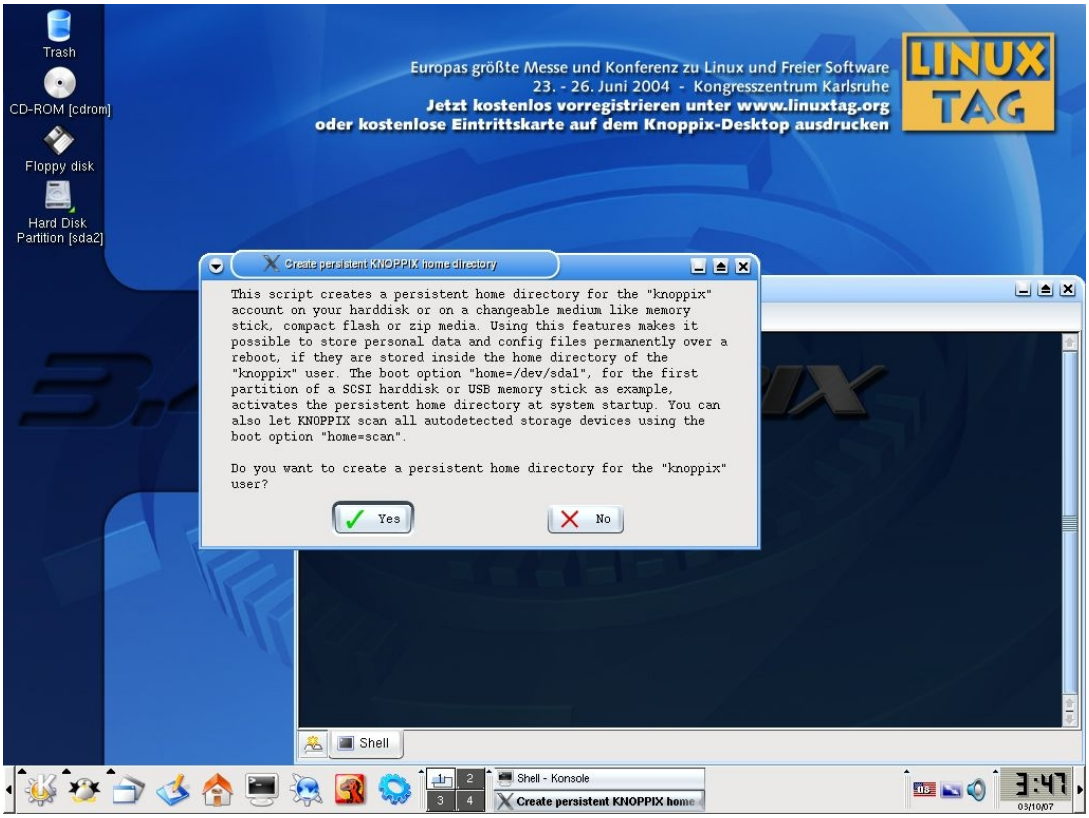
Στη διανομή Linux knoppix τα περισσότερα flash disks έχουν μια κωδική ονομασία συσκευής όπως /dev/sda1 ή /dev/sda2, όμως για να είμαστε σίγουροι για την ονομασία της συσκευής την τοποθετούμε στον υπολογιστή μας. Θα εμφανισθεί ένα εικονίδιο με ονομασία Hard Disk Partion και μέσα σε παρένθεση θα υπάρχει η κωδική ονομασία της συσκευής [sda2], όπως παρατηρούμε στην παρακάτω εικόνα

Γνωρίζοντας το όνομα του flash disk είμαστε έτοιμοι να χρησιμοποιήσουμε την γραμμή εντολών έτσι ώστε να το δεσμεύσουμε για μόνιμο αποθηκευτικό χώρο του

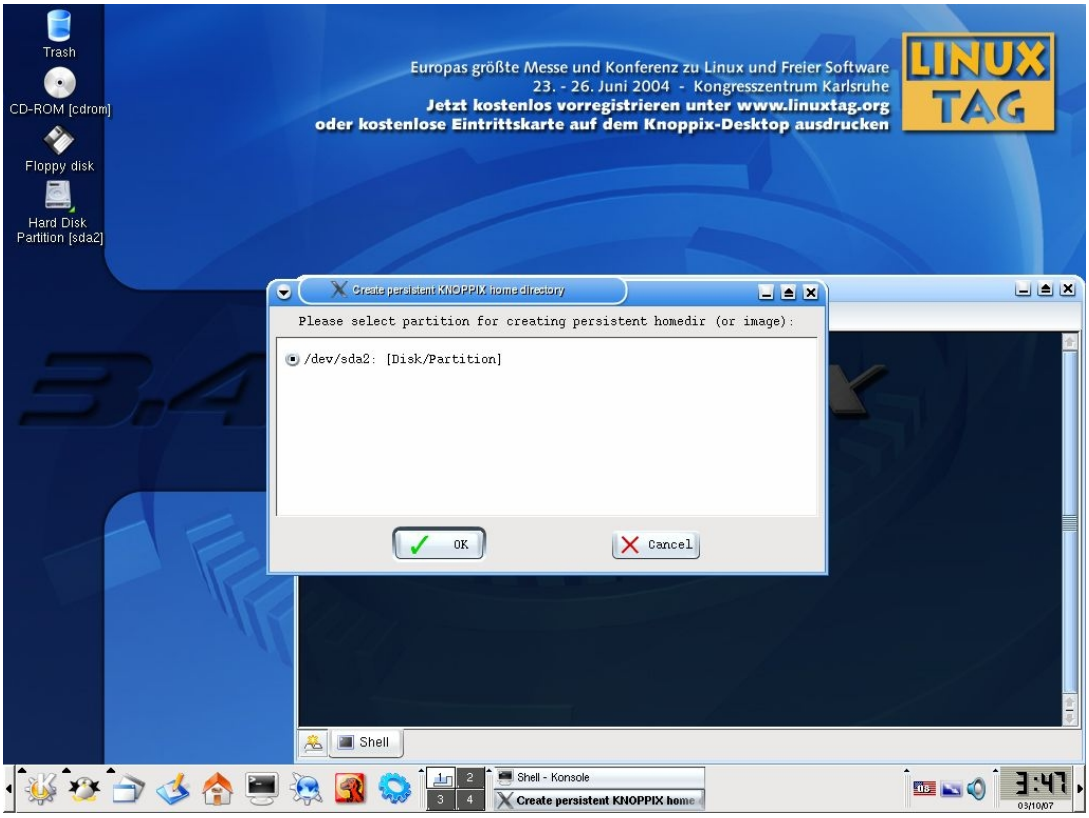
συστήματος. Πηγαίνουμε στην γραμμή εντολών και δίνουμε **sudo mkpersistenthome**, όπως βλέπουμε και στην παρακάτω οθόνη



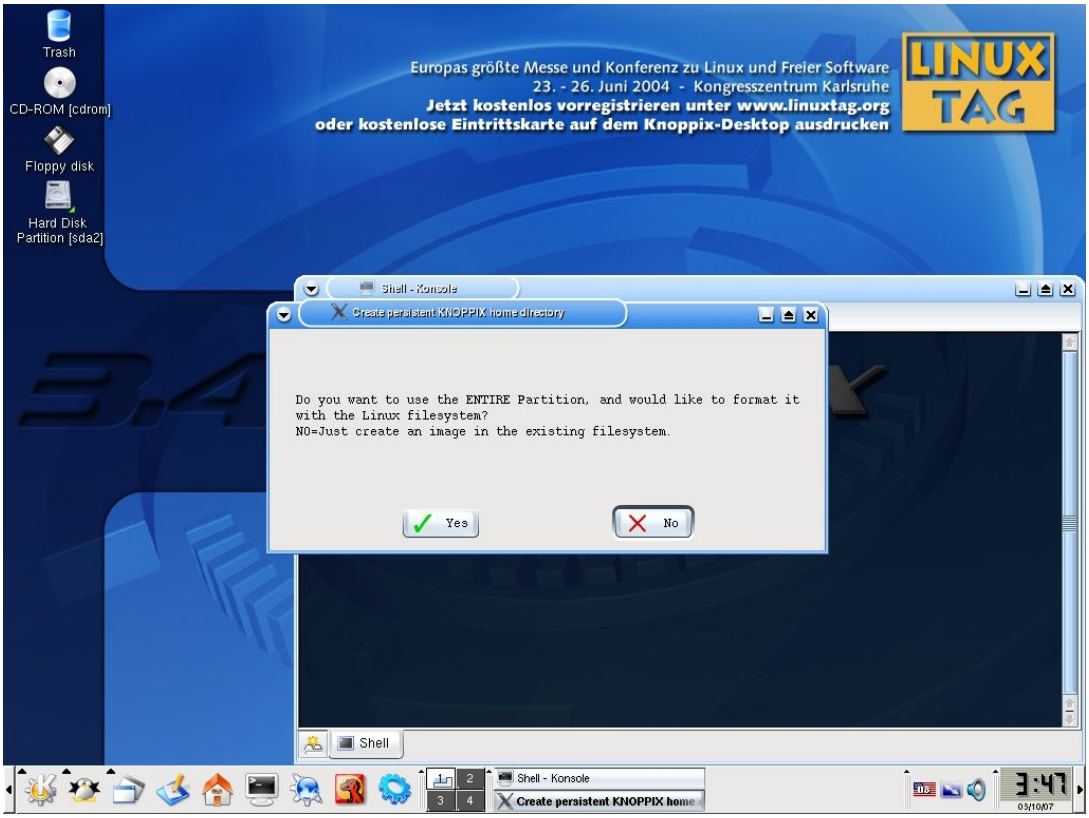
Δίνοντας την εντολή θα μας εμφανισθεί ένα μήνυμα όπως το παρακάτω



Επιλέγοντας “Yes” εμφανίζεται μια οθόνη στην οποία προσδιορίζουμε την ονομασία της συσκευής αποθήκευσης (flash disk). Είναι πιθανό να υπάρχει παραπάνω από μια επιλογές, εμείς τσεκάρουμε την κωδική ονομασία /dev/sda2 που αφορά το flash disk.

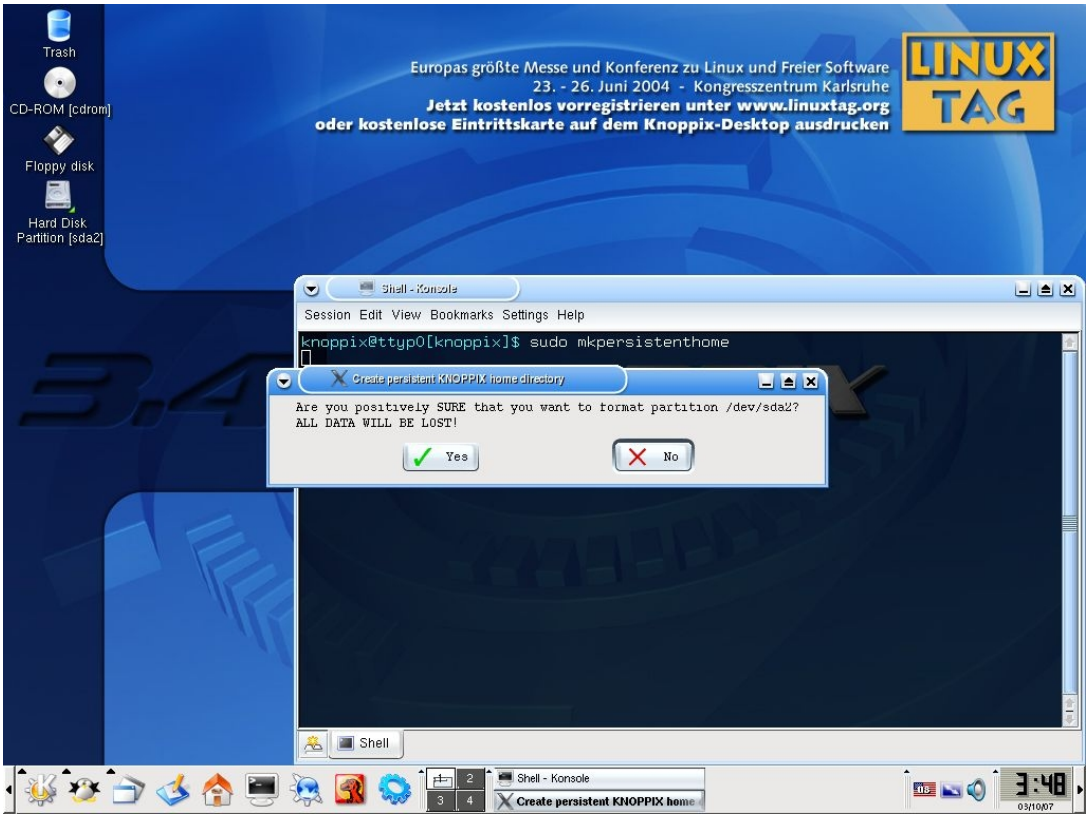


Στη συνέχεια θα πρέπει να επιλέξουμε αν θα χρησιμοποιήσουμε ολόκληρο το χώρο του flash disk για μόνιμη αποθήκευση του συστήματος ή θα δημιουργήσουμε ένα image το οποίο θα περιέχεται στο flash disk και μέσα του θα αποθηκευονται τα δεδομένα μας.

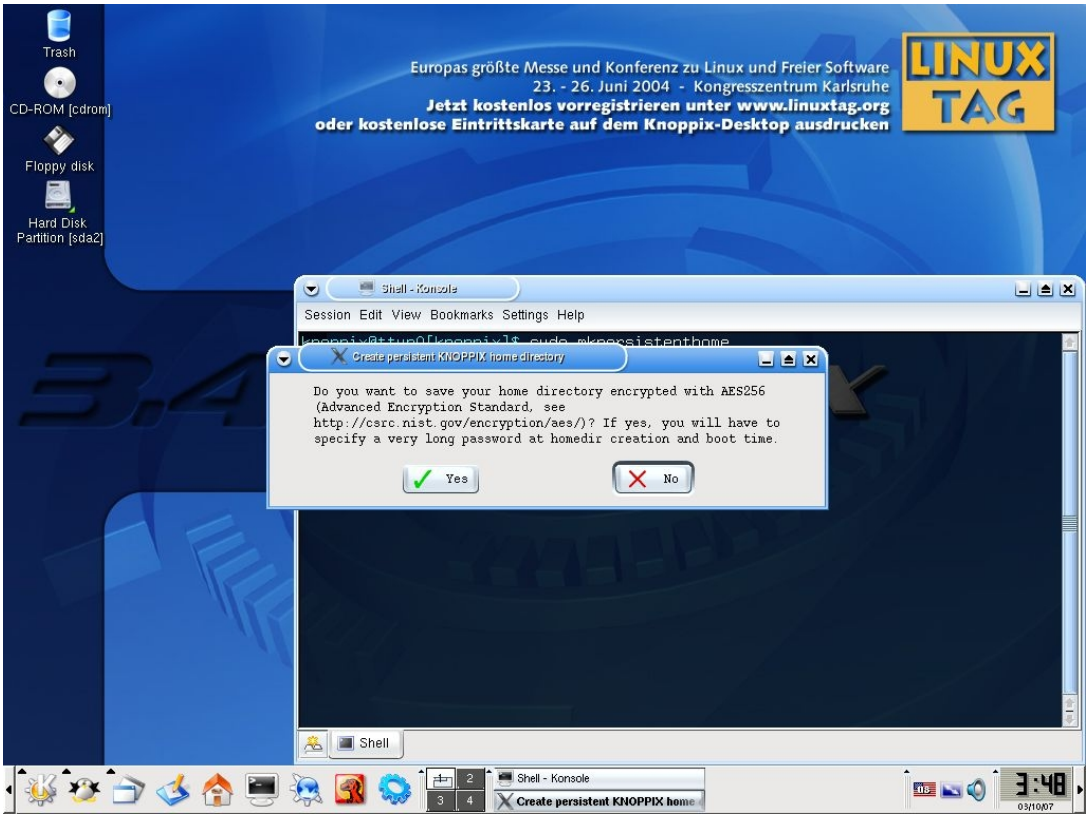


Είναι ευκολότερο και πιο ασφαλές να δεσμεύσουμε ολόκληρο το flash disk αποκλειστικά για το OpenCA, με αυτό τον τρόπο διασφαλιζουμε τα δεδομένα μας χωρίς να υπάρχει ο φόβος να γίνει επικάλυψη δεδομένων ή κάποιο format στο flash disk. Πρέπει να είμαστες σίγουροι ότι το το flash disk δεν περιέχει πολύτιμα δεδομένα δίοτι επιλέγοντας “Yes” θα γίνει format και θα χάθουν όλα τα περιεχόμενα του.

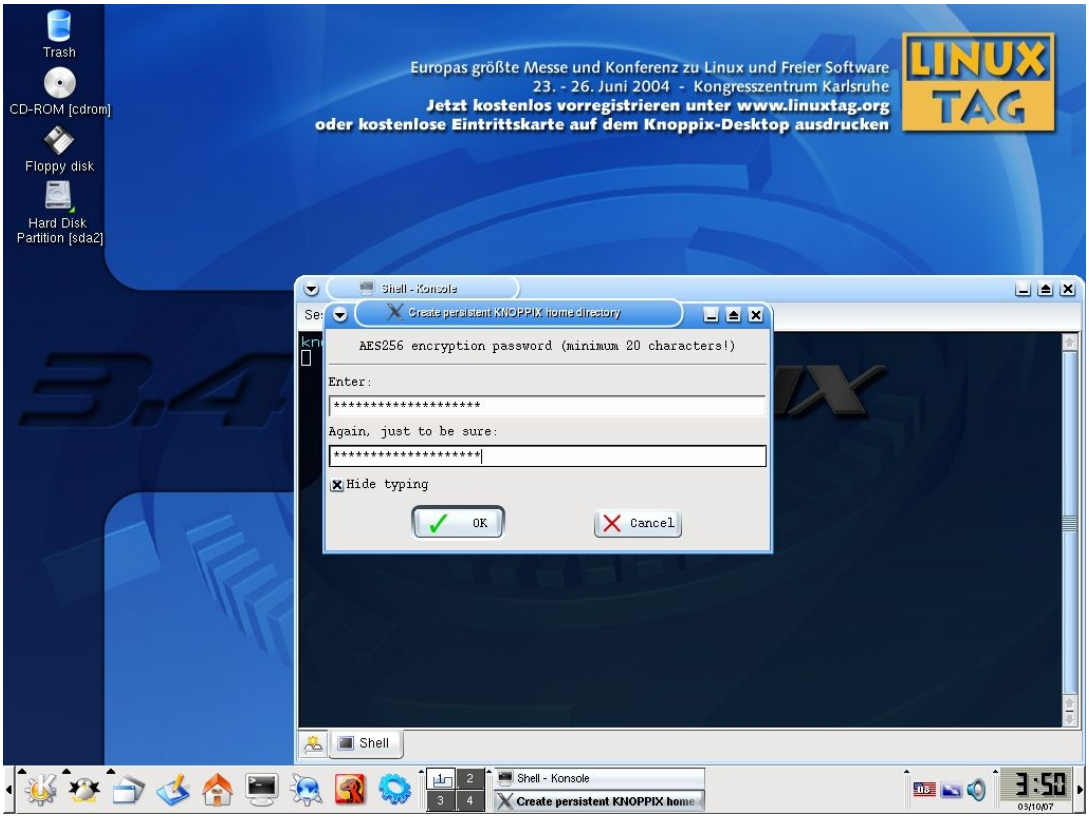
Σε αυτό το βήμα μας ζητάει το σύστημα να επιβεβαιώσουμε ότι θέλουμε να διαγραφεί το περιεχόμενο του flash disk.



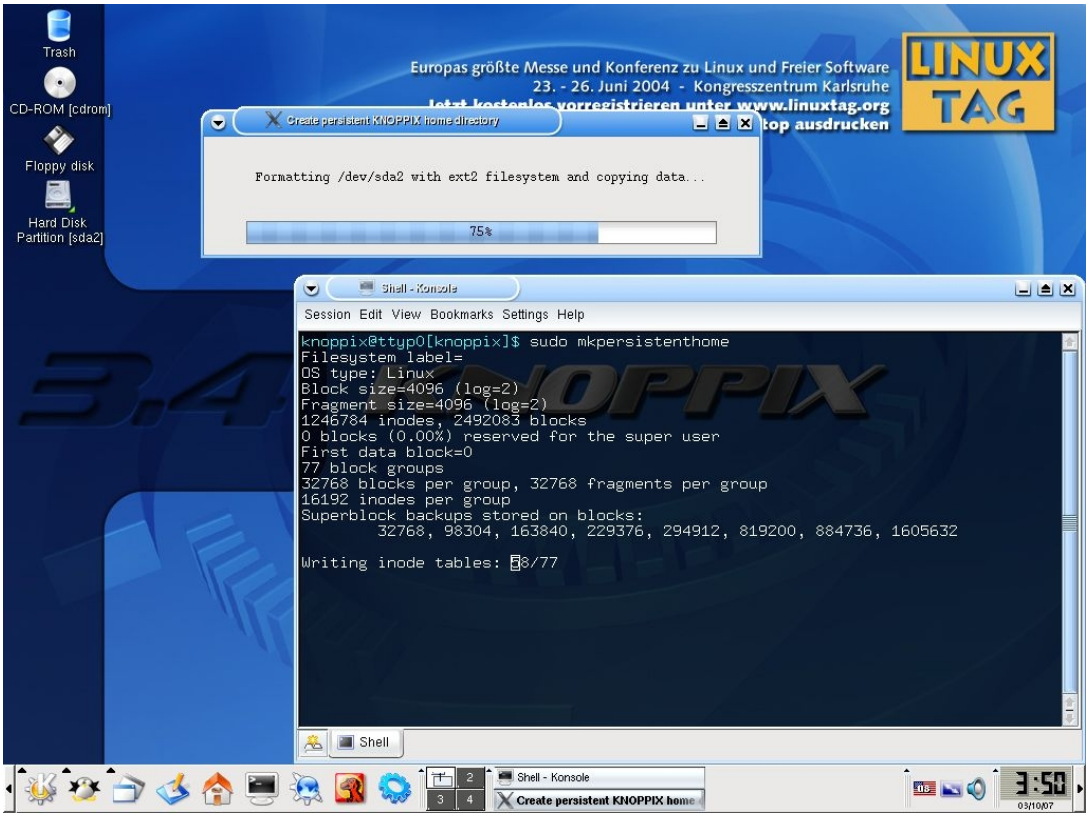
Επιλέγουμε “Yes” για να επιβεβαιώσουμε ότι θέλουμε να διαγραφουν τα περιεχόμενα του flash disk. Πρίν αρχίσει η διαδικασία του format θα μας γίνει ερώτηση αν θέλουμε να κρυπτογραφίσουμε το file system το οποίο θα δημιουργηθεί στο flash disk. Με αυτό τον τρόπο διασφαλίζουμε την ασφάλεια της συσκευής στην περίπτωση που θα χαθεί ή θα κλάπει.



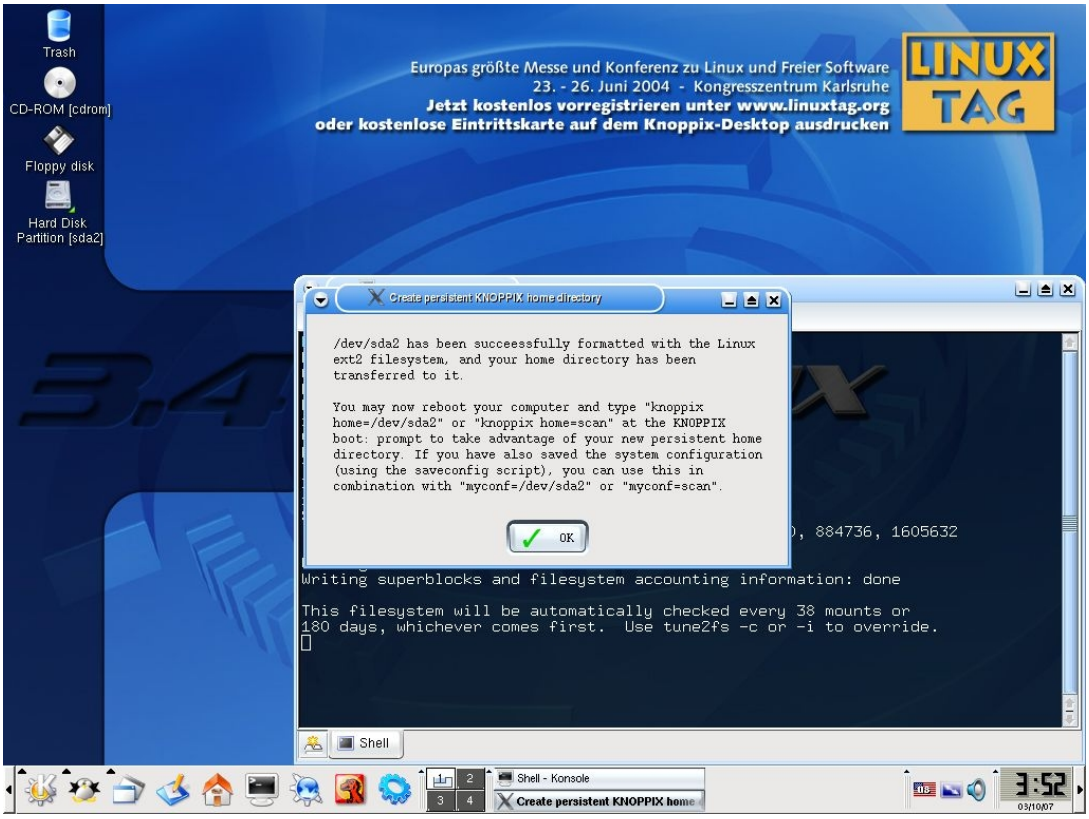
Θεωρητικά υπάρχει μια ένσταση για το αν θα πρέπει να υπάρχουν τόσες δικλείδες ασφαλείας σε ένα σύστημα που ο σκοπός του είναι εκπαιδευτικός, παράλα αυτά εμείς θα ενεργόποιήσουμε όλες τις δικλείδες που μας παρέχονται από το σύστημα. Επίλεγοντας “Yes” θα μας ζητηθεί ένα password το οποίο θα πρέπει να έχει τουλάχιστον 20 χαρακτήρες. Πρεπει να το εισάγουμε στο παρακάτω παράθυρο.



Όταν επιλέξουμε “Ok” το password θα ταυτοποιηθεί με το δεύτερο που δώσαμε, αν είναι τα ίδια τότε το flash disk θα φορμαριστεί. Θα εμφανίστει μια οθόνη με πληροφορίες σχετικά με το format.



Όταν το format ολοκληρωθεί θα εμφανισθεί ένα παράθυρο επιβεβαίωσης



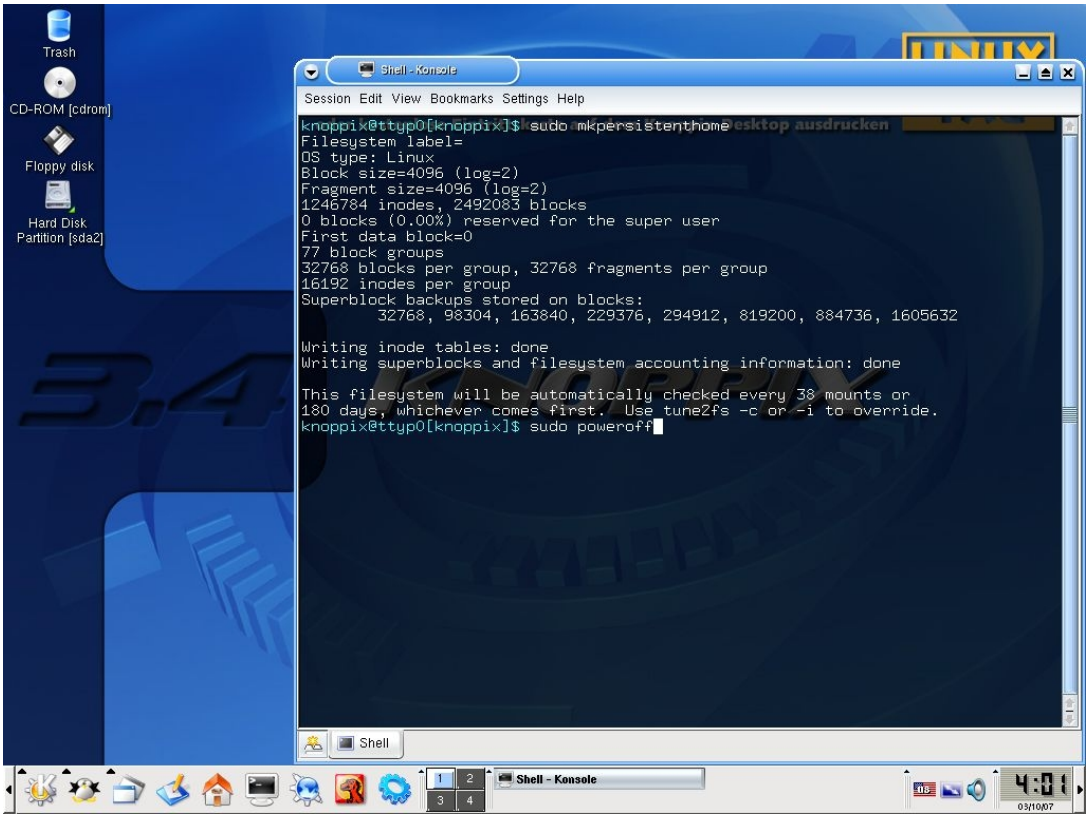
Θα πρέπει να έχουμε υπόψιν ότι η συσκευή η οποία στην δική μας περίπτωση έχει την ονομασία /dev/sda2 από συστήμα σε σύστημα ενδέχεται η κωδική ονομασία να αλλάζει.

Σε αυτό το σημείο το OpenCa είναι έτοιμο να παράγει και να διανέμει πιστοποιητικά. Η εντολή mkpersistenthome αυτόματα αντιγράφει τα περιεχόμενα του /home/Knoppix στο flash disk, άρα όσες αιτήσεις για πιστοποιητικά γίνουν στο OpenCa και όσα πιστοποιητικά δημιουργηθούν θα αποθηκευόνται από τώρα και στο εξής στο flash disk. Πρίν ξεκινήσουμε να πούμε για το πώς θα χειριζόμαστε το OpenCA θα πούμε για το πώς θα κλείνουμε το σύστημα το οποίο υπενθυμίζουμε ότι τρέχει σε livecd.

2.1 Πως κάνουμε Shut Down το OpenCA liveCD

Το OpenCa/Knoppix σύστημα είναι ένα συστήμα Linux το οποίο έχει προεγκατεστημένο το OpenCa. Κατά την διάρκεια του boot το σύστημα ψάχνει να βρεί τον αποθηκευτικό χώρο που του έχουμε ορίσει, αν δεν έχουμε ορίσει αποθηκευτικό χώρο θα σεταριστεί από την αρχή. Θα μας γίνουν τρείς ερωτήσεις οι οποίες σχετίζονται με το όνομα του Οργανισμό μας, με τα αρχικά γράμματα της χώρας μας (GR) καθώς και το email μας.

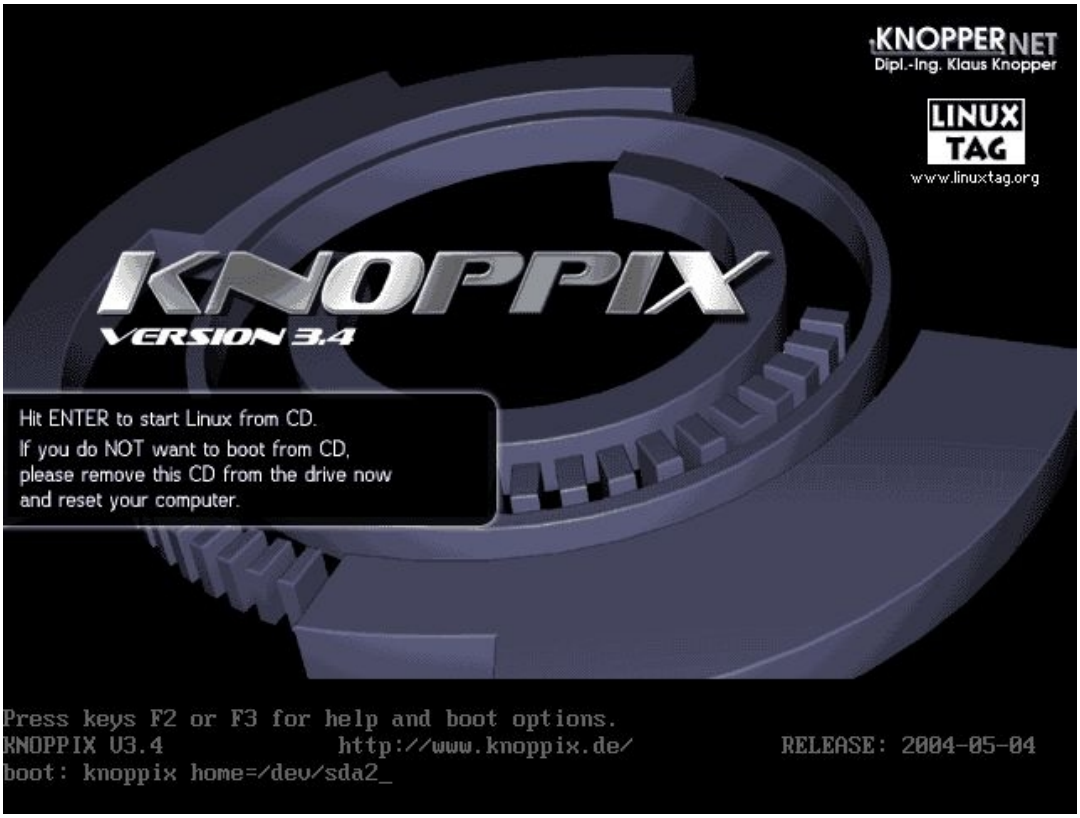
Μέχρι στιγμής έχουμε δεσμέυσει το flash disk για μόνιμο αποθηκευτικό χώρο του συστήματος μας, όμως ακόμα δεν έχει ενεργοποιηθεί, για να συμβεί αυτό θα πρέπει να κλείσουμε το σύστημα μας και να το ξεκινήσουμε από την αρχή. Αυτό θα συμβεί δίνοντας σε γραμμή εντολών την εντολή **sudo poweroff** (λόγω του ότι η εντολή poweroff για να εκτελεστει χρειάζεται δικαιώματα administrator, βάζουμε μπροστά sudo) ή επιλέγοντας από το κάτω μέρος του Desktop στα αριστερά το εικονίδιο με το γράμμα K, επιλεγοντάς το μας εμφανίζεται ένα menu έκει επιλέγουμε Log off και στην συνέχεια Shut down. Παρακάτω φαίνεται η γραμμή εντολών.



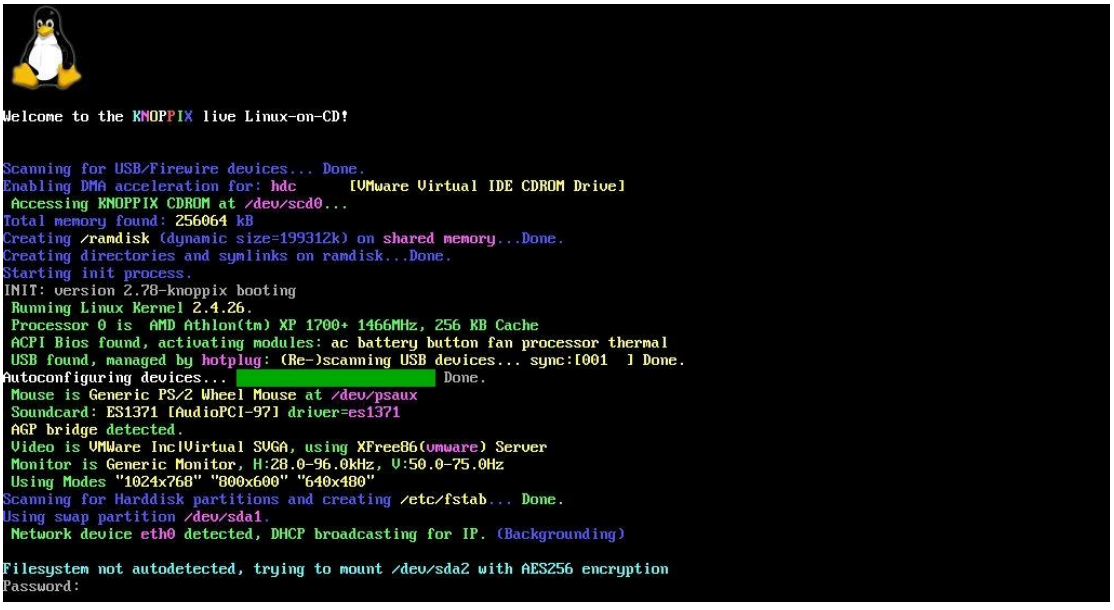
Με αυτό τον τρόπο κάνουμε Shut Down στο σύστημα, βγάζουμε το CD (πιθανόν να έχει ήδη βγεί) και μαζί με το flash disk τα αποθηκεύουμε σε ασφαλές μέρος για να τα χρησιμοποιήσουμε αργότερα.

2.2 Ξεκινόντας το OpenCa έχοντας δεσμεύσει μόνιμο χώρο

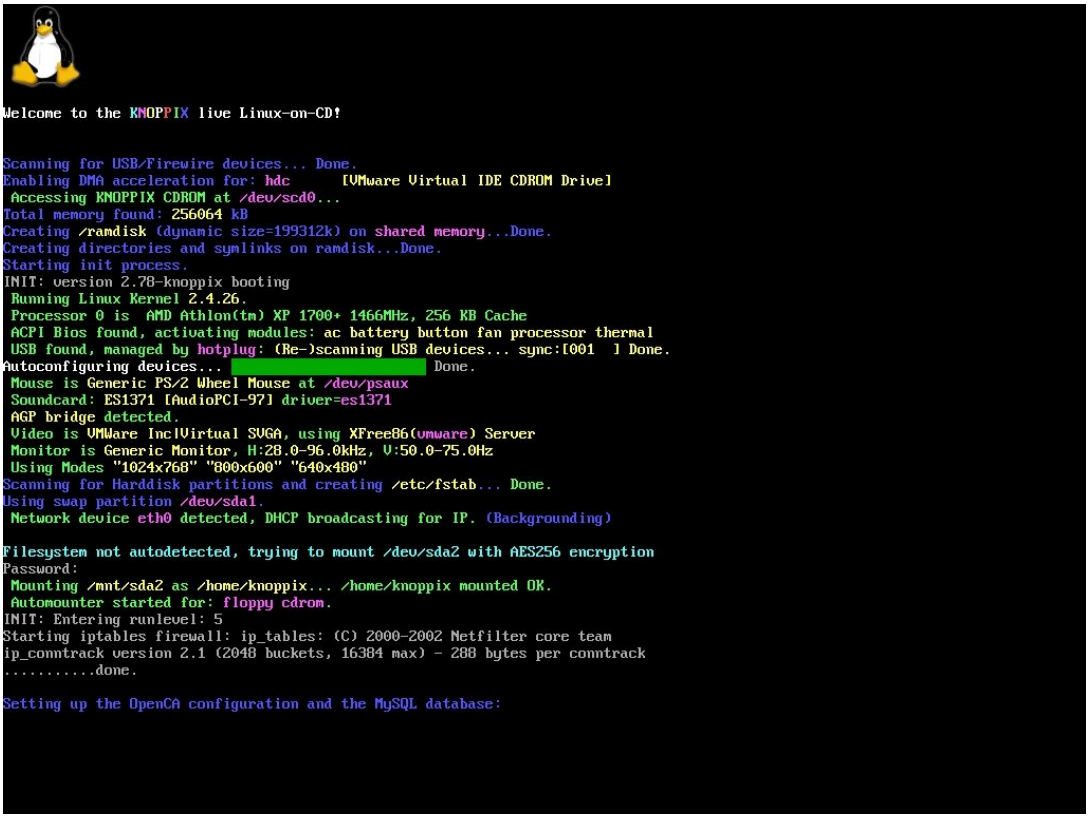
Όταν ξεκινάμε το σύστημα θα πρέπει να του δίνουμε την διαδρόμη στην οποία βρίσκεται η συσκευή flash disk που χρησιμοποιούμε για μόνιμο αποθηκευτικό χώρο. Η διαδικασία αυτή γίνεται αμέσος το boot από το OpenCA Livecd. Χρειάζεται λοιπόν να πούμε στο σύστημα που βρίσκεται η συσκευή αποθήκευσης και αυτό γίνεται με την εντολή **Boot: Knoppix home=/dev/sda2**



Το σύστημα θα κάνει boot με τον ίδιο τρόπο που έκανε και πρίν με μία μόνο διαφορά, θα δεσμεύσει πρώτα τον αποθηκευτικό χώρο από το flash disk και μετά θα δώσει στο σύστημα διεύθυνση ip μέσω του DHCP. Σε αυτό το σημείο θα πρέπει να δώσουμε τον κωδικό των 20 χαρακτήρων με τον οποίο είχαμε ασφαλίσει το flash disk.



Αν το password δεν είναι σωστό θα πρέπει να το ξαναδώσουμε, αν όλα πάνε καλά το σύστημα θα φορτώσει το OpenCA



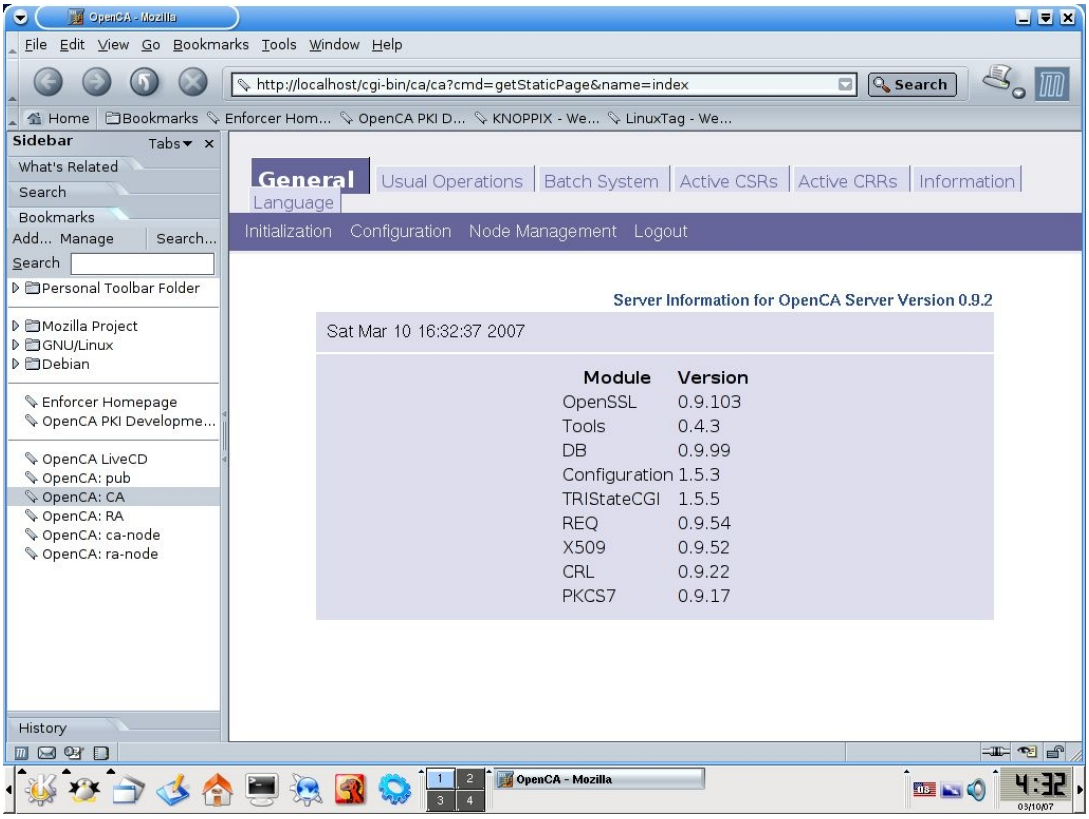
Όταν τελειώσει το boot το σύστημα θα είναι έτοιμο για χρήση

3. Διαμόρφωση της εγκατάστασης του Openca Live cd

3.1 Αρχικοποιόντας το OpenCA

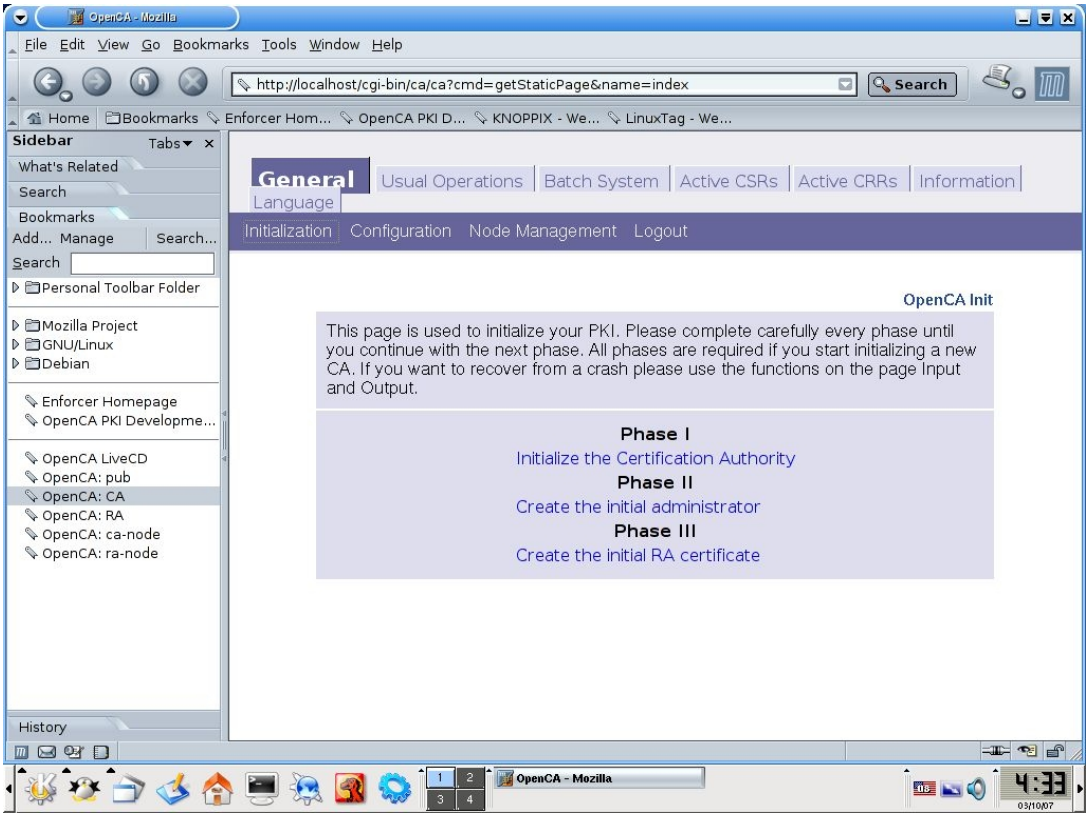
Πρίν αρχίσουμε να χρησιμοποιούμε το OpenCa για την παραγωγή πιστοποιητικών ή οποία άλλη λειτουργία θέλουμε θα πρέπει να το αρχικοποιήσουμε ακολουθόντας τα παρακάτω βήματα.

1. Για να συνδεθούμε στο OpenCa Live Cd ανοίγουμε ένα browser (τα Knoppix έχουν ενσωματωμένο το Mozilla) επιλέγοντας από την μπάρα στο κάτω μέρος του desktop το κόκκινο εικονίδιο με το δεινόσαυρο, και δίνουμε **http://localhost/ca/** για να συνδεθούμε στις σελίδες της αρχής πιστοποίησης ή επιλέγουμε από την αριστερή στήλη των bookmarks το link OpenCA:CA.

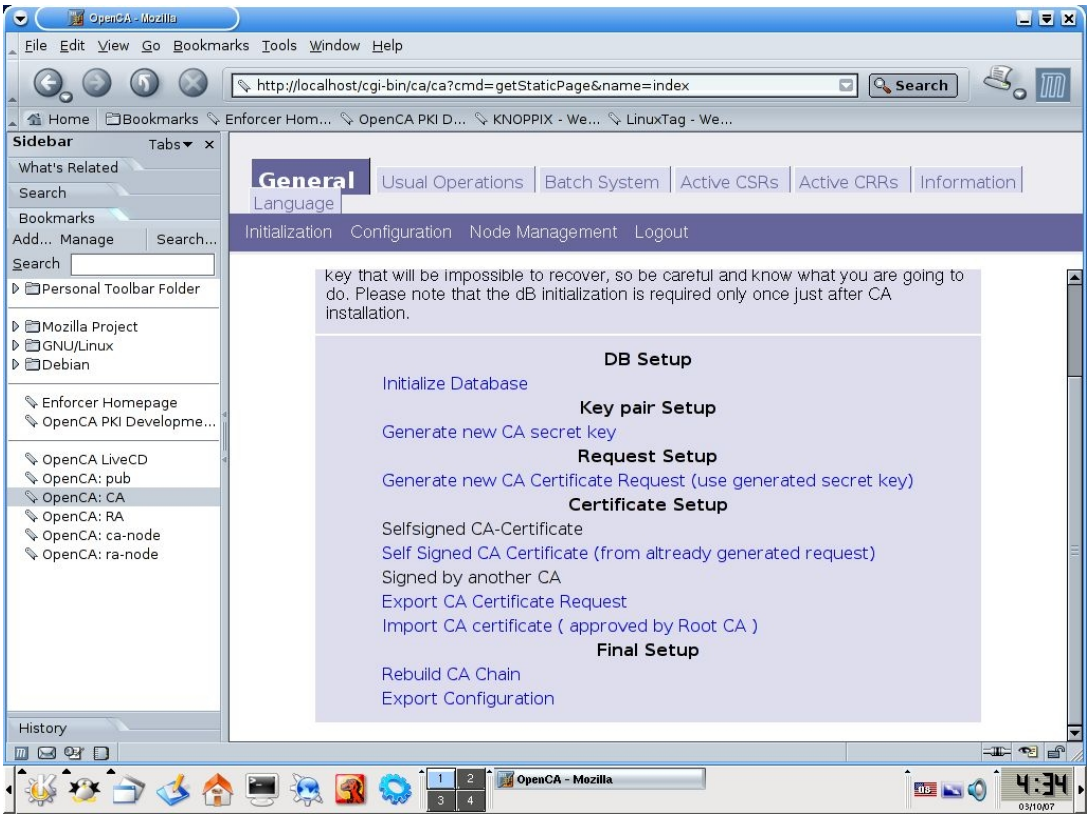


3.1.1 Φάση I : Αρχικοποίηση της Αρχής πιστοποίησης (CA)

Επιλέγουμε την καρτέλα General και μετά “**Initialization**” στην μπλέ μπάρα κάτω από την καρτέλα, θα εμφανισθεί η παρακάτω οθόνη

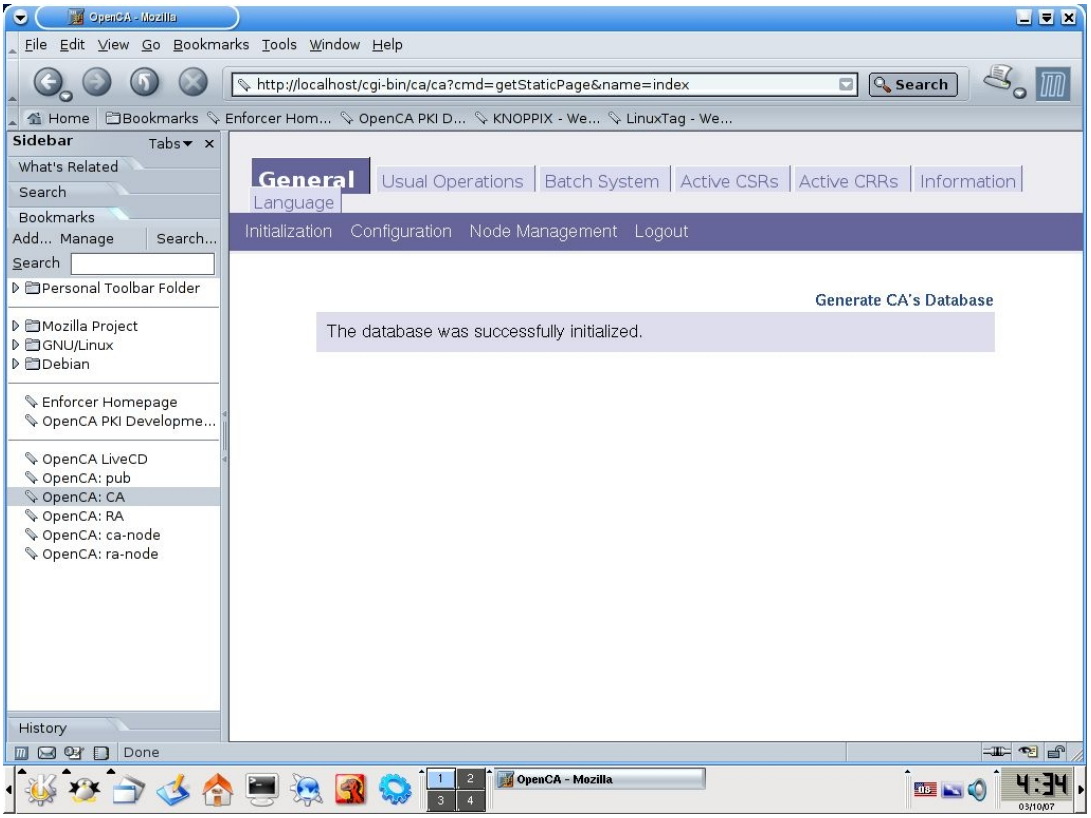


2. Παρατηρούμε ότι υπάρχουν τρείς φάσεις αρχικοποίησης, στην πρώτη φάση επιλέγουμε "**Initialize the Certificate Autority"** και βλέπουμε την παρακάτω οθόνη

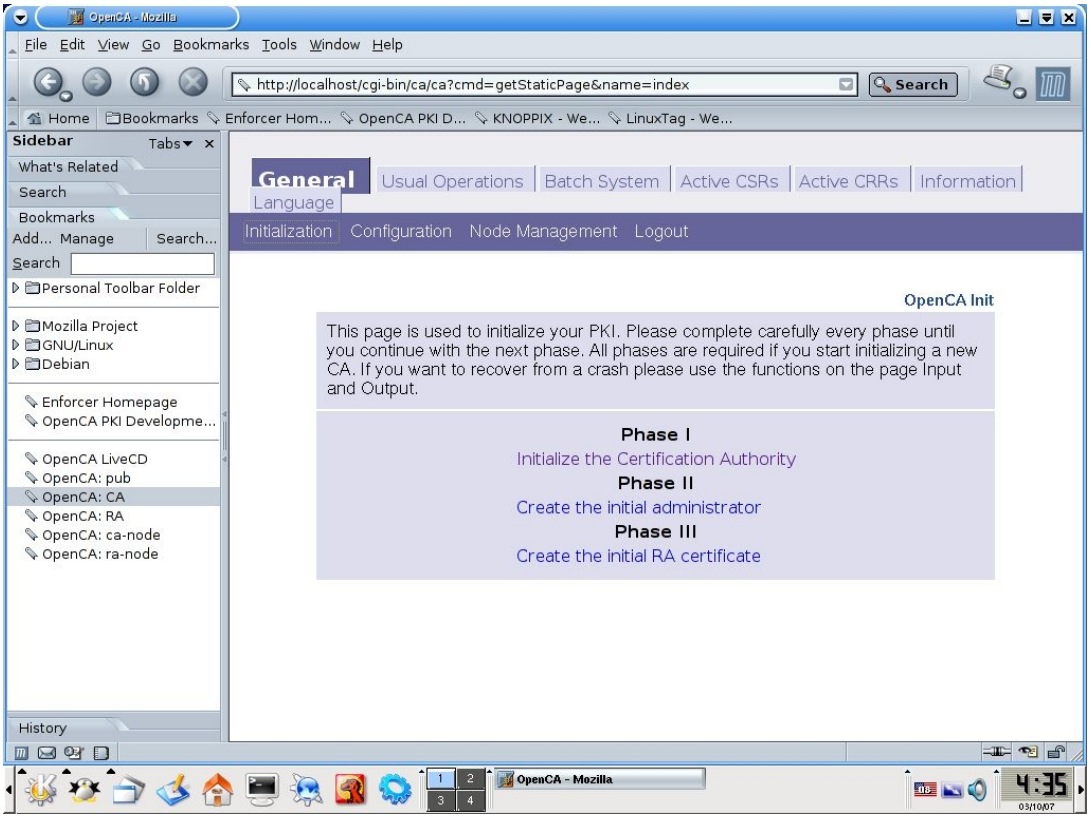


Η παραπάνω οθόνη σχετίζεται με την αρχικοποίηση του Ca και η διαδικασία γίνεται μόναχα την πρώτη φορά που ξεκινάμε την αρχή πιστοποίησης.

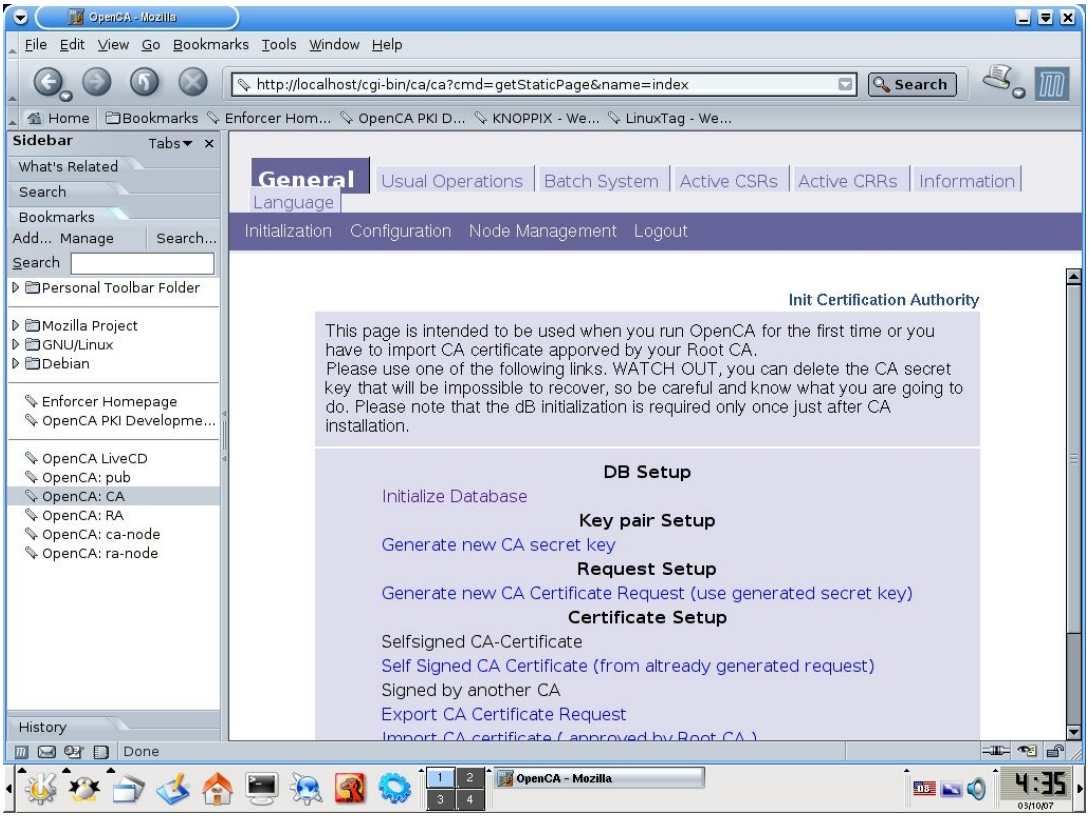
3. Το πρώτο βήμα είναι να αρχικοποιήσουμε την βάση δεδομένων η οποία είναι κοινή για τον CA και τον RA. Επιλέγουμε **Initialize Database**, θα μας αναφερθεί επιτυχής αρχικοποίηση.



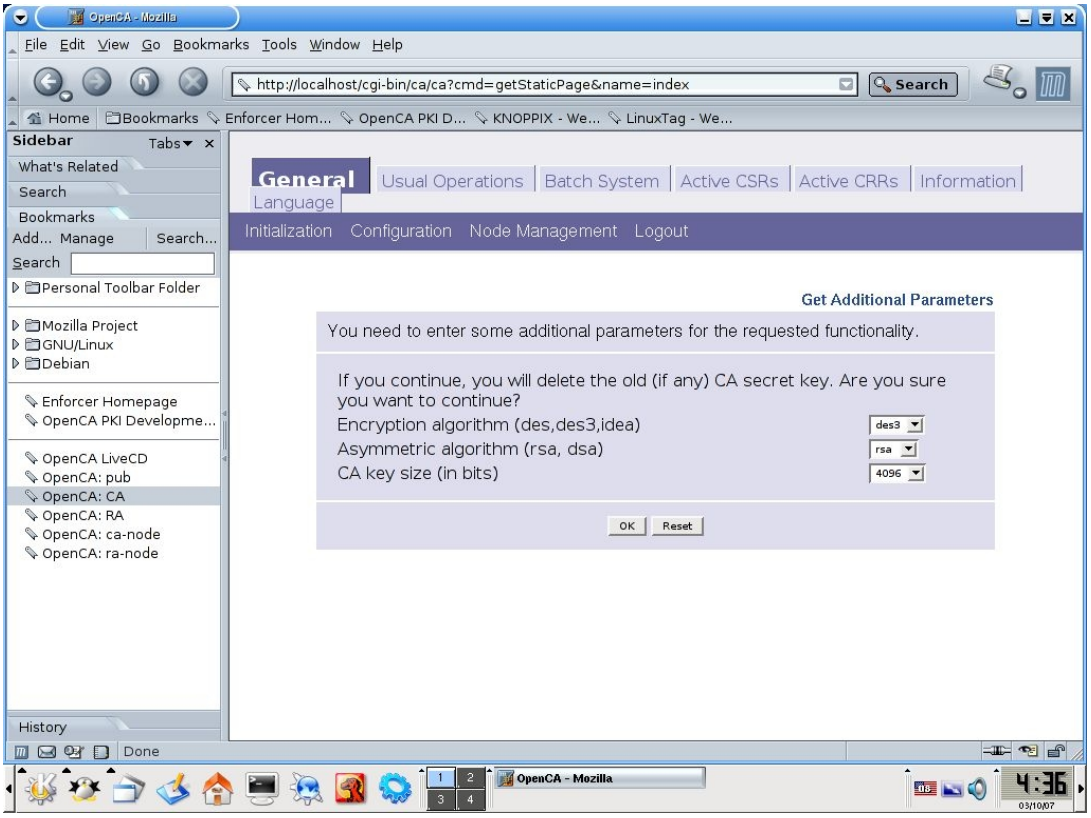
Θα πρέπει να πάμε πίσω στην προηγούμενη οθόνη, για να γίνει αυτό επιλέγουμε **Initialization** απο την καρτέλα **General** και γυρνάμε πίσω



4. Επιλέγουμε ξανά "**Initialize the Certificate Authority**"το επόμενο βήμα είναι να δημιουργήσουμε το μυστικό κλειδί για τον CA, επιλέγουμε "**Generate new CA secret key"** κάτω από το "**Key pair Setup**".

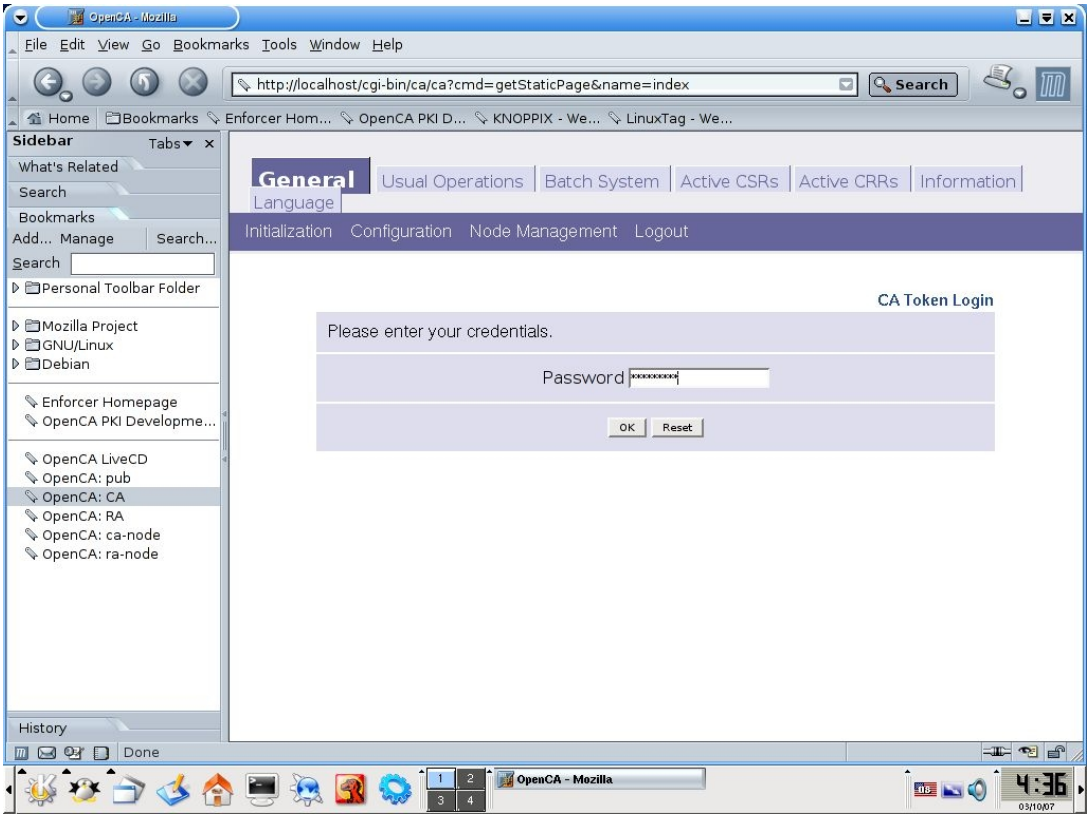


Μας εμφανίζεται η οθόνη " **Get The Additional Parameters** "



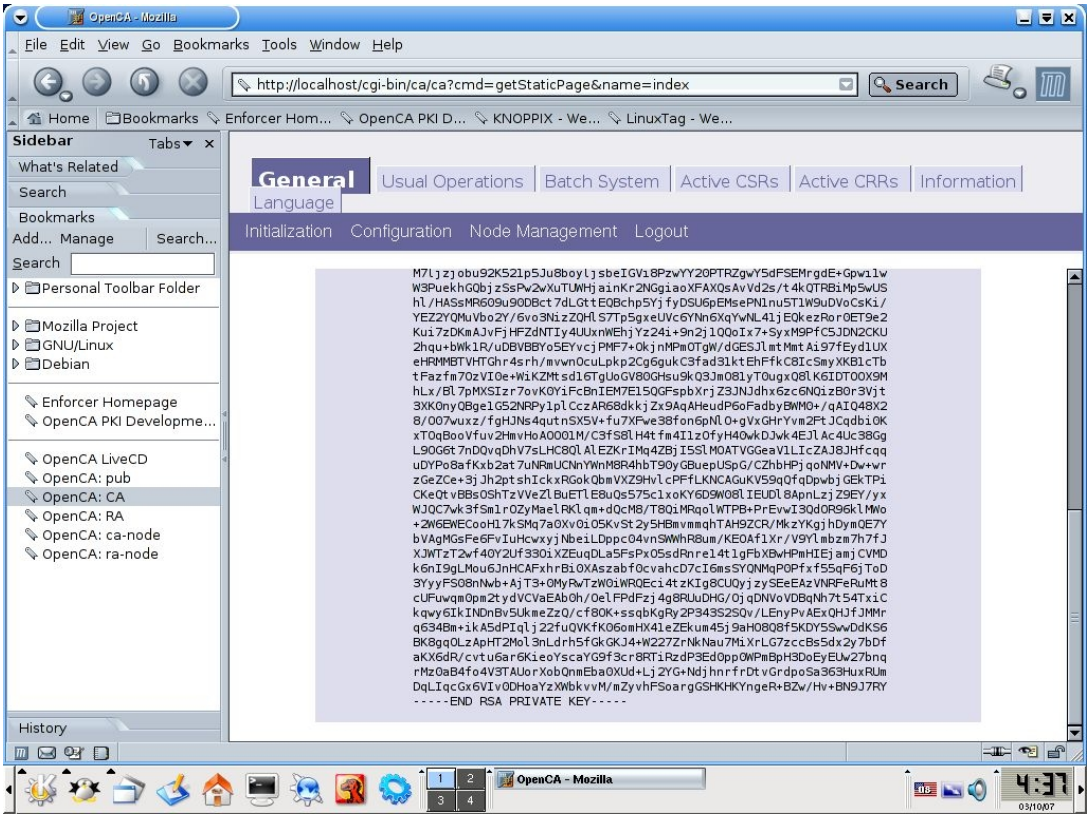
Οι προεπιλεγμένες τιμές **είναι Encryption algorithm (des,des3,idea):**des3 **Asymmetric algorithm (rsa, dsa):**rsa, **CA key size (in bits):**4096. Τις αφήνουμε ως έχουν και επιλέγουμε "OK".

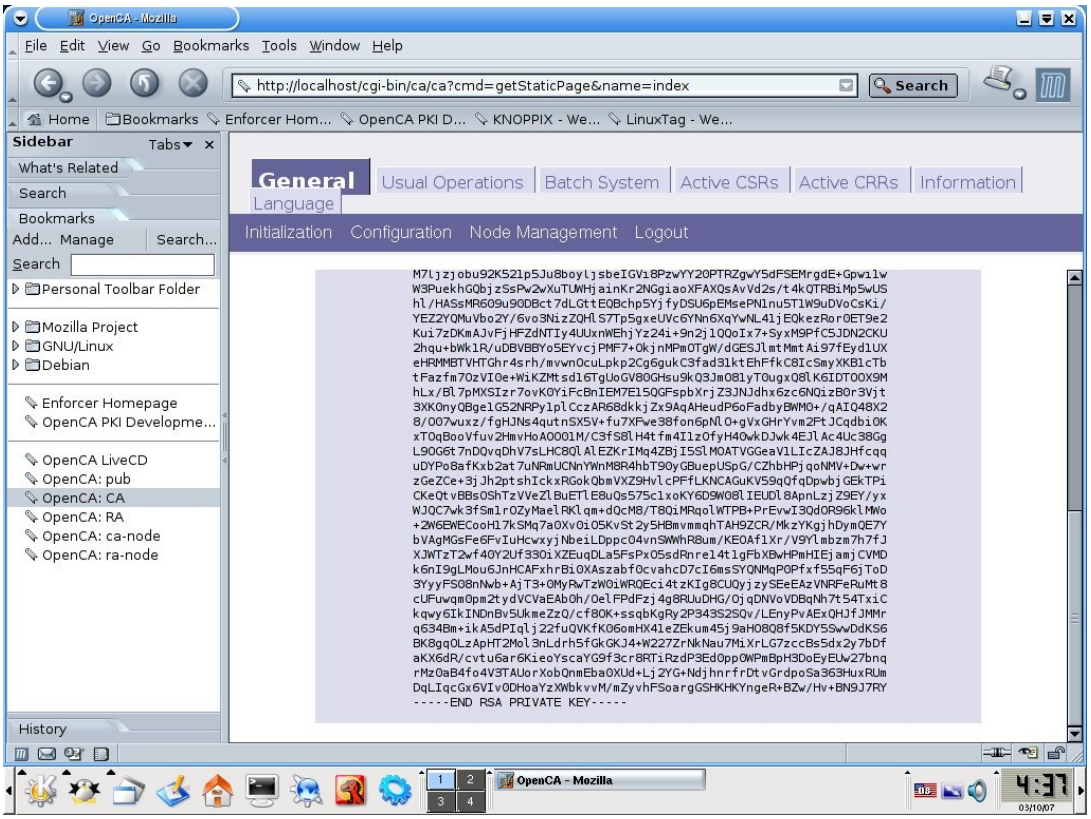
5. Δίνουμε ένα Password για το CA certificate Key στην σελίδα "**Ca Token Login Page**"



Το password θα προστατεύει το ιδιωτικό κλειδί και θα χρειαστεί για την λειτουργία του CA. Πρέπει να προσέξουμε σε αυτό το σημείο όταν πληκτρολογούμε το password διότι δεν υπάρχει επιβεβαίωση και θα πρέπει να θυμόμαστε τι έχουμε δώσουμε διότι θα το χρησιμοποιήσουμε αρκετές φορές στην συνέχεια.

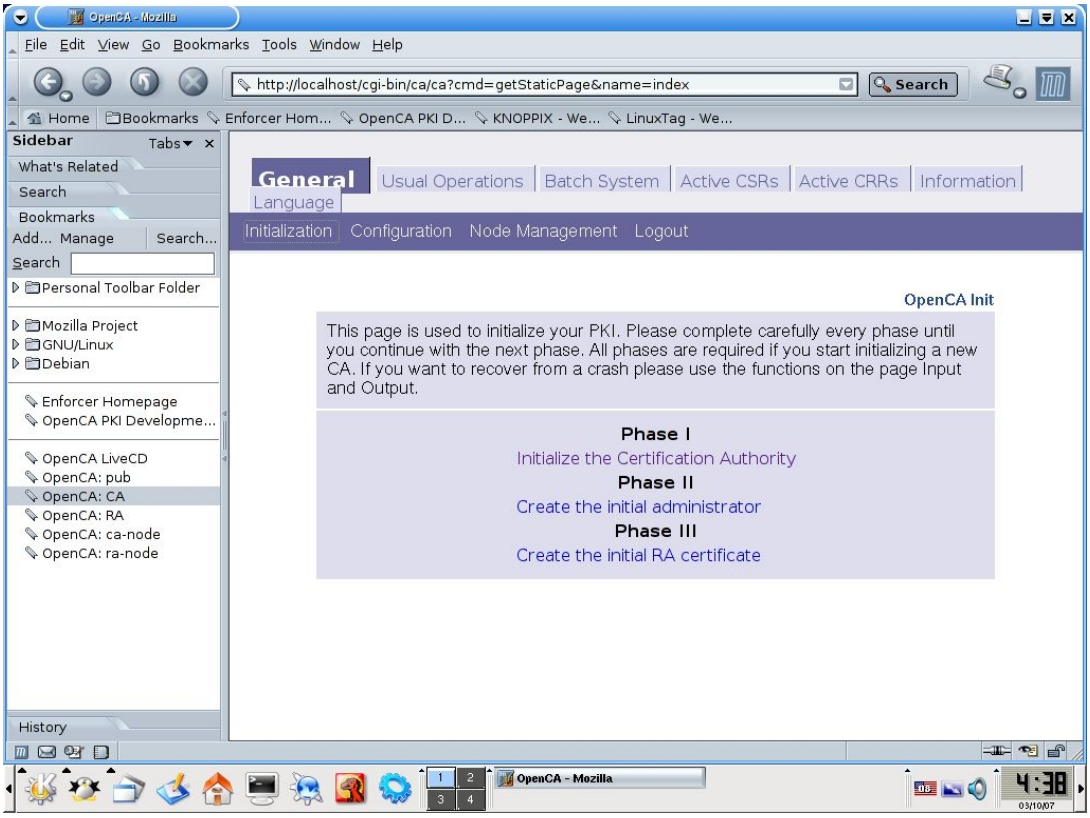
Αφού δώσουμε το password επιλέγουμε "Ok" ο Server θα δημιουργήσει ένα ζευγάρι κλειδιών βασισμένο στις παραμέτρους που του έχουμε δώσει, όταν το κλειδί δημιουργηθεί θα μας εμφανισθεί η παρακάτω οθόνη επιβεβαίωσης



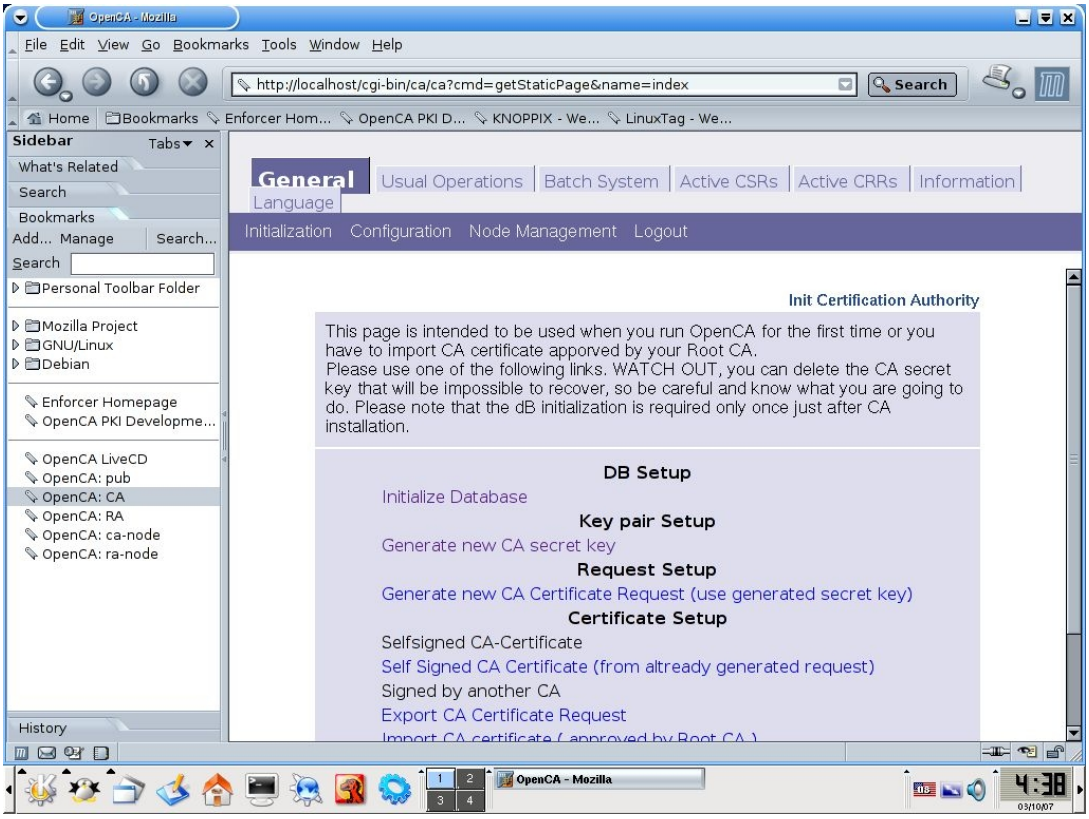


Με αυτό το βήμα ολοκληρώνεται η φάση της δημιουργίας του μυστικού κλειδίου. Το επόμενο βήμα είναι να δημιουργήσουμε ένα νέο αίτημα που σχετίζεται με το

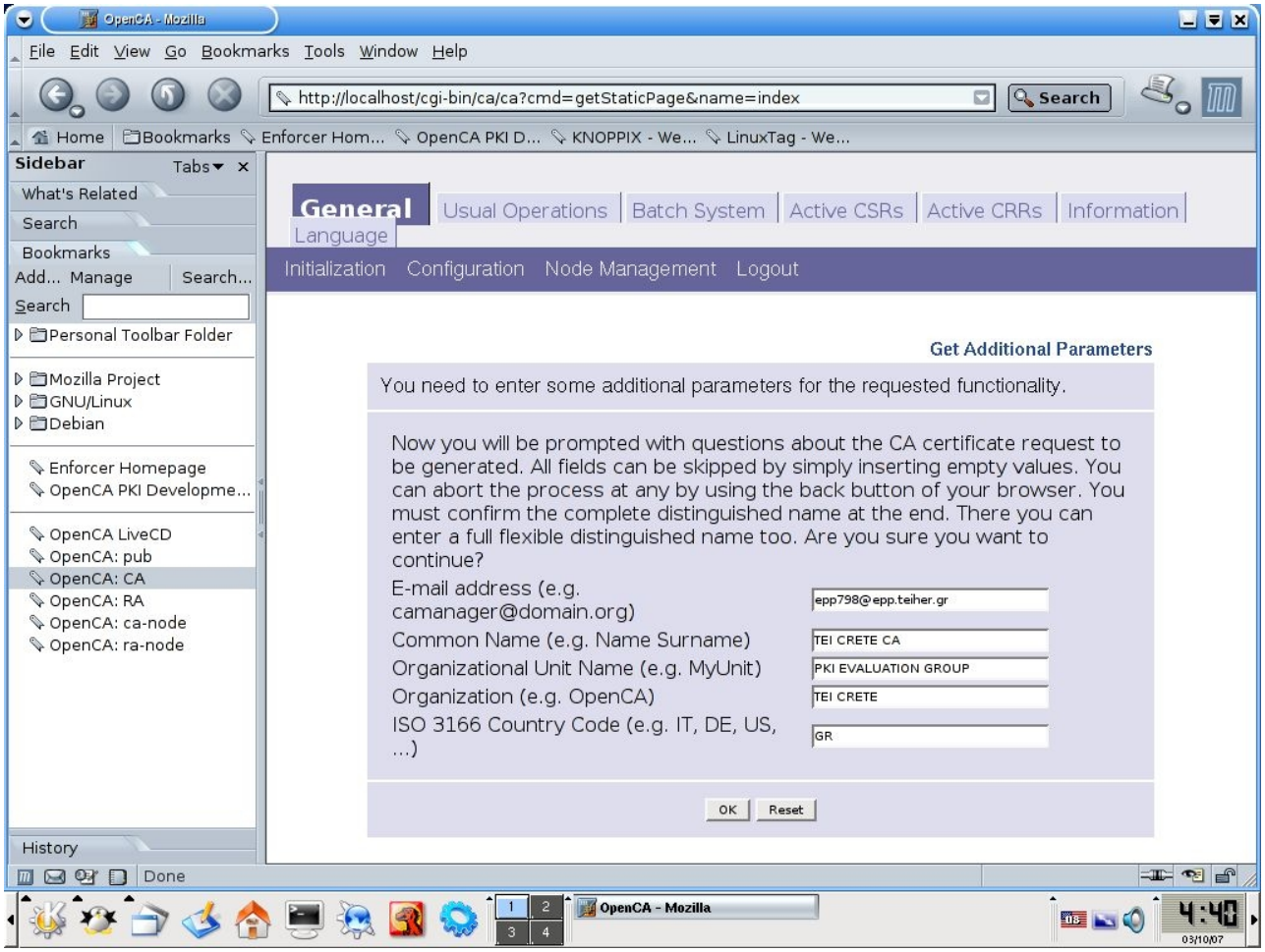
πιστοποιητικό του πάροχο μας (root), επιλέγουμε **Initialization** και επιστρέφουμε πίσω στην Φάση I.



Επιλέγουμε "**Initialize the Certification Authority**".



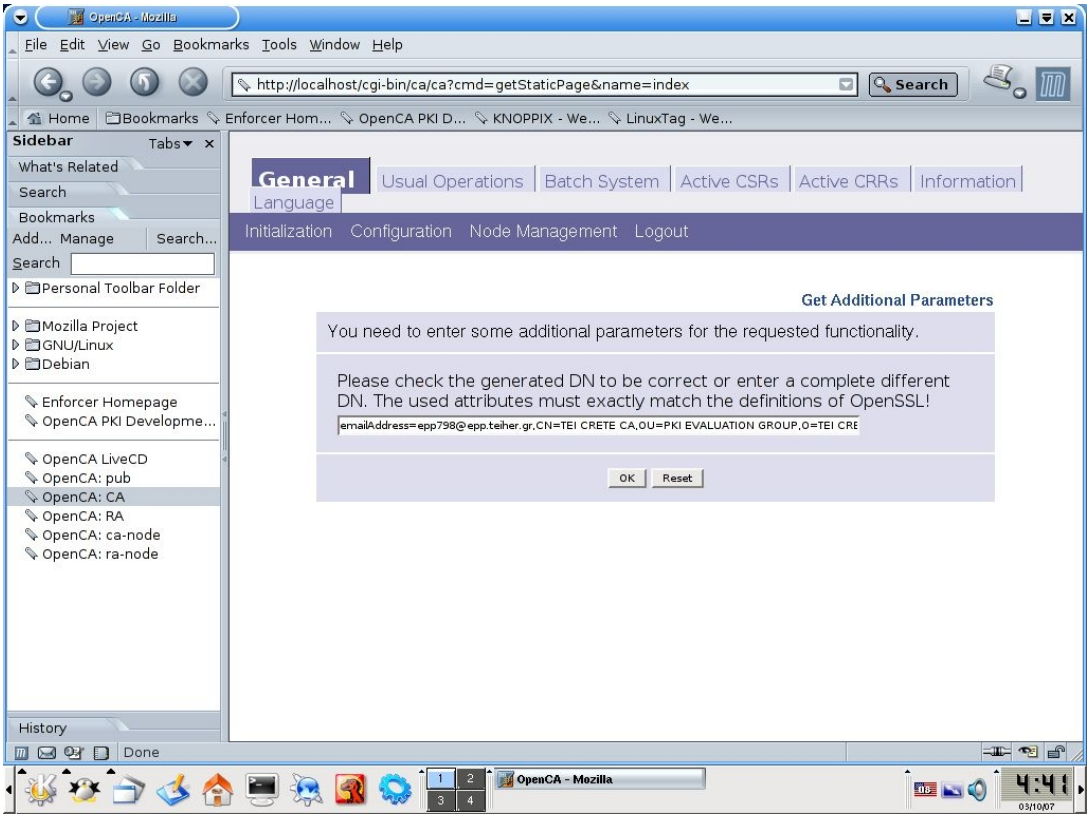
6. Επιλέγουμε “**Generate new CA Certificate Request (use generated secrete key)**



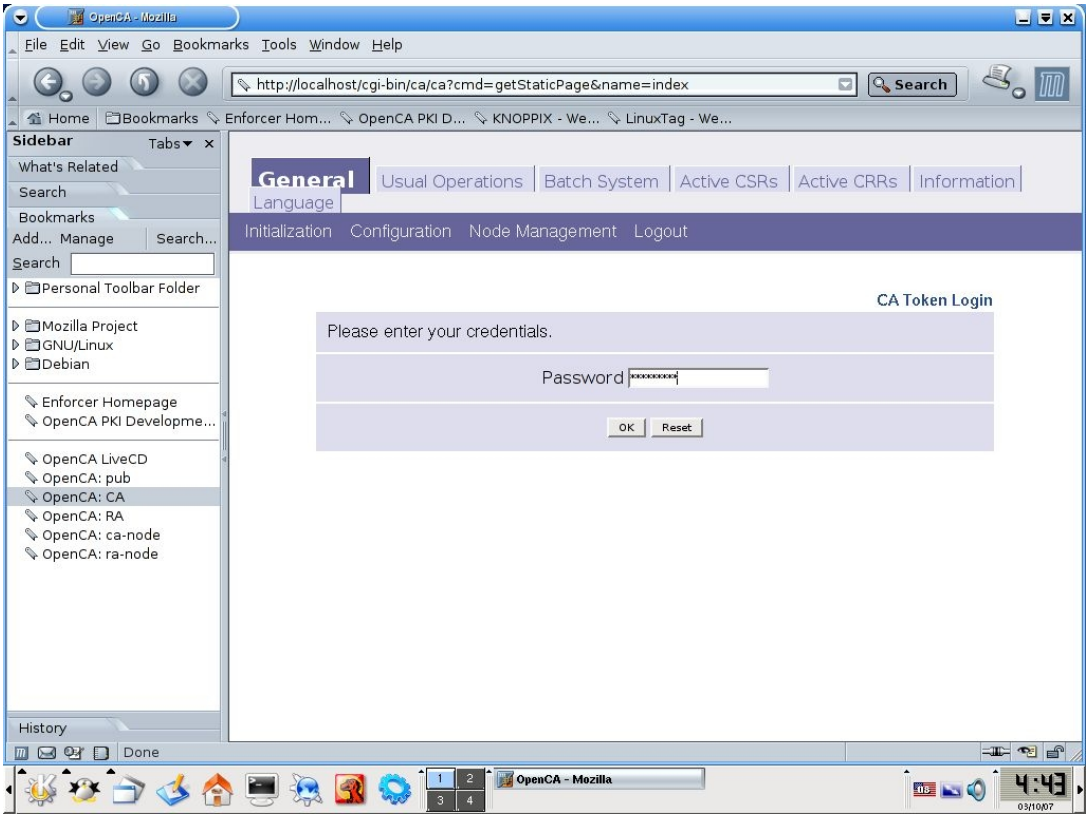
Σε αυτή την οθόνη συμπληρώνουμε τις βασικές πληροφορίες που έχουν να κάνουν με τον παρόχο μας (root CA). Αν και το Common Name καθώς και το email δεν σχετίζονται με καμία λειτουργία του CA ή του RA, παρόλα αυτά αυτές οι πληροφορίες εμφανίζονται σε κάποιες φόρμες παρακάτω, οπότε θα πρέπει να είναι κατανοητές.

Το όνομα του οργανισμού θα πρέπει να είναι το ίδιο με αυτο που είχαμε δώσει στην αρχή όταν το σύστημα έκανε boot.

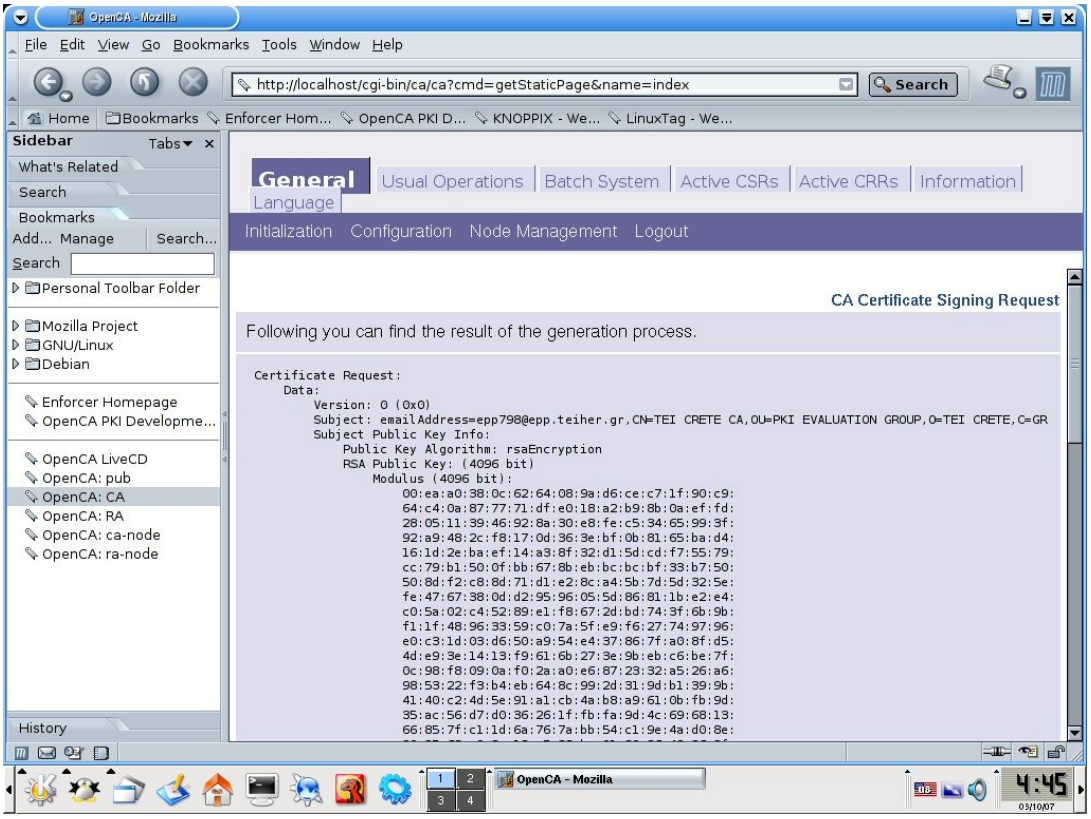
Είναι καλύτερα να μην χρησιμοποιήσουμε χαρακτήρες όπως “\” ή “,”. Το OpenCa θα συνδέσει τον οργανισμό που είχαμε δώσει όταν το σύστημα έκανε boot με τον οργανισμό που δίνουμε τώρα οπότε θα πρέπει να είναι ίδια. Επιλέγοντας “Ok” εμφανίζεται η παρακάτω οθόνη επιβεβαίωσης η οποία μας δείχνει πως ο DN δημιουργήθηκε με βάση τα στοιχεία που του δώσαμε.

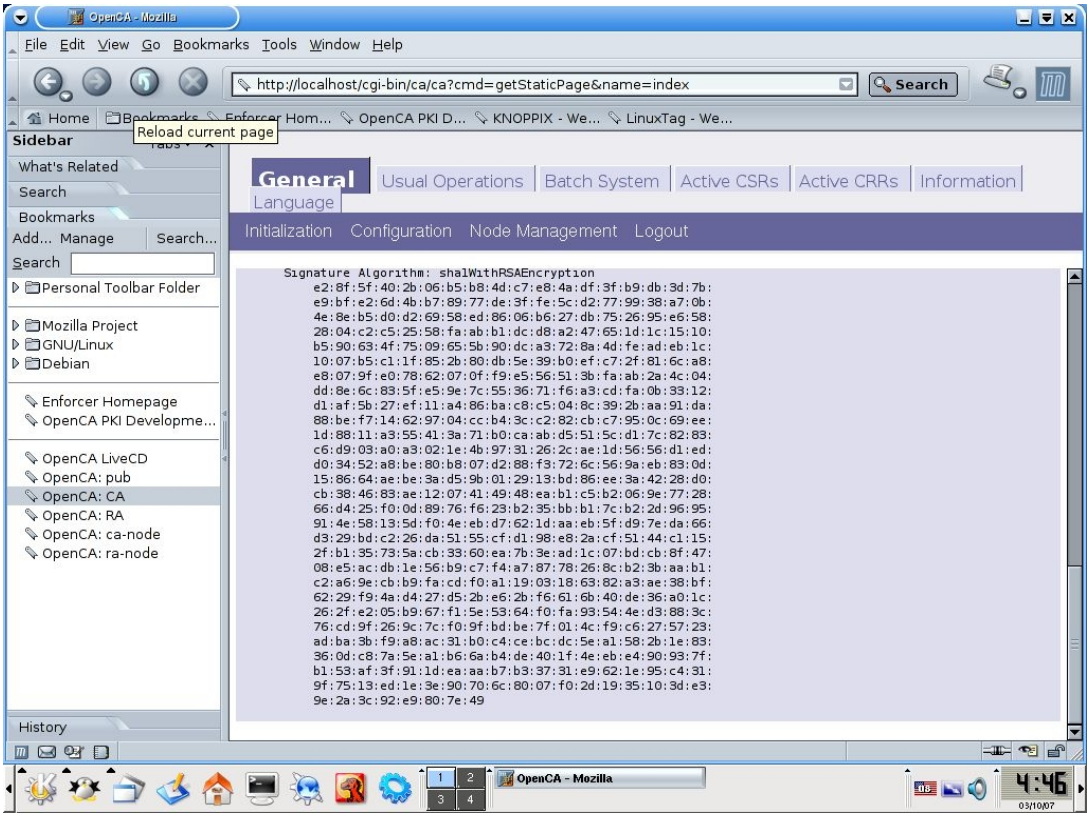


Αν είμαστε ευχαριστημένοι με τα δεδομένα που έχει ο DN επιλέγουμε "Ok" και συνεχίζουμε, θα μας ζητηθεί το password για το μυστικό κλείδι του CA.

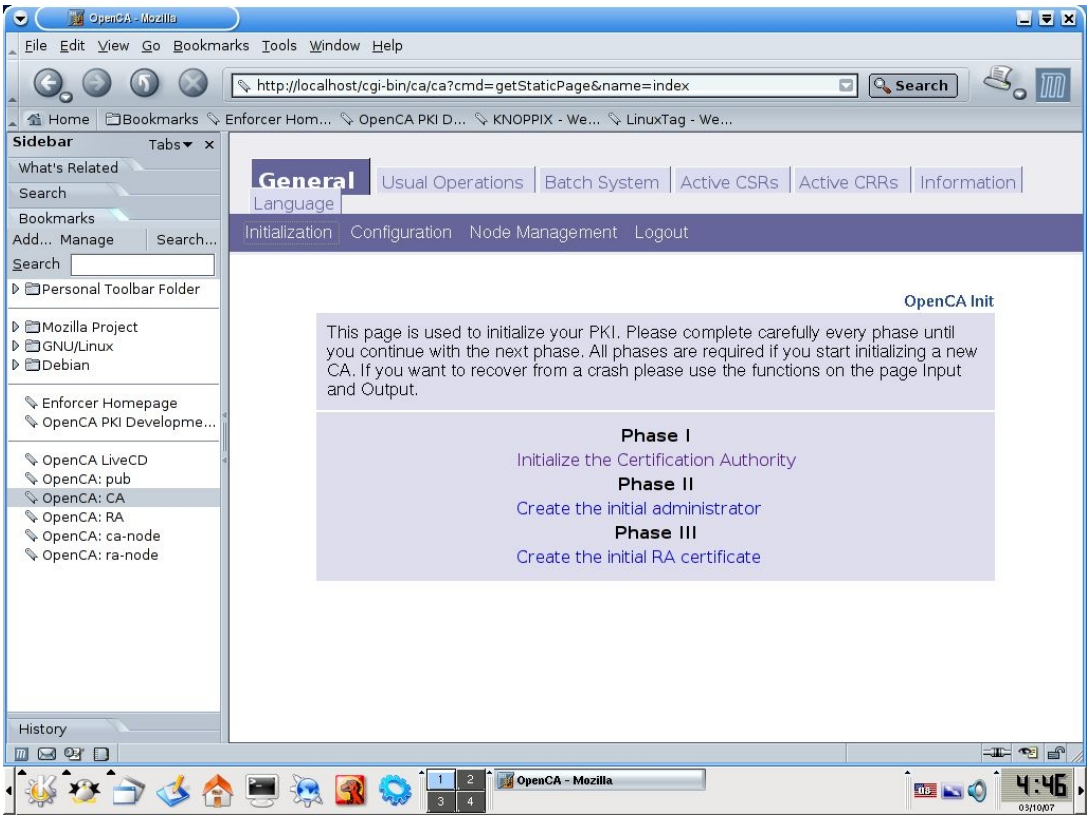


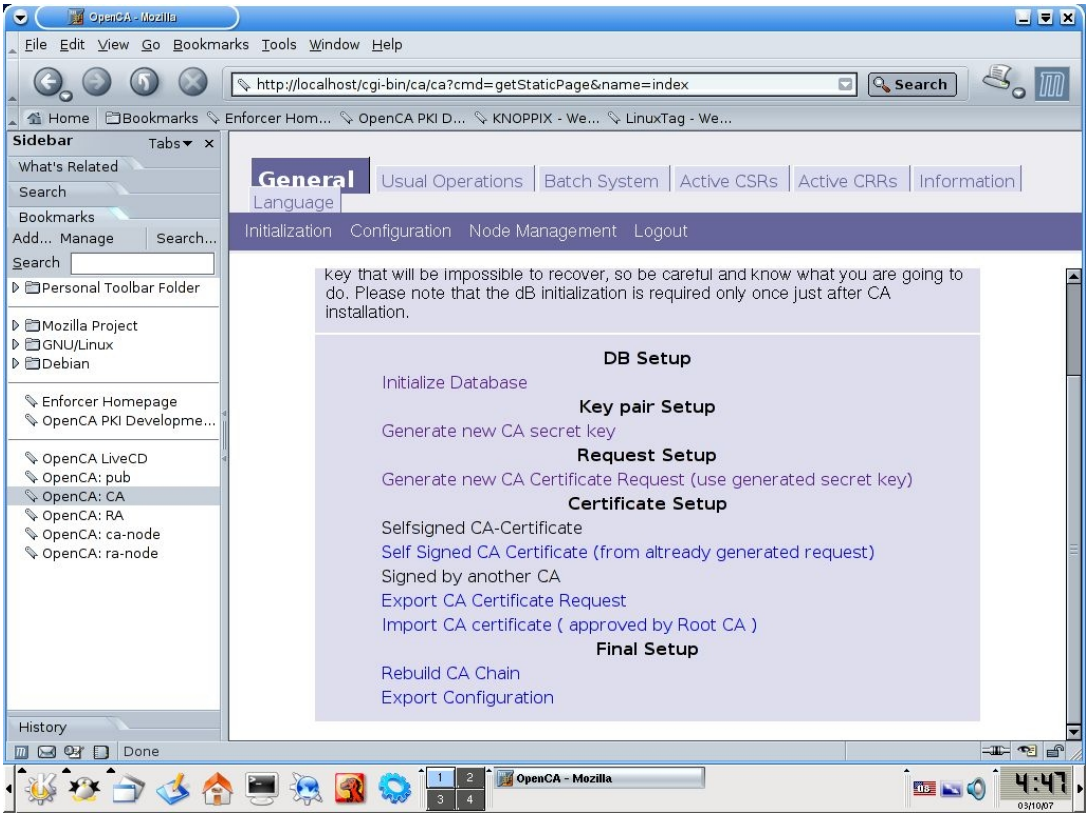
Δίνουμε το password του κλειδιού που δημιουργήσαμε στο βήμα 5 και μας εμφανίζεται μια οθόνη επιβεβαίωσης η οποία δείχνει λεπτομέρειες σχετικά με την αίτηση που κάναμε για το πιστοποιητικό που μόλις δημιουργήθηκε.





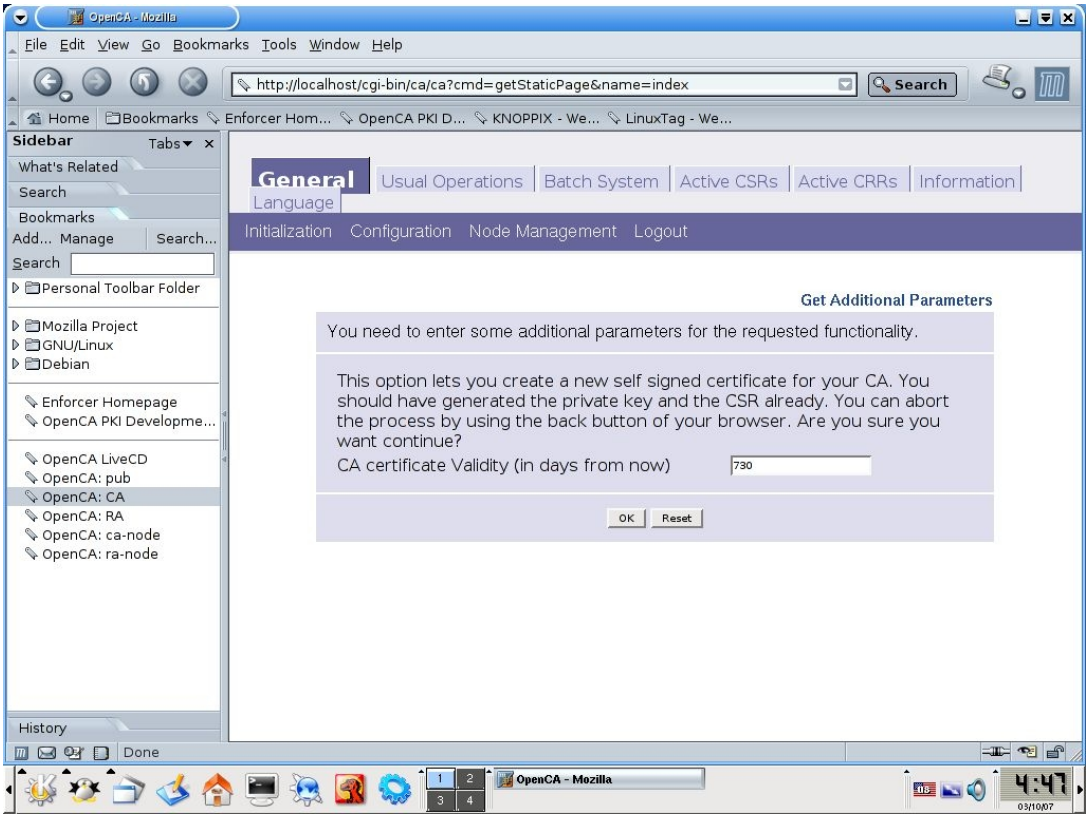
Σε αυτό το σημείο θα πρέπει να ασχοληθούμε λίγο με την αίτηση για το πιστοποιητικό που μόλις δημιουργήσαμε για τον αρχικό χρήστη “root” της αρχής πιστοποίησης. Επιλέγουμε **Initialization** και γυρνάμε στην Φάση I, επιλέγουμε “**Initialize the Certificate Authority**”



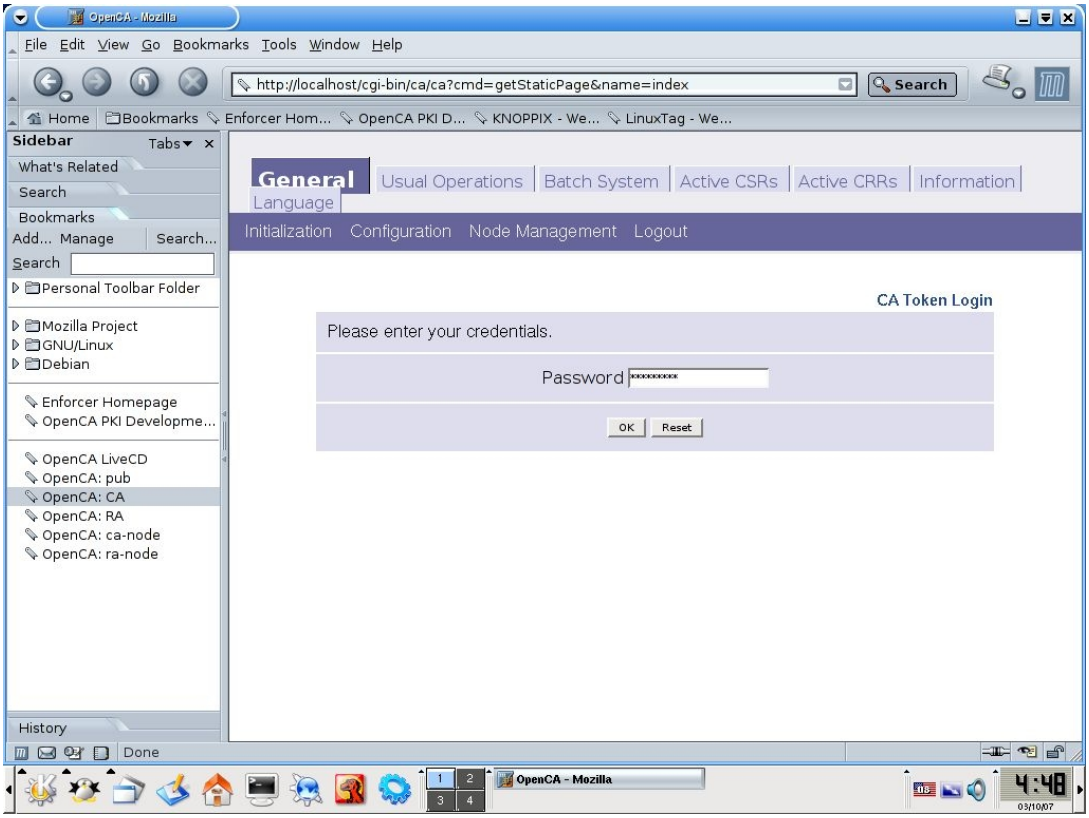


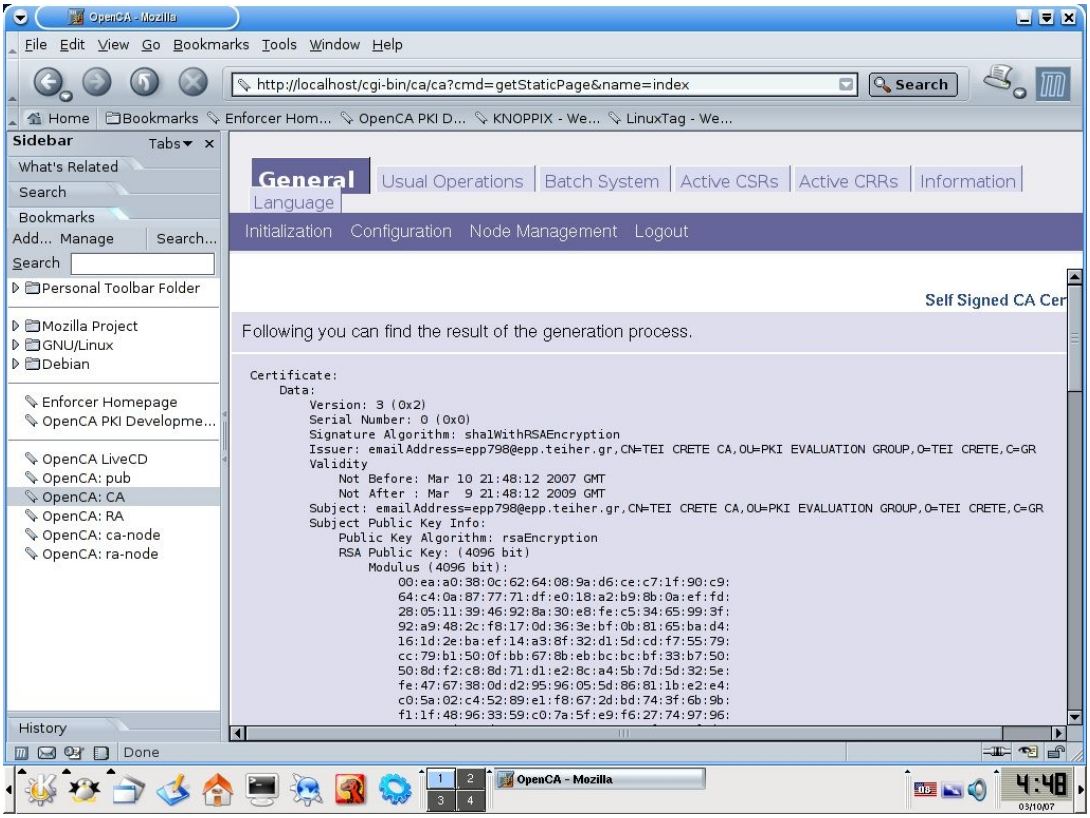
7. Στην παραπάνω οθόνη επιλέγουμε "**Self Signed CA Certificate (from already generated request)**". Αυτό σημαίνει ότι θα υπογράψουμε το πιστοποιητικό για την δική μας αρχή πιστοποίησης μόνοι μας, δεν θα προμηθευτούμε δηλαδή ένα επίσημο πιστοποιητικό απο κάποια άλλη αρχή όπως η VeriSign ή η Thawte.

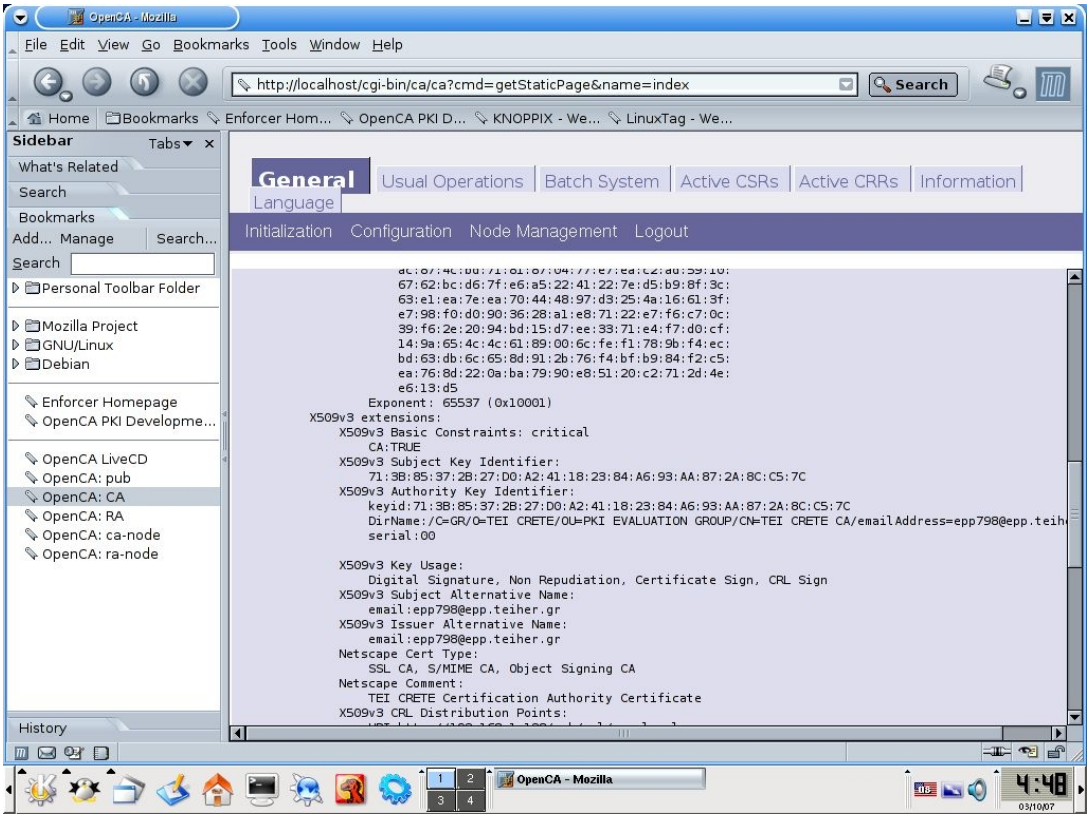
Στη συνέχεια του δίνουμε μια τιμή για το πόσες μέρες θα ισχύει ο CA. Η προεπιλεγμένη περίοδος είναι δύο χρόνια (730 ημέρες) το αφήνουμε ως έχει και επιλέγουμε "Ok".

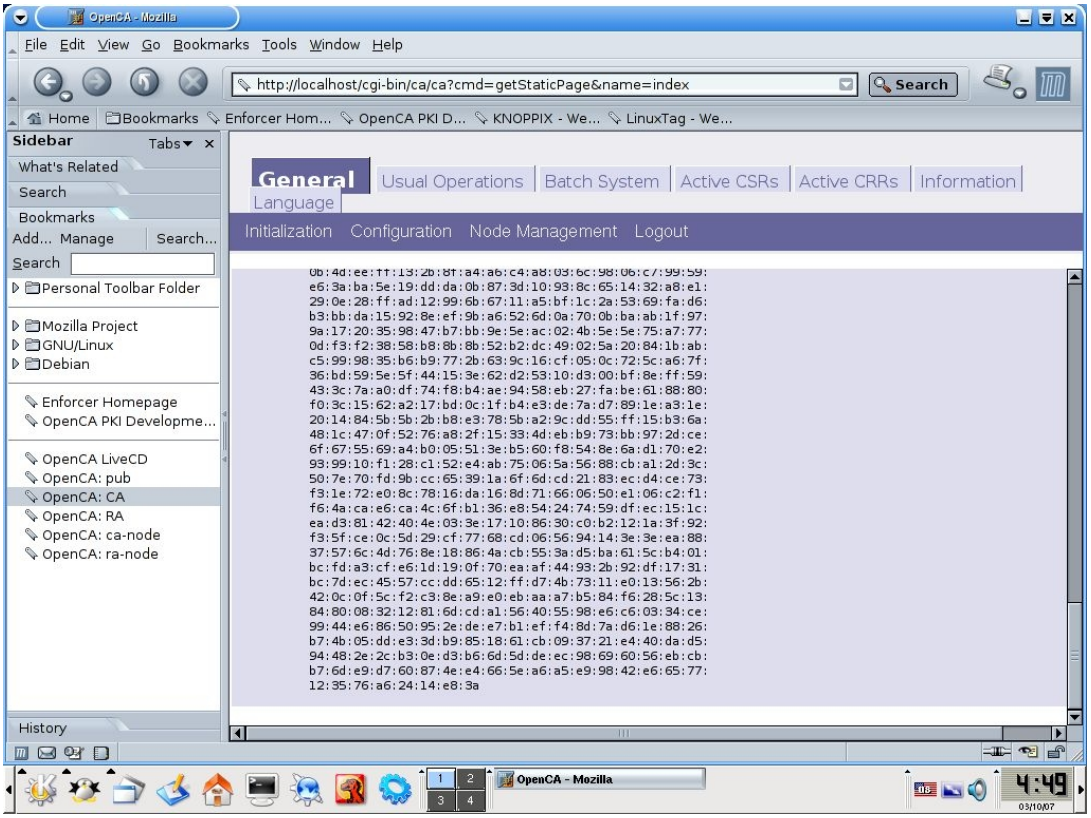


Δίνουμε το password του ιδιωτικού κλειδιού και το πιστοποιητικό δημιουργείται , όπως φαίνεται στις παρακάτω οθόνες.

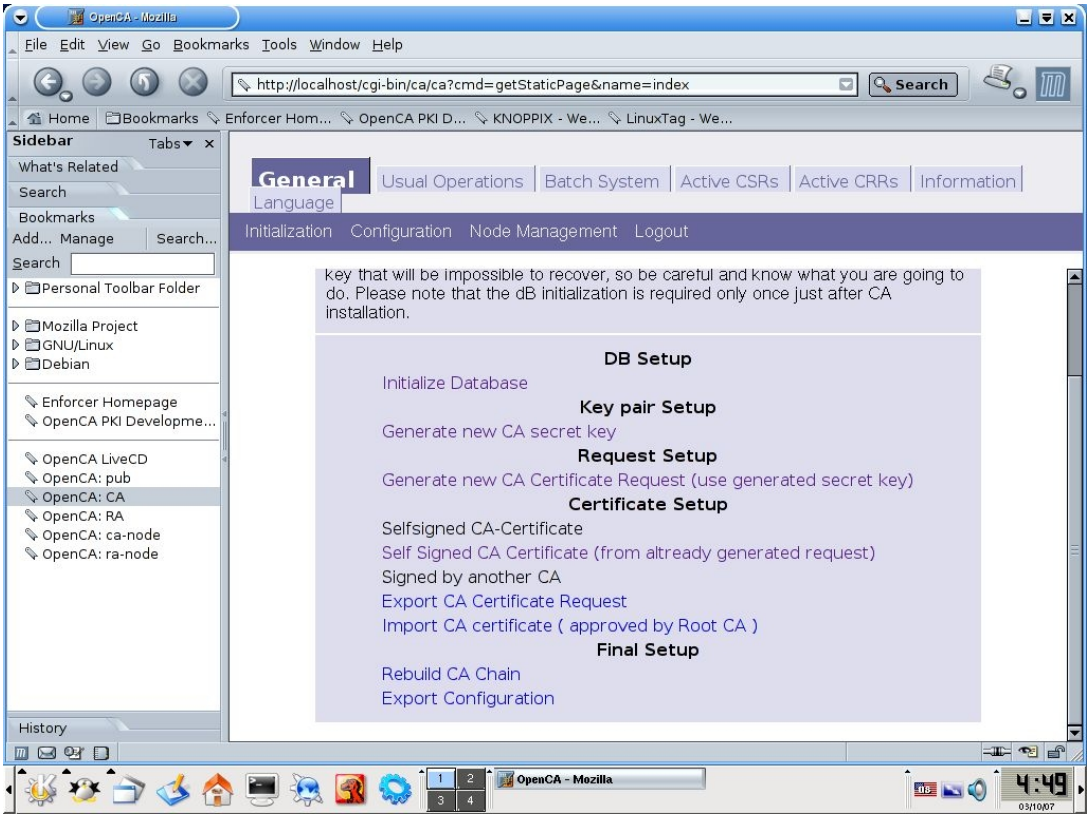

OpenCA - Mozilla
File Edit View Go Bookmarks Tools Window Help
http://localhost/cgi-bin/ca/ca?cmd=getStaticPage&name=index
Search
Home Bookmarks Enforcer Hom... OpenCA PKI D... KNOPPIX - We... LinuxTag - We...
Sidebar Tabs
What's Related
Search
Bookmarks
Add... Manage Search...
Search
Personal Toolbar Folder
Mozilla Project
GNU/Linux
Debian
Enforcer Homepage
OpenCA PKI Developme...
OpenCA LiveCD
OpenCA: pub
OpenCA: CA
OpenCA: RA
OpenCA: ca-node
OpenCA: ra-node
History
General
Usual Operations
Batch System
Active CSRs
Active CRRs
Information
Language
Initialization Configuration Node Management Logout
CA Token Login
Please enter your credentials.
Password
OK Reset
Done
OpenCA - Mozilla
4:48
03/10/07


OpenCA - Mozilla
File Edit View Go Bookmarks Tools Window Help
http://localhost/cgi-bin/ca/ca?cmd=getStaticPage&name=index
Search
Home Bookmarks Enforcer Hom... OpenCA PKI D... KNOPPIX - We... LinuxTag - We...
Sidebar Tabs
What's Related
Search
Bookmarks
Add... Manage Search...
Search
Personal Toolbar Folder
Mozilla Project
GNU/Linux
Debian
Enforcer Homepage
OpenCA PKI Developme...
OpenCA LiveCD
OpenCA: pub
OpenCA: CA
OpenCA: RA
OpenCA: ca-node
OpenCA: ra-node
History
General
Usual Operations
Batch System
Active CSRs
Active CRRs
Information
Language
Initialization Configuration Node Management Logout
Self Signed CA Cer
Following you can find the result of the generation process.
Certificate:
Data:
Version: 3 (0x2)
Serial Number: 0 (0x0)
Signature Algorithm: sha1WithRSAEncryption
Issuer: emailAddress=epp798@epp.teiher.gr,CN=TEI CRETE CA,OU=PKI EVALUATION GROUP,O=TEI CRETE,C=GR
Validity
Not Before: Mar 10 21:48:12 2007 GMT
Not After : Mar 9 21:48:12 2009 GMT
Subject: emailAddress=epp798@epp.teiher.gr,CN=TEI CRETE CA,OU=PKI EVALUATION GROUP,O=TEI CRETE,C=GR
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (4096 bit)
Modulus (4096 bit):
00:ea:a0:38:0c:62:64:08:9a:d6:ce:c7:1f:90:c9:
64:c4:0a:87:77:71:df:e0:18:a2:b9:8b:0a:ef:fd:
28:05:11:39:46:92:8a:30:e8:fe:c5:34:65:99:3f:
92:a9:48:2c:f8:17:0d:36:3e:bf:0b:81:65:ba:d4:
16:1d:2e:ba:ef:14:a3:8f:32:d1:5d:cd:f7:55:79:
cc:79:b1:50:0f:bb:67:8b:eb:bc:bc:bf:33:b7:50:
50:8d:f2:c8:8d:71:d1:e2:8c:a4:5b:7d:5d:32:5e:
fe:47:67:38:0d:d2:95:96:05:5d:86:81:1b:e2:e4:
c0:5a:02:c4:52:89:e1:f8:67:2d:bd:74:3f:6b:9b:
f1:1f:48:96:33:59:c0:7a:5f:e9:f6:27:74:97:96:
Done
OpenCA - Mozilla
4:48
03/10/07

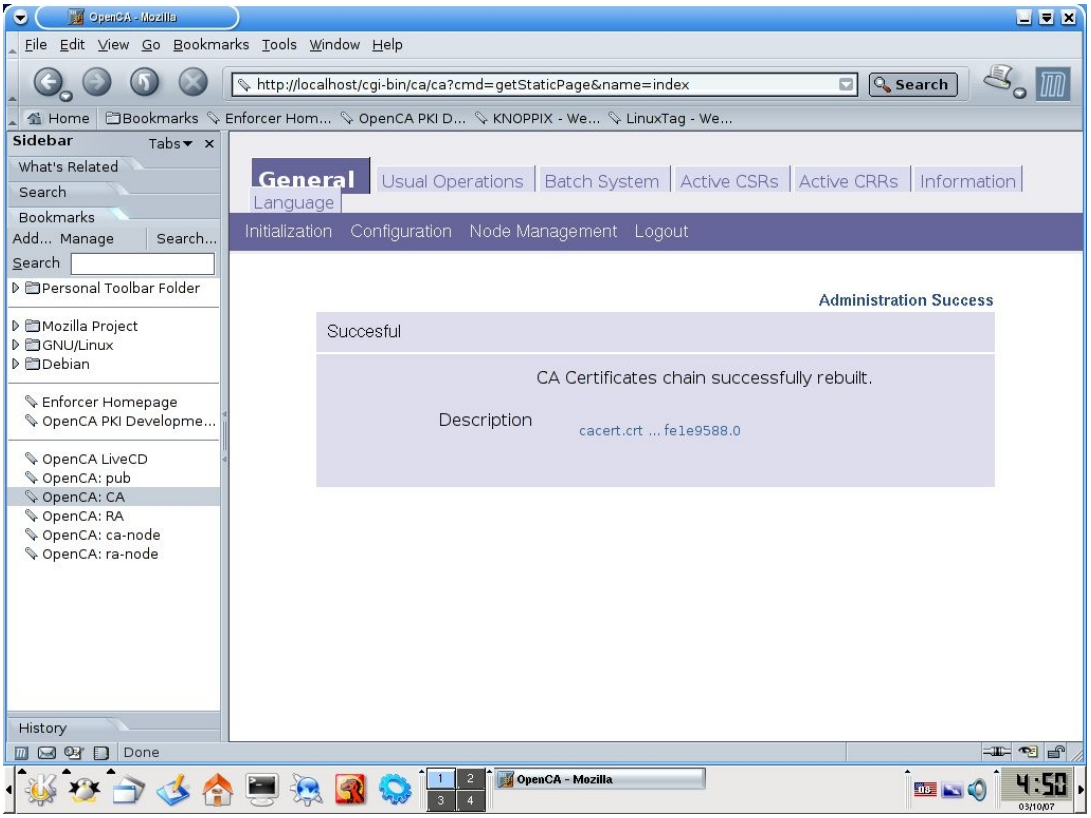




Τέλος γυρνάμε πίσω στην φάση I και πηγαίνουμε στη "**Init Certificate Authority**"

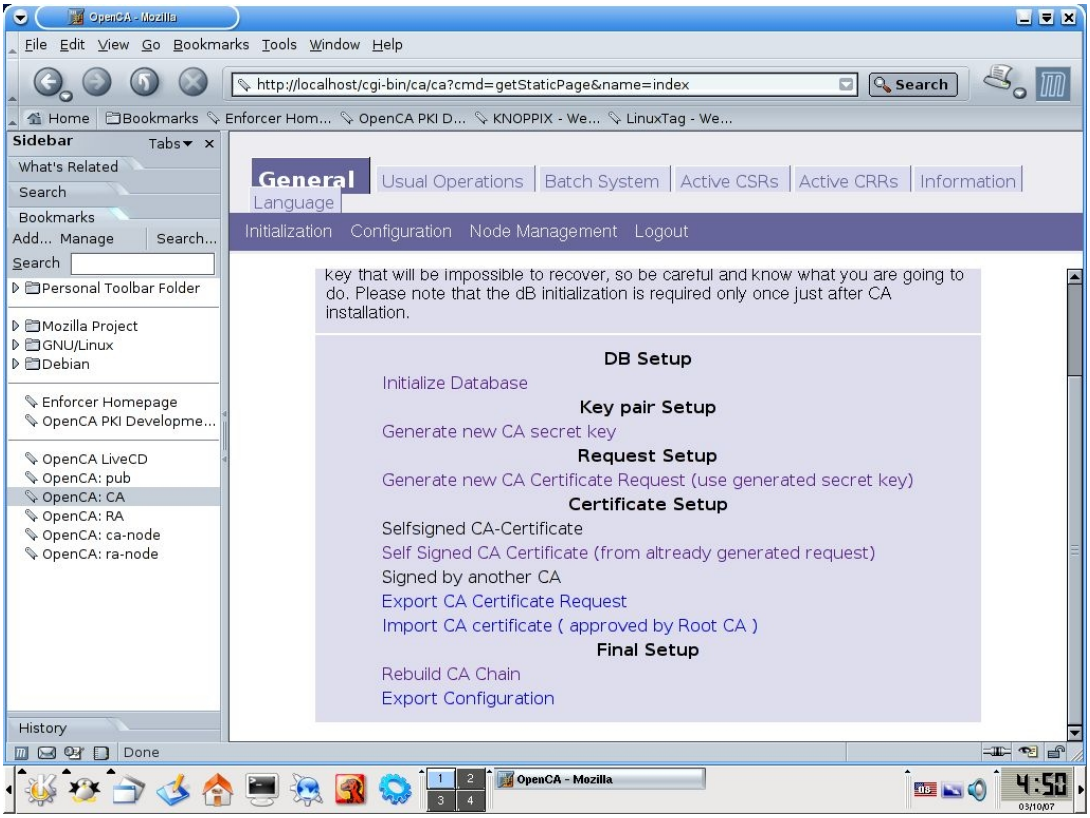


8. Επιλέγουμε "**Rebuilt CA Chain**" και θα πρέπει να πάρουμε μια επιτυχής απάντηση.

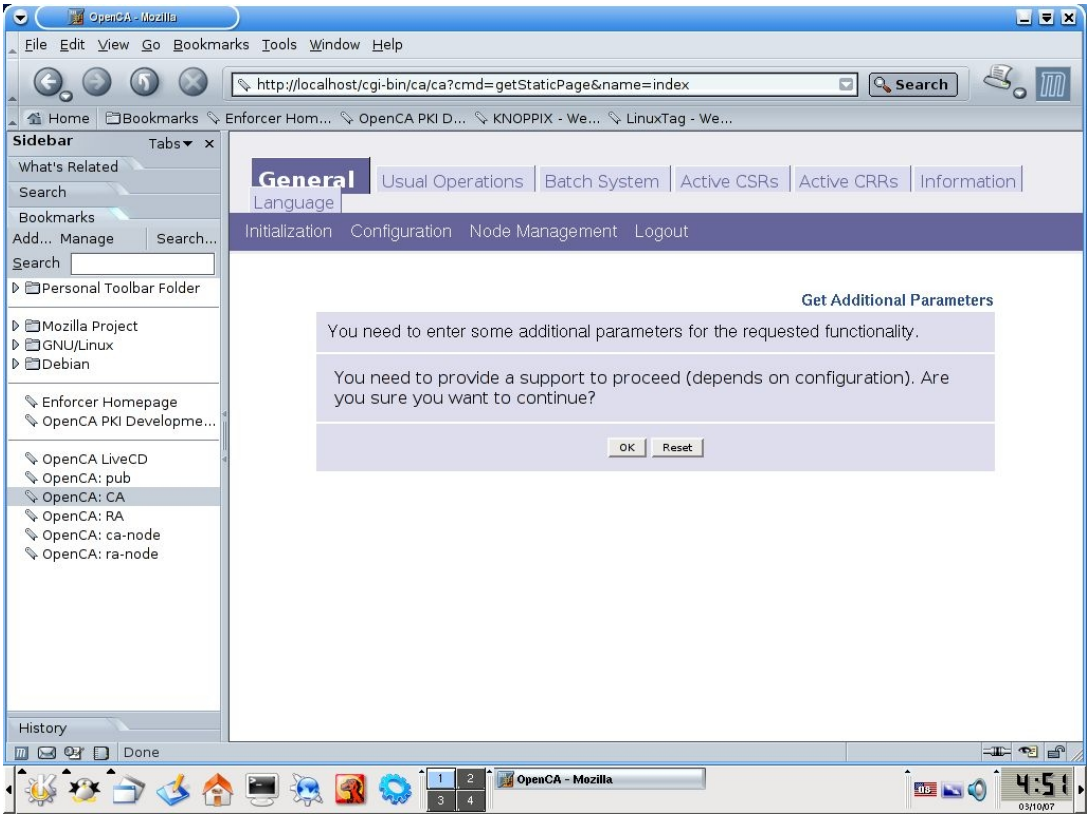


Αυτό το βήμα είναι απαραίτητο για να βεβαιωθούμε ότι τα πιστοποιητικό που θα εκδίδονται απο την αρχή πιστοποίησης θα είναι έγκυρα.

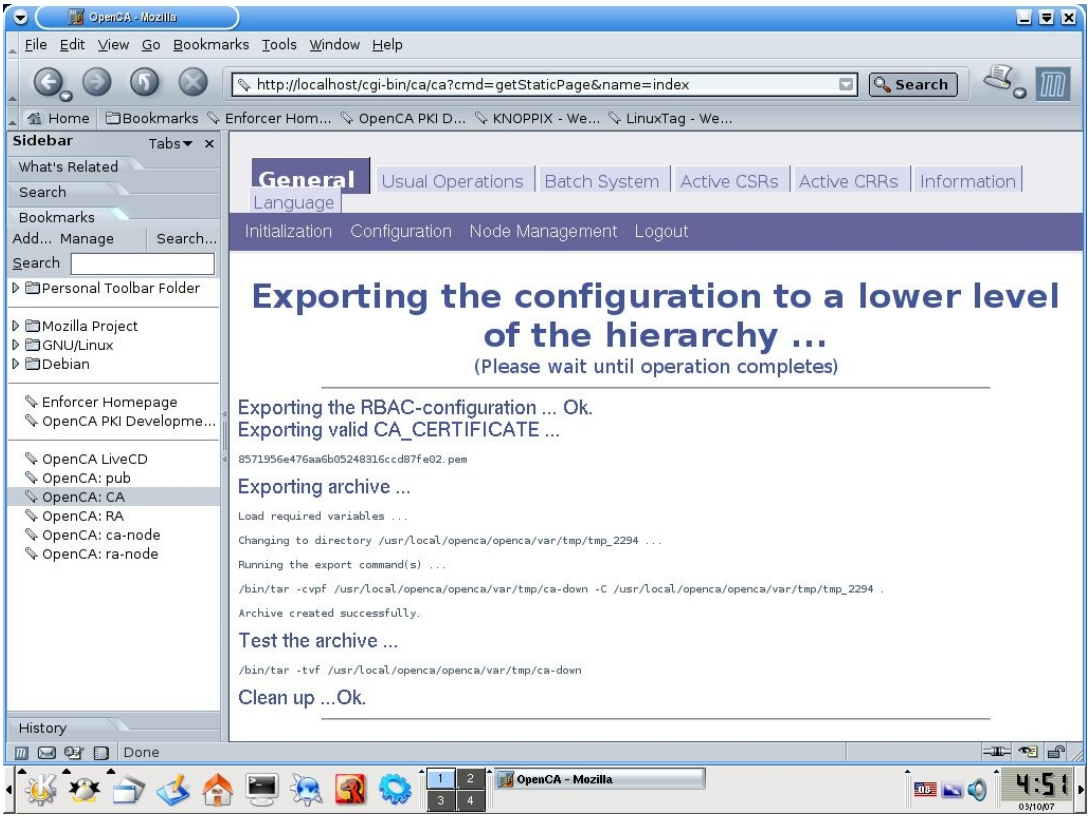
Επιστρέφουμε στο "**Init Certificate Authority**" της φάσης I



9. Επιλέγουμε "**Export Configuration**", κάτω απο την επιλογή "**Final Step**" και βλέπουμε το παρακάτω μήνυμα



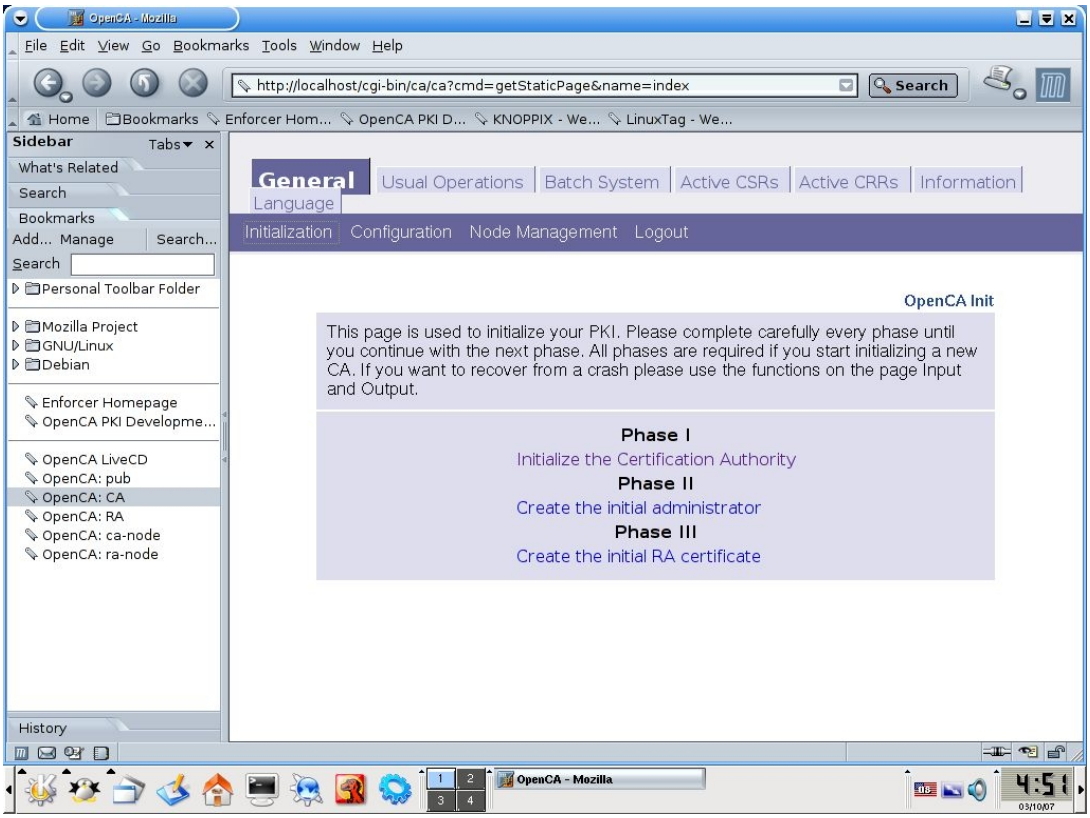
Αυτή η έκδοση του Openca δεν χρειάζεται συμπληρωματική υποστήριξη επιλέγουμε "OK" και παίρνουμε ένα μήνυμα επιτυχής επιβεβαίωσης



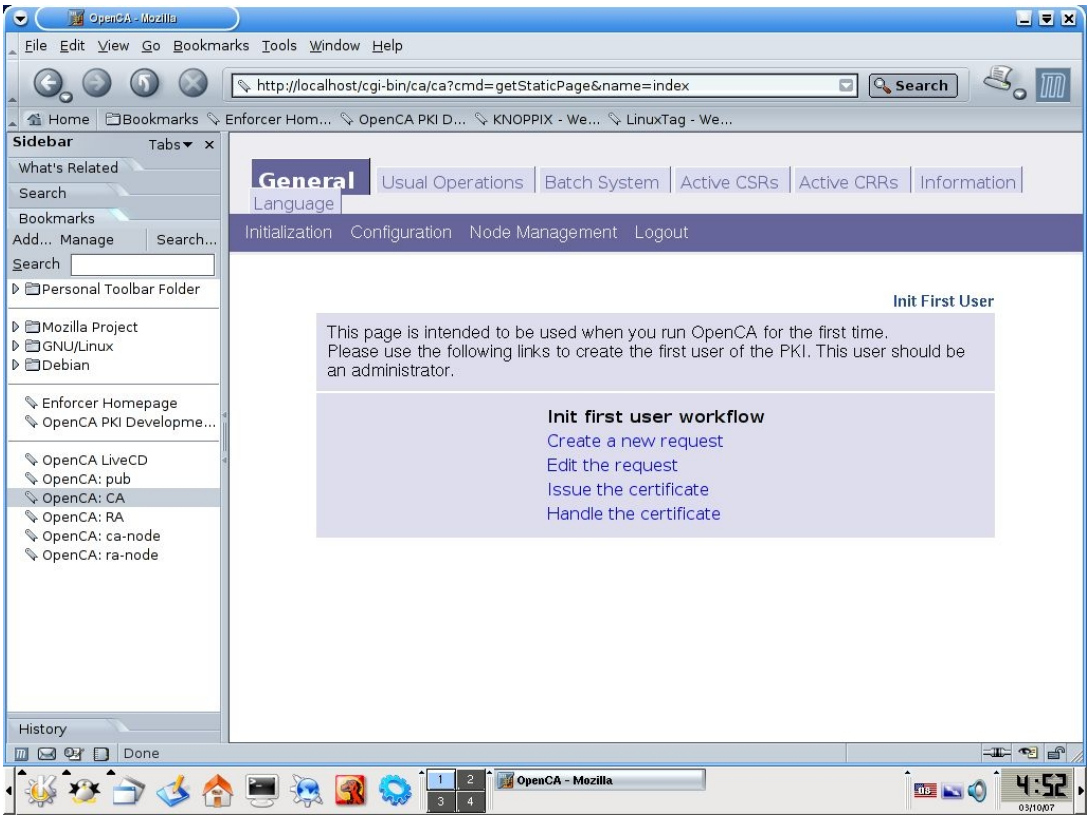
Μόλις τελειώσαμε με την αρχικοποίηση της αρχής πιστοποίησης.

3.1.2 Φαση ΙΙ: Δημιουργία αρχικού διαχειριστή

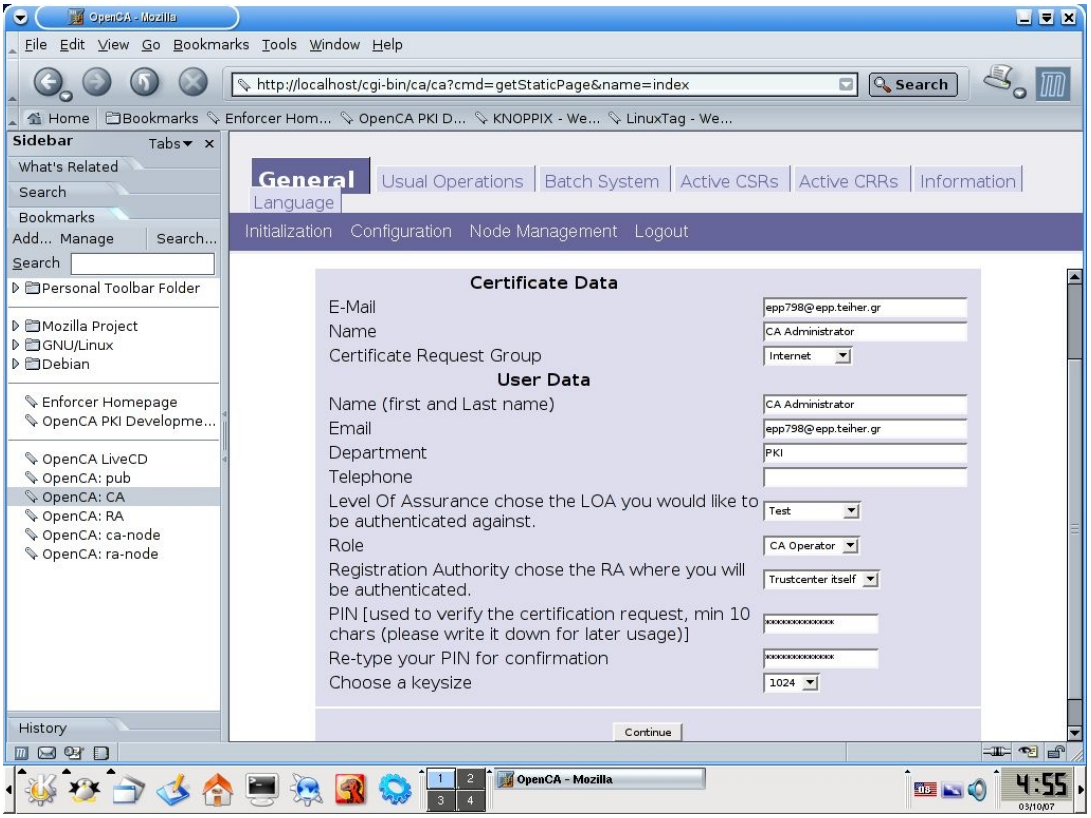
10. Πηγαίνουμε στην καρτέλα **General** και επιλέγουμε **Initialization**, επιλέγουμε την Φάση ΙΙ "**Create the Initial administrator**" για να δημιουργήσουμε ένα πιστοποιητικό για τον βασικό διαχειριστή του OpenCa.



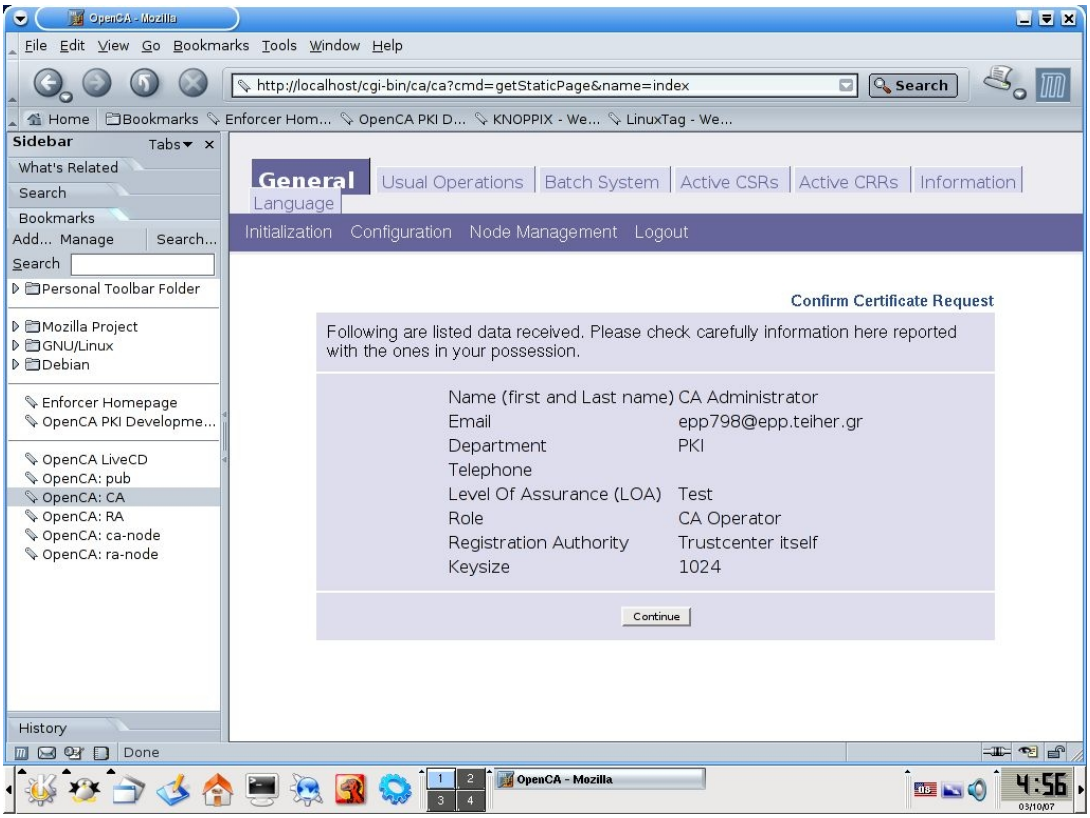
Εμφανίζεται η σελίδα "**Init First User**"



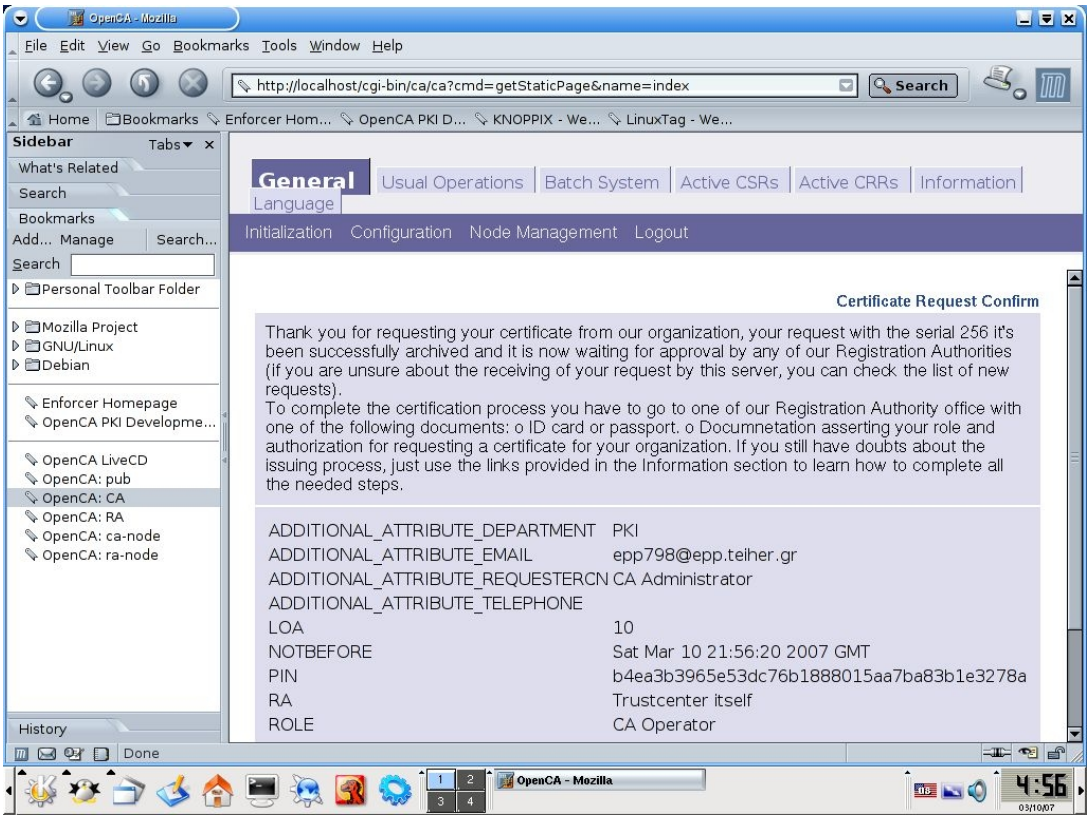
11. Επιλέγουμε "**Create a new request**", συμπληρώνουμε τα πεδία του πιστοποιητικού, ο ρόλος θα πρέπει να είναι "**CA Operator**", ο κωδικός PIN χρησιμοποιείται για την ασφάλεια του ιδιωτικού κλειδιού του πιστοποιητικού .

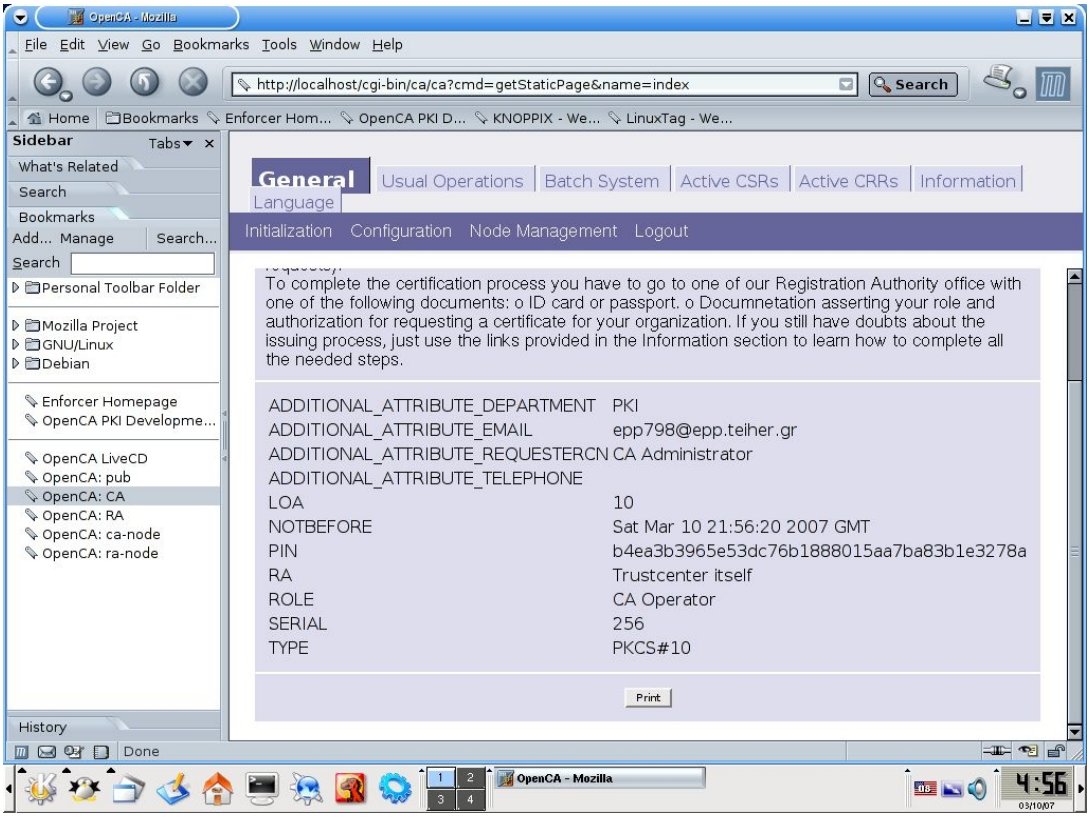


Δεν θα πρέπει να βάλουμε χαρακτήρες όπως "\" ή "," δίοτι είναι πιθανό να ύπαρχει πρόβλημα στην καταχώριση των στοιχείων. Επιλέγουμε μέγεθος κλειδιου 1024 και στη συνέχεια "continue" και εμφανίζεται μια οθόνη επιβεβαίωσης

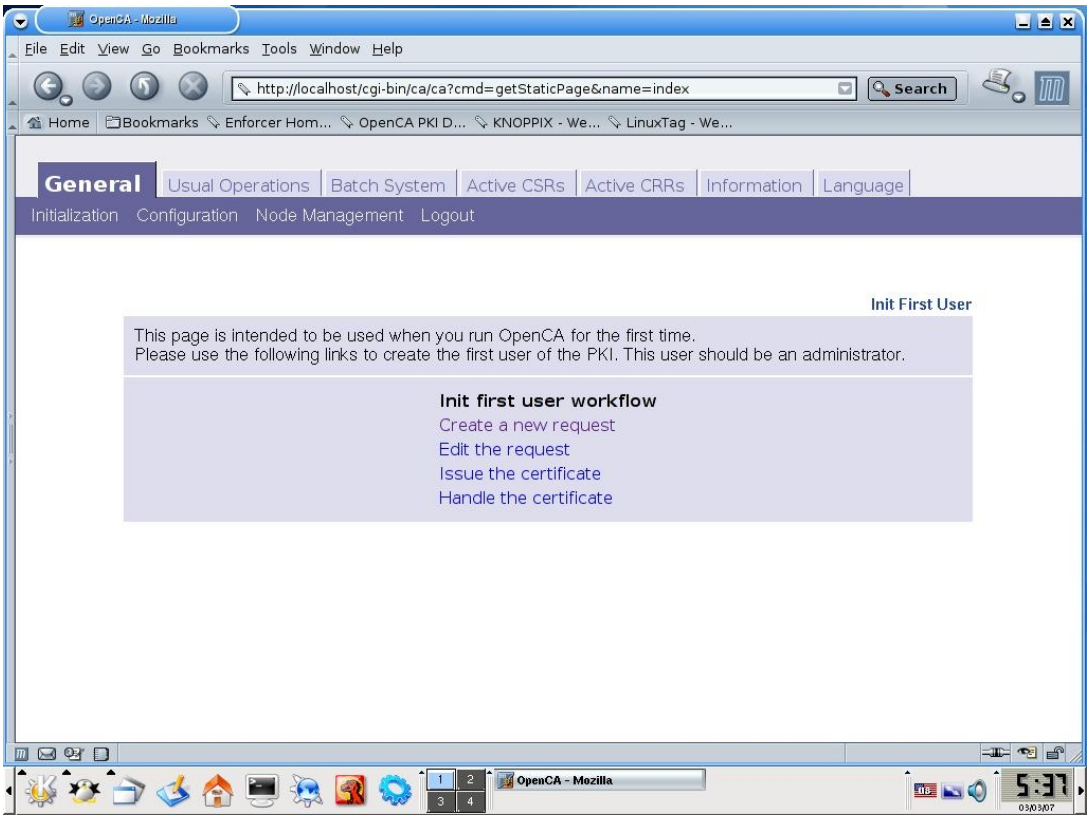


Επιλέγουμε πάλι continue και εμφανίζεται μια δεύτερη οθόνη επιβεβαίωσης

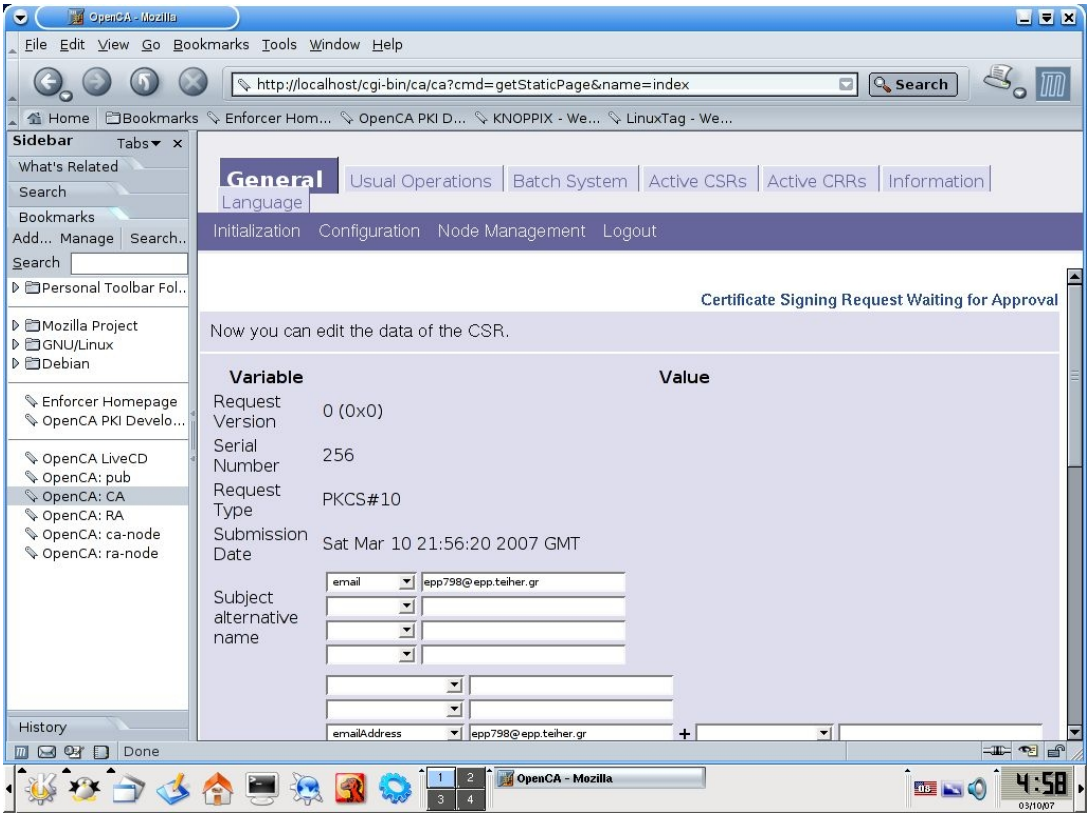


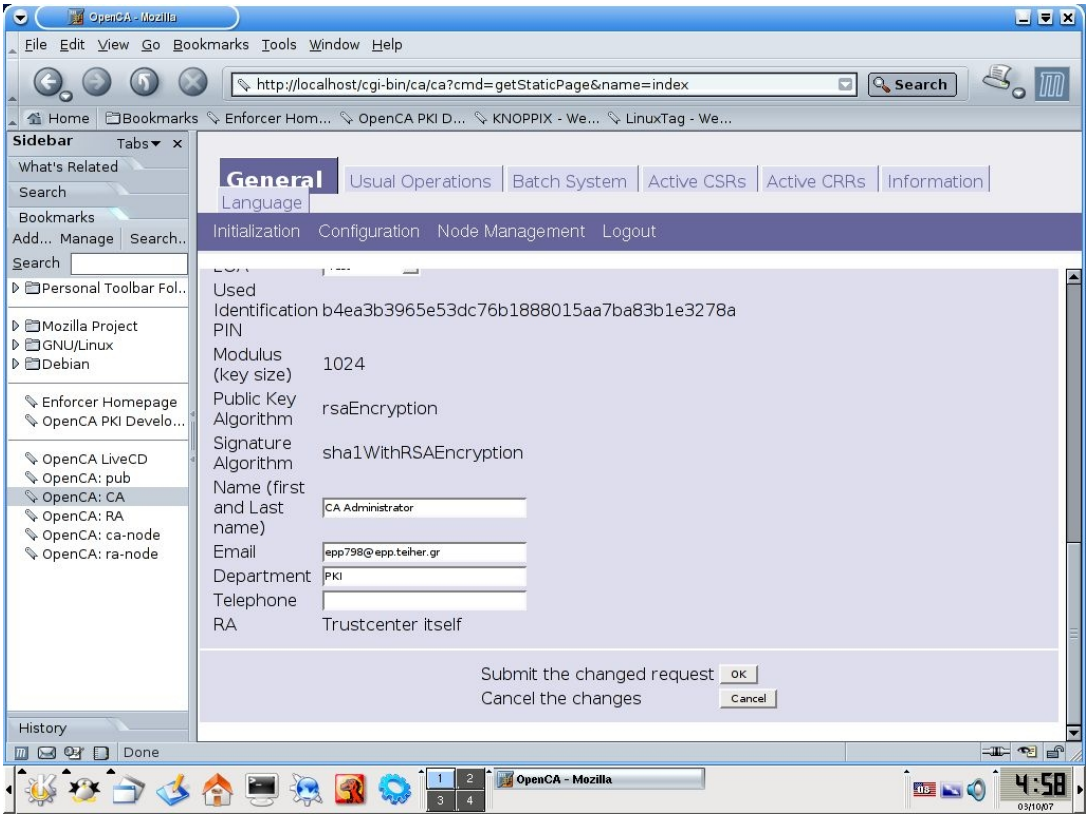


Επιστρέφουμε **Initialization -> Phase II** (Create an initial administrator)

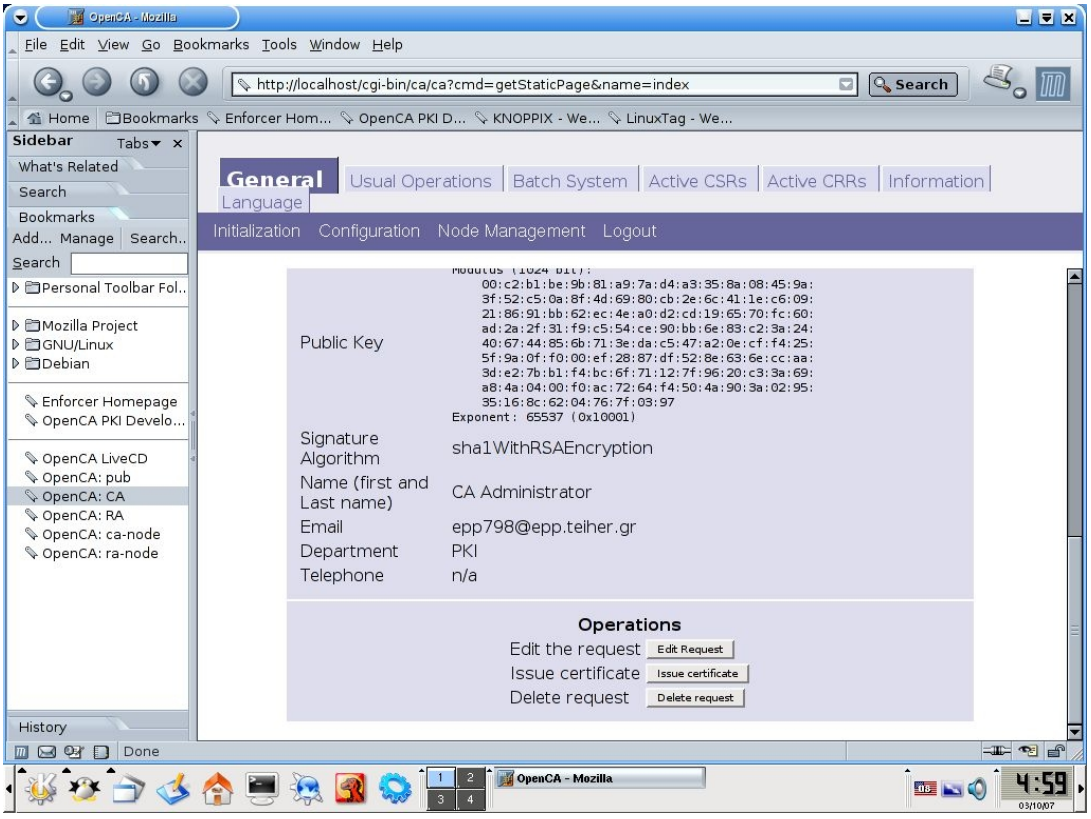


12. Επιλέγουμε **Edit the request,** δεν υπάρχει ουσιασικά κάποια διόρθωση αλλά για να προχωρίση η διαδικασία πρέπει να το επιλέξουμε. Στο τέλος της σελίδας επιλέγουμε "**Submit the changed request** " (ασχέτως αν δεν έχουν γίνει αλλαγές το επιλέγουμε)



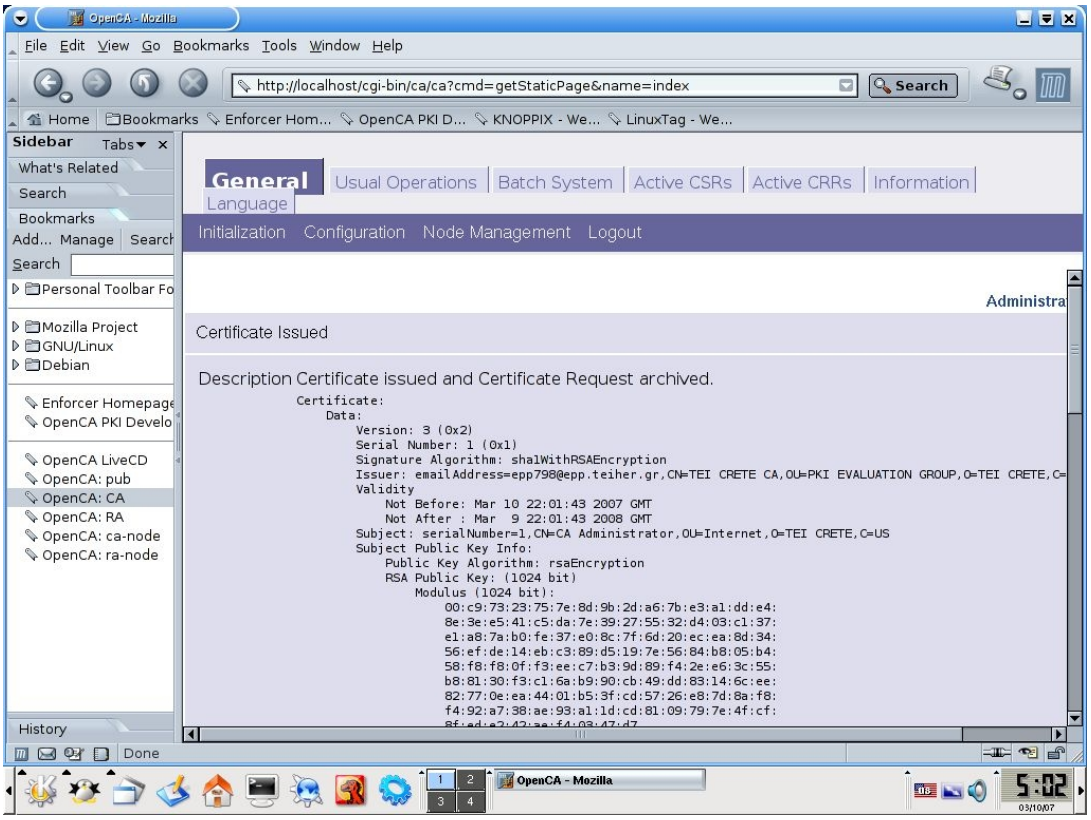


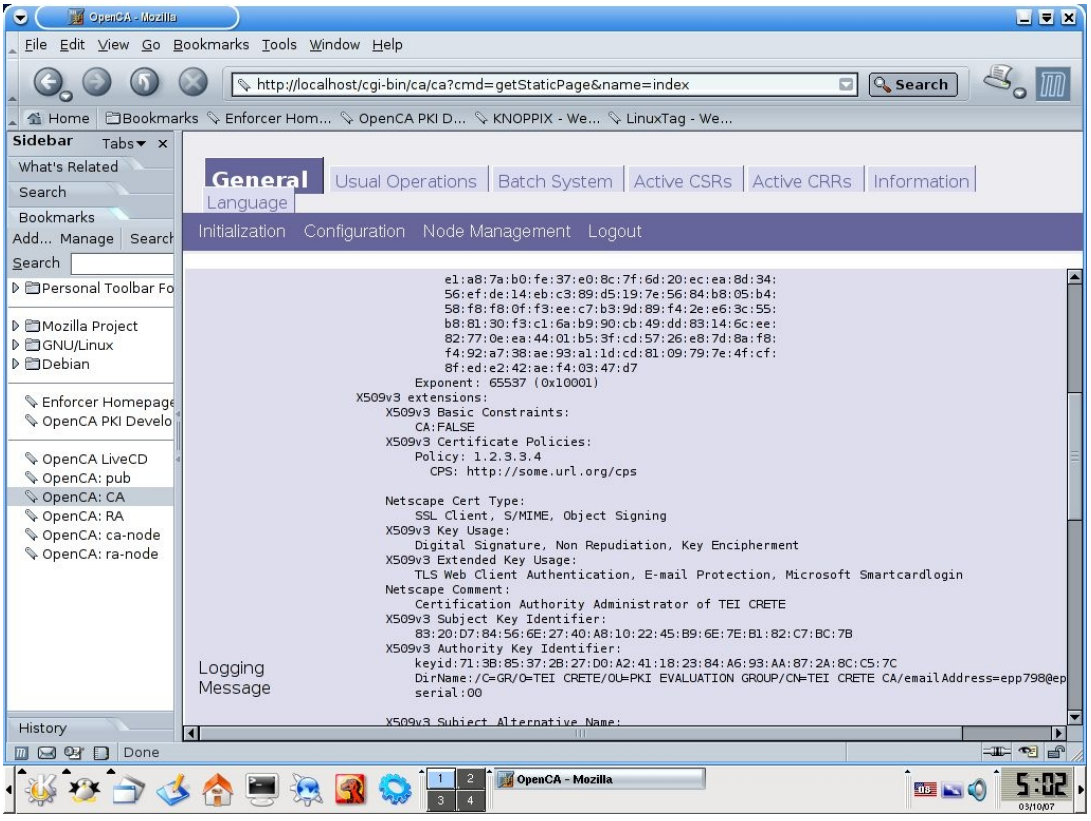
Επιλέγουμε στο τέλος της σελίδας “Οκ” και εμφανίζεται η παρακάτω οθόνη

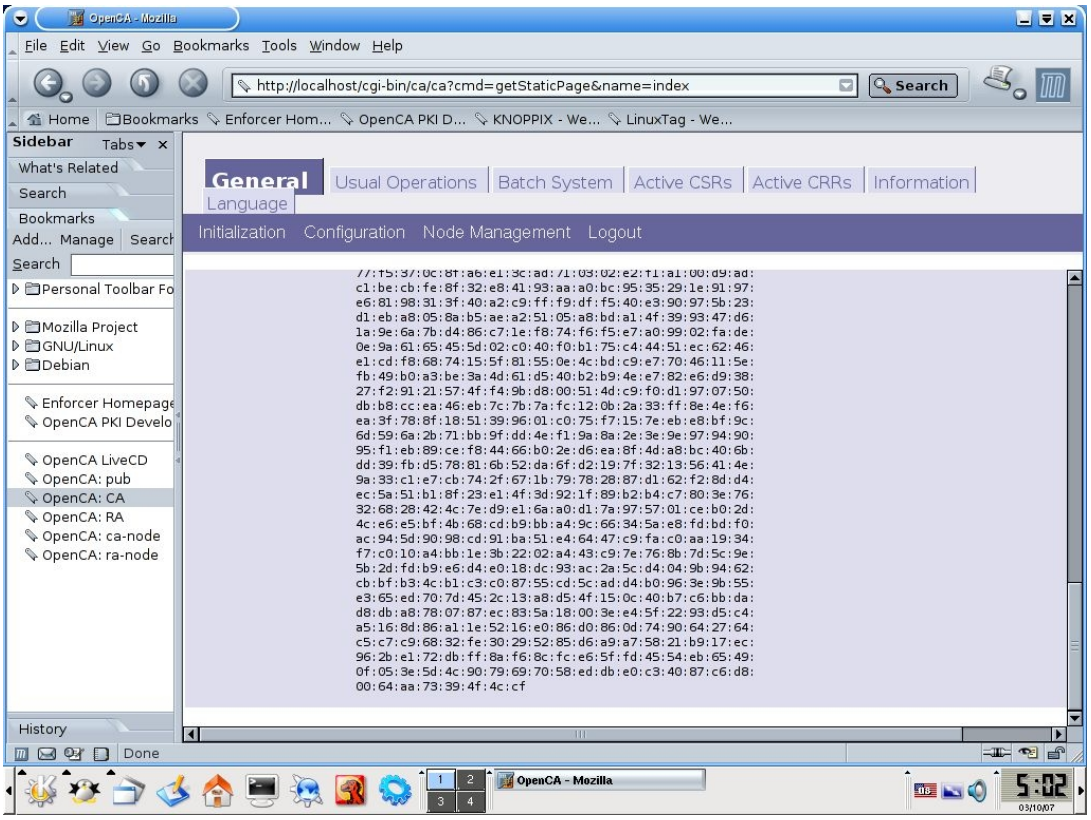


Έχουμε ήδη κάνει "**Edit**" οπότε επιλέγουμε "**Issue certificate**" για να εκδώσουμε το πιστοποιητικό. Δίνουμε το password του ιδιωτικού κλειδιού

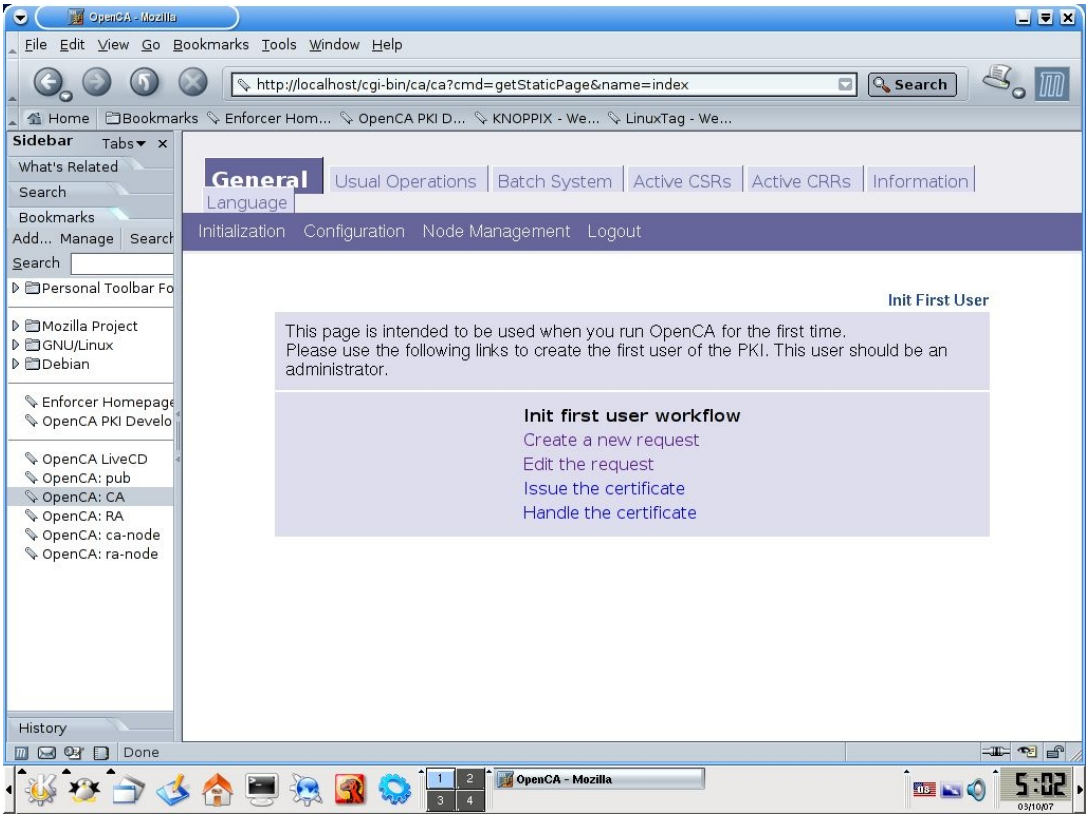
Εμφανίζεται μια οθόνη ενημέρωσης όπως η παρακάτω


OpenCA - Mozilla
File Edit View Go Bookmarks Tools Window Help
http://localhost/cgi-bin/ca/ca?cmd=getStaticPage&name=index
Search
Home Bookmarks Enforcer Hom... OpenCA PKI D... KNOPPIX - We... LinuxTag - We...
Sidebar Tabs
What's Related
Search
Bookmarks
Add... Manage Search
Search
Personal Toolbar Fo
Mozilla Project
GNU/Linux
Debian
Enforcer Homepage
OpenCA PKI Develo
OpenCA LiveCD
OpenCA: pub
OpenCA: CA
OpenCA: RA
OpenCA: ca-node
OpenCA: ra-node
History
General
Language
Usual Operations
Batch System
Active CSRs
Active CRRs
Information
Initialization
Configuration
Node Management
Logout
Administra
Certificate Issued
Description Certificate issued and Certificate Request archived.
Certificate:
Data:
Version: 3 (0x2)
Serial Number: 1 (0x1)
Signature Algorithm: sha1WithRSAEncryption
Issuer: emailAddress=epp798@epp.teiher.gr,CN=TEI CRETE CA,OU=PKI EVALUATION GROUP,O=TEI CRETE,C=
Validity
Not Before: Mar 10 22:01:43 2007 GMT
Not After : Mar 9 22:01:43 2008 GMT
Subject: serialNumber=1,CN=CA Administrator,OU=Internet,O=TEI CRETE,C=US
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (1024 bit)
Modulus (1024 bit):
00:c9:73:23:75:7e:8d:9b:2d:a6:7b:e3:a1:dd:e4:
8e:3e:e5:41:c5:da:7e:39:27:55:32:d4:03:c1:37:
e1:a8:7a:b0:fe:37:e0:8c:7f:6d:20:ec:ea:8d:34:
56:ef:de:14:eb:c3:89:d5:19:7e:56:84:b8:05:b4:
58:f8:f8:0f:f3:ee:c7:b3:9d:89:f4:2e:e6:3c:55:
b8:81:30:f3:c1:6a:b9:90:cb:49:dd:83:14:6c:ee:
82:77:0e:ea:44:01:b5:3f:cd:57:26:e8:7d:8a:f8:
f4:92:a7:38:ae:93:a1:1d:cd:81:09:79:7e:4f:cf:
Done
OpenCA - Mozilla
5:02
03/10/07

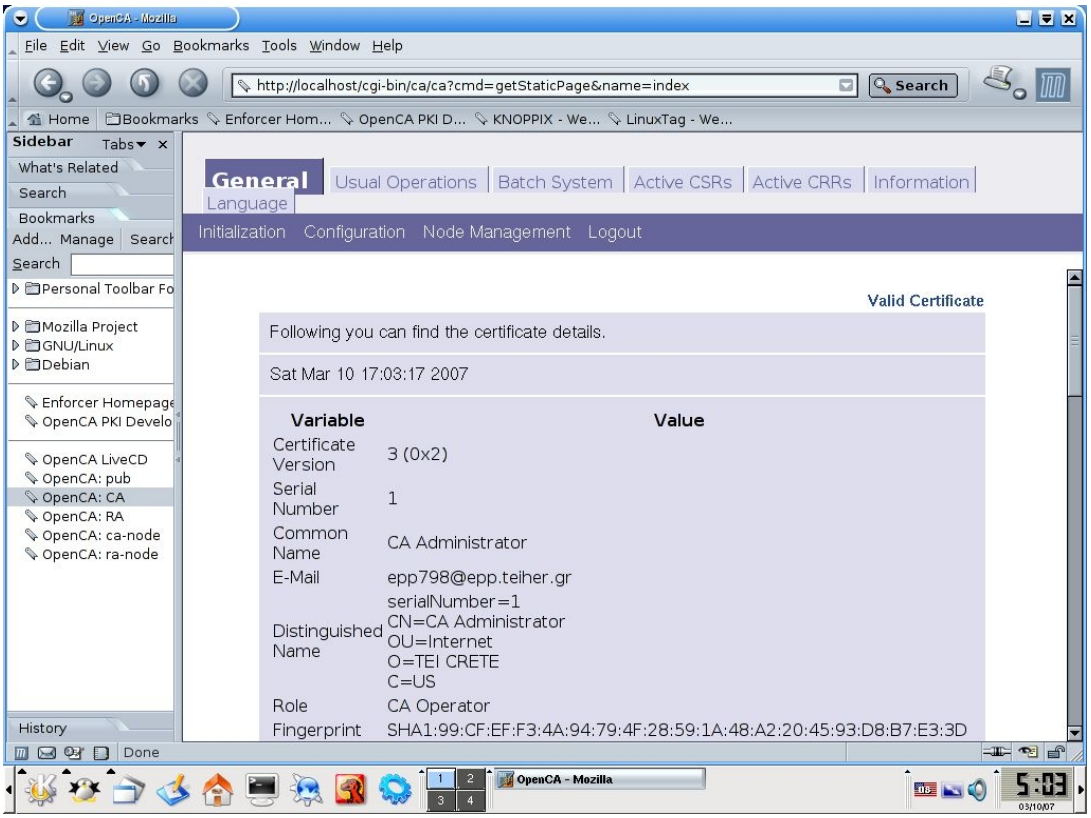

OpenCA - Mozilla
File Edit View Go Bookmarks Tools Window Help
http://localhost/cgi-bin/ca/ca?cmd=getStaticPage&name=index
Search
Home Bookmarks Enforcer Hom... OpenCA PKI D... KNOPPIX - We... LinuxTag - We...
Sidebar Tabs
What's Related
Search
Bookmarks
Add... Manage Search
Search
Personal Toolbar Fo
Mozilla Project
GNU/Linux
Debian
Enforcer Homepage
OpenCA PKI Develo
OpenCA LiveCD
OpenCA: pub
OpenCA: CA
OpenCA: RA
OpenCA: ca-node
OpenCA: ra-node
History
General
Language
Usual Operations
Batch System
Active CSRs
Active CRRs
Information
Initialization
Configuration
Node Management
Logout
e1:a8:7a:b0:fe:37:e0:8c:7f:6d:20:ec:ea:8d:34:
56:ef:de:14:eb:c3:89:d5:19:7e:56:84:b8:05:b4:
58:f8:f8:0f:f3:ee:c7:b3:9d:89:f4:2e:e6:3c:55:
b8:81:30:f3:c1:6a:b9:90:cb:49:dd:83:14:6c:ee:
82:77:0e:ea:44:01:b5:3f:cd:57:26:e8:7d:8a:f8:
f4:92:a7:38:ae:93:a1:1d:cd:81:09:79:7e:4f:cf:
8f:ed:e2:42:ae:f4:03:47:d7
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
X509v3 Certificate Policies:
Policy: 1.2.3.3.4
CPS: http://some.url.org/cps
Netscape Cert Type:
SSL Client, S/MIME, Object Signing
X509v3 Key Usage:
Digital Signature, Non Repudiation, Key Encipherment
X509v3 Extended Key Usage:
TLS Web Client Authentication, E-mail Protection, Microsoft Smartcardlogin
Netscape Comment:
Certification Authority Administrator of TEI CRETE
X509v3 Subject Key Identifier:
83:20:D7:84:56:6E:27:40:A8:10:22:45:B9:6E:7E:B1:82:C7:BC:7B
X509v3 Authority Key Identifier:
keyid:71:3B:85:37:2B:27:D0:A2:41:18:23:84:A6:93:AA:87:2A:8C:C5:7C
DirName:/C=GR/O=TEI CRETE/OU=PKI EVALUATION GROUP/CN=TEI CRETE CA/emailAddress=epp798@ep
serial:00
Logging
Message
X509v3 Subject Alternative Name:
Done
OpenCA - Mozilla
5:02
03/10/07



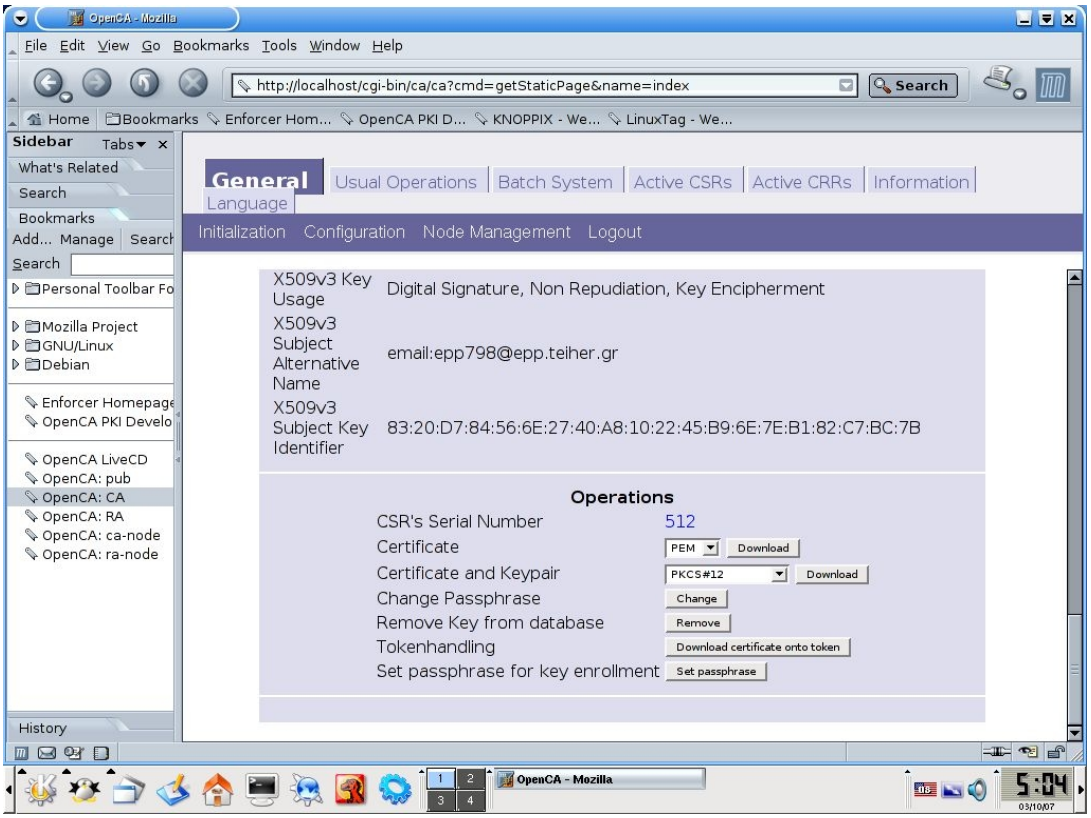
Επιστρέφουμε **Initialization -> Phase II** (Create an initial administrator)-> **Init first user**



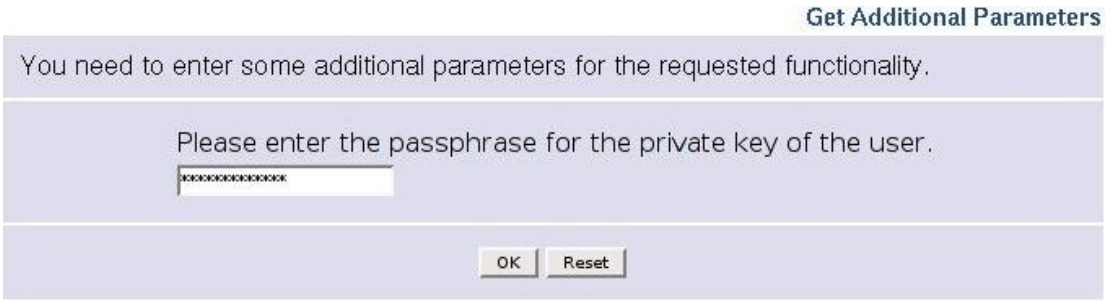
13. Επιλέγουμε “**Handle the certificate**” και εμφανίζεται η παρακάτω οθόνη



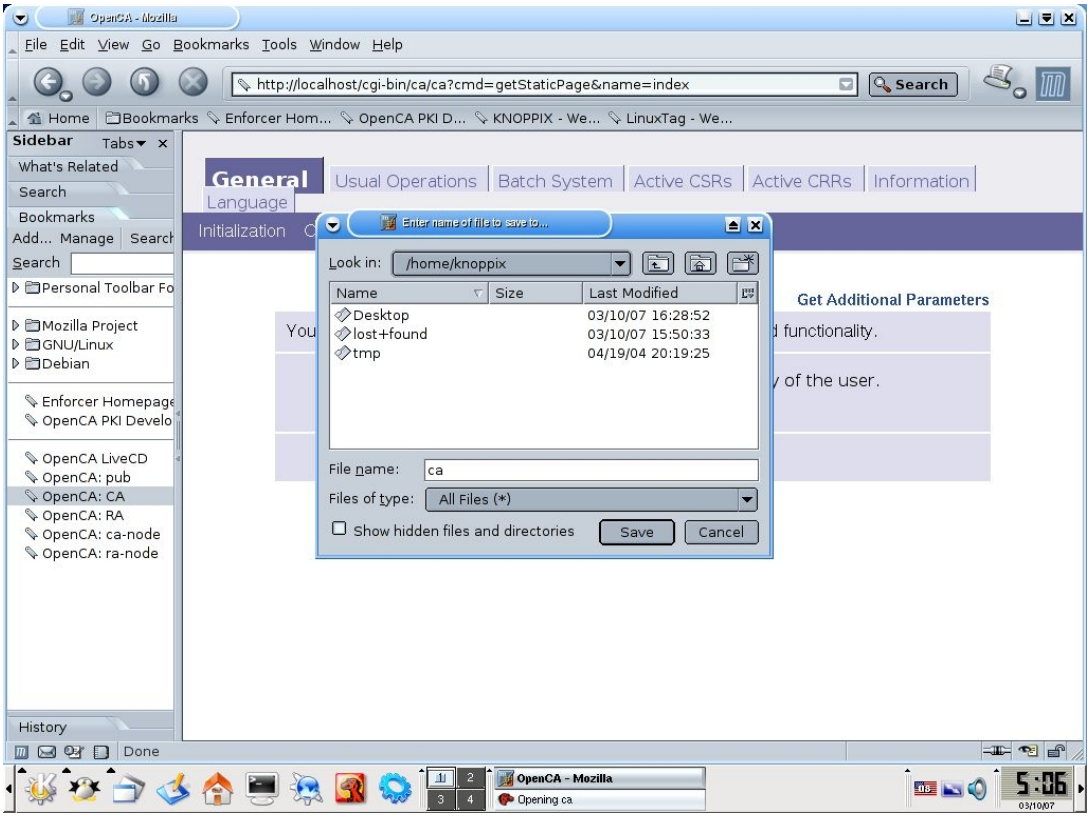
Στο κάτω μέρος της σελίδας στην επιλογή “**Certificate and keypair**”επιλέγουμε PKCS#12



Επιλέγουμε “**Download**” δίπλα στο “**Certificate and Keypair**”, θα μας ζητηθεί το PIN που δώσαμε στο βήμα 11. Επαναλαμβάνουμε ότι το password δεν έχει να κάνει το ιδιωτικό κλειδί αλλά σχετίζεται με την αίτηση για το πιστοποιητικό.



το δίνουμε και στη συνέχεια μπορούμε να το αποθηκεύσουμε και να το εισάγουμε στον Browser.

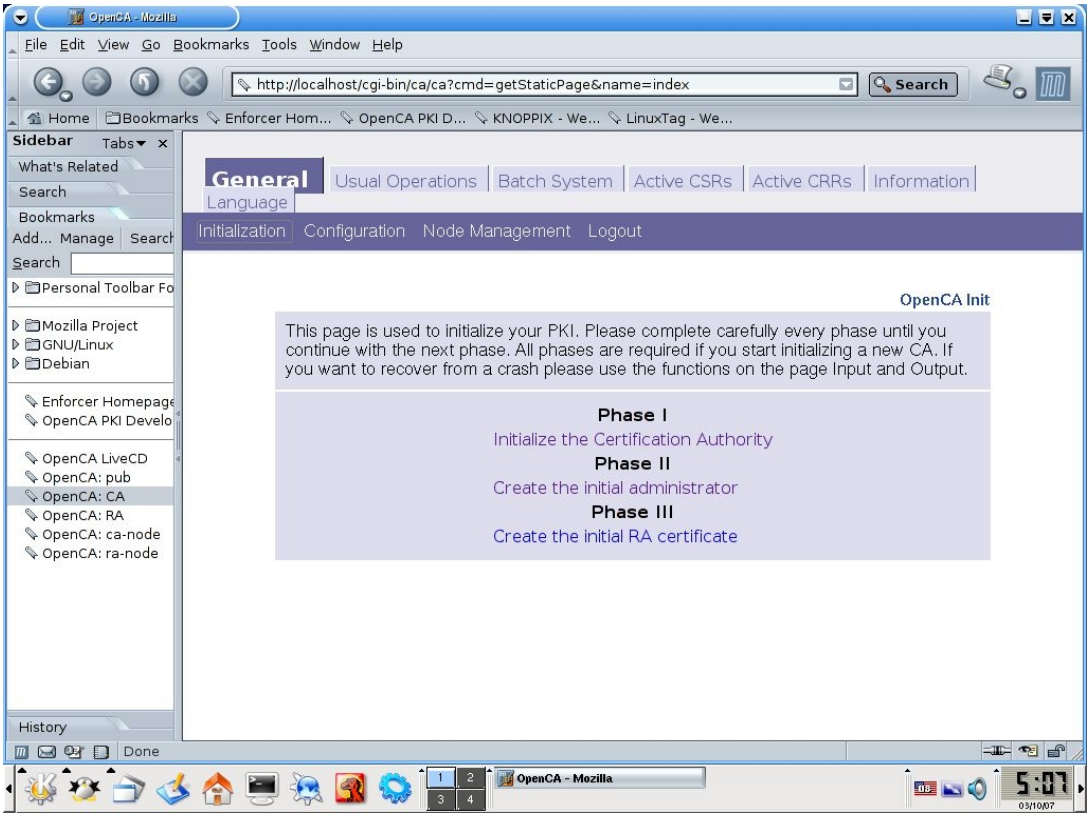


Όπως παρατηρούμε το πιστοποιητικό το αποθηκεύουμε στο /home/knoppix το οποίο έμεις του έχουμε ορίσει να είναι το dev/sda2 δηλαδή το flash disk μας.

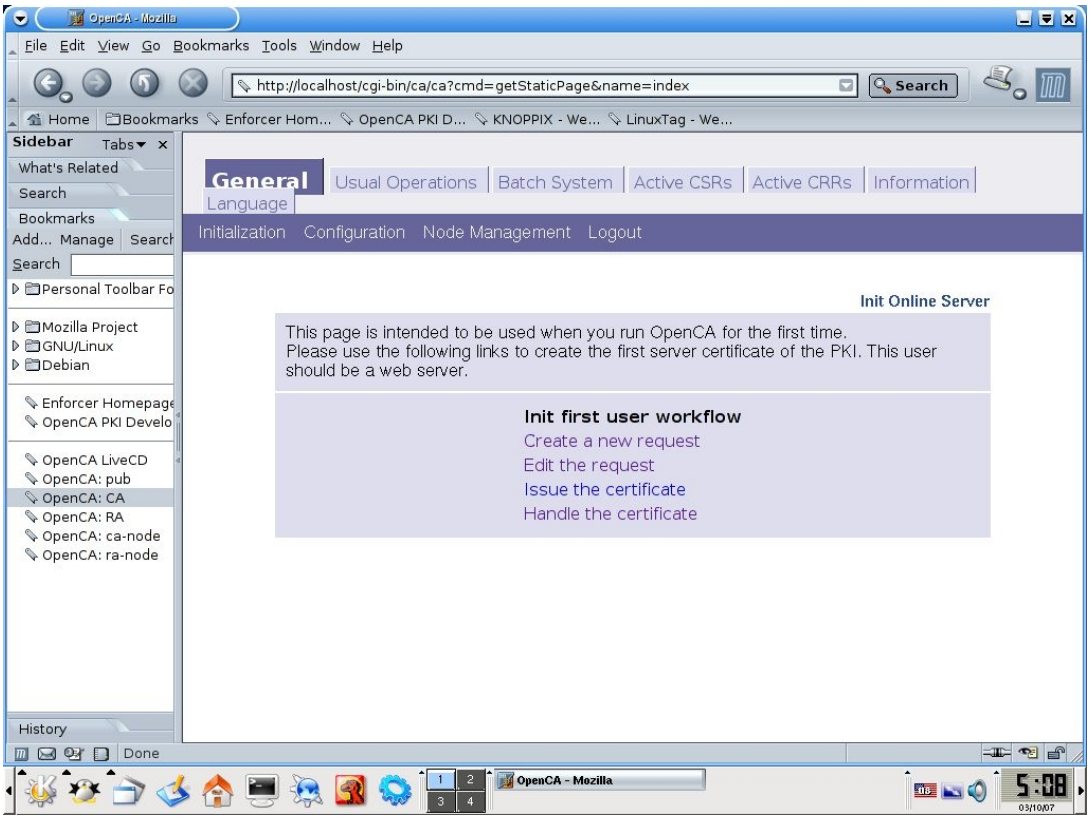
3.1.3 Φάση ΙΙΙ: Δημιουργία του αρχικού πιστοποιητικού RA

Ο RA διαχειριστής διαχειρίζεται την αρχή πιστοποίησης και ιεραρχικά είναι ένα επίπεδο κάτω από τον root CA. Μπορούμε να του δώσουμε περιορισμένα ή όχι δικαιώματα. Ο RA διαχειριστής χρησιμοποιείται στο σύστημα για περισσότερη ασφάλεια έτσι ώστε να μην δίδεται απευθείας πρόσβαση στον root CA. Έτσι οι αίτησεις για πιστοποιητικά πρώτα περνούν από τον RA και μετα προωθούνται στον CA για πιστοποίηση. Η διαδικασία δημιουργίας αρχικού χρήστη RA και του πιστοποιητικού του είναι παρόμοια με την διαδικασία που ακολουθήσαμε παραπάνω για τον CA.

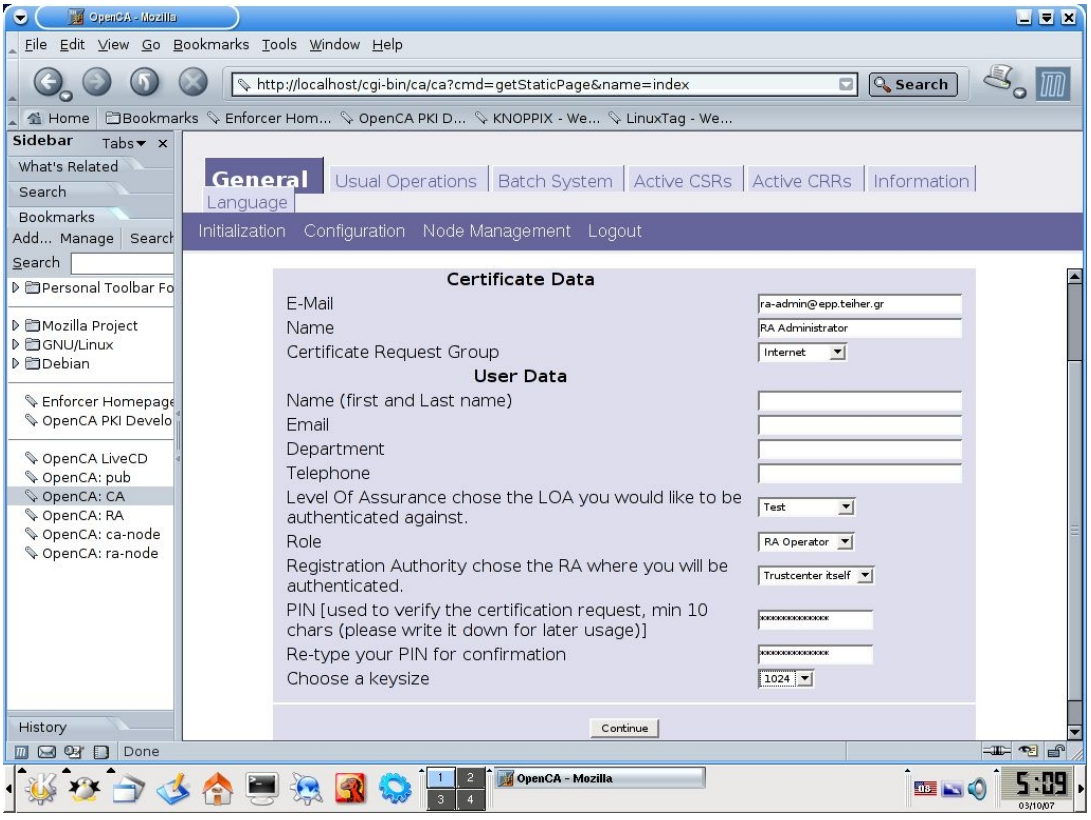
14. Επιλέγουμε "**Create the initial RA certificate**"



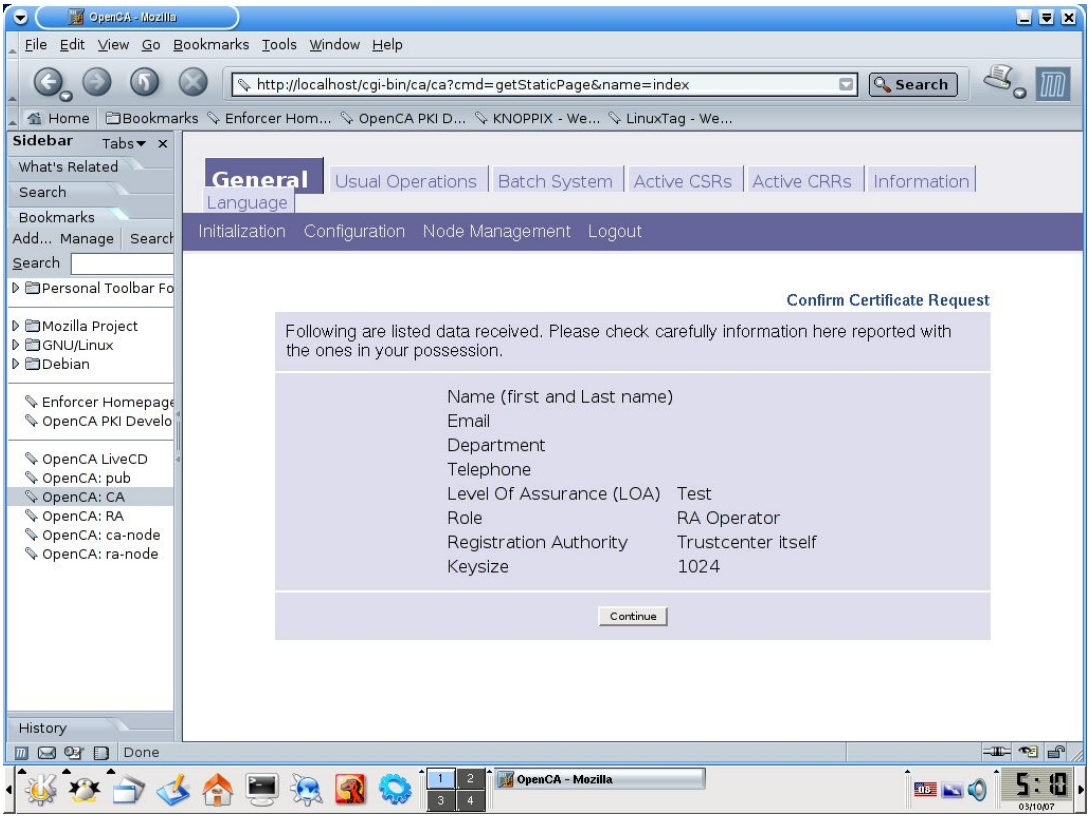
Εμφανίζεται η σελίδα "**Init First user workflow**"



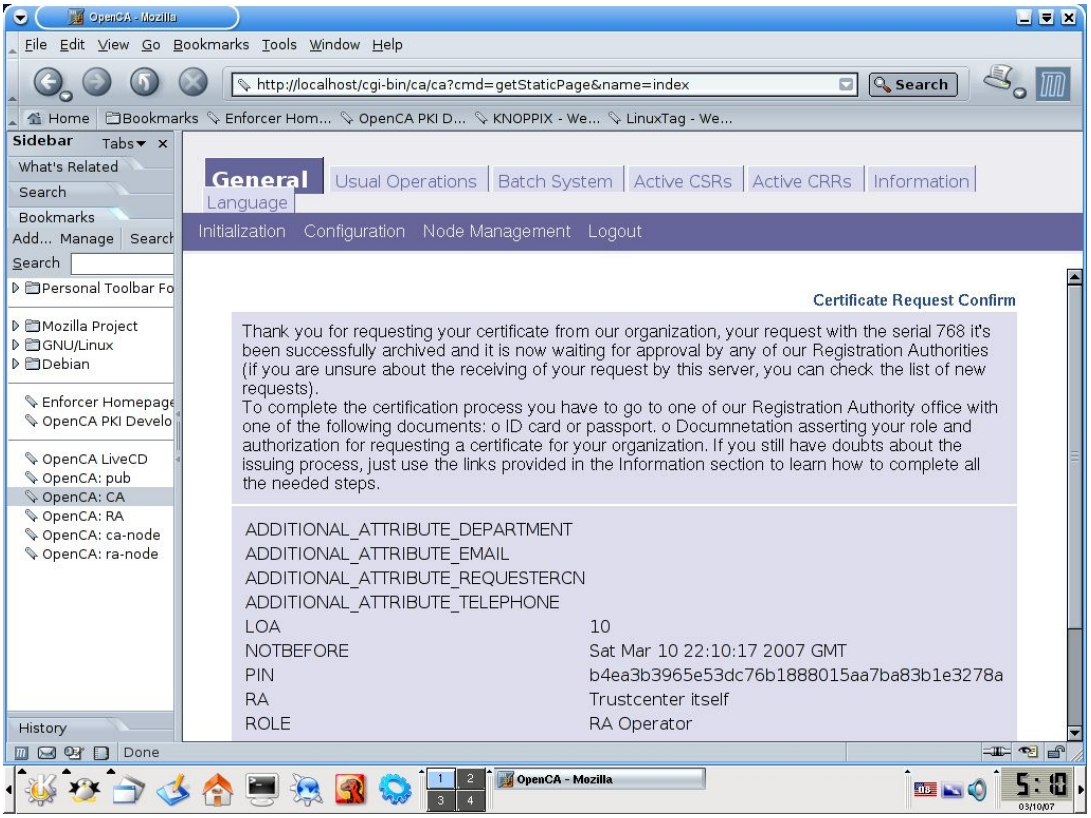
15. Σε αυτό το βήμα δημιουργούμε ένα πιστοποιητικό (και ένα ζευγάρι κλειδιού) για την αναγνώριση του RA διαχειριστή, επιλέγουμε "**Create a new request**" και συμπληρώνουμε τα πεδία δίνοντας επίσης ένα pin το οποίο θα προστατεύει το ιδιωτικό κλειδί.



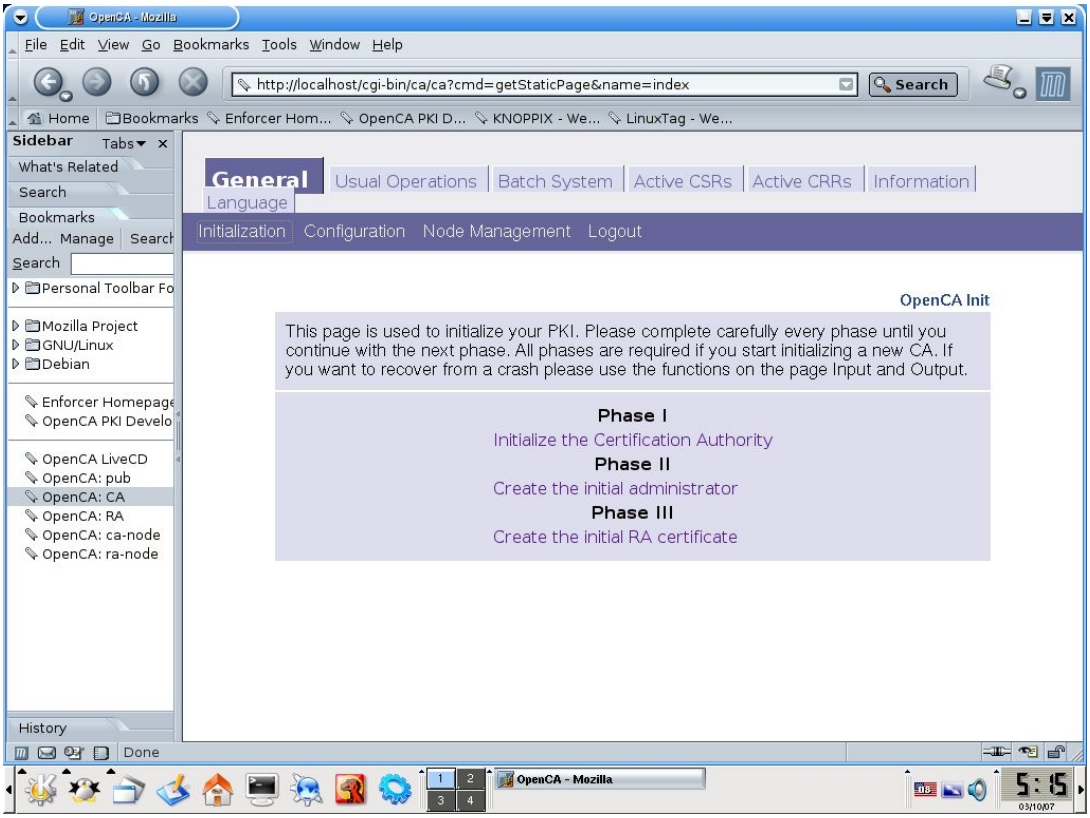
Στη συνέχεια εμφανίζεται μια οθόνη επιβεβαίωσης



Επιλέγοντας "**Continue**" εμφανίζεται ακόμη μια επιβεβαίωση

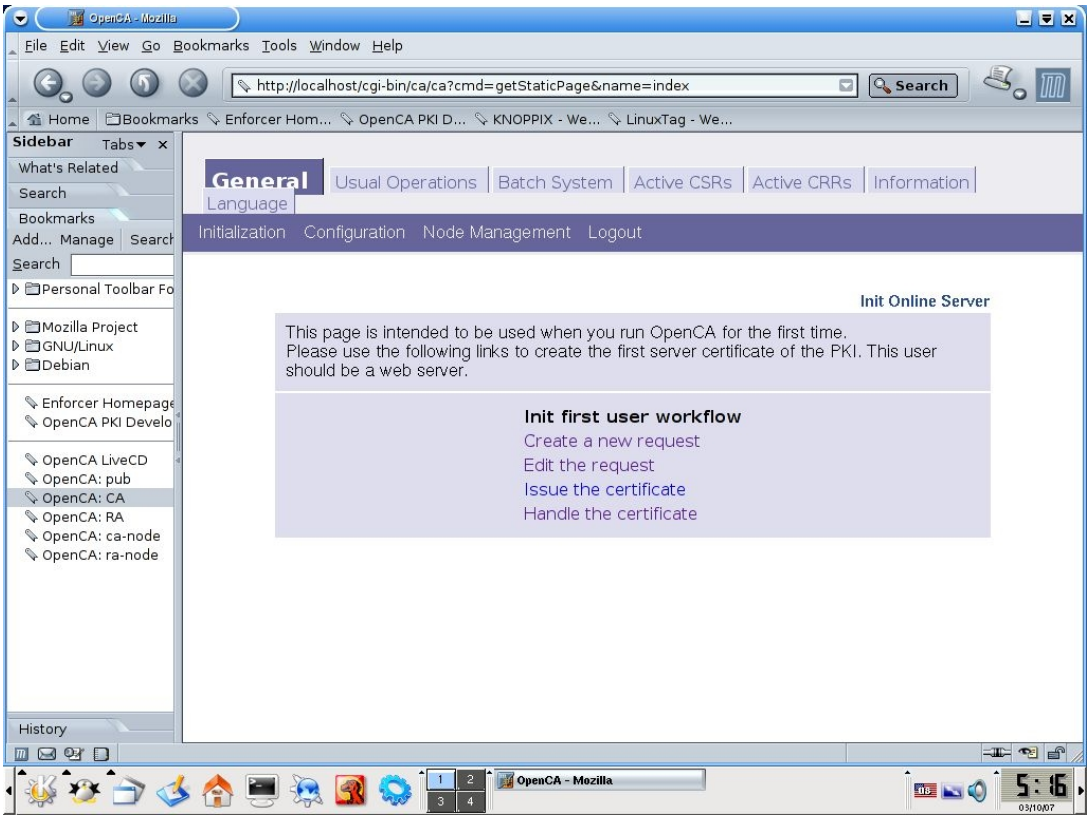


Σε αυτό το σημείο θα πρέπει να ξεκαθαρίσουμε ότι τον χρήστη RA-administrator που μόλις δημιουργήσαμε δεν θα τον χρησιμοποιήσουμε ποτέ άμεσσα. Οι κύριες εργασίες μεταξύ του RA και του CA στο OpenCA πιστοποιούνται με τη χρήση του ιδιωτικού κλείδιου (που έχουμε δημιουργήσει σε προηγούμενα βήματα).

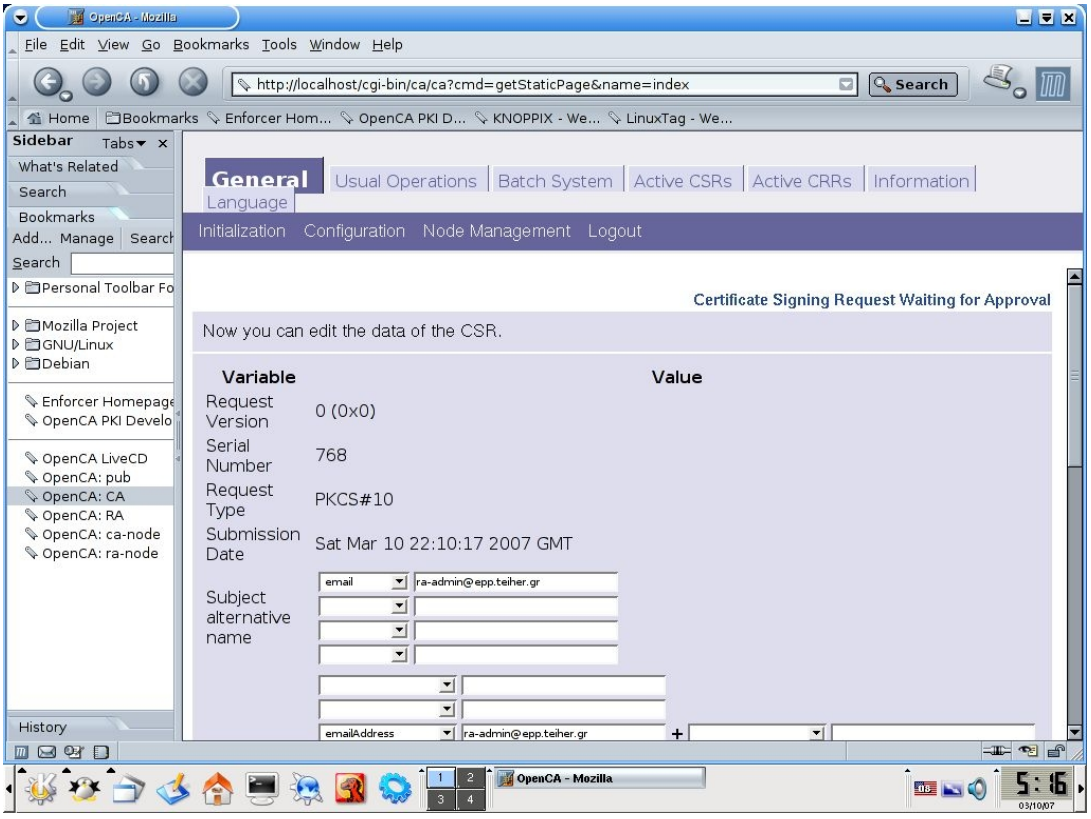


Επιστρέφουμε στη **φάση III**

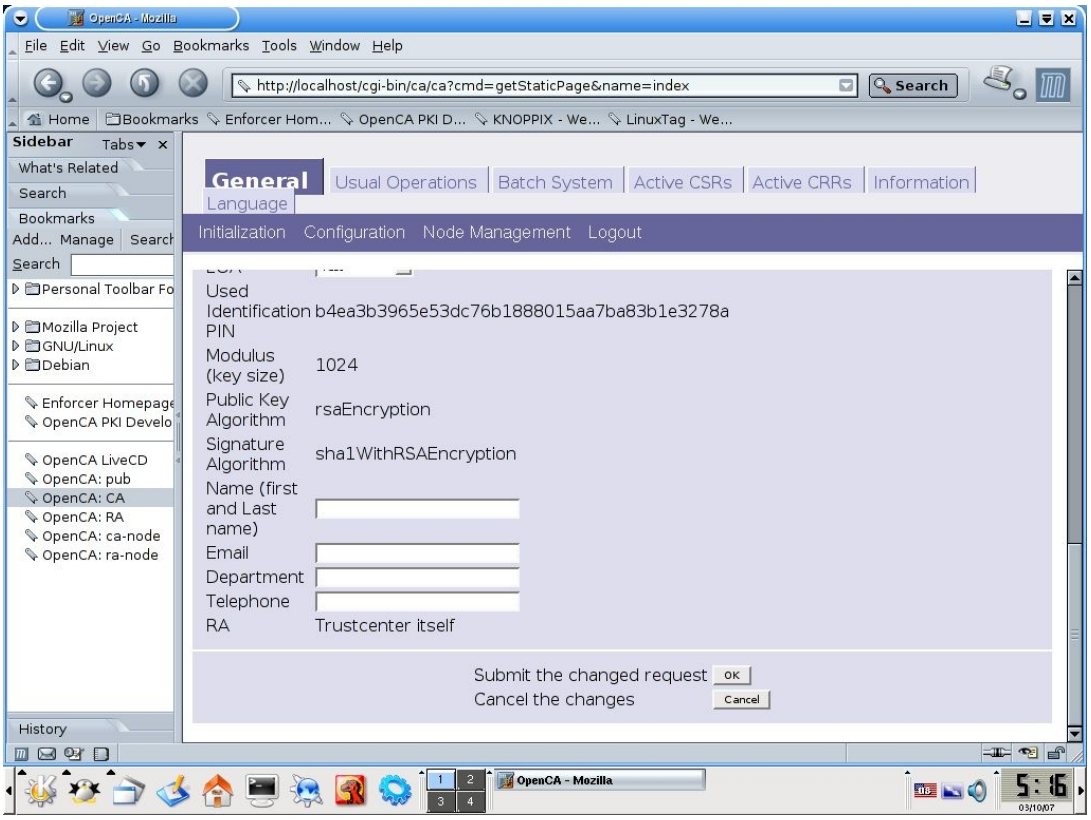
16. Επιλέγουμε "**Edit the request**"



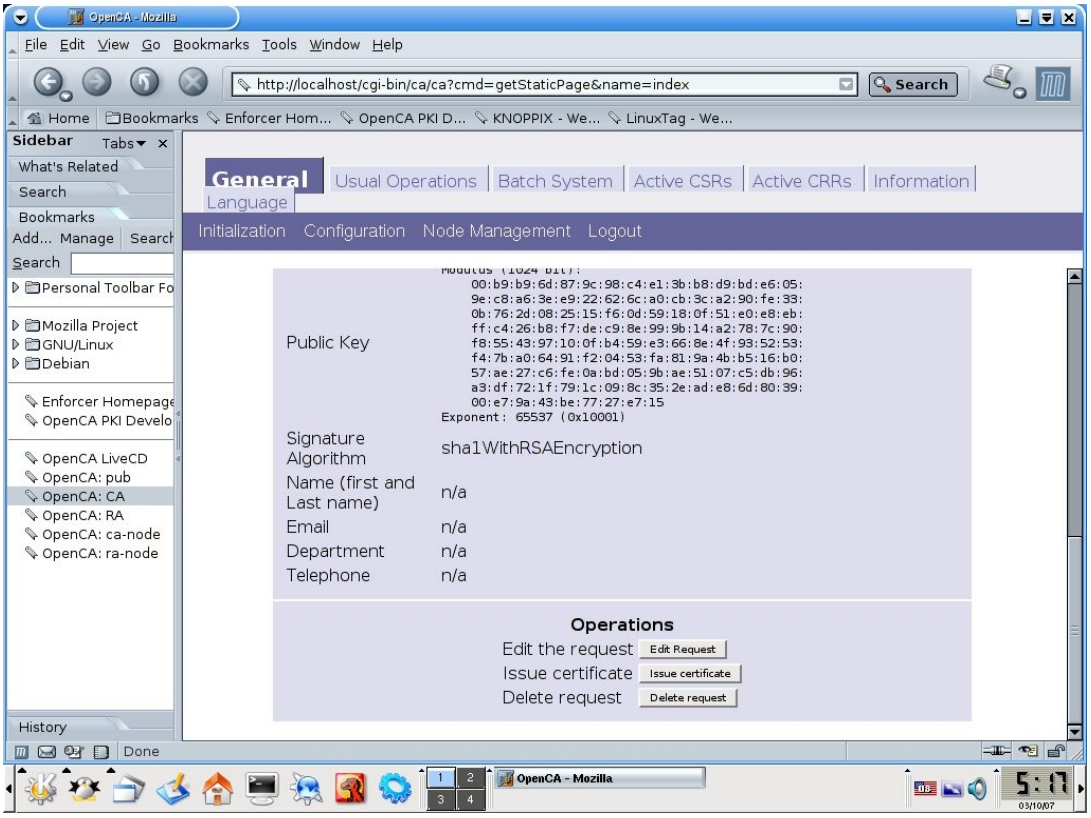
Εμφανίζεται η παρακάτω οθόνη



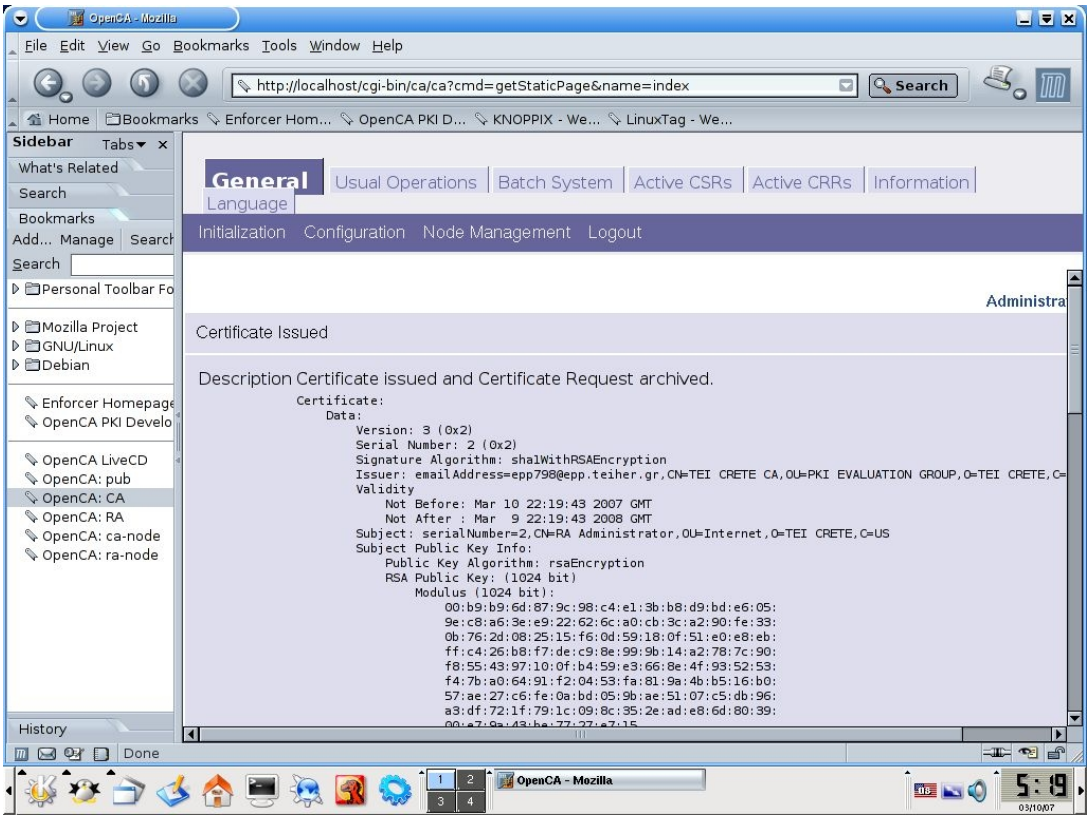
Δεν χρειάζεται να κάνουμε καποία αλλαγή στην παραπάνω οθόνη, αν θέλουμε όμως να αλλάξουμε κάτι το σύστημα το επιτρέπει. Στο κάτω μέρος της σελίδας επιλέγουμε "**Submit the changed request**" (ασχέτως αν δεν έχουν γίνει αλλαγές το επιλέγουμε)

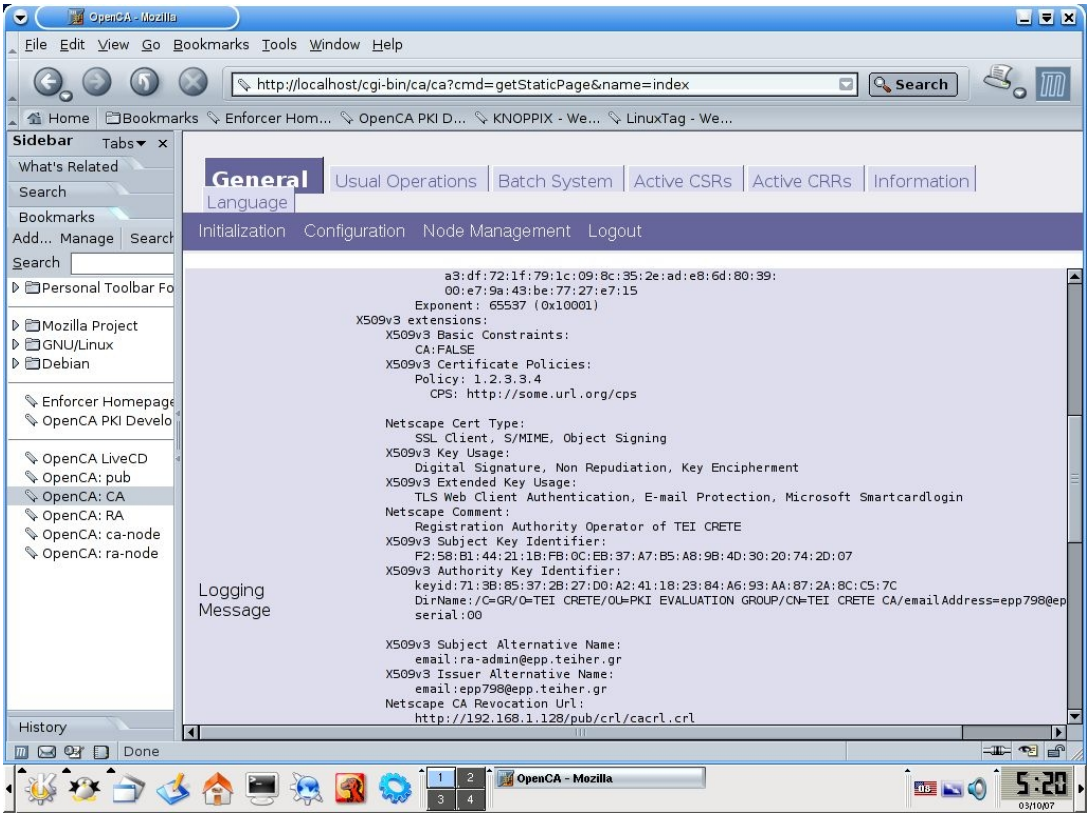


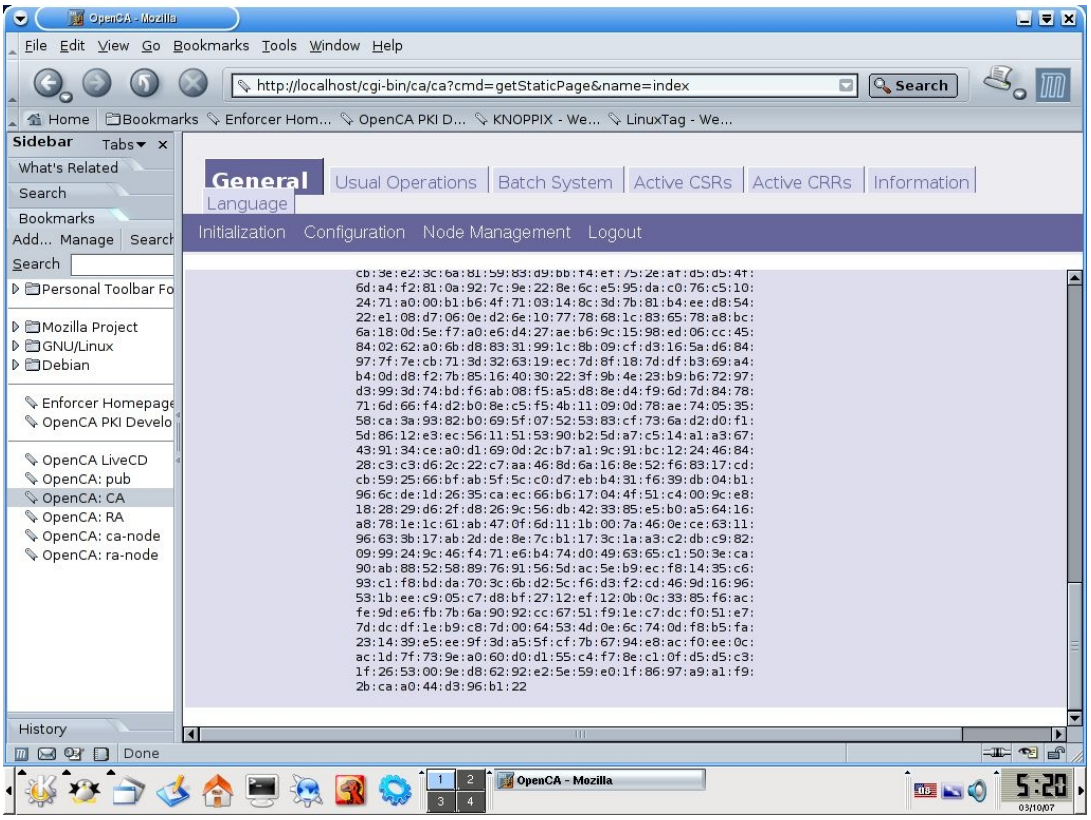
Επιλέγουμε "OK", εμφανίζεται η παρακάτω οθόνη (φάινεται το scrolling)



Επιλέγουμε "**Issue Certificate**", θα ζητηθεί το password του private key, το εισάγουμε και βλέπουμε την παρακάτω οθόνη.

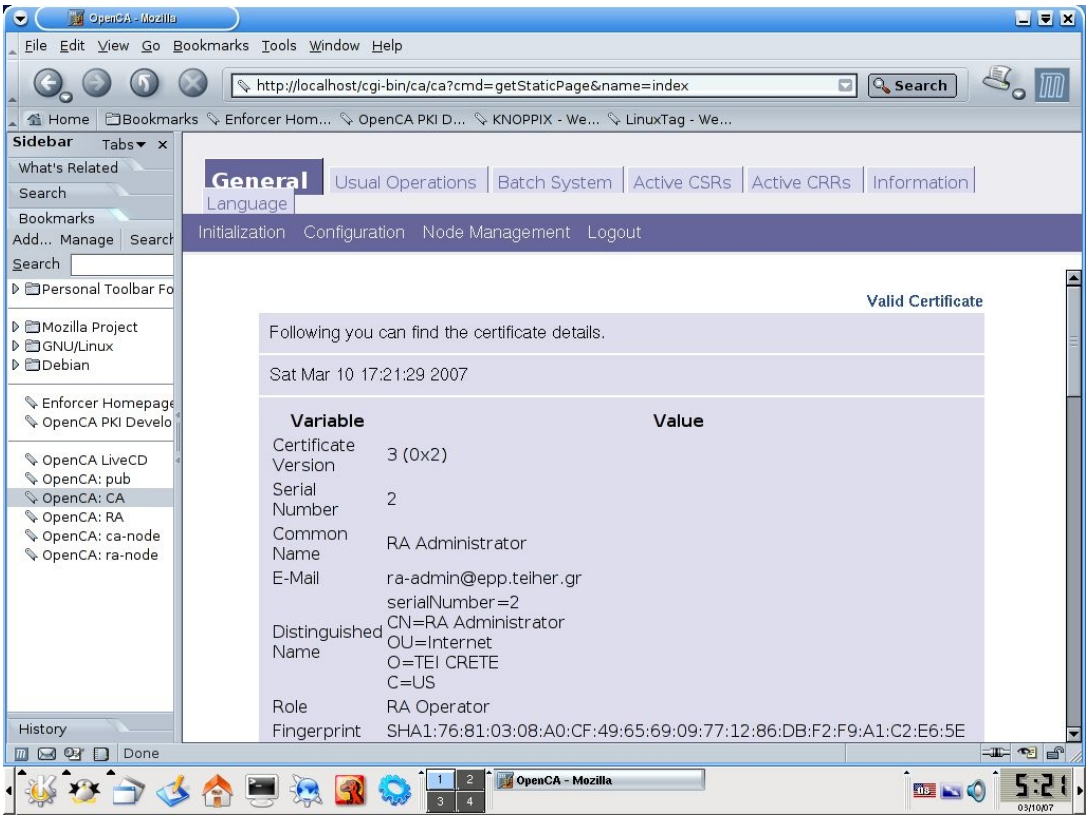

OpenCA - Mozilla
File Edit View Go Bookmarks Tools Window Help
http://localhost/cgi-bin/ca/ca?cmd=getStaticPage&name=index
Search
Home Bookmarks Enforcer Hom... OpenCA PKI D... KNOPPIX - We... LinuxTag - We...
Sidebar Tabs
What's Related
Search
Bookmarks
Add... Manage Search
Search
Personal Toolbar Fo
Mozilla Project
GNU/Linux
Debian
Enforcer Homepage
OpenCA PKI Develo
OpenCA LiveCD
OpenCA: pub
OpenCA: CA
OpenCA: RA
OpenCA: ca-node
OpenCA: ra-node
History
Done
General
Usual Operations
Batch System
Active CSRs
Active CRRs
Information
Language
Initialization
Configuration
Node Management
Logout
Administra
Certificate Issued
Description Certificate issued and Certificate Request archived.
Certificate:
Data:
Version: 3 (0x2)
Serial Number: 2 (0x2)
Signature Algorithm: sha1WithRSAEncryption
Issuer: emailAddress=epp798@epp.teiher.gr,CN=TEI CRETE CA,OU=PKI EVALUATION GROUP,O=TEI CRETE,C=
Validity
Not Before: Mar 10 22:19:43 2007 GMT
Not After : Mar 9 22:19:43 2008 GMT
Subject: serialNumber=2,CN=RA Administrator,OU=Internet,O=TEI CRETE,C=US
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (1024 bit)
Modulus (1024 bit):
00:b9:b9:6d:87:9c:98:c4:e1:3b:b8:d9:bd:e6:05:
9e:c8:a6:3e:e9:22:62:6c:a0:cb:3c:a2:90:fe:33:
0b:76:2d:08:25:15:f6:0d:59:18:0f:51:e0:e8:eb:
ff:c4:26:b8:f7:de:c9:8e:99:9b:14:a2:78:7c:90:
f8:55:43:97:10:0f:b4:59:e3:66:8e:4f:93:52:53:
f4:7b:a0:64:91:f2:04:53:fa:81:9a:4b:b5:16:b0:
57:ae:27:c6:fe:0a:bd:05:9b:ae:51:07:c5:db:96:
a3:df:72:1f:79:1c:09:8c:35:2e:ad:e8:6d:80:39:
OpenCA - Mozilla
5:19
03/10/07

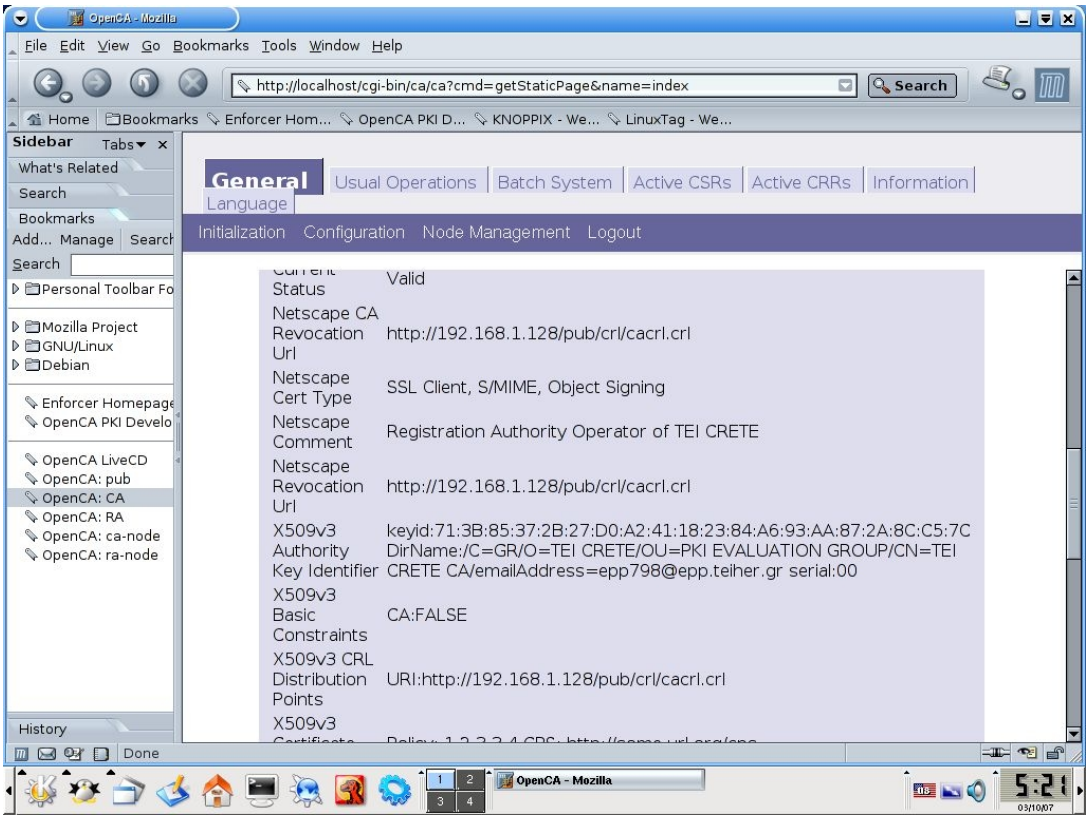

OpenCA - Mozilla
File Edit View Go Bookmarks Tools Window Help
http://localhost/cgi-bin/ca/ca?cmd=getStaticPage&name=index
Search
Home Bookmarks Enforcer Hom... OpenCA PKI D... KNOPPIX - We... LinuxTag - We...
Sidebar Tabs
What's Related
Search
Bookmarks
Add... Manage Search
Search
Personal Toolbar Fo
Mozilla Project
GNU/Linux
Debian
Enforcer Homepage
OpenCA PKI Develo
OpenCA LiveCD
OpenCA: pub
OpenCA: CA
OpenCA: RA
OpenCA: ca-node
OpenCA: ra-node
History
Done
General
Usual Operations
Batch System
Active CSRs
Active CRRs
Information
Language
Initialization
Configuration
Node Management
Logout
a3:df:72:1f:79:1c:09:8c:35:2e:ad:e8:6d:80:39:
00:e7:9a:43:be:77:27:e7:15
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
X509v3 Certificate Policies:
Policy: 1.2.3.3.4
CPS: http://some.url.org/cps
Netscape Cert Type:
SSL Client, S/MIME, Object Signing
X509v3 Key Usage:
Digital Signature, Non Repudiation, Key Encipherment
X509v3 Extended Key Usage:
TLS Web Client Authentication, E-mail Protection, Microsoft Smartcardlogin
Netscape Comment:
Registration Authority Operator of TEI CRETE
X509v3 Subject Key Identifier:
F2:58:B1:44:21:1B:FB:0C:EB:37:A7:B5:A8:9B:4D:30:20:74:2D:07
X509v3 Authority Key Identifier:
keyid:71:3B:85:37:2B:27:D0:A2:41:18:23:84:A6:93:AA:87:2A:8C:C5:7C
DirName:/C=GR/O=TEI CRETE/OU=PKI EVALUATION GROUP/CN=TEI CRETE CA/emailAddress=epp798@ep
serial:00
X509v3 Subject Alternative Name:
email:ra-admin@epp.teiher.gr
X509v3 Issuer Alternative Name:
email:epp798@epp.teiher.gr
Netscape CA Revocation Url:
http://192.168.1.128/pub/crl/cacrl.crl
Logging
Message
OpenCA - Mozilla
5:20
03/10/07



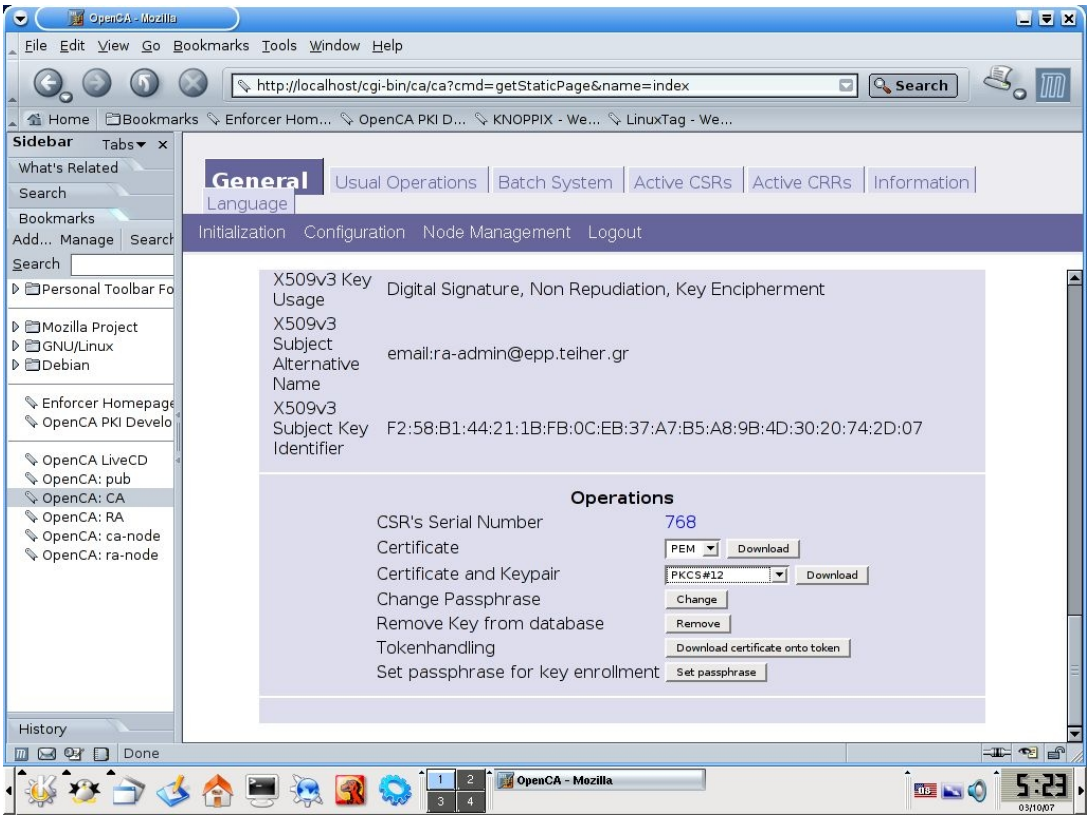
Επιστρέφουμε στη φάση III, στο τελευταίο βήμα για να ολοκληρωθεί το πιστοποιητικό του RA.

17. Επιλέγουμε "**Handle the Request**"

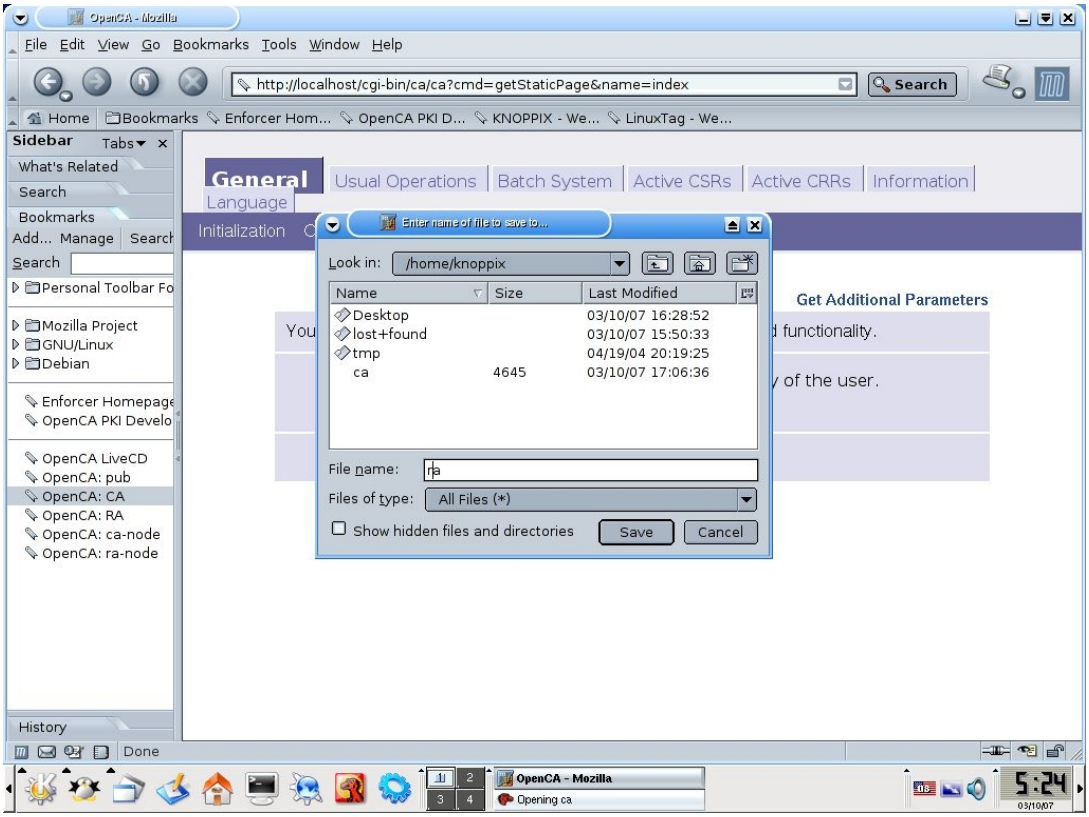




Στη συνέχεια στο τομέα "**Operations**" στο πεδίο "**Certificate and keypair**" επιλέγουμε PKCS#12 και "**Download**" θα ζητηθεί το pin που δώσαμε παραπάνω.



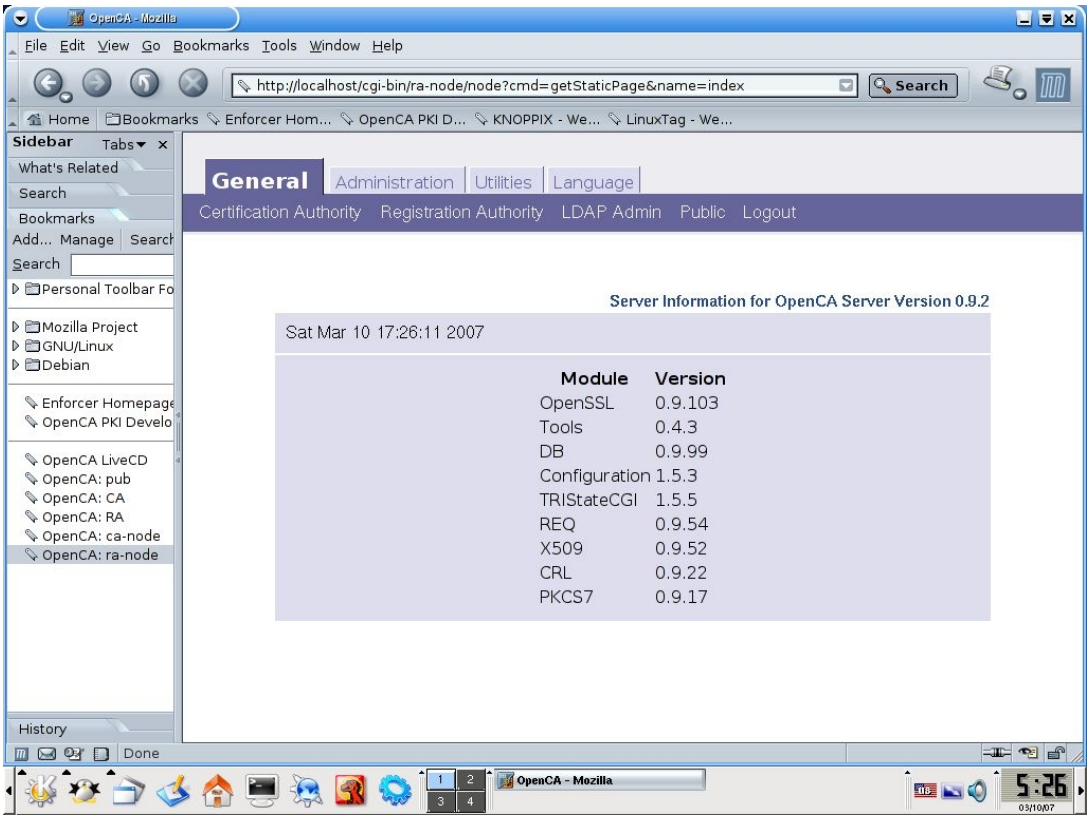
Στη συνέχεια το αποθηκεύουμε στο /home/Knoppix το οποίο ουσιαστικά είναι το flash disk (/dev/sda2) και μπορούμε να το εντάξουμε στο browser όποτε θέλουμε.



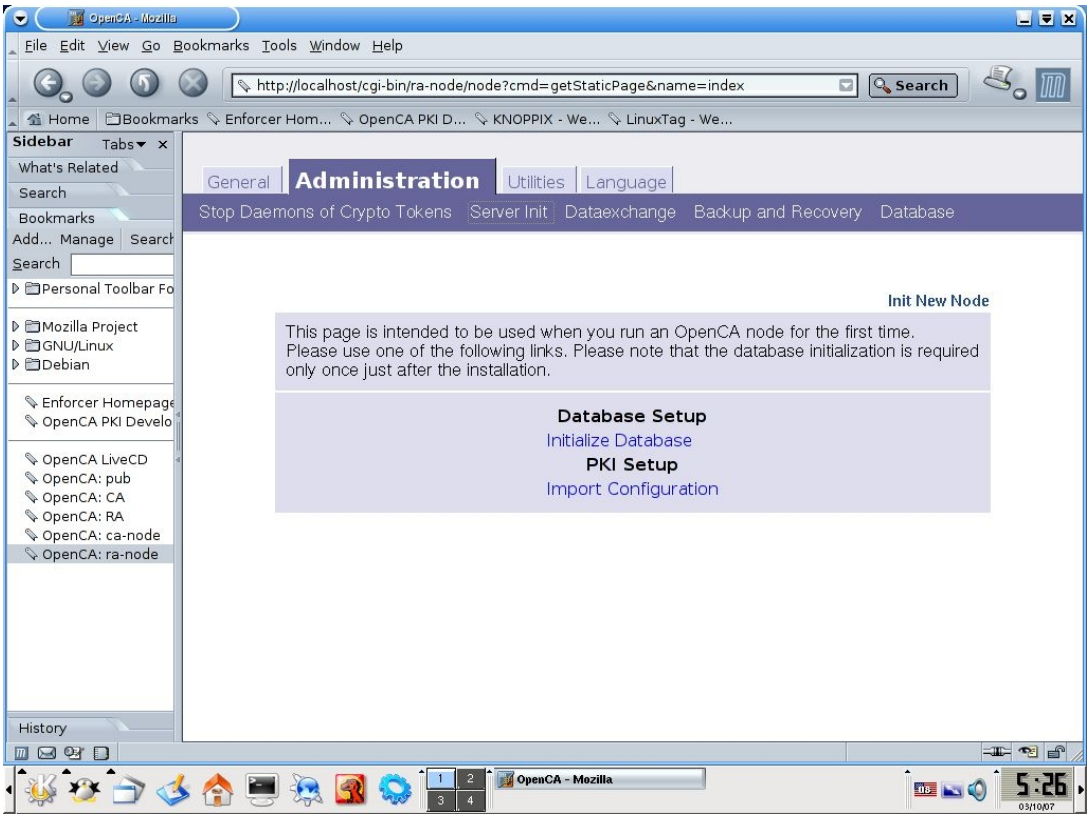
3.2 Αρχικοποίηση του RA

Έχουμε αρχικοποιησεί τον διαχειριστή του συστήματος RA και έχουμε δημιουργήσει και ένα πιστοποιητικό γι αυτόν. Αυτό που μας μένει είναι να αρχικοποίησουμε το ίδιο το σύστημα RA έτσι ώστε να είναι έτοιμο για λειτουργία. Αύτη η αρχικοποίηση του RA συστήματος μεταξύ των άλλων θα επιτρέπει τα πιστοποιητικά που δημιουργούνται από την αρχή μας, να είναι διαθέσιμα στους χρήστες για download από το http://CAip/pub η οποία CAip είναι μια ψεύτικη ip που δίνει ο DHCP στο δίκτυο μας και κατά συνέπεια στον CA, αλλά αυτά θα τα δούμε λίγο αργότερα.

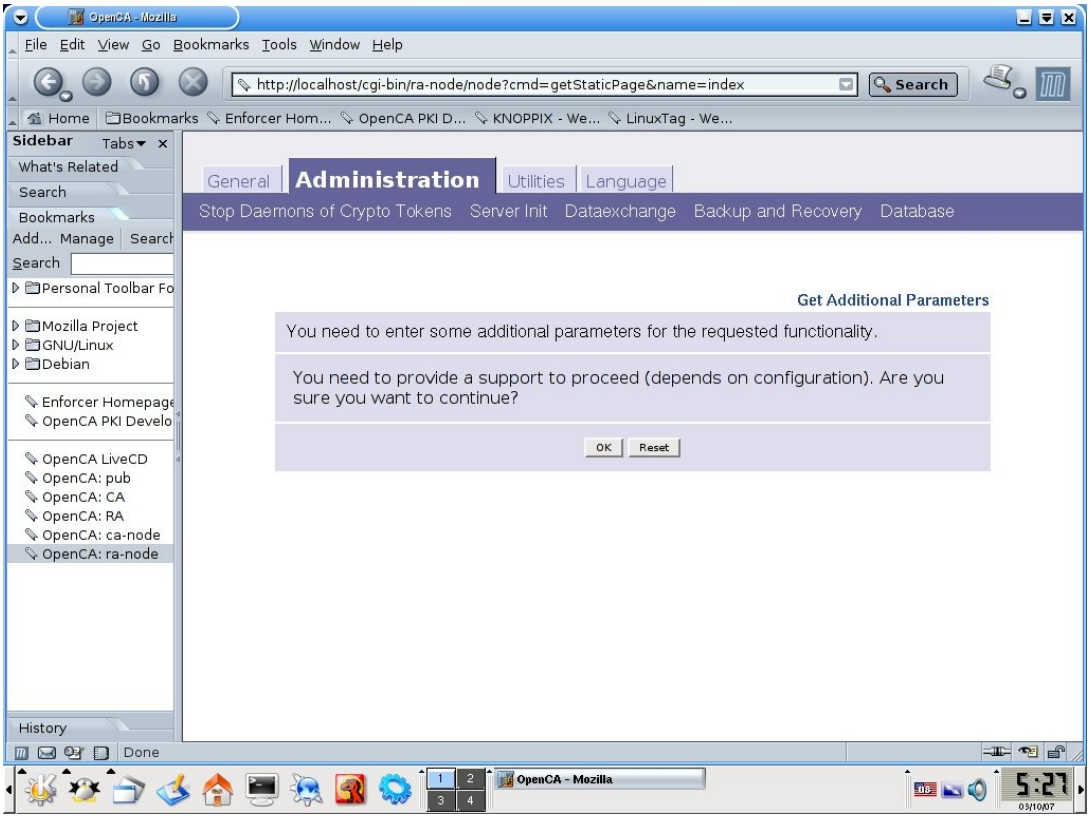
18. Συνδεόμαστε στο **http://localhost/ra-node** ή επιλέγουμε απο τα bookmarks αριστερά στον mozilla OpenCA:ra-node

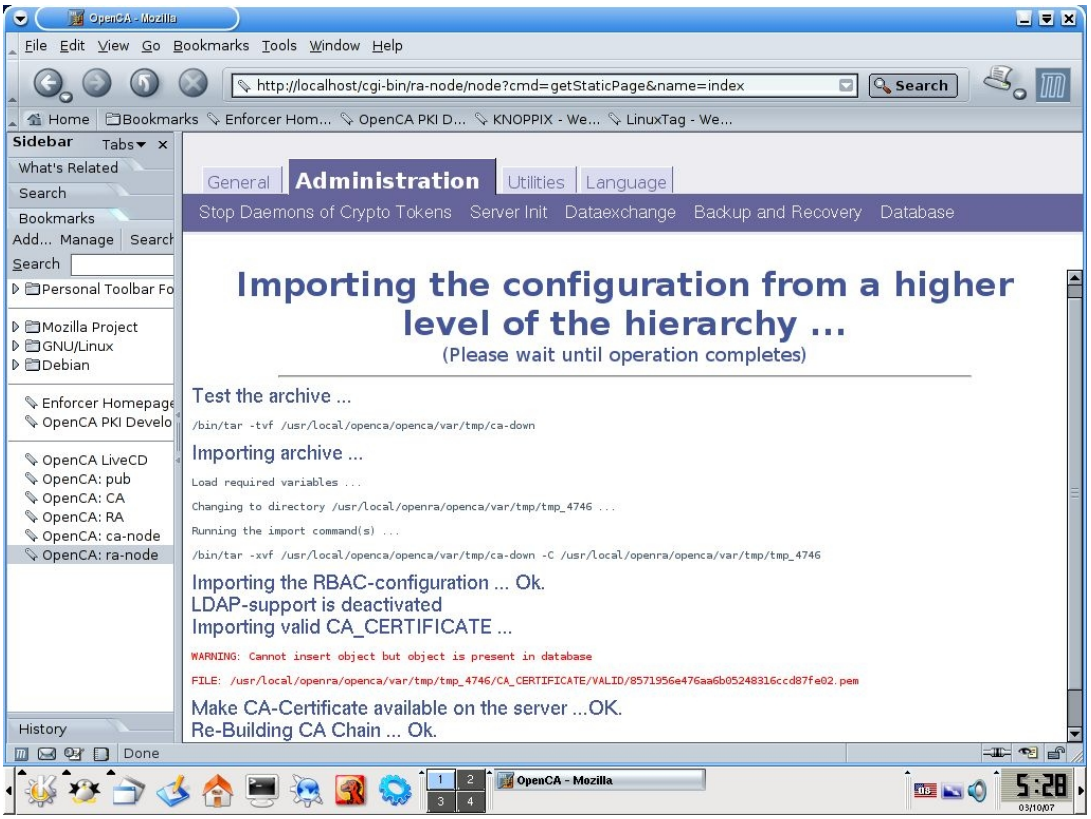


Και επιλέγουμε την καρτέλα **Administration** και από την μπάρα την επιλογή **Server Init** εμφανίζεται ένα παράθυρο με τίτλο **Init New Node** και δύο επιλογές



19. Επιλέγουμε “**Import Configuration**” κάτω από το **PKI Setup,** θα πρέπει να μας εμφανισθεί ένα επιτυχές μήνυμα αφού πρώτα θα μας έχει προτρέψει για επιβεβαίωση





Σε αυτό το βήμα το πιστοποιητικό CA διατίθεται στον RA και σε δημόσιους χρήστες.

Στην παραπάνω οθόνη εμφανίζεται ένα μήνυμα με κόκκινα γράμματα "**Cannot insert object but object is present in database**" το οποίο δεν είναι κάτι ανησυχητικό και το παραβλέπουμε.

Η διαδικασία των ρυθμίσεων της αρχής πιστοποίησης έχει ολοκληρωθεί. Μπορούμε να κλείσουμε το σύστημα (power off από γραμμή εντολών ή log off από το είκονίδιο K του desktop) ή να το αφήσουμε ανοιχτό και να συνδεθούμε σαν χρήστες πια στην αρχή πιστοποίησης ώστε να πάρουμε ένα πιστοποιητικό.

4. Δημιουργία πιστοποιητικού ssl για χρήση σε Server

Υπάρχουν δύο είδη πιστοποιητικών, για clients (πελάτες) και για Servers (εξυπηρετητές). Τα πιστοποιητικά σε γενικές γραμμές είναι παρόμοια όμως σε αυτό που διαφέρουν είναι ο τρόπος που δημιουργείται το ιδιωτικό κλειδί. Σε αυτό το κεφάλαιο θα δούμε πως μπορούμε να δημιουργήσουμε ένα πιστοποιητικό συμβατό για χρήση με τον webserver Apache (2.0.59-Openssl_0.9.8d-Win32) σε ένα συστήμα windows xp, το λειτουργίκο σύστημα στο οποίο θα εγκαταστήσουμε τον Apache μπορεί να είναι οποιοδήποτε λειτουργίκο σύστημα αφού υποστηρίζεται σχεδόν απο όλα τα λείτουργικα συστήματα. Για να το επιτύχουμε αυτό θα πρέπει να δημιουργήσουμε ένα ιδιωτικό κλειδί σαν τμήμα μιας "Αίτηση υπογραφής πιστοποιητικού" (Certificate signing request) χρησιμοποιόντας το Openssl.

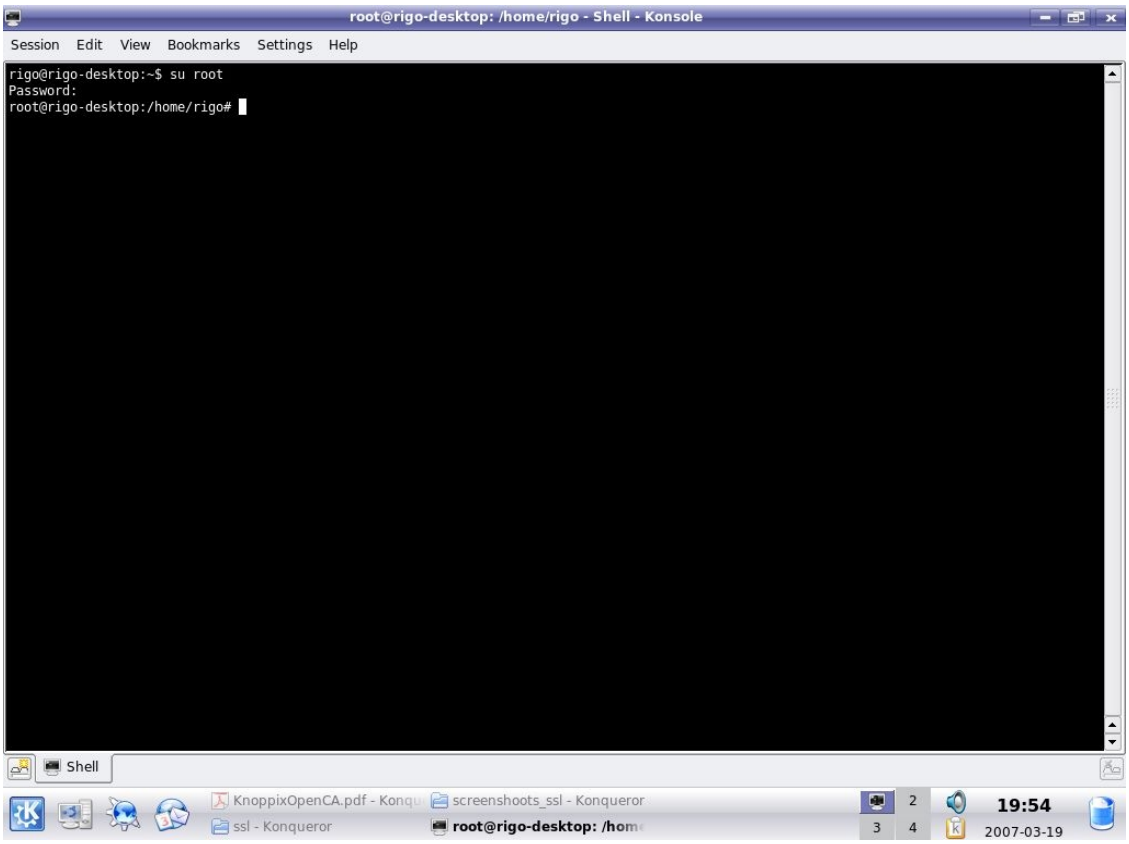
Στην περίπτωση μας χρησιμοποιήσαμε την έκδοση Openssl 0.9.7ι και την εγκαταστήσαμε σε ένα σύστημα Linux με διανομή kubuntu. Μόλις βγεί το πιστοποιητικό και το ιδιωτικό κλειδί, θα τα πάρουμε και θα τα μεταφέρουμε στο λειτουργικό σύστημα Windows, απο το οποίο θα κάνουμε τις απαραίτητες ενέργειες ώστε να πιστοποιηθεί το πιστοποιητικο μας απο το OpenCA. Επιλέξαμε δύο διαφορετικά λειτουργικά συστήματα ώστε να δείξουμε ότι δεν υπάρχουν προβλήματα ασυμβατότητας. Το πιστοποιητικό όταν βγεί απο το OpenCA θα είναι έτοιμο για χρήση.

Όταν περάσουμε το πιστοποιητικό μας και ενεργοποιήσουμε τον Apache, ο οποίος θα έχει εγκατασταθεί σε λειτουργικό σύστημα windows xp, θα έχουμε επιτύχει να συνδεθούμε σε κανάλι επικοινωνιας ssl με τον server μας (https) και θα μπορούμε να παρατηρήσουμε τα στοιχεία του πιστοποιητικού καθώς και ένα λουκετακί στο δεξί κάτω μέρος του browser μας.

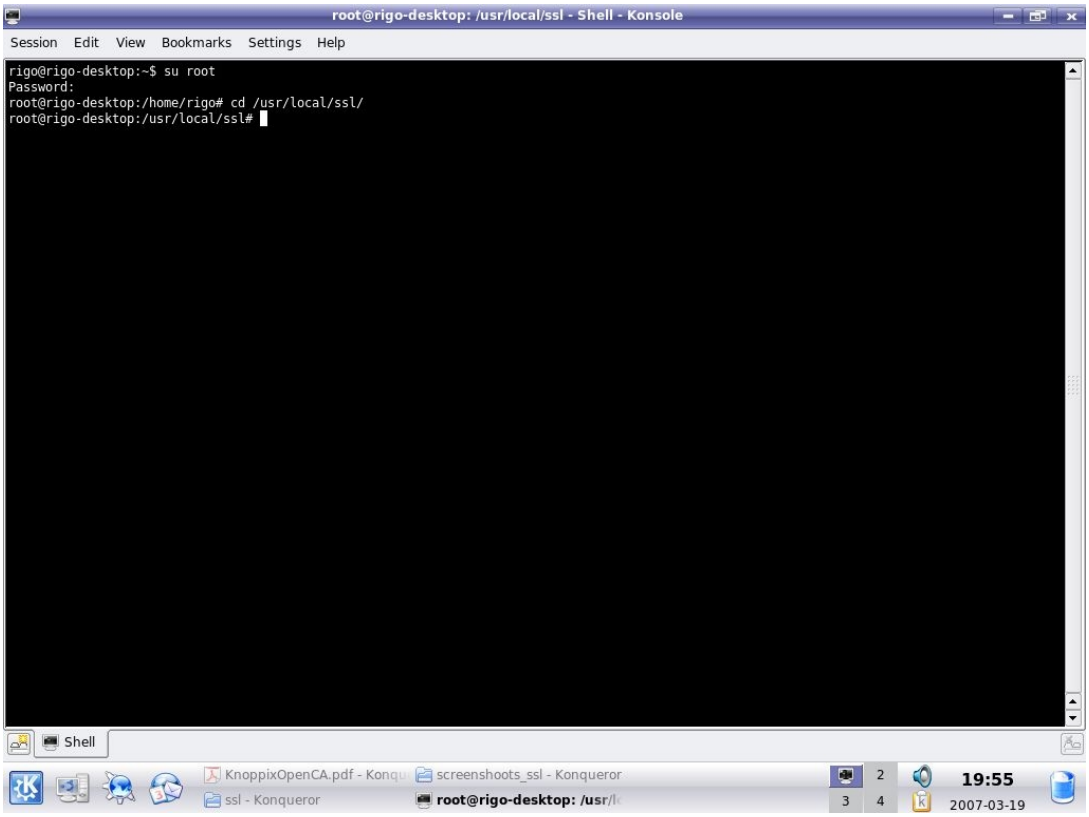
4.1 Δημιουργία μίας Αίτησης υπογραφής πιστοποιητικού (CSR)

Ξεκινόντας θα πρέπει να δημιουργήσουμε μια "Αίτηση υπογραφής πιστοποιητικού" από το Openssl και στην συνέχεια θα την επικυρώσουμε από το OpenCa. Λόγω του ότι η διανομή Kubuntu είναι εγκατεστημένη σε διαφορετική τοποθεσία και δεν έχει άμεσσα πρόσβαση με την αρχή πιστοποίησης μας, θα πρέπει να μεταφέρουμε το .csr αρχείο που θα πάρουμε από το openssl και να το πάμε σε ένα υπολογιστή που "βλέπει" απευθείας το Openca.

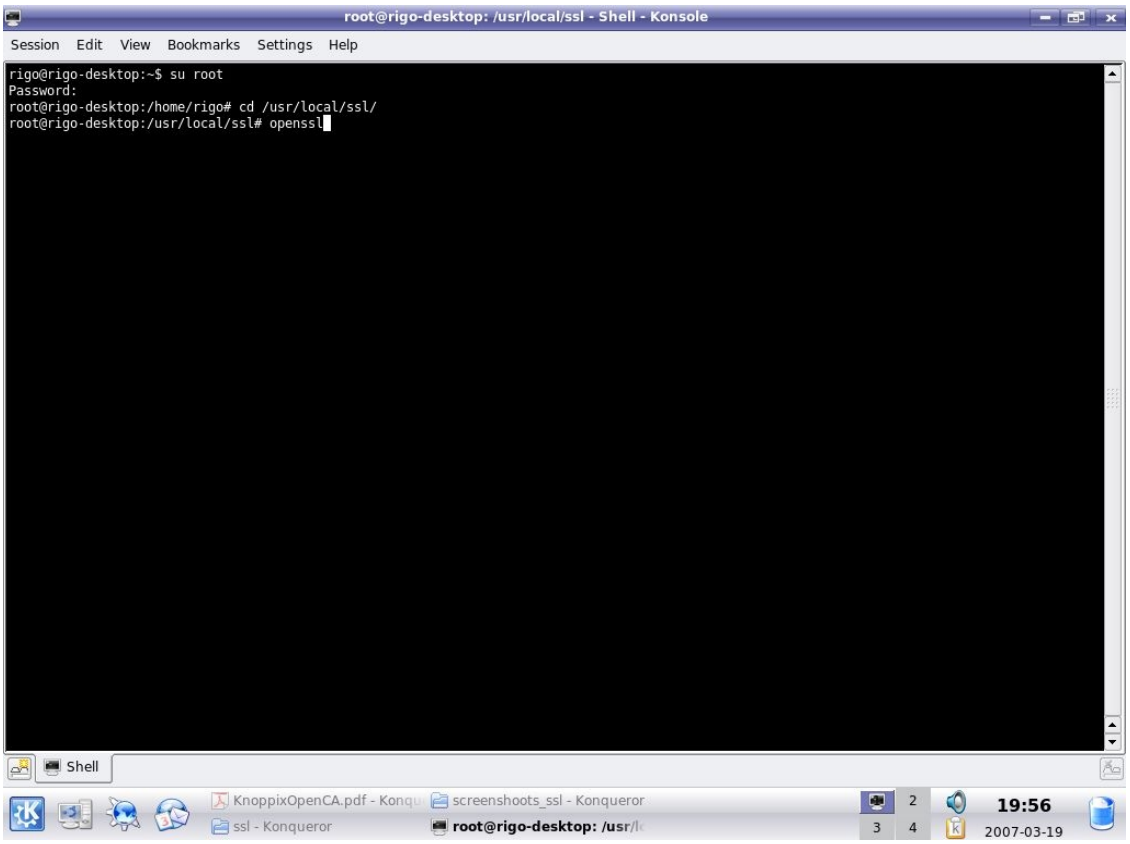
Αρχικά θα πρέπει να μπούμε στο σύστημα Linux (διανομή Kubuntu) και μέσω ενός terminal να τρέξουμε το openssl. Για να το επιτύχουμε αυτό θα πρέπει να είμαστε administrator. Οπότε με την εντολή su root σε περιβάλλον Kubuntu και δίνοντας το password του root μπαίνουμε στο σύστημα σαν administrator, όπως βλέπουμε και παρακάτω



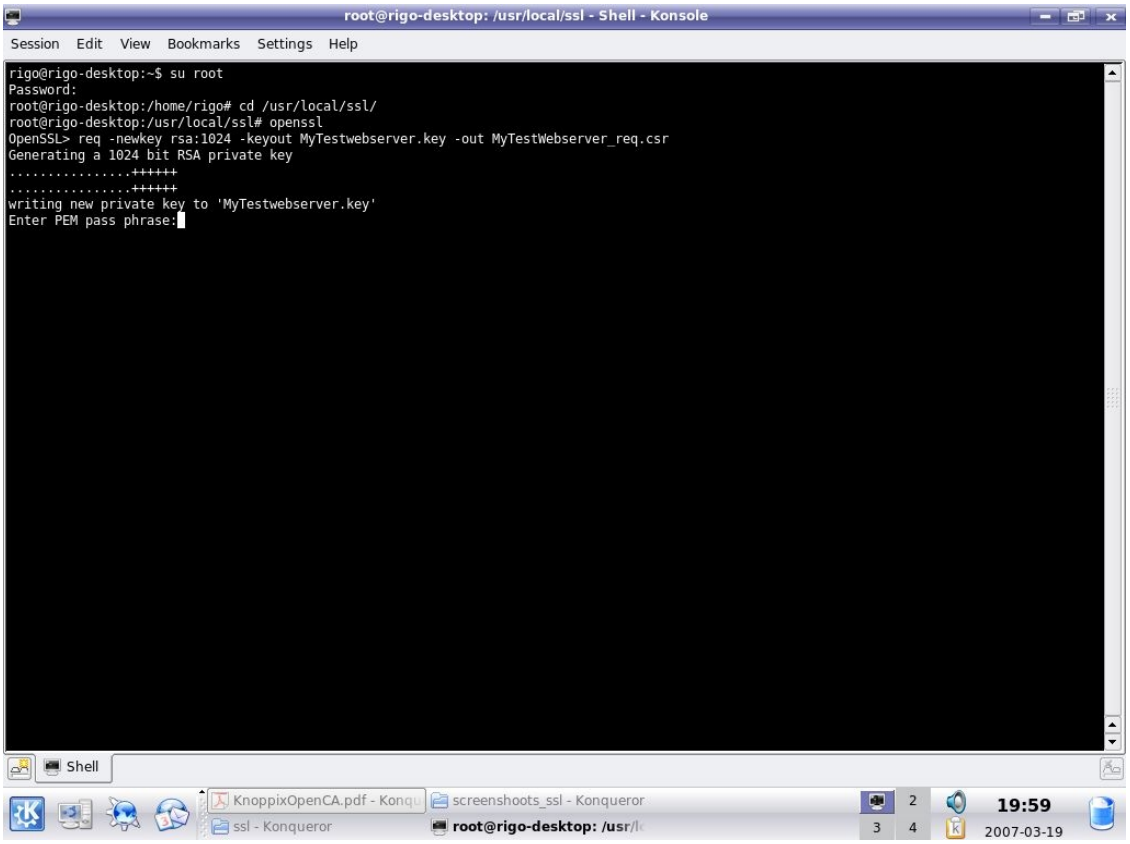
Στη συνέχεια πηγαίνουμε στο path όπου έχει εγκατασταθεί το openssl



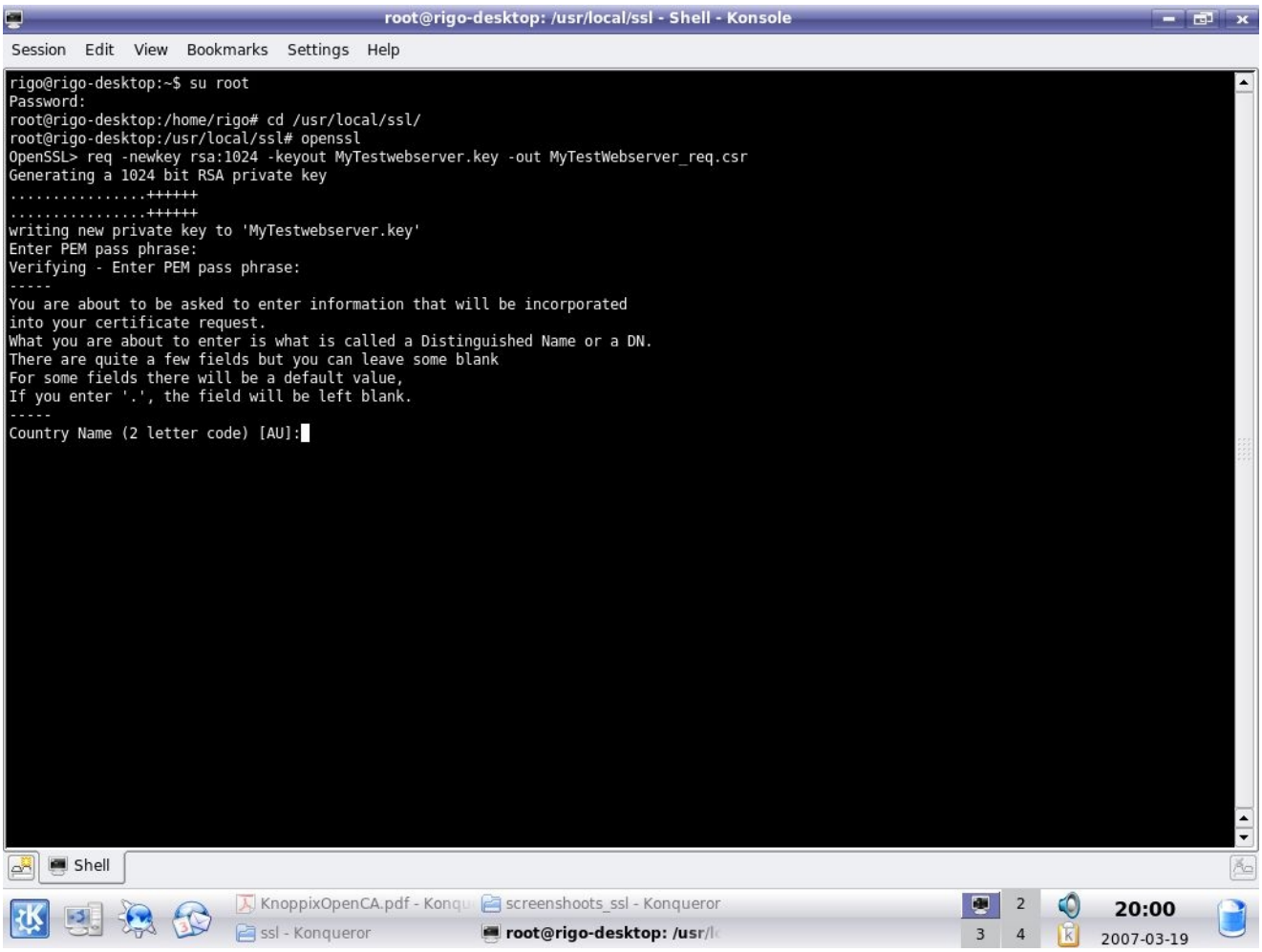
και δίνουμε την εντολή openssl



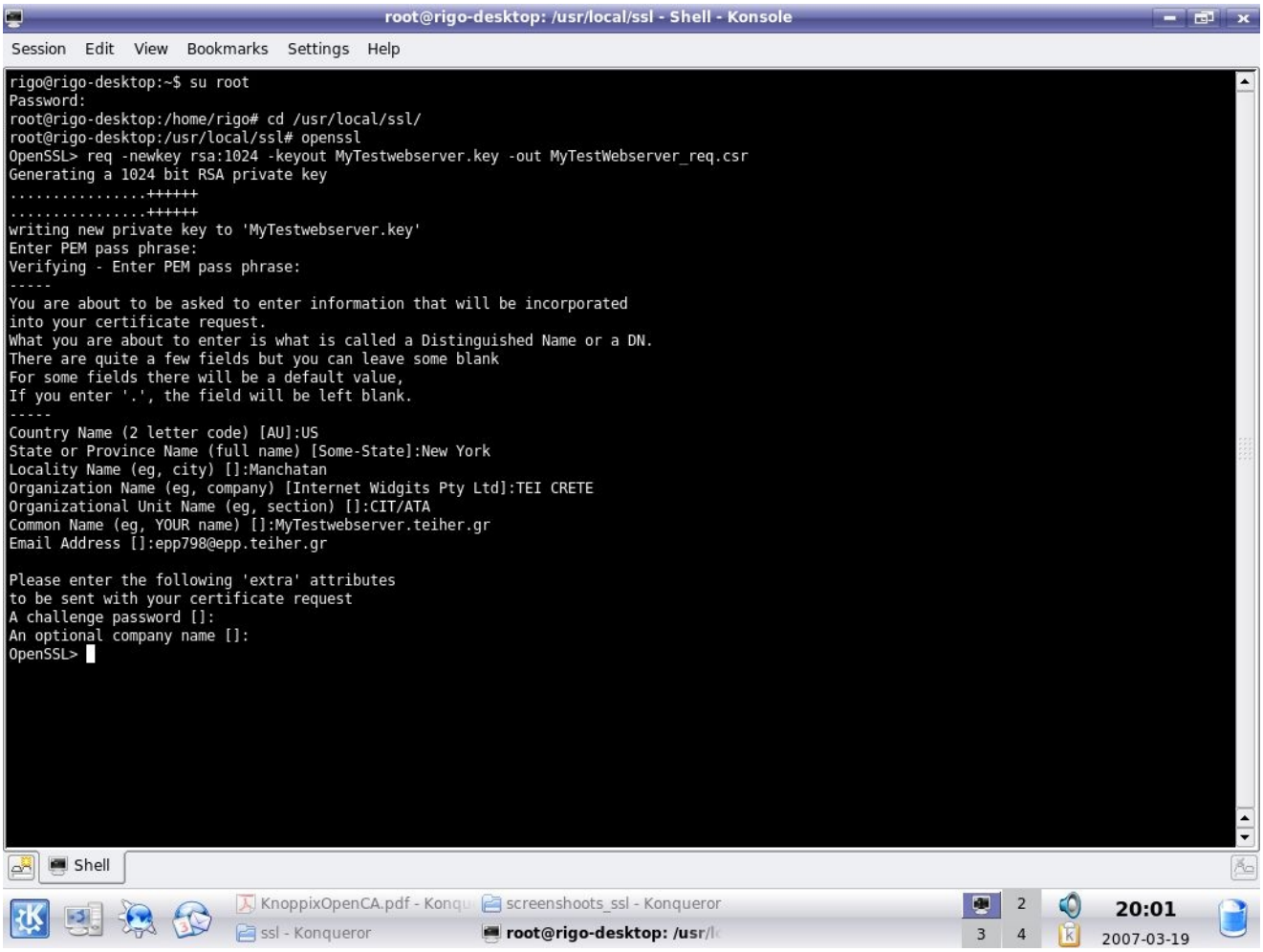
Τώρα άφου είμαστε μέσα στο ssl δίνουμε την εντολή παρακάτω εντολή



Με αυτή την εντολή το openssl δημιουργεί ένα νέο CSR και ένα αντίστοιχο ιδιωτικό κλειδί. Το ιδιωτικό κλειδί θα αποθηκευτεί στο αρχείο MyTestWebserver.key και η "Αίτηση υπογραφής πιστοποιητικού" θα αποθηκευτεί στο αρχείο MyTestWebserver_req.csr. Για να δημιουργηθεί το CSR θα πρέπει το openssl να κάνει κάποιες ερωτήσεις. Αρχικά θα πρέπει να δωθεί ένα password για την προστασία του ιδιωτικού κλειδίου το οποίο θα χρειάζεται να το δίνουμε όποτε θέλουμε να έχουμε πρόσβαση στο πιστοποιητικό (π.χ όταν ξεκιναεί ο Apache).

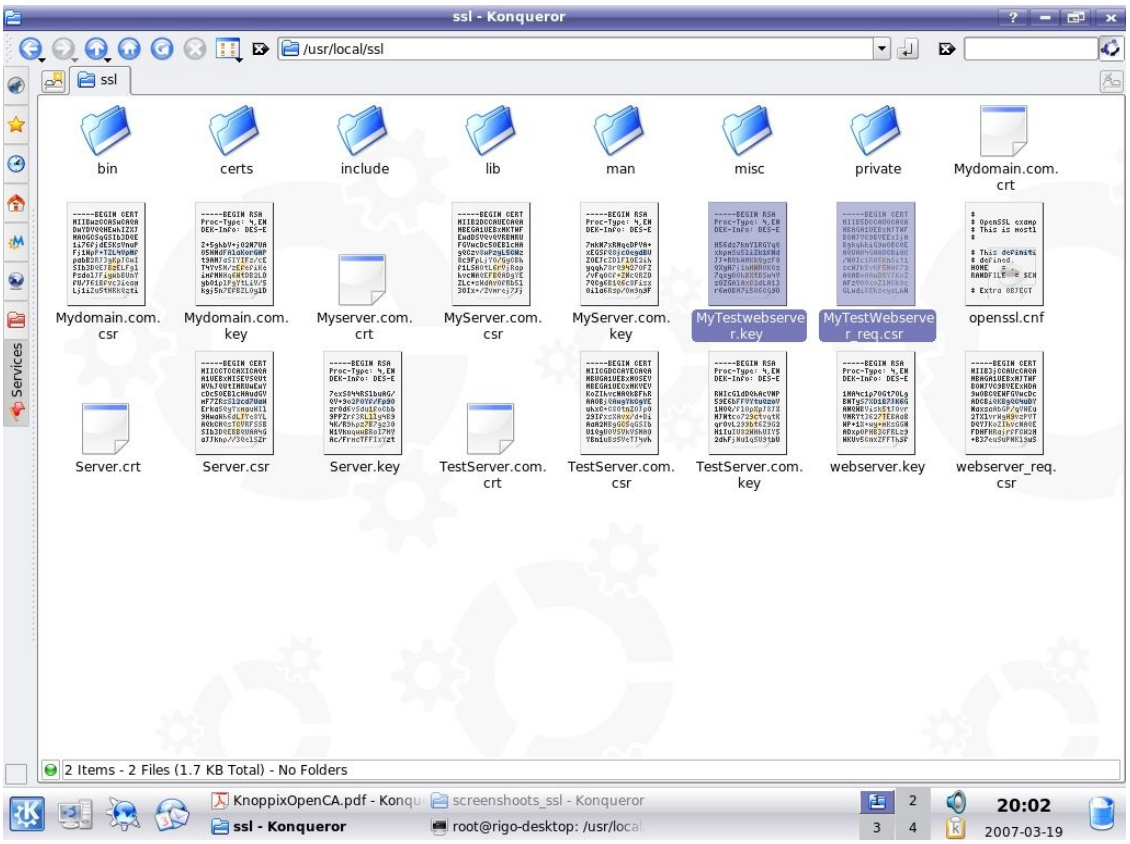


Στη συνέχεια απαντάμε στις παρακάτω ερωτήσεις. Οι απαντήσεις μας θα αποθηκευτούν στο πιστοποιητικό και θα είναι προσβάσιμες σε όποιον θέλει να μάθει περισσότερα στοιχεία για αυτό.



Στο κάτω μέρος υπάρχουν ερωτήσεις οι οποίες αφορούν τον υπεύθυνο για την αίτηση, μπορούμε να μην τις απαντήσουμε. Το “Common Name” θα πρέπει να είναι το ίδιο με το DNS name που έχει ο server μας. Αν ο server μας δεν έχει DNS name και είναι προσβάσιμος με http://localhost τότε το Common Name δεν έχει και τόση σημασία. Ότι και αν συμπληρώσουμε πάντως θα πρέπει να το θυμόμαστε για να το χρησιμοποιήσουμε στο αρχείο httpd.conf του apache.

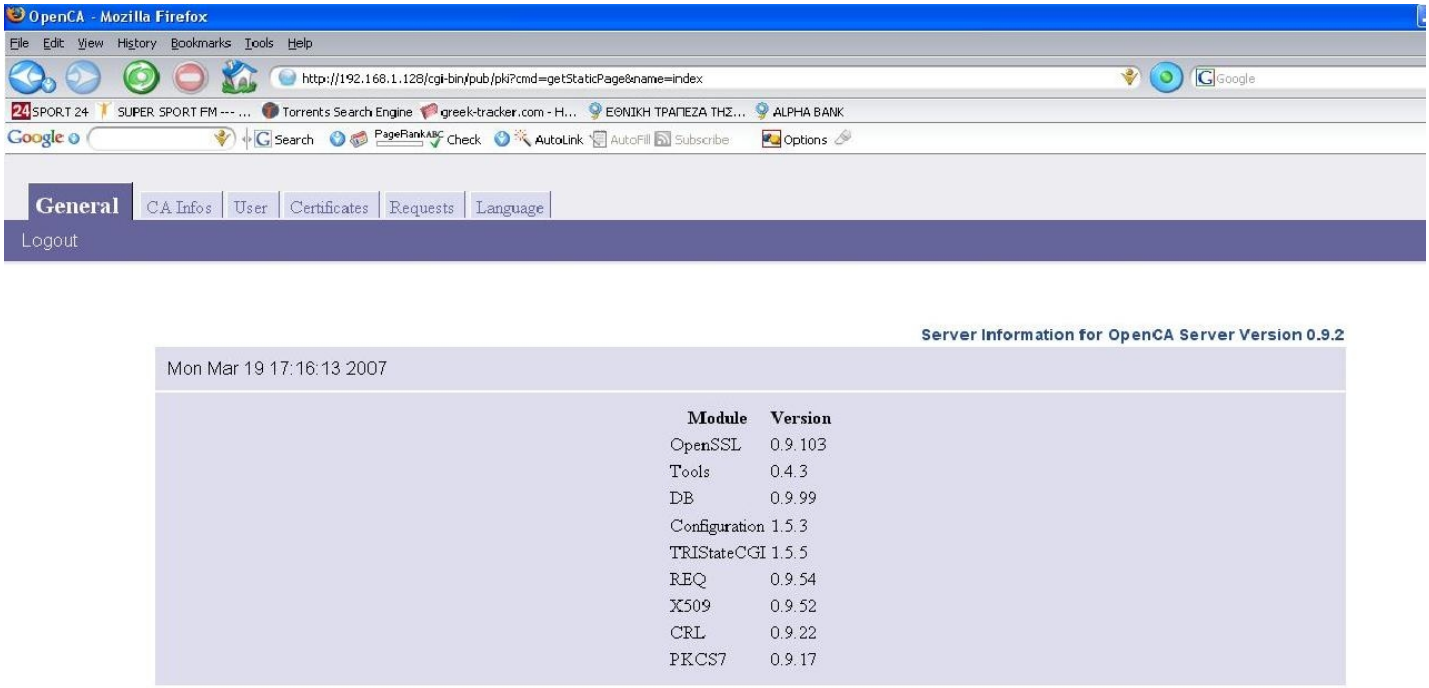
Τελειώνοντας με τα παραπάνω θα περάσουμε στο γραφικό περιβάλλον του Kubuntu, θα πάμε στο path του ssl (/usr/local/ssl) στο οποίο έχει αποθηκευτεί η αίτηση και το ιδιωτικό κλειδί, θα τα αντιγράψουμε και θα τα πάμε στο μηχάνημα με λειτουργικό Windows το οποίο έχει πρόσβαση στην αρχή πιστοποίησης μας (βλέπει το τοπικό δίκτυο) έτσι ώστε να συνεχίσουμε την διαδικασία από το OpenCA.



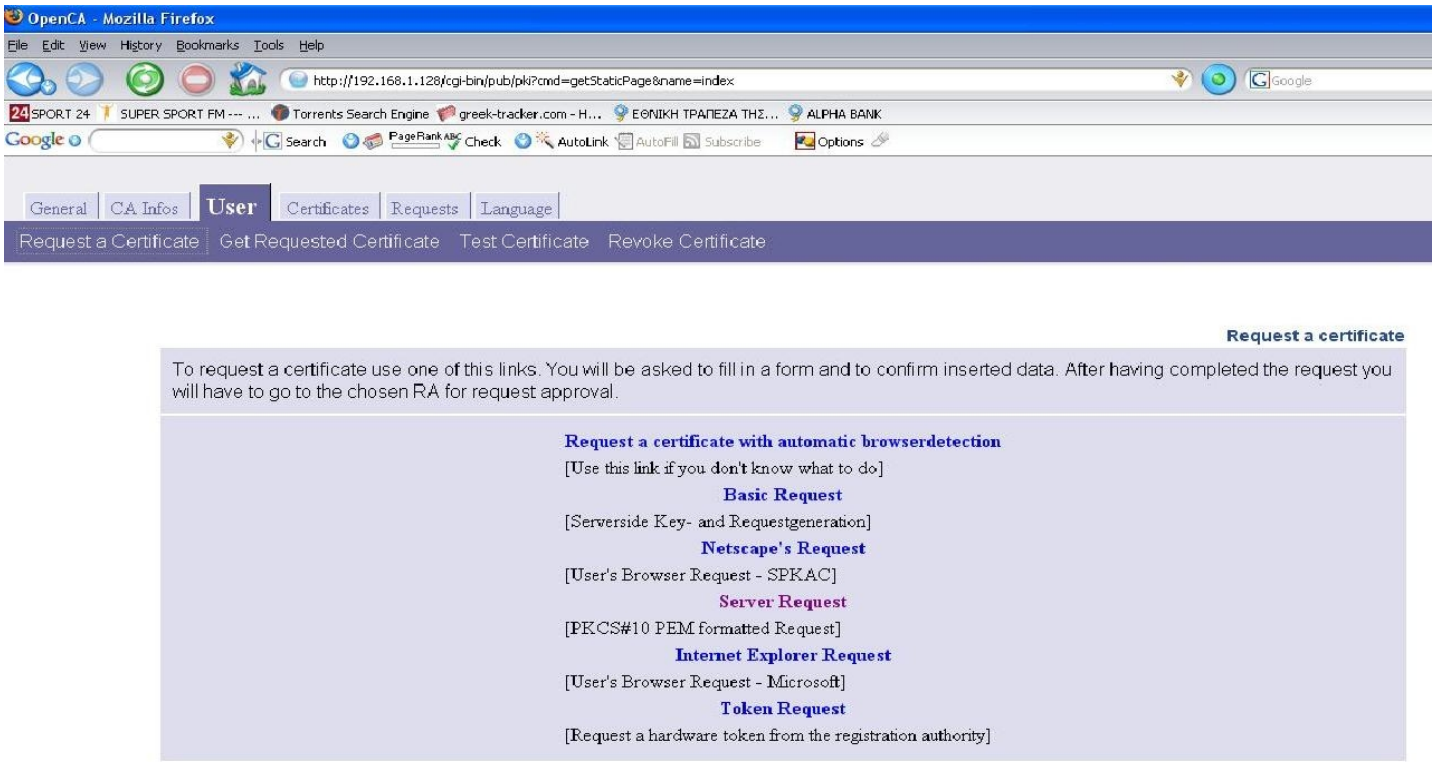
4.2 Χρησιμοποιόντας το CSR ώστε να ζητησόυμε ένα πιστοποιητικό ssl

Αρχίζοντας θα πρέπει να γνωρίζουμε την ip του σύστηματος στο οποίο είναι εγκατεστημένη η αρχή πιστοποίησης μας. Μπαίνουμε στο σύστημα Knoppix και με την εντολή ifconfig βλεπούμε την ip η οποία στην περιπτωσή μας είναι 192.168.1.128. Θα χρησιμοποιήσουμε ένα υπολογιστή με λειτουργικό Windows Xp Professional με SP2 και browser τον mozilla version 2.

Έχοντας ανοιχτό το σύστημα με την αρχή πιστοποίησης, ανοίγουμε τον browser μέσα από τα windows και δίνουμε http://192.168.1.128/pub, έτσι έχουμε πρόσβαση στην αρχή πιστοποίησης (τομέας public).

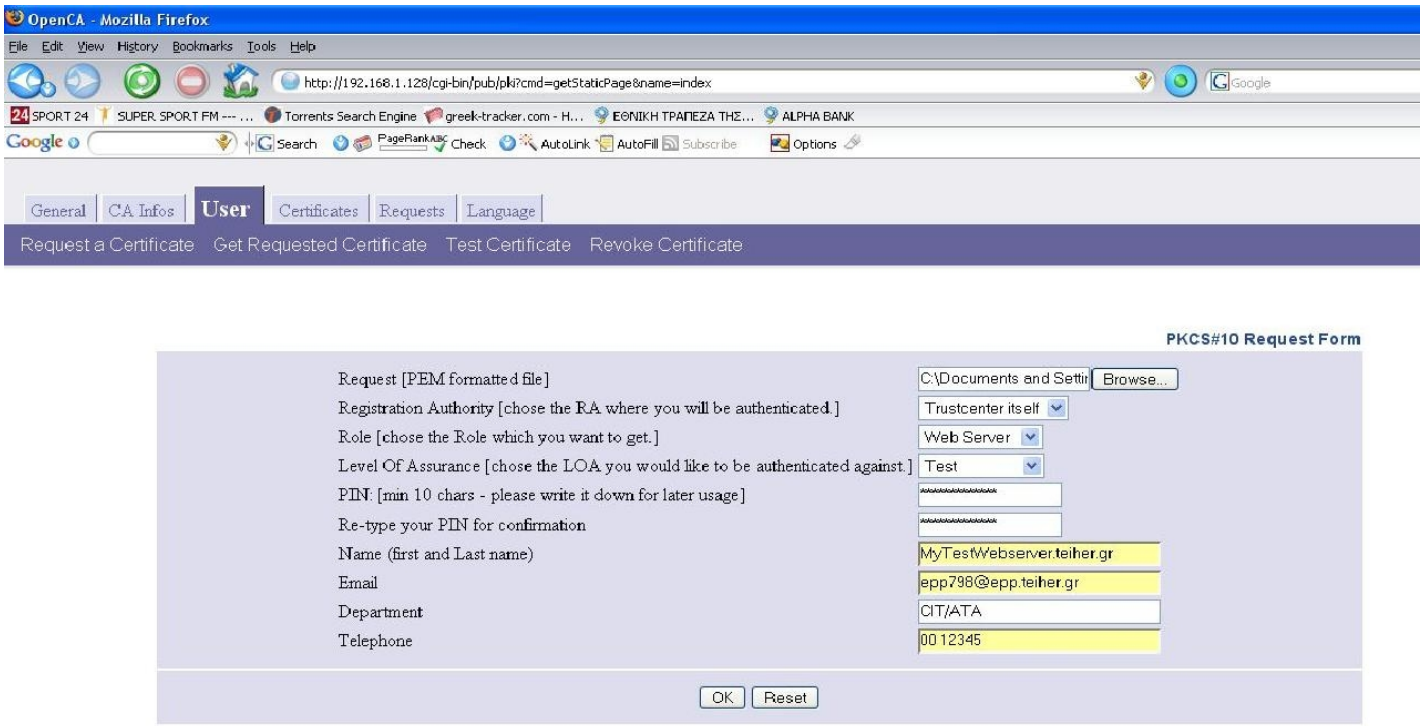


Επιλέγουμε την καρτέλα "**User**" και μετά "**Request a Certificate**"

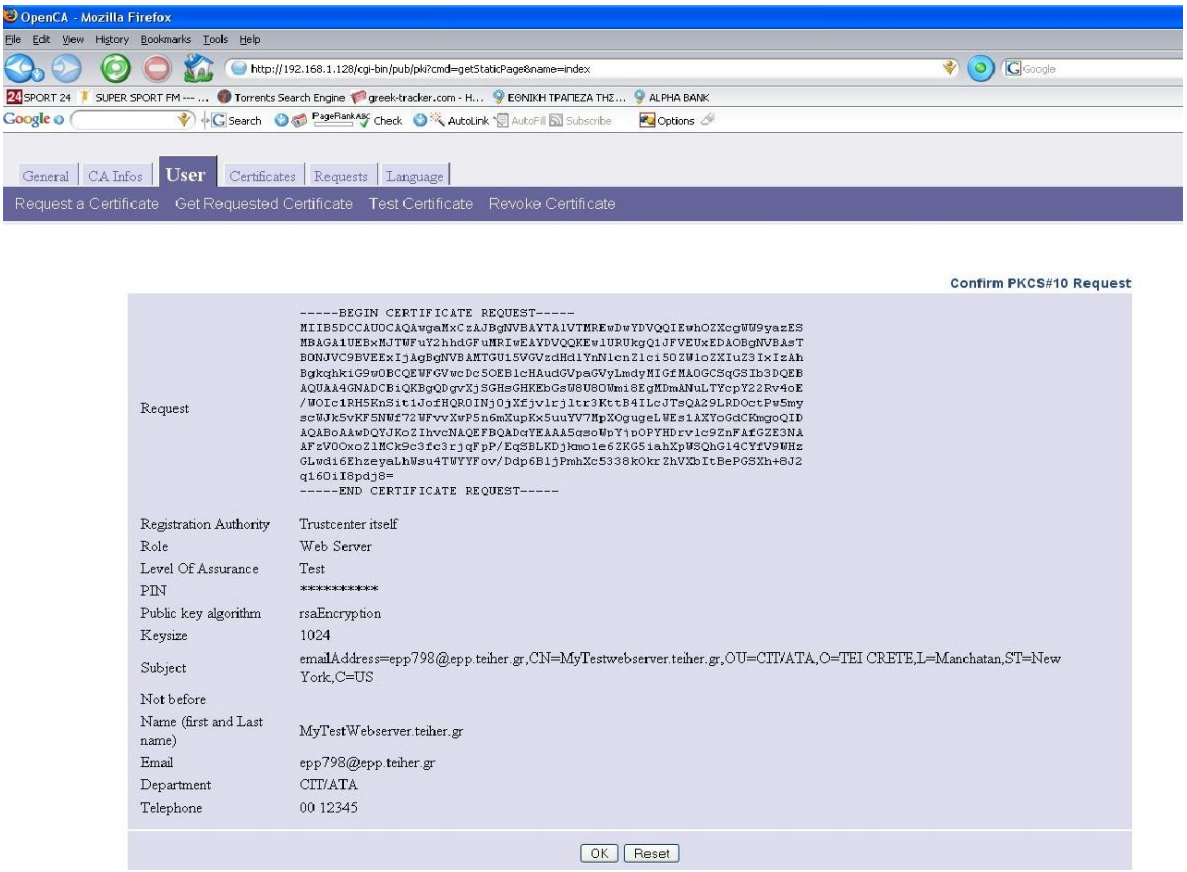


Για να προχωρίσουμε επιλέγουμε "**Server Request**", θα μας εμφανισθεί μια οθόνη στην οποία θα πρέπει να συμπληρώσουμε κάποια πεδία με πληροφορίες σχετικά με την αίτηση μας. Οι ερωτήσεις είναι παρόμοιες με αυτές που απαντήσαμε στο Openssl γι αυτό και οι απαντήσεις μας θα πρεπέι να είναι ίδιες.

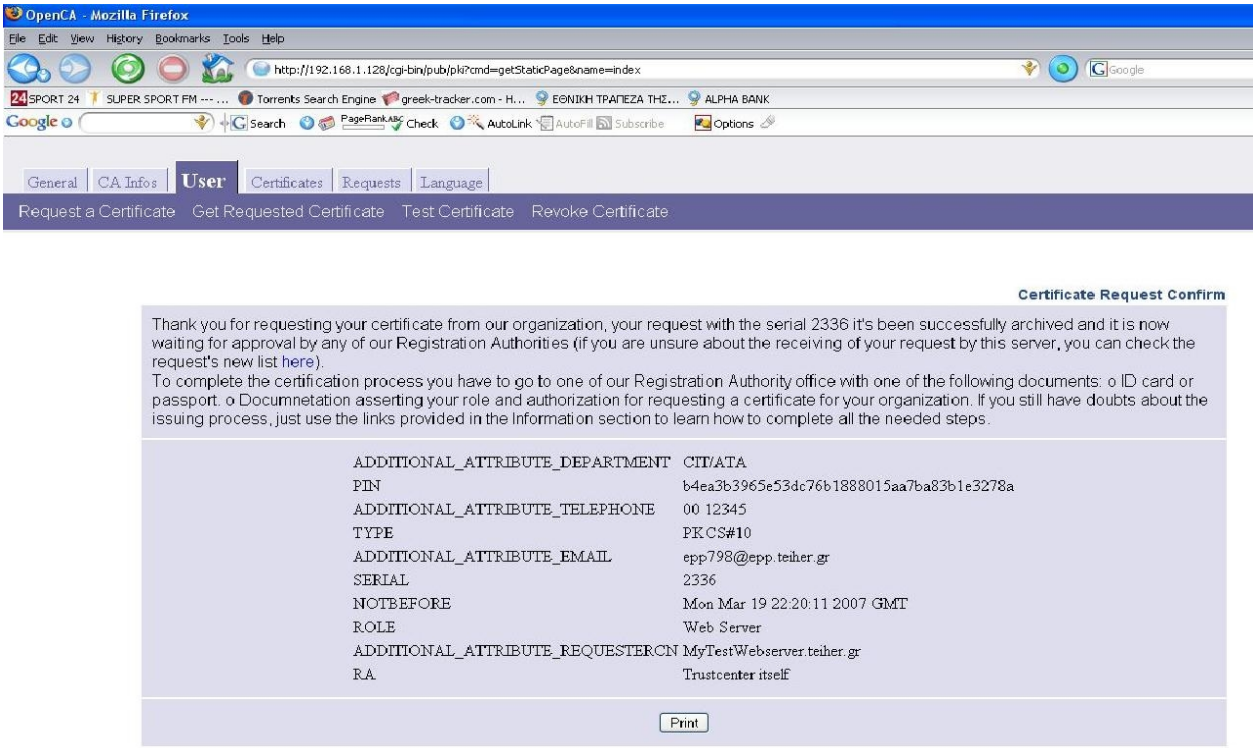
Στο πρώτο πεδίο "**Request [PEM formatted file]**" κάνουμε browse το αρχείο .csr το οποίο δημιουργήσαμε με την βοήθεια του openssl. Αλλάζουμε το "**Role**" σε "**Web Server**", δίνουμε ένα PIN και το επιβεβαιώνουμε, δίνουμε το DNS όνομα (ίδιο με αυτό που είχαμε δώσει στην διαδικασία του openssl) η οθόνη θα πρέπει να είναι κάπως έτσι.



Επιλέγουμε "Ok" και εμφανίζεται μια οθόνη επιβεβαίωσης



Επιλέγουμε “Ok” και εμφανίζεται μια τελική οθόνη επιβεβαίωσης

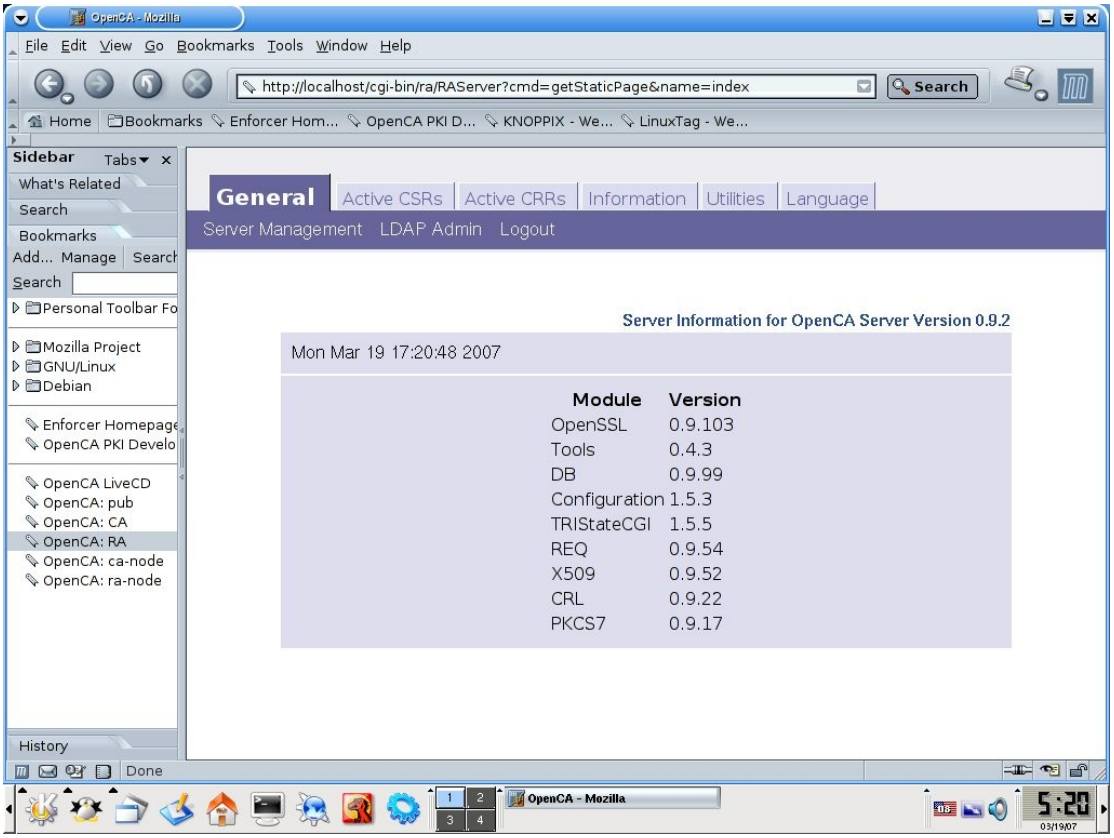


Καταφέραμε να ζητήσουμε ένα πιστοποιητικό για τον webserver μας. Το επόμενο βήμα είναι να πάμε στην άρχη πιστοποίησης να εγκρίνουμε και να εκδώσουμε το

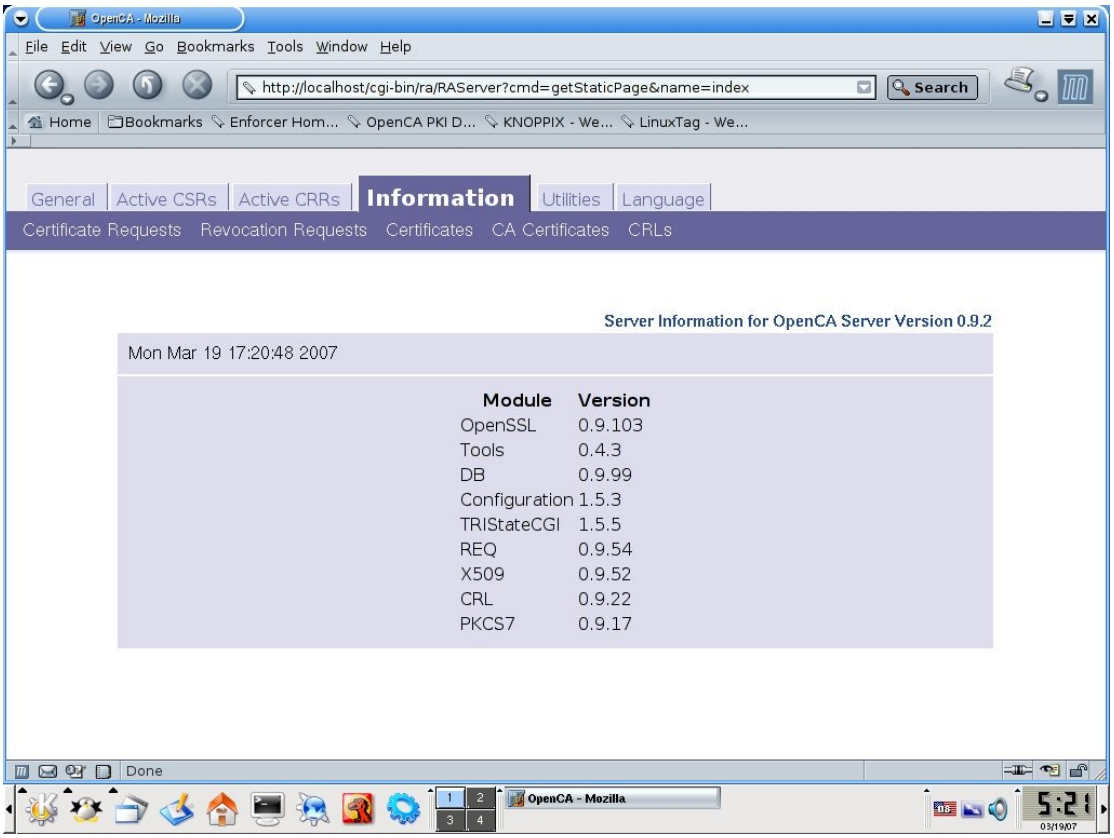
πιστοποιητικό. Πρέπει να σημειώσουμε το serial number (στην περιπτωσή μας 2336) της αιτήσης μας γιατί θα το χρησιμοποιήσουμε στην συνέχεια στη διαδικασία της έκδοσης του πιστοποιητικού.

4.3 Έγκριση της αίτησης πιστοποιητικού

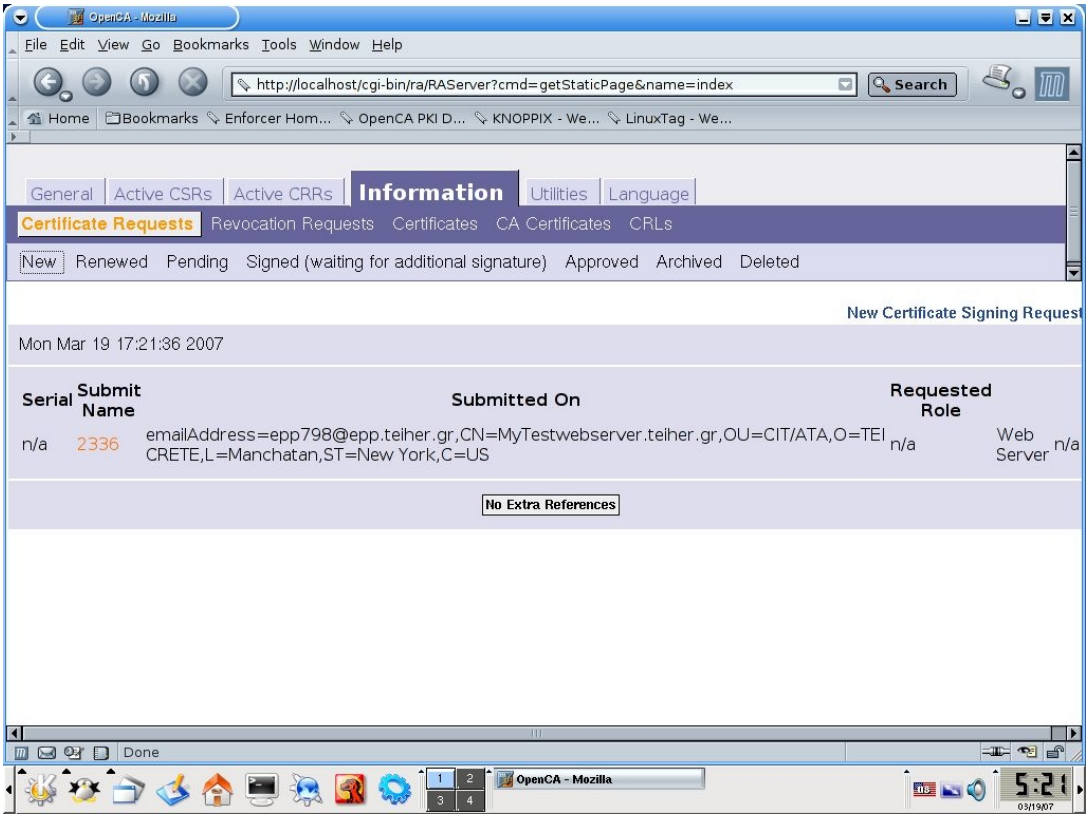
Πηγαίνουμε στον υπολογιστή στον οποίο έχουμε εγκαταστήσει την αρχή πιστοποίησης (OpenCa) και ανοίγουμε τον browser mozilla. Επιλέγουμε από την στήλη στα αριστερά με τα Bookmarks το OpenCA:RA, εμφανίζεται μια οθόνη όπως η παρακάτω



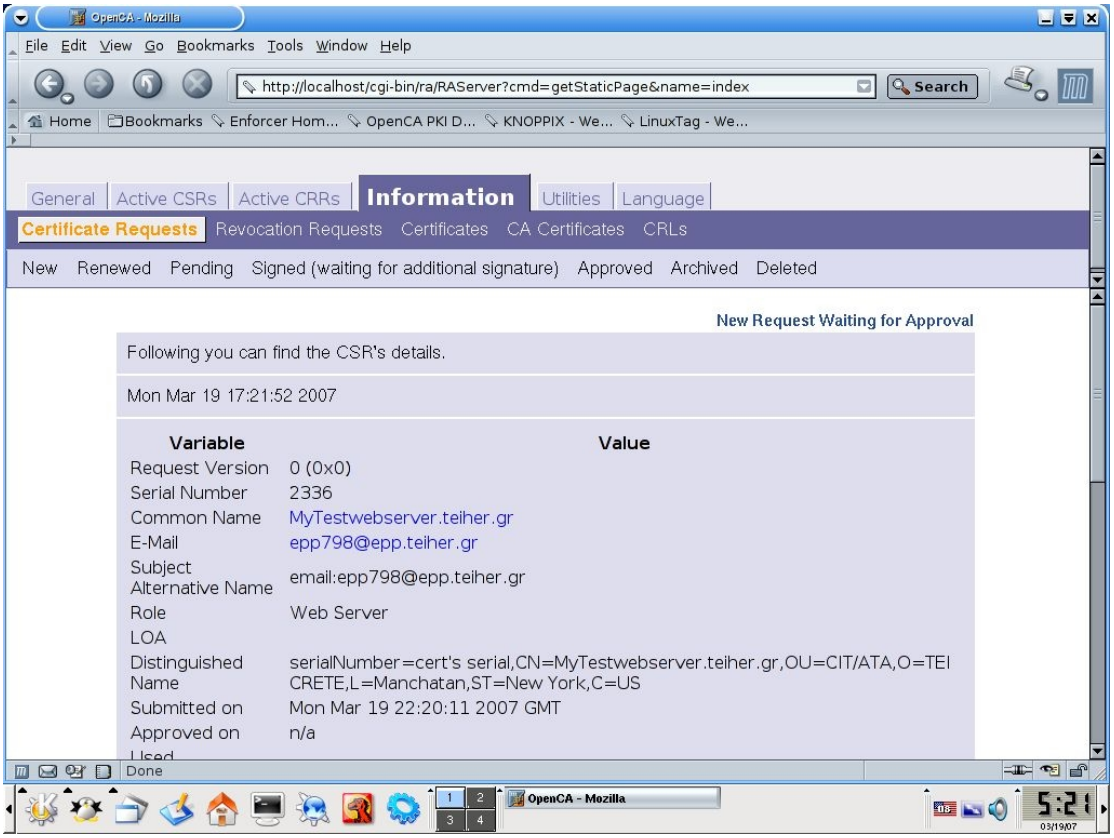
Επιλέγουμε την καρτέλα "**Information**" και βλέπουμε την παρακάτω οθόνη

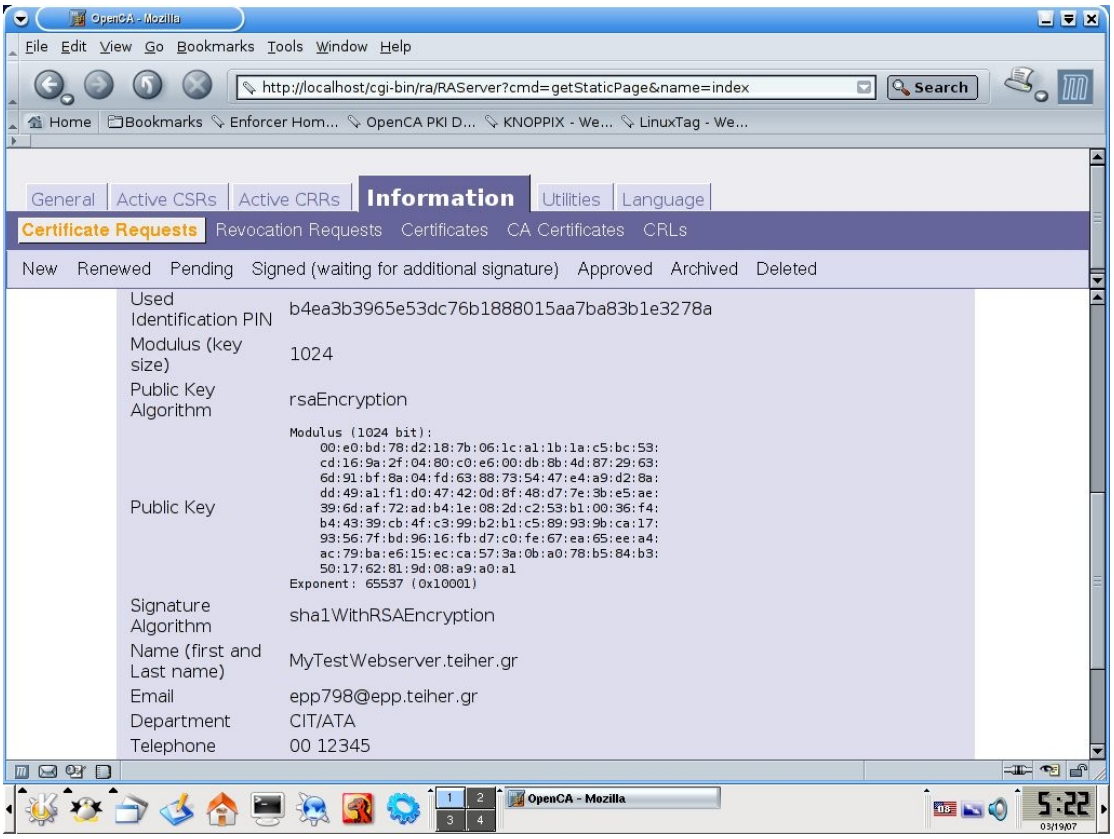


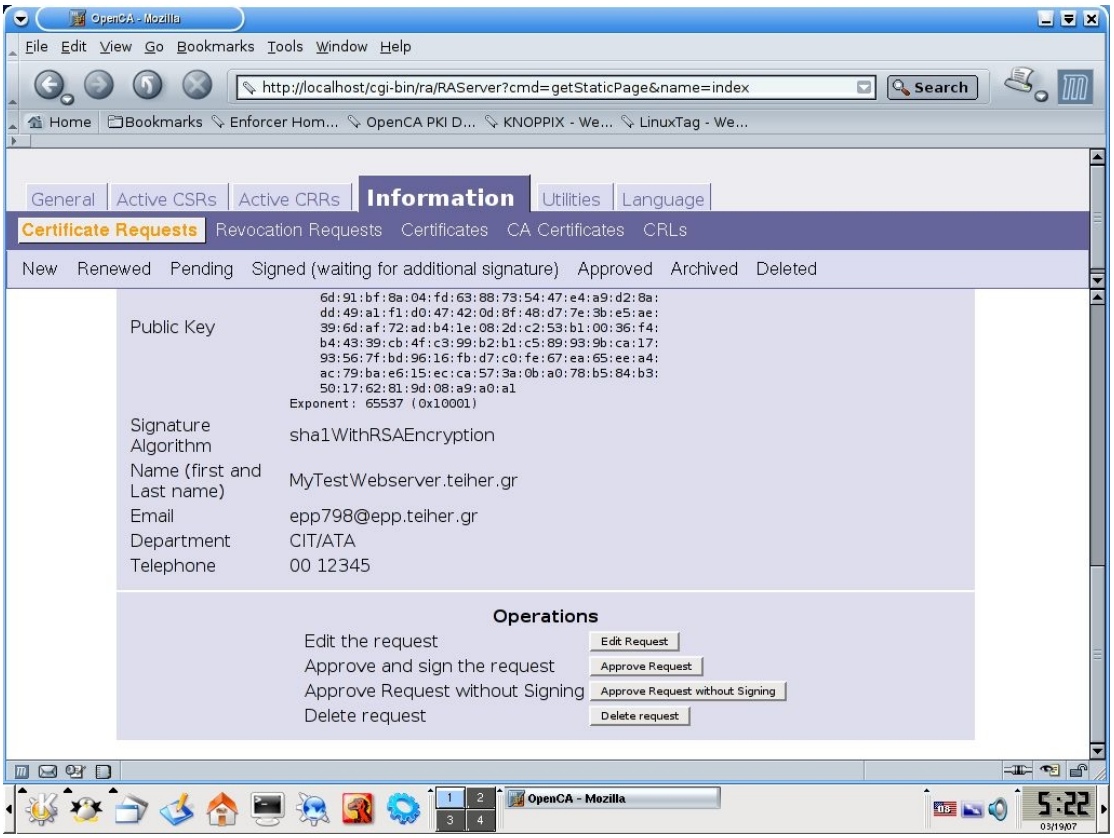
Στη συνέχεια επιλέγουμε "**Certificate Request**" και μετά "**New**", εμφανίζεται η παρακάτω οθόνη



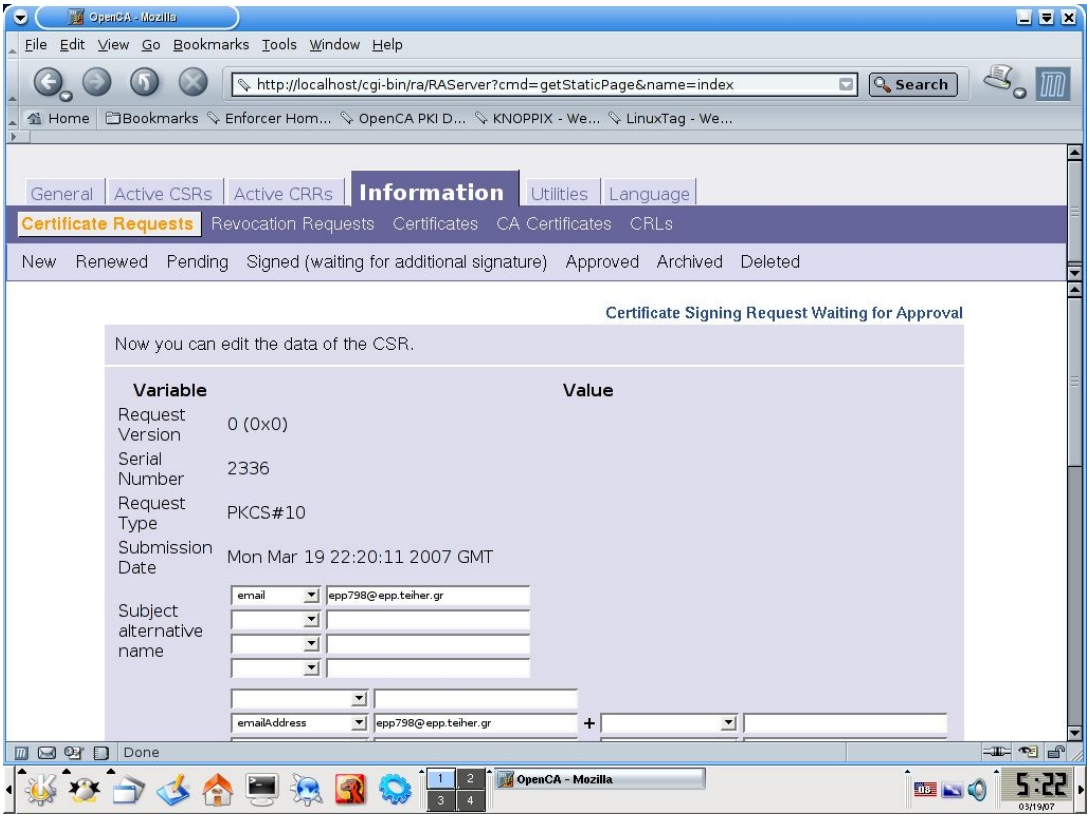
Επιλέγουμε τον κωδικό της αίτησης ο οποίος είναι ο 2336 και εμφανίζεται η παρακάτω οθόνη

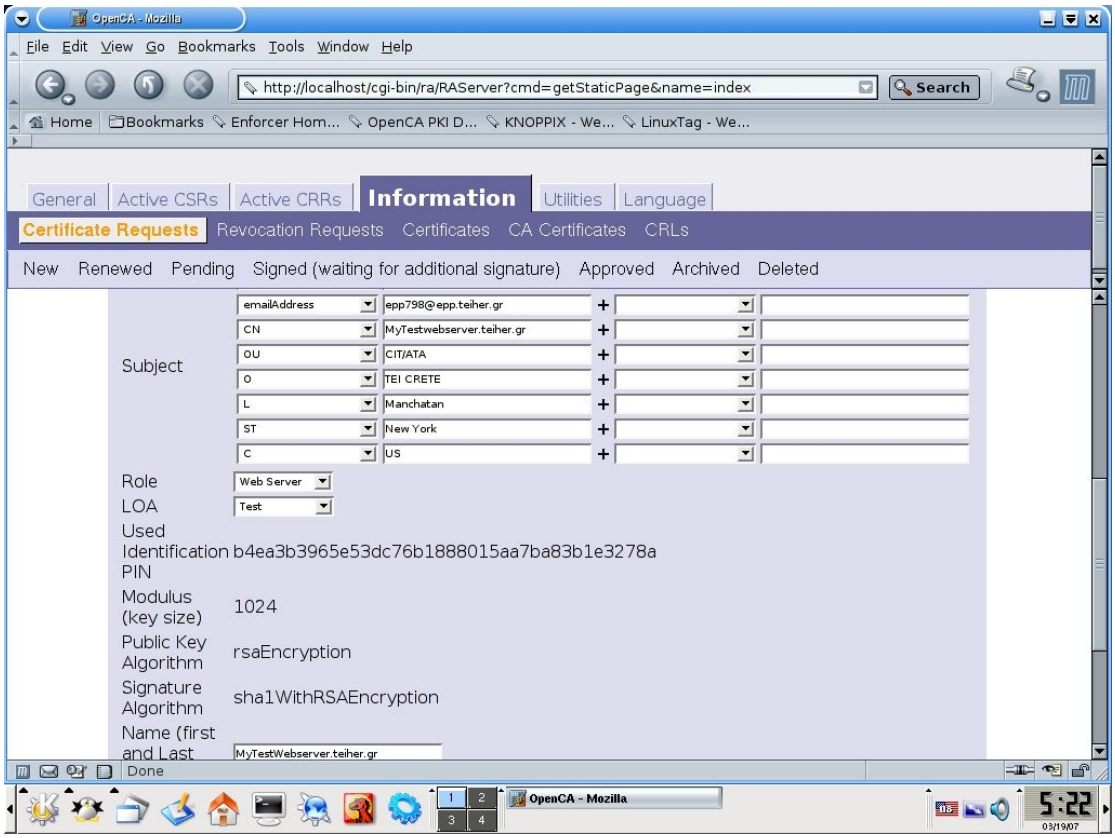


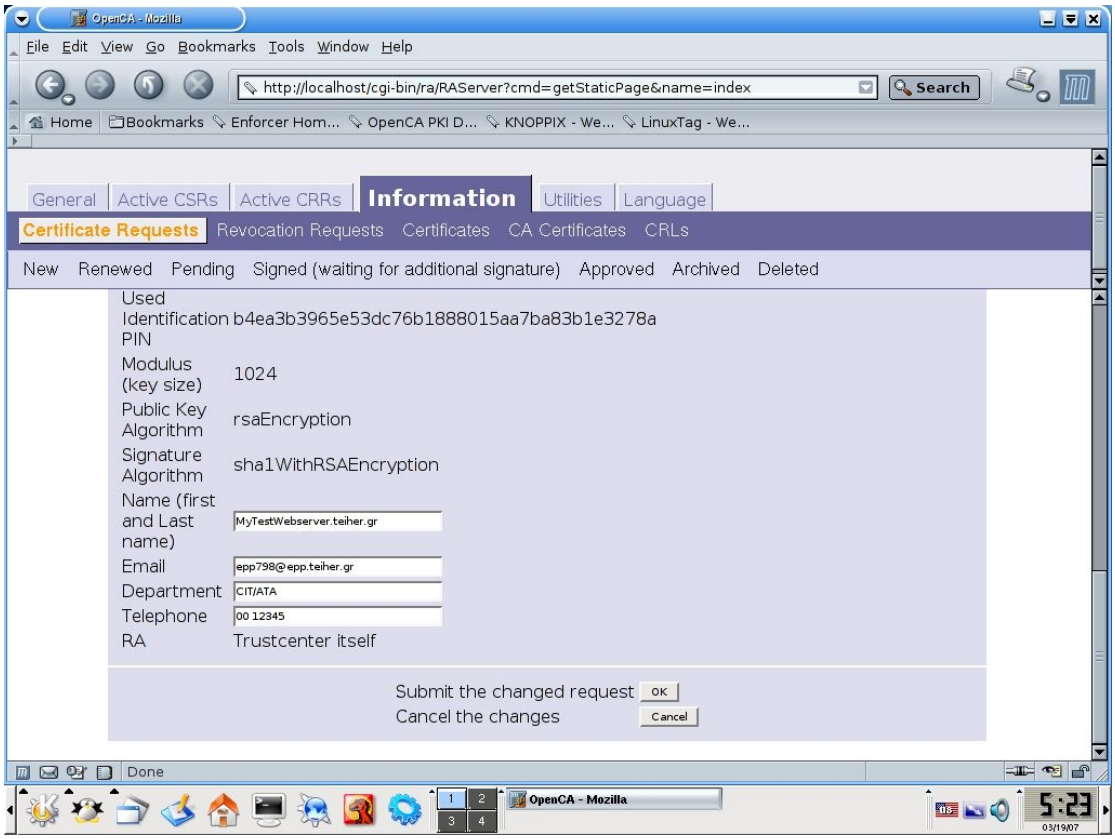




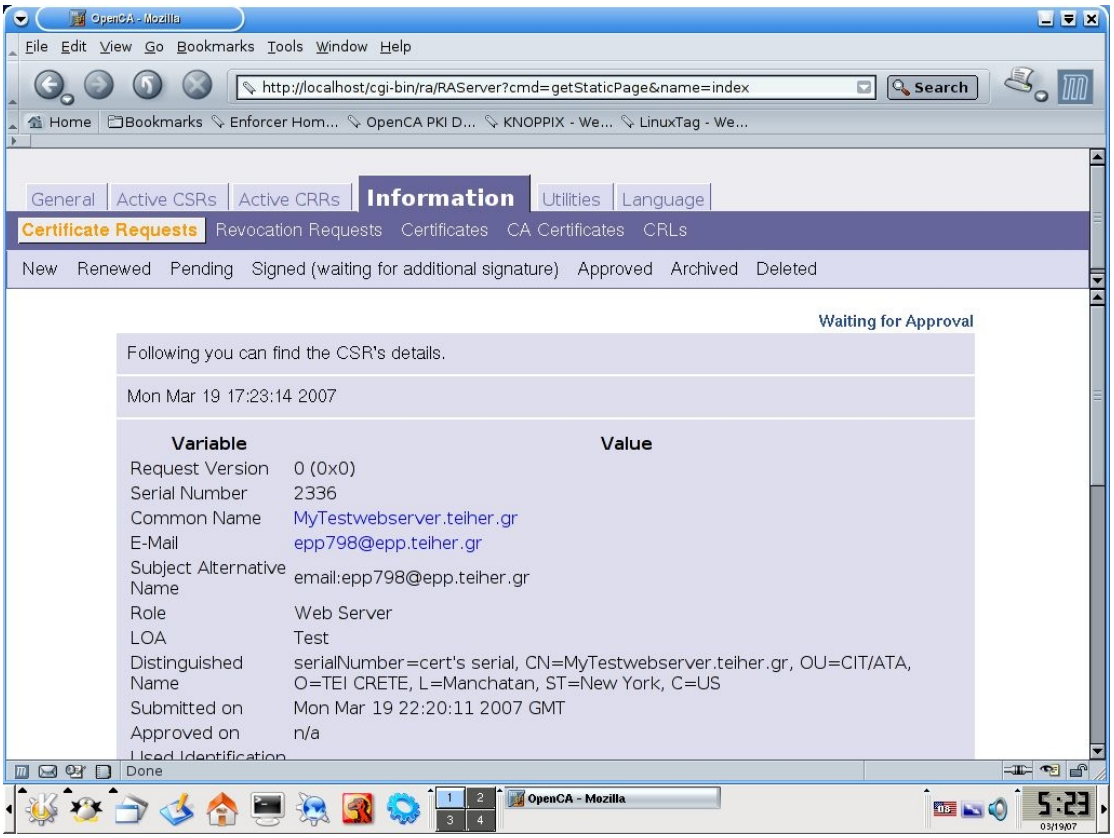
Το επόμενο βήμα της διαδικασίας είναι το "**Edit Request**". Με αυτή την επιλογή μας δίνεται η δυνατότητα να δούμε την αίτηση του χρήστη και αν δεν συμφωνούμε με κάτι να κάνουμε αλλαγές αν κριθεί απαραίτητο.

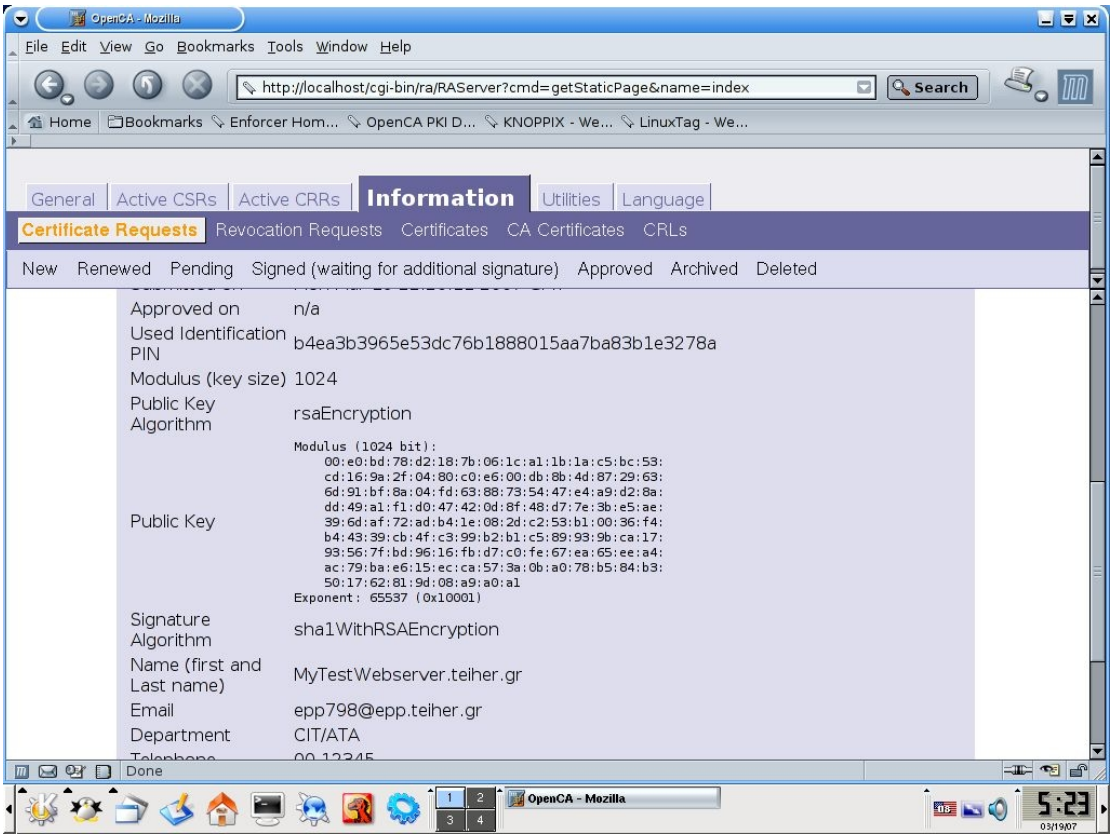


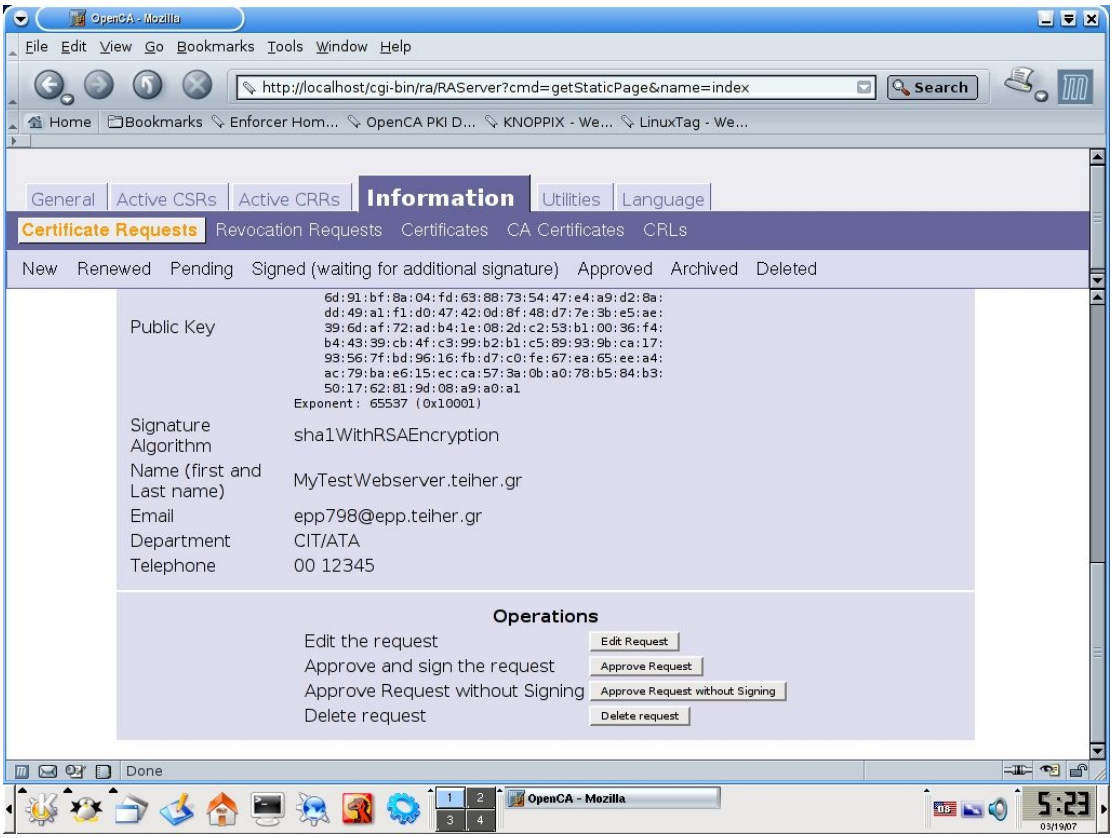




Για να συνεχίσουμε επιλέγουμε “Ok” και γυρίζουμε στην προηγούμενη οθόνη


OpenCA - Mozilla
File Edit View Go Bookmarks Tools Window Help
http://localhost/cgi-bin/ra/RAServer?cmd=getStaticPage&name=index
Search
Home Bookmarks Enforcer Hom... OpenCA PKI D... KNOPPIX - We... LinuxTag - We...
General Active CSRs Active CRRs Information Utilities Language
Certificate Requests Revocation Requests Certificates CA Certificates CRLs
New Renewed Pending Signed (waiting for additional signature) Approved Archived Deleted
Waiting for Approval
Following you can find the CSR's details.
Mon Mar 19 17:23:14 2007
Variable
Value
Request Version
0 (0x0)
Serial Number
2336
Common Name
MyTestwebserver.teiher.gr
E-Mail
epp798@epp.teiher.gr
Subject Alternative Name
email:epp798@epp.teiher.gr
Role
Web Server
LOA
Test
Distinguished Name
serialNumber=cert's serial, CN=MyTestwebserver.teiher.gr, OU=CIT/ATA, O=TEI CRETE, L=Manchatan, ST=New York, C=US
Submitted on
Mon Mar 19 22:20:11 2007 GMT
Approved on
n/a
Done
OpenCA - Mozilla
5:23
03/19/07


OpenCA - Mozilla
File Edit View Go Bookmarks Tools Window Help
http://localhost/cgi-bin/ra/RAServer?cmd=getStaticPage&name=index
Search
Home Bookmarks Enforcer Hom... OpenCA PKI D... KNOPPIX - We... LinuxTag - We...
General Active CSRs Active CRRs Information Utilities Language
Certificate Requests Revocation Requests Certificates CA Certificates CRLs
New Renewed Pending Signed (waiting for additional signature) Approved Archived Deleted
Approved on
n/a
Used Identification PIN
b4ea3b3965e53dc76b1888015aa7ba83b1e3278a
Modulus (key size)
1024
Public Key Algorithm
rsaEncryption
Public Key
Modulus (1024 bit):
00:e0:bd:78:d2:18:7b:06:1c:a1:1b:1a:c5:bc:53:
cd:16:9a:2f:04:80:c0:e6:00:db:8b:4d:87:29:63:
6d:91:bf:8a:04:fd:63:88:73:54:47:e4:a9:d2:8a:
dd:49:a1:f1:d0:47:42:0d:8f:48:d7:7e:3b:e5:ae:
39:6d:af:72:ad:b4:1e:08:2d:c2:53:b1:00:36:f4:
b4:43:39:cb:4f:c3:99:b2:b1:c5:89:93:9b:ca:17:
93:56:7f:bd:96:16:fb:d7:c0:fe:67:ea:65:ee:a4:
ac:79:ba:e6:15:ec:ca:57:3a:0b:a0:78:b5:84:b3:
50:17:62:81:9d:08:a9:a0:a1
Exponent: 65537 (0x10001)
Signature Algorithm
sha1WithRSAEncryption
Name (first and Last name)
MyTestWebserver.teiher.gr
Email
epp798@epp.teiher.gr
Department
CIT/ATA
Done
OpenCA - Mozilla
5:23
03/19/07



Σε αυτό το σημείο είμαστε έτοιμοι να εγκρίνουμε την αίτηση. Το OpenCa μας δίνει δύο επιλογές, "**Approve request**" και μετά την υπογράφουμε με το πιστοποιητικό του Administrator του OpenCa και "**Approve Request without Signing**". Σε επαγγελματικό περιβάλλον θα επιλέξουμε την πρώτη επιλογή. Ωστόσο ο mozilla που παρέχει το live cd δεν προσφέρει το SecCLAB plugin ώστε να επιλέξουμε "**Approve and signing request**".

Το SecCLAB plugin είναι ένα **XPCOM** (Cross Platform Component Object Model) component το οποίο περιλαμβάνει κάποιες συναρτήσεις οι οποίες σχετίζονται με την Αρχή Πιστοποίησης και την συμβατότητα της με τον web browser mozilla. Το παραπάνω plugin ενσωματώνεται στο mozilla και του δίνει τη δυνατότητα να υπογράφει και να πιστοποιεί ψηφιακές υπογραφές οι οποίες παράγονται από εφαρμογές XUL (Openca Live cd), επίσης μπορεί να κωδικοποιεί και να αποκωδικοπεί κρυπογραφημένα μηνύματα από εφαρμογες XUL (Openca Live cd).

Στην δική μας περίπτωση ο Mozilla του OpenCA Live cd δεν έχει αυτό το plugin. Αυτό όμως δεν μας εμποδίζει καθόλου να προχωρίσουμε την διαδικασία μας για την δημιουργία του ψηφιακού πιστοποιητικού.

Επιλέγουμε "**Approve Request without Signing**", θα ζητηθεί να επιβεβαιώσουμε την επιλογή μας και στη συνέχεια εμφανίζεται η παρακάτω οθόνη

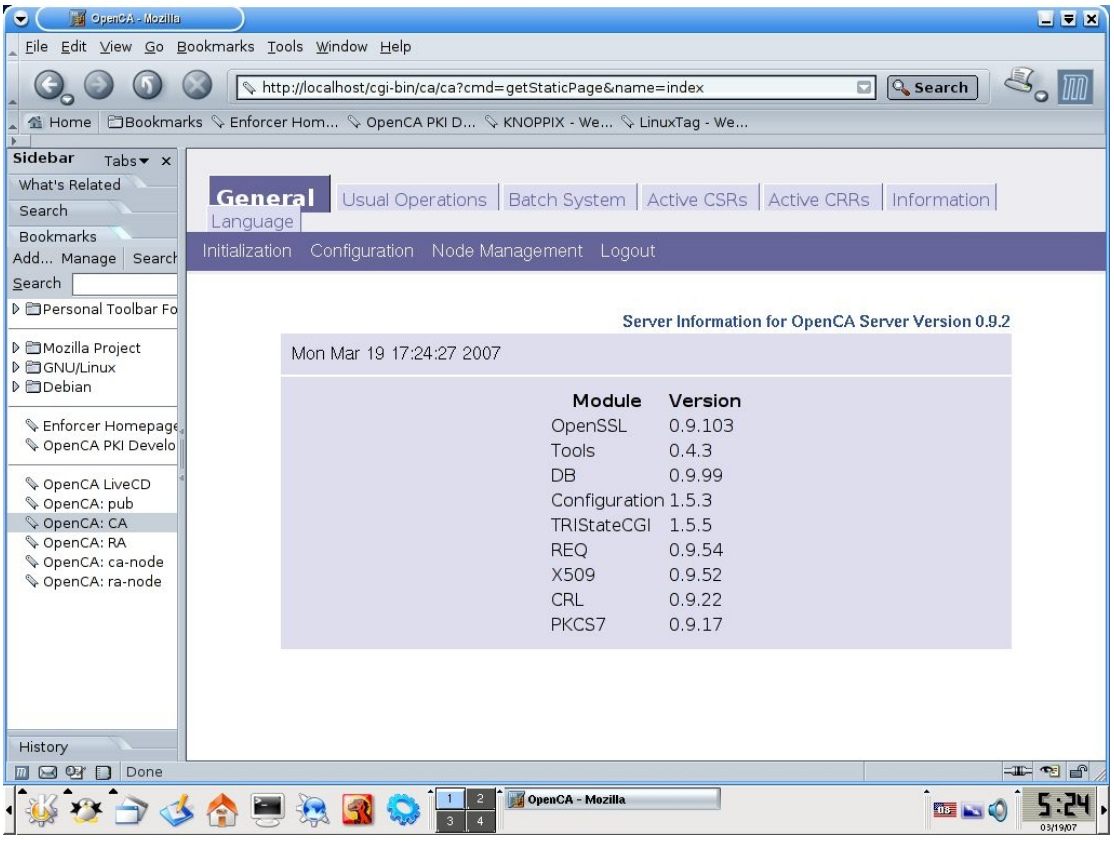
Certificate Signing Request Approved

Certificate Request Successfully approved.
Signature: not available because the request was not signed

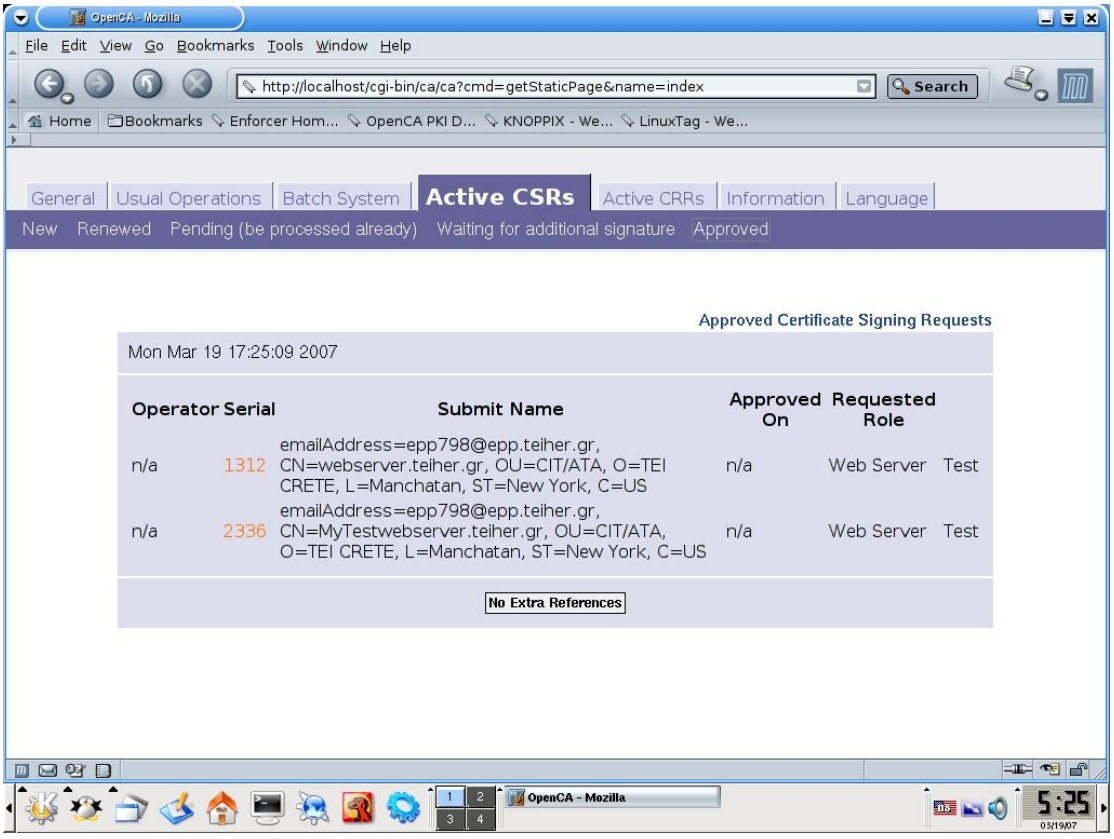
Mon Mar 19 17:24:01 2007

4.4 Έκδοση του πιστοποιητικού

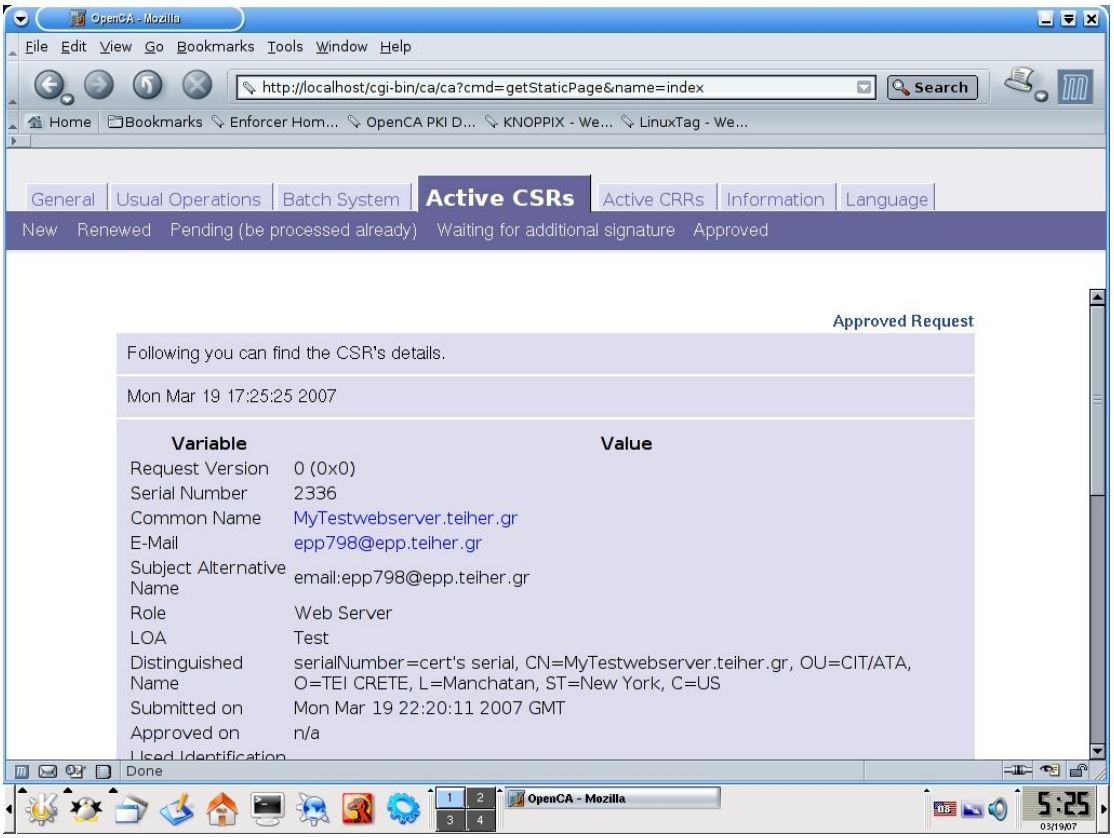
Μέχρι στιγμής η αίτηση που έχει κάνει ο χρήστης για να παραλάβει ένα πιστοποιητικό για χρήση σε Server έχει εγκριθεί. Ήρθε η ώρα να γίνει η έκδοση του από την αρχή πιστοποίησης. Οπότε επιλέγουμε από την αριστερή στήλη του Mozilla με τα Bookmarks το OpecCA:CA, εμφανίζεται κάτι παρόμοιο

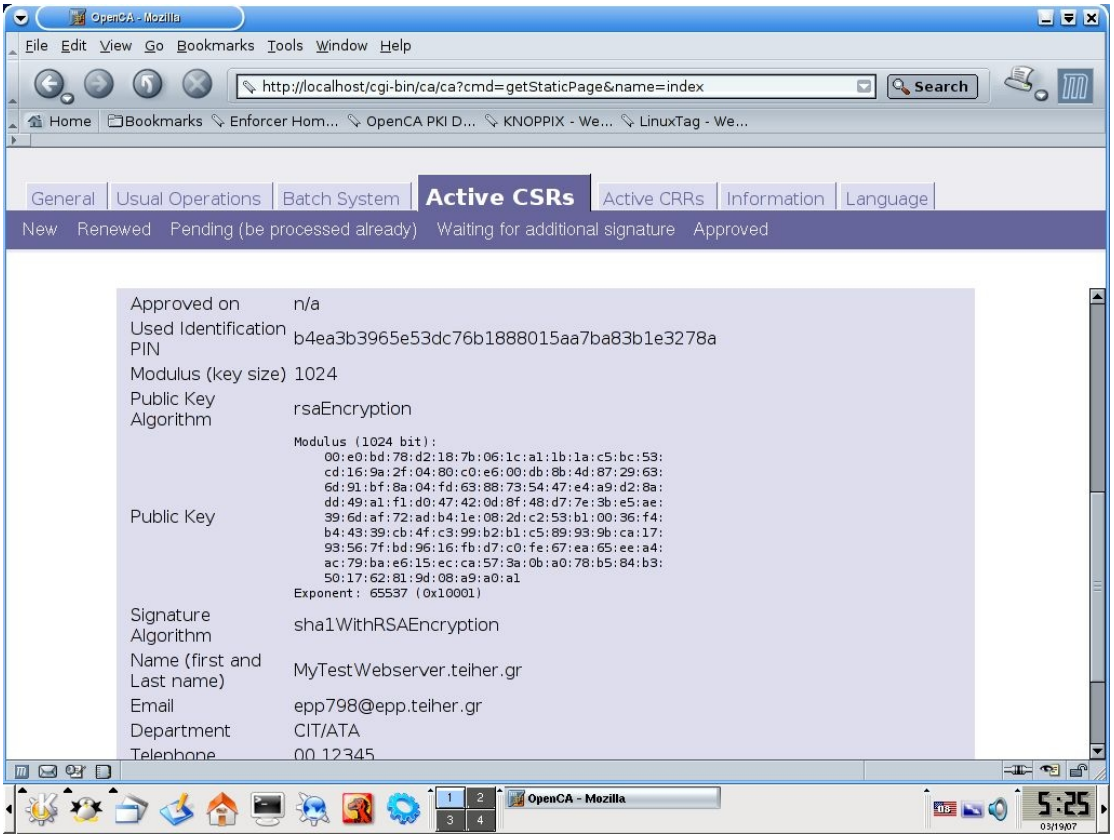


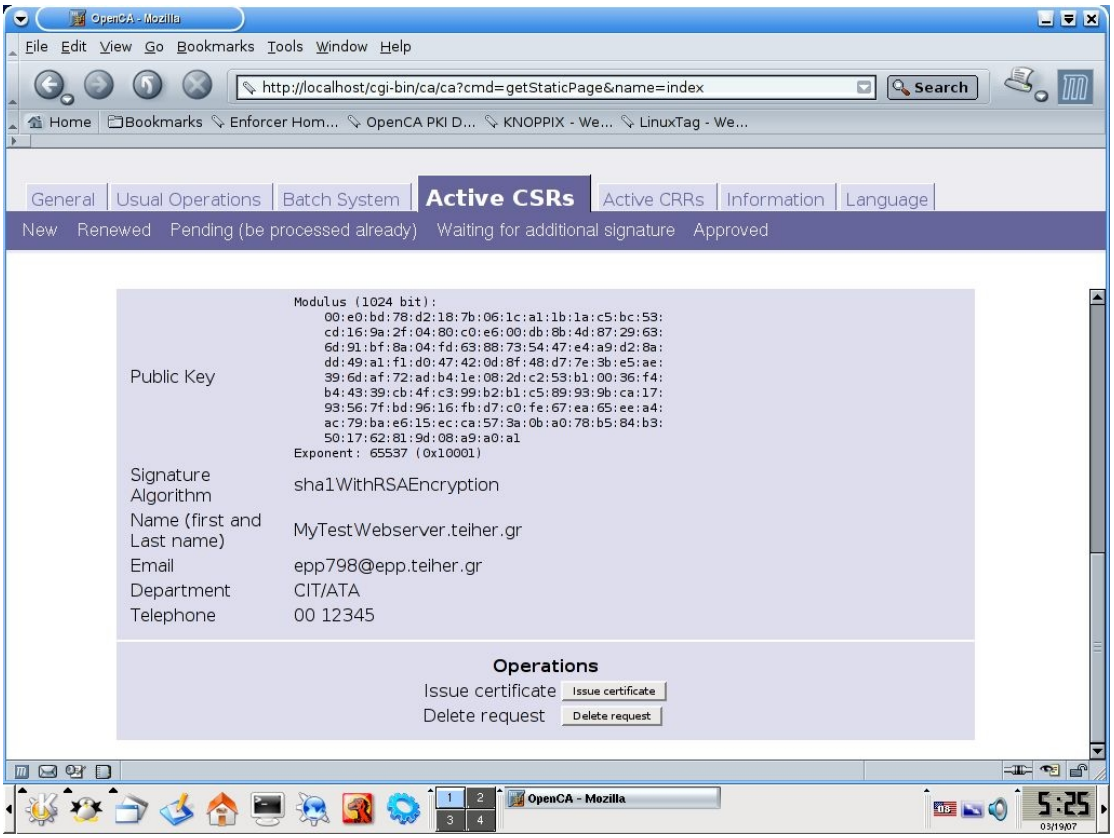
Επιλέγουμε την καρτέλα "**Active CSRs**" και μετά "**New**" και εμφανίζεται η παρακάτω οθόνη στην οποία υπάρχουν οι αιτήσεις των πιστοποιητικών τα οποία έχουν εγκριθεί και είναι έτοιμα να εκδωθούν.



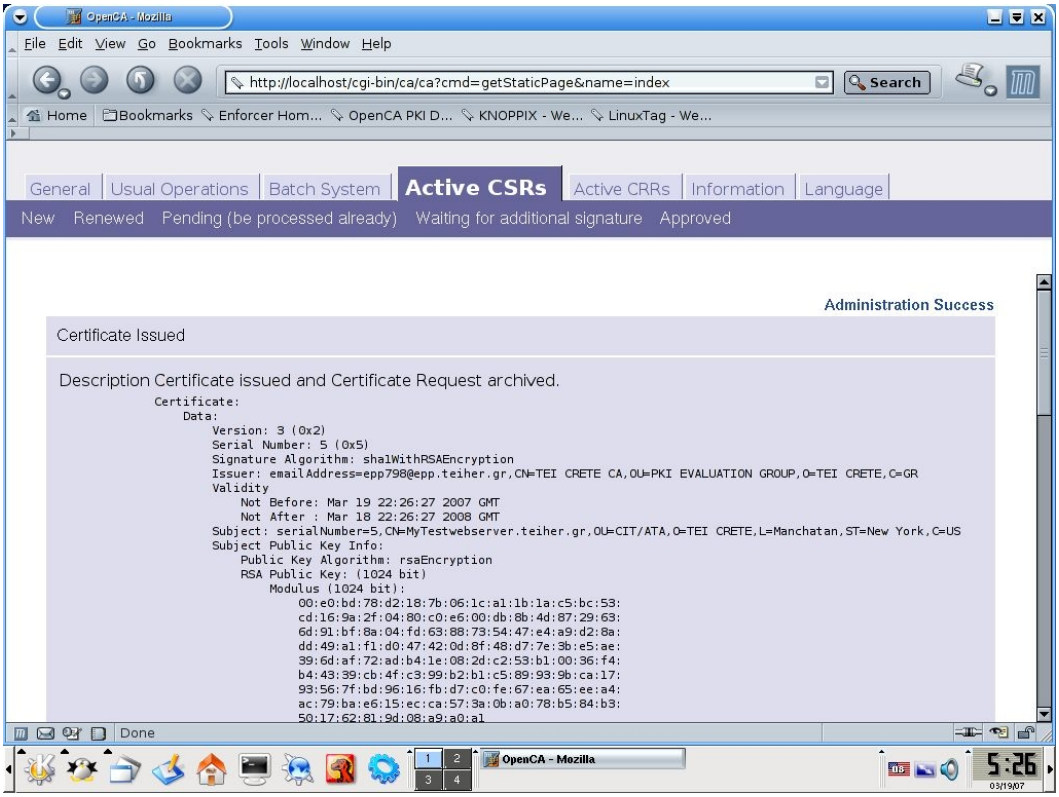
Η δική μας αίτηση έχει κωδικό 2336 οπότε την επιλέγουμε για να περάσουμε στην παρακάτω οθόνη την οποία βλέπουμε σε πλήρης μεγεθός

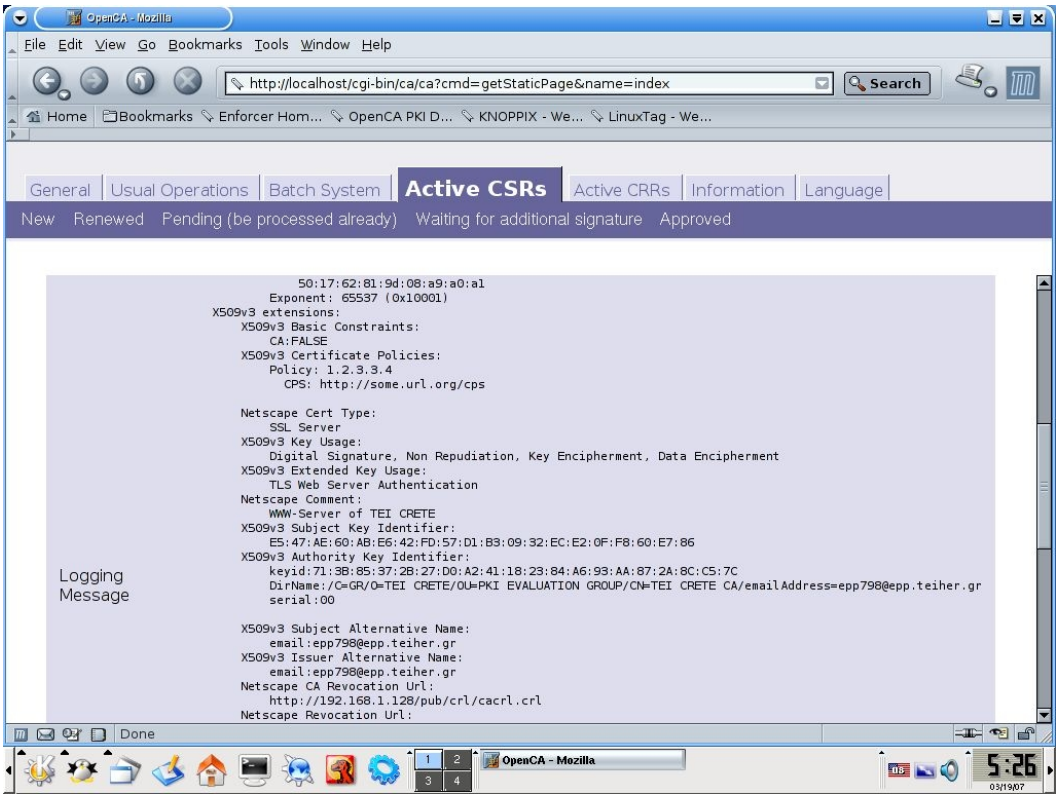

OpenCA - Mozilla
File Edit View Go Bookmarks Tools Window Help
http://localhost/cgi-bin/ca/ca?cmd=getStaticPage&name=index
Search
Home Bookmarks Enforcer Hom... OpenCA PKI D... KNOPPIX - We... LinuxTag - We...
General Usual Operations Batch System Active CSRs Active CRRs Information Language
New Renewed Pending (be processed already) Waiting for additional signature Approved
Approved Request
Following you can find the CSR's details.
Mon Mar 19 17:25:25 2007
Variable Value
Request Version 0 (0x0)
Serial Number 2336
Common Name MyTestwebserver.teiher.gr
E-Mail epp798@epp.teiher.gr
Subject Alternative Name email:epp798@epp.teiher.gr
Role Web Server
LOA Test
Distinguished Name serialNumber=cert's serial, CN=MyTestwebserver.teiher.gr, OU=CIT/ATA, O=TEI CRETE, L=Manchatan, ST=New York, C=US
Submitted on Mon Mar 19 22:20:11 2007 GMT
Approved on n/a
Done
OpenCA - Mozilla
5:25
03/19/07

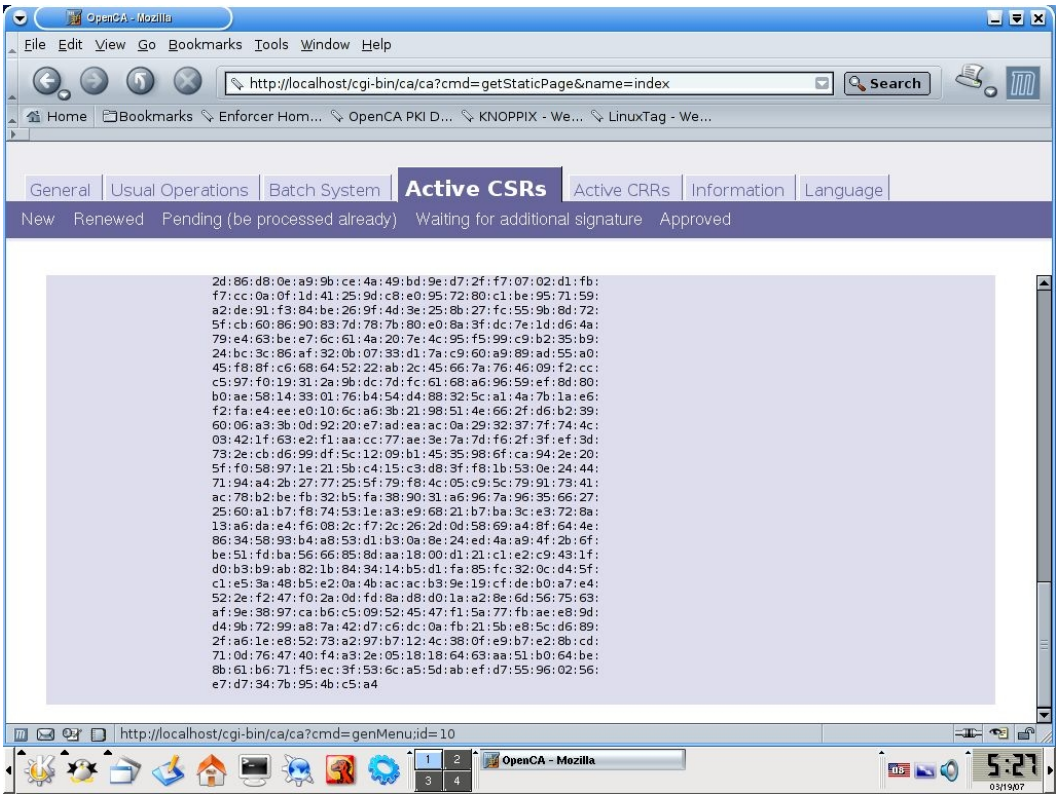

OpenCA - Mozilla
File Edit View Go Bookmarks Tools Window Help
http://localhost/cgi-bin/ca/ca?cmd=getStaticPage&name=index
Search
Home Bookmarks Enforcer Hom... OpenCA PKI D... KNOPPIX - We... LinuxTag - We...
General Usual Operations Batch System Active CSRs Active CRRs Information Language
New Renewed Pending (be processed already) Waiting for additional signature Approved
Approved on n/a
Used Identification PIN b4ea3b3965e53dc76b1888015aa7ba83b1e3278a
Modulus (key size) 1024
Public Key Algorithm rsaEncryption
Public Key
Modulus (1024 bit):
00:e0:bd:78:d2:18:7b:06:1c:a1:1b:1a:c5:bc:53:
cd:16:9a:2f:04:80:c0:e6:00:db:8b:4d:87:29:63:
6d:91:bf:8a:04:fd:63:88:73:54:47:e4:a9:d2:8a:
dd:49:a1:f1:d0:47:42:0d:8f:48:d7:7e:3b:e5:ae:
39:6d:af:72:ad:b4:1e:08:2d:c2:53:b1:00:36:f4:
b4:43:39:cb:4f:c3:99:b2:b1:c5:89:93:9b:ca:17:
93:56:7f:bd:96:16:fb:d7:c0:fe:67:ea:65:ee:a4:
ac:79:ba:e6:15:ec:ca:57:3a:0b:a0:78:b5:84:b3:
50:17:62:81:9d:08:a9:a0:a1
Exponent: 65537 (0x10001)
Signature Algorithm sha1WithRSAEncryption
Name (first and Last name) MyTestWebserver.teiher.gr
Email epp798@epp.teiher.gr
Department CIT/ATA
Telephone 00 12345
OpenCA - Mozilla
5:25
03/19/07



Για να εκδωθεί το πιστοποιητικό επιλέγουμε "**Issue Certificate**", δίνουμε το password το οποίο προστατεύει το ιδιωτικό κλειδί του CA (το οποίο είχαμε δώσει στην διαδικασία της αρχικοποίησης του CA) και εμφανίζεται η παρακάτω οθόνη επιβεβαίωσης.



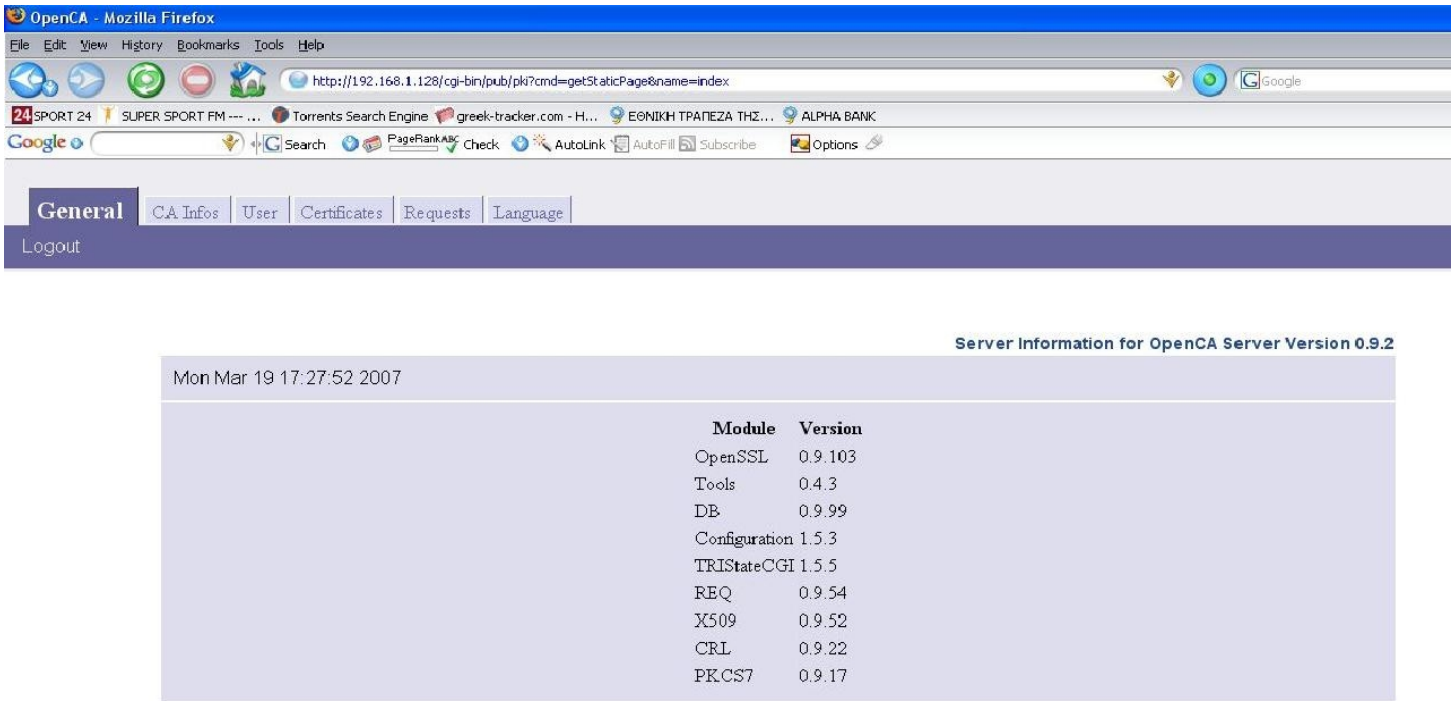




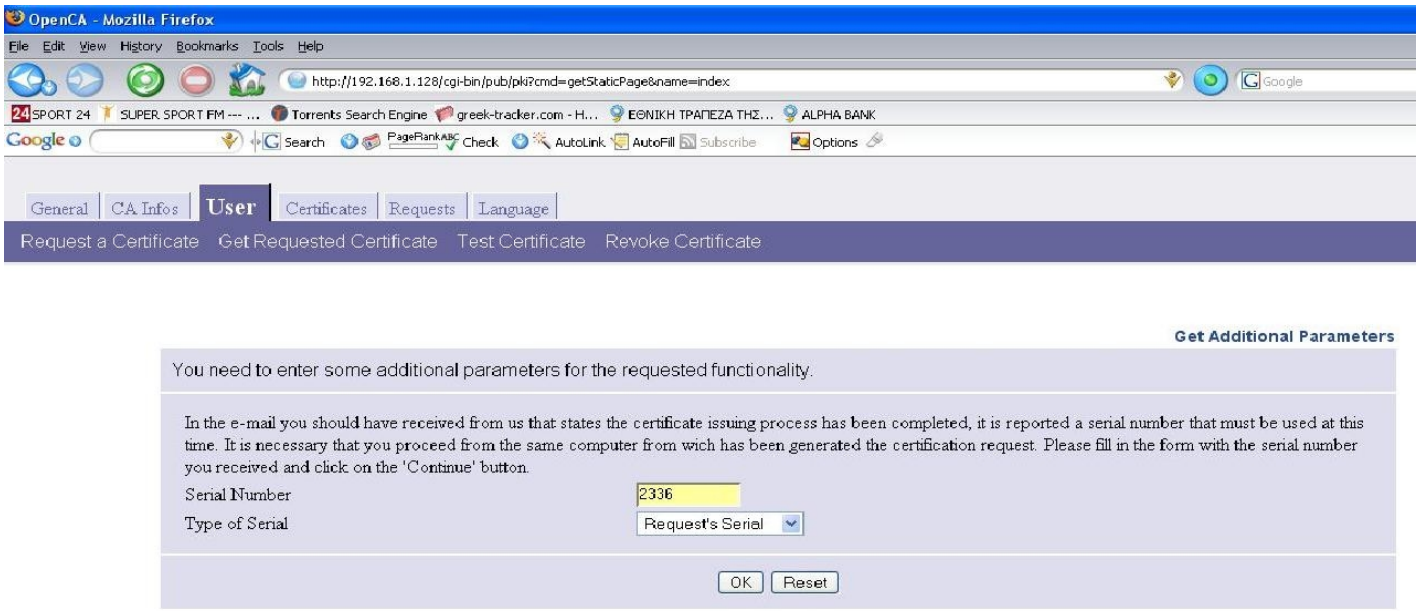
Το πιστοποιητικό θεωρείται πια ότι έχει εκδοθεί από την αρχή πιστοποίησης. Αυτό που μένει είναι να το ανακτήσουμε από τον ίδιο υπολογιστή από τον οποίο το ζητήσαμε.

4.5 Ανάκτηση του πιστοποιητικού

Για να πάρουμε το πιστοποιητικό το οποίο έχει εκδωθεί από το OpenCa θα πρέπει να πάμε στον υπολογιστή client τον οποίο χρησιμοποιήσαμε για να κάνουμε την αίτηση. Ανοίγουμε το mozilla και δίνουμε την ip του συστήματος στο οποίο βρίσκεται το OpenCA (την ip σε Knoppix την βλέπουμε με την εντολή ifconfig). Στην περίπτωση μας http://192.168.1.128/pub, επίσης ο υπολογιστής ο οποίος φιλοξενεί την το OpenCA θα πρέπει να είναι on-line και να "βλέπει" το δικτύο μας. Θα πρέπει να μας εμφανισθεί η παρακάτω οθόνη

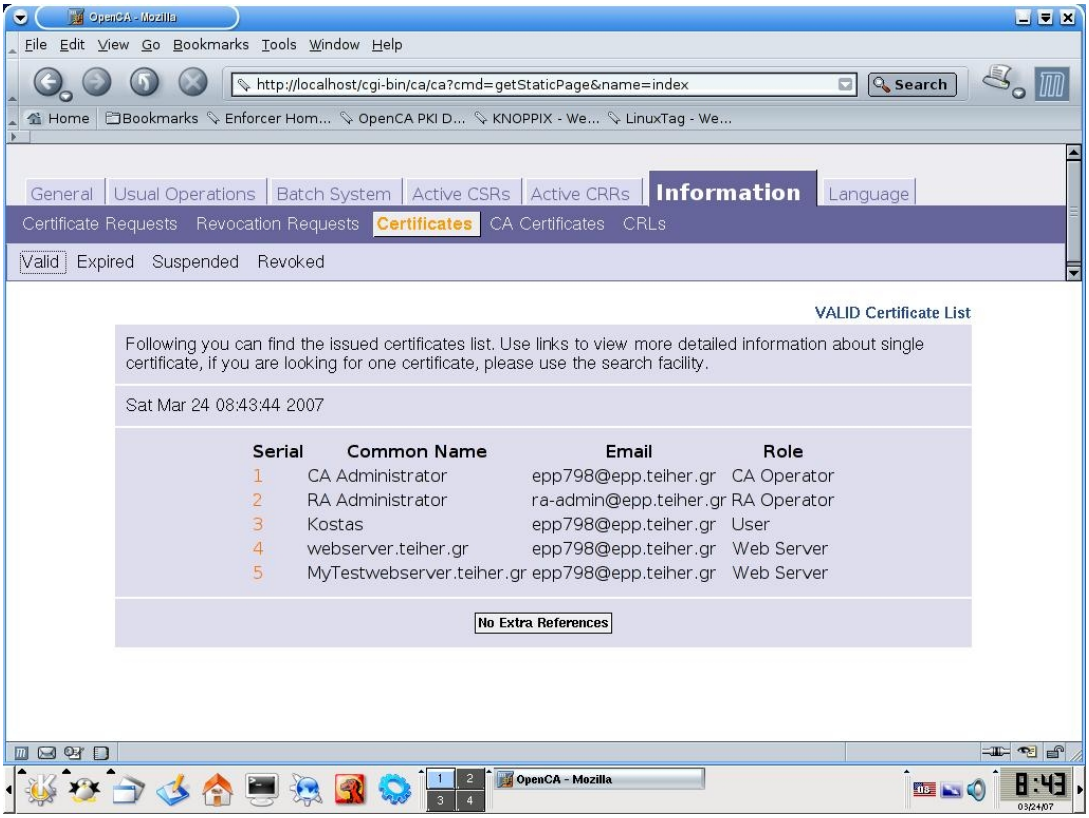


Επιλέγουμε "**User**" και μετά "**Get Requested Certificate**", εμφανίζεται το παρακάτω

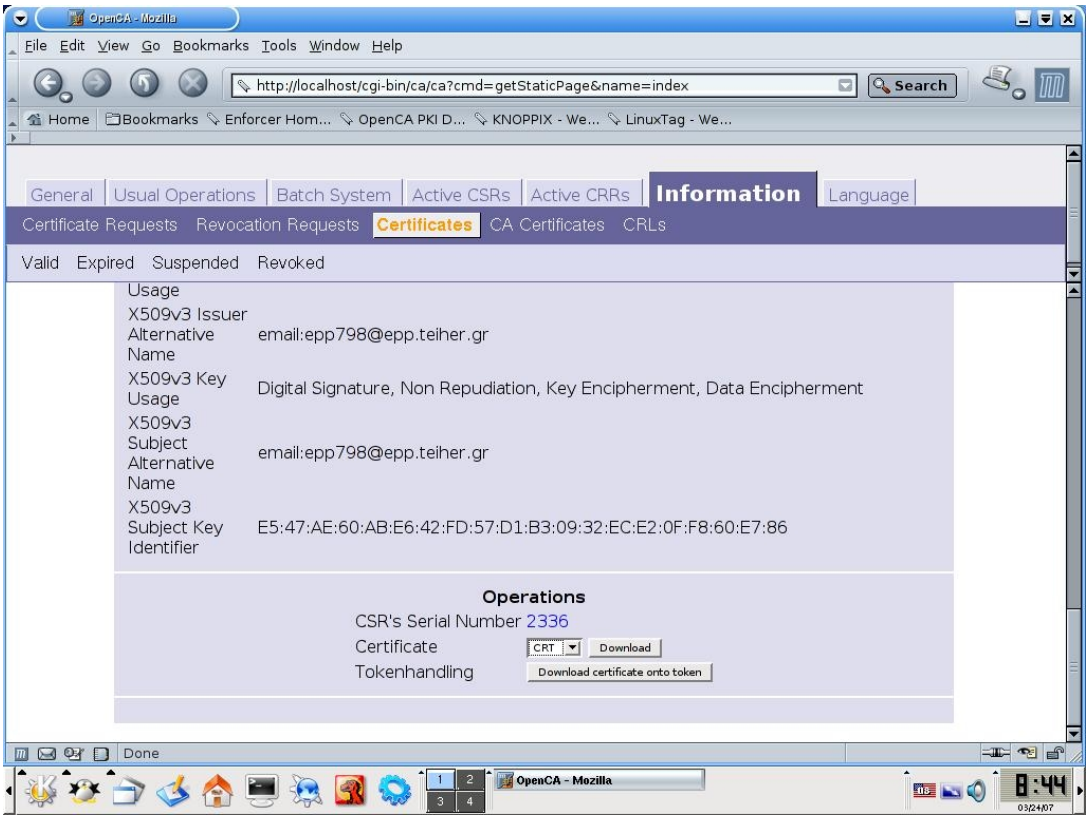


Στο πεδίο "**Serial Number**" δίνουμε το κωδικό της αίτησης μας, στην συγκεκριμένη περίπτωση είναι το 2336 και στο πεδίο "**Type of Serial** " επιλέγουμε "**Request's Serial",** θα πρέπει να εμφανισθεί ένα παράθυρο για download το κατεβάζουμε και το σώζουμε.

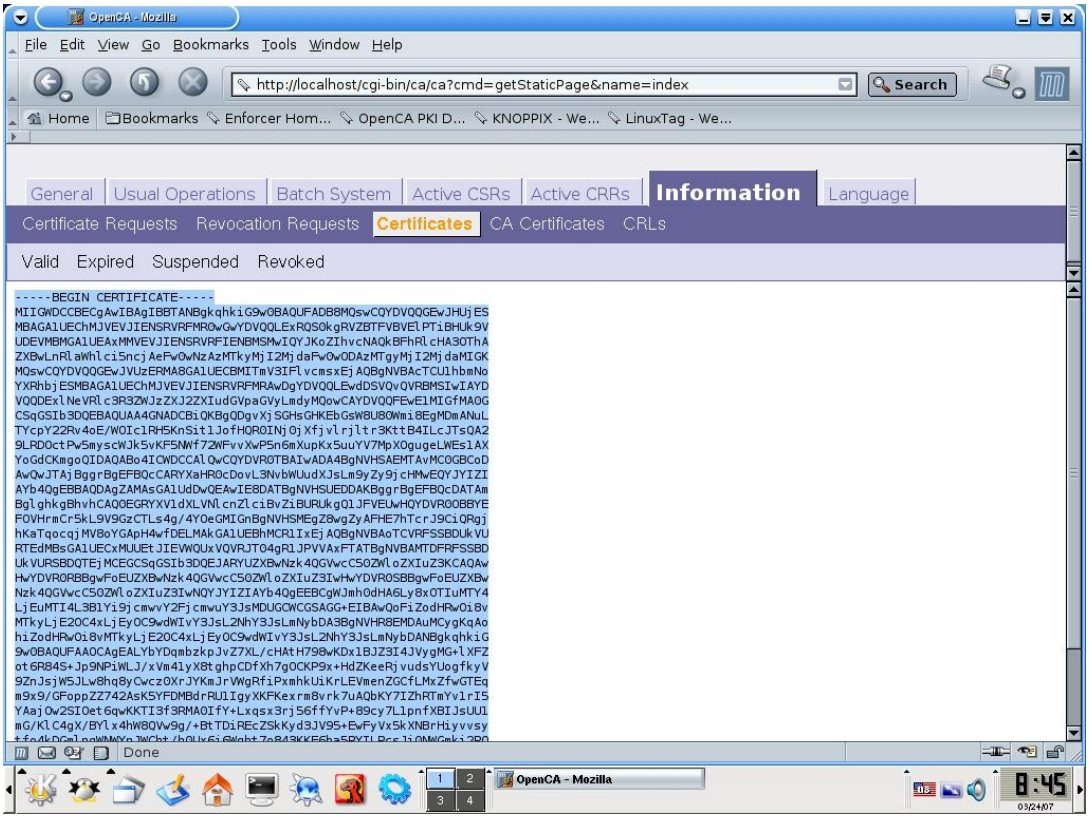
Στην περίπτωση που δεν μπορέσουμε να κάνουμε κατεβάσουμε το πιστοποιητικό του server, υπάρχει πιθανότητα να υπάρχει ασυμβατότητα του browser μας. Στην περίπτωση αυτη και λόγω του ότι όλη η διαδικασία έχει εκπαιδευτικό σκοπό θα κάνουμε μια μικρή παράκαμψη και θα πάμε στο σύστημα στο οποίο λειτουργεί η αρχή πιστοποίησης, θα μπούμε στον CA και θα επιλέξουμε "**Informations**"μετά "**Certificates**" και τέλος "**Valid**". Εδώ θα εμφανισθούν όλα τα πιστοποιητικά τα οποία έχει εκδώσει η αρχή πιστοποίησης και είναι έγκυρα.



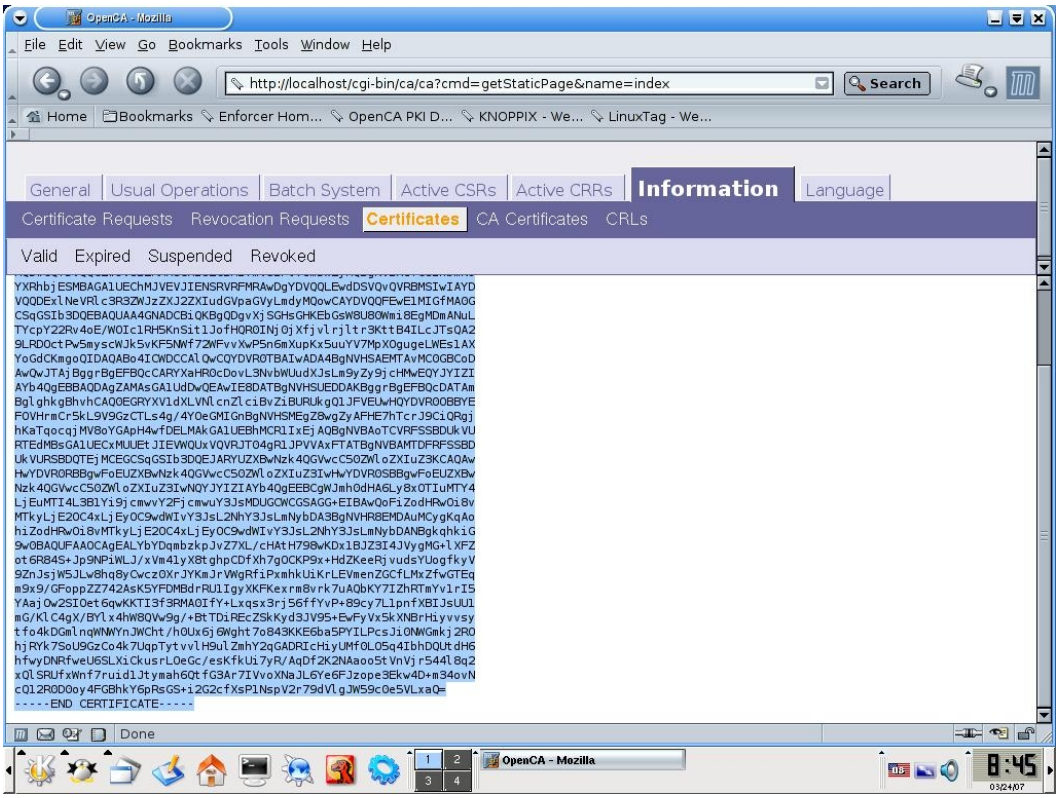
Παρατηρούμε ότι το πιστοποιητικό με αριθμό σειράς πέντε είναι το πιστοποιητικό που μας αφορά. Επιλέγοντας τον αριθμό πέντε περνάμε στην παρακάτω οθόνη



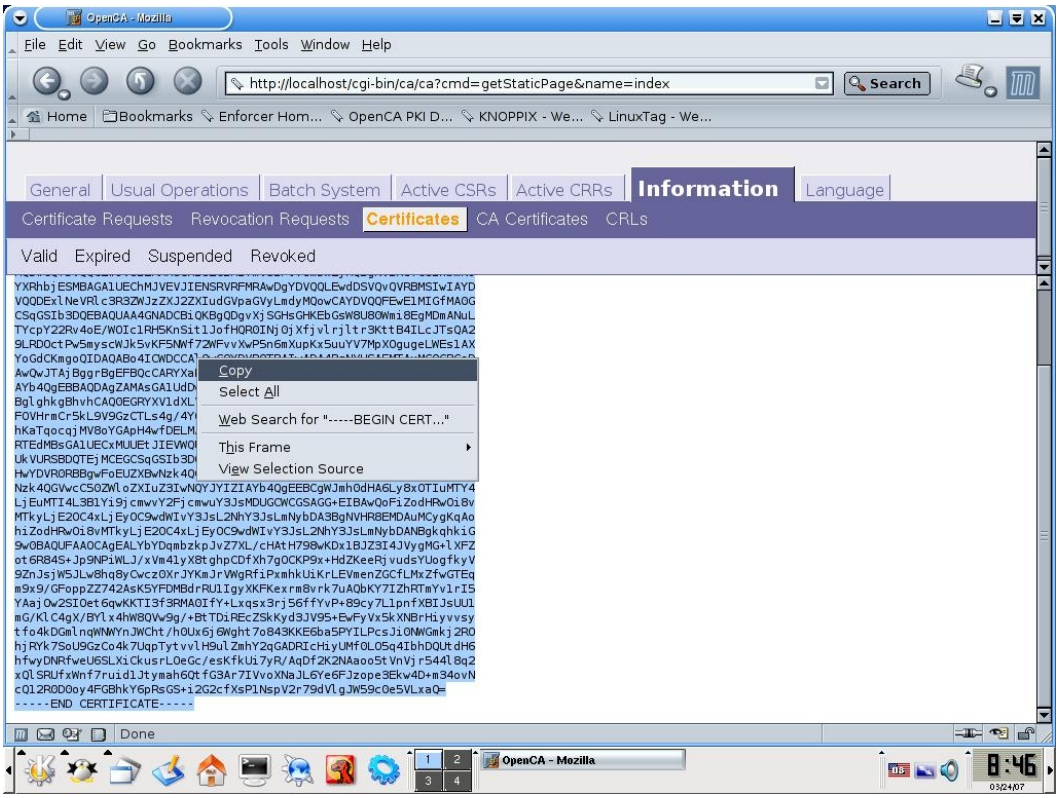
Παρατηρούμε ότι ο αριθμός της αίτησης είναι ο ίδιος με πρίν 2336, στο πεδίο "**Certificates**" επιλέγουμε "**CRT**" και μετα "**Download**". Εμφανίζεται το πιστοποιητικό



Επιλέγουμε όλο το κείμενο απο το "**Begin Certificate**" έως το "**End Certificate**" συμπεριλαμβανομένων και αυτών των τίτλων.



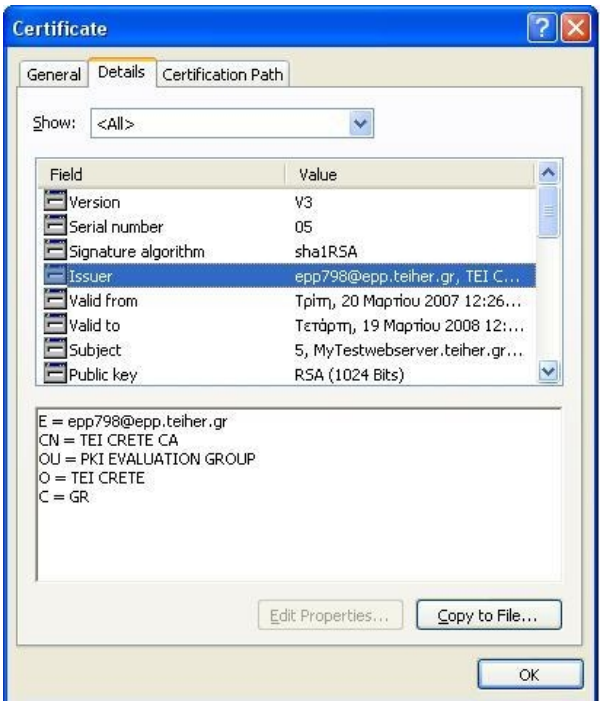
Και επιλέγουμε με δεξί κλίκ "**Copy**"

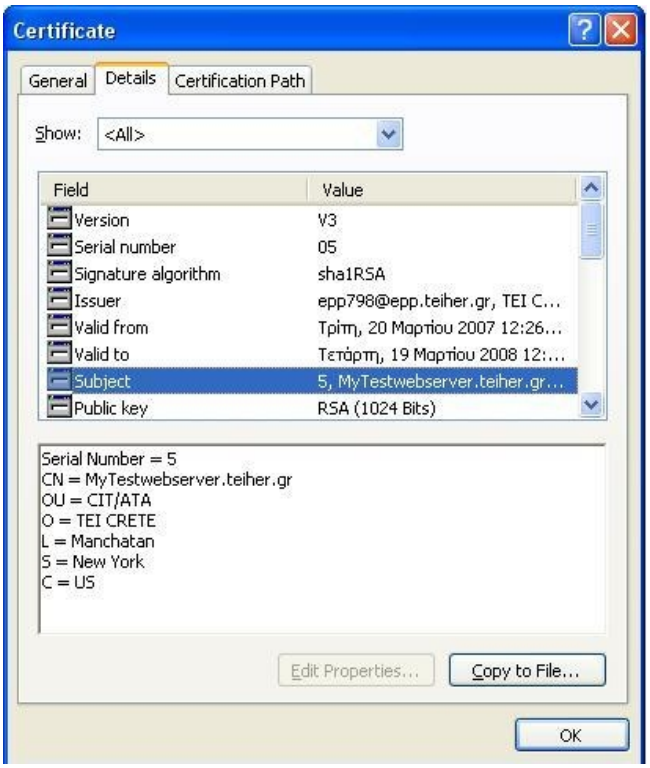


Τώρα ανοίγουμε ένα αρχείο .txt κάνουμε "**Paste**" το περιεχόμενο και μετά το σώζουμε με όνομα "MyWebserver.crt".

Με τον πρώτο τρόπο ή αν είχαμε κάποια επιπλοκή με τον δεύτερο τρόπο, καταφέραμε να δημιουργήσουμε ένα πιστοποιητικό για webserver. Αυτό που μένει είναι να δούμε στην πράξη αν πραγματικά ο Apache Webserver μαζί με το πιστοποιητικό μπορουν να συνεργαστούν ώστε να επιτευχθεί μια ασφαλής σύνδεση μεταξύ webserver και client.

Παρακάτω βλέπουμε κάποιες εικόνες με πληροφορίες του πιστοποιητικού που μόλις δημιουργήσαμε

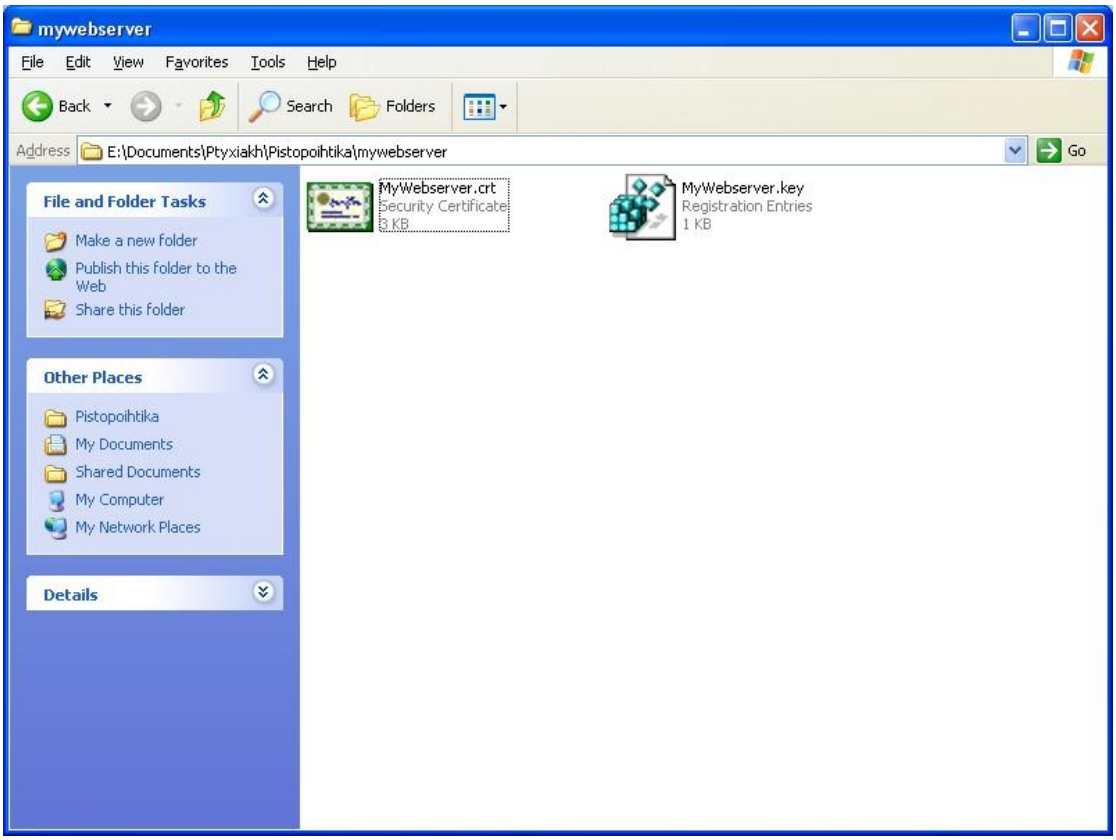




5. Χρησιμοποιόντας το πιστοποιητικό SSL με τον Apache 2.0.59-Openssl_0.9.8d-Win32 σε περιβάλλον windows.

Αρχικά έχουμε εγκαταστήσει τον Apache (2.0.59-Openssl_0.9.8d-Win32) σε περιβάλλον windows xp, οπότε θα πρέπει να κάνουμε τις απαραίτητες ρυθμίσεις ώστε να τον κάνουμε ασφαλή με την χρήση του ssl πιστοποιητικού.

Οπότε αρχικά θα πρέπει να έχουμε στο σύστημα μας δύο αρχεία, το πιστοποίητικο το οποίο έχει δημιουργηθεί απο το Openssl και έχει υπογραφεί απο την αρχή πιστοποίησης μας και το ιδιωτικό κλειδί το οποίο είχαμε δημιουργήσει στην αρχή απο το Openssl.



Θα δημιουργήσουμε δύο φακέλους στο directory του Apache c:/apache/conf. Ο πρώτος θα ονομάζεται ssl.key και θα περιέχει το ιδιωτικό κλειδί και ο δεύτερος θα ονομάζεται ssl.crt και θα περιέχει το πιστοποιητικό.

Στην συνέχεια θα πρέπει να κάνουμε κάποιες αλλαγές στα δύο .conf αρχεία του Apache, το httpd.conf και το ssl.conf. Αρχίζουμε απο το httpd.conf αρχείο (c:/apache/conf) και σβήνουμε τα σχόλια απο το εξής :

LoadModule ssl_module **modules/mod_ssl.so**

Στη συνέχεια δημιουργούμε ένα αρχείο .txt στο οποίο γράφουμε μέσα τα εξής :

@ echo Το password που είχαμε δώσει για το ιδιωτικό κλειδί

Το σώζουμε με όνομα **passphrase.bat** (προσέχουμε να έχει αφαιρεθεί η κατάληξη .txt και να υπάρχει μονάχα η .bat) στο path c:/apache/conf

Πηγαίνουμε στο ssl.conf (c:/apache/conf) προσθέτουμε την εξής γραμμή

SSLPassPhraseDialog "exec:c:/apache/conf/passphrase.bat"

Με το παραπάνω καταφέραμε ο Apache να παίρνει το συνθηματικό του ιδιωτικού κλειδίου κατευθείαν απο το αρχειο passphrase.bat.

Στο ίδιο αρχείο (ssl.conf) συμπληρώνουμε το Path στο οποίο βρίσκεται το ιδιωτικό μας κλειδί καθώς και το πιστοποιητικό μας .

SSLCertificateFile c:/apache/conf/ssl.crt/Mywebserver.crt
SSLCertificateKeyFile c:/apache/conf/ssl.key/Mywebserver.key

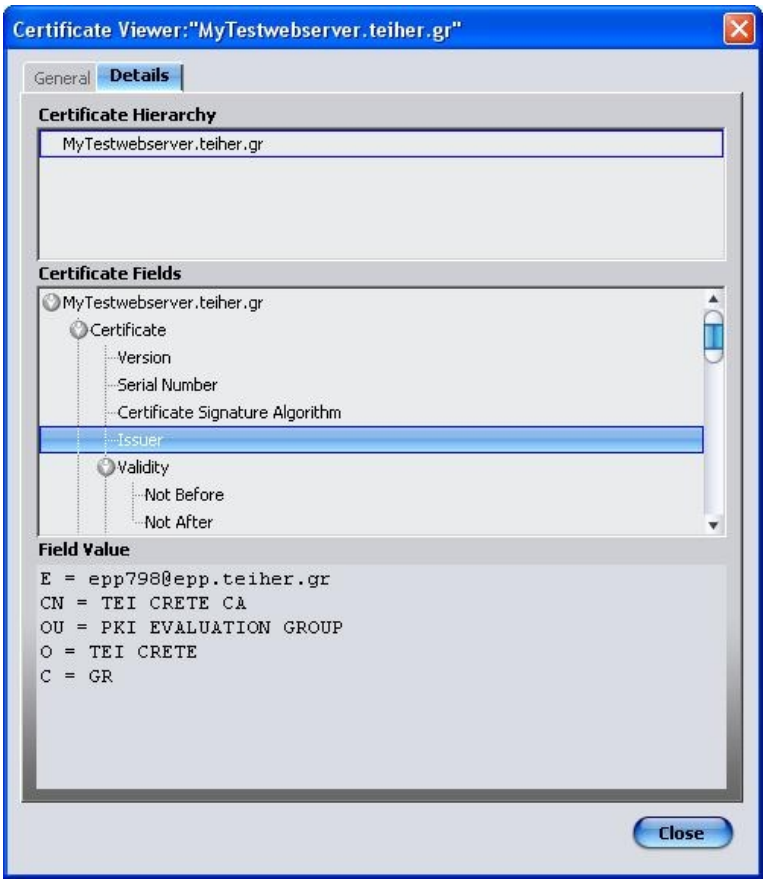
Τέλος σβήνουμε τα σχολία απο την γραμμή ***SSLMutex file:logs/ssl_mutex*** και προσθέτουμε την γραμμή ***SLMutex default.***

Αυτές είναι οι βασικές αλλάγες που θα πρέπει να κάνουμε στα δύο αρχεία .conf ώστε να λειτουργήσει ο Apache. Ανοίγουμε λοιπόν την γραμμή εντολών των Windows (start->Run->Cmd) πάμε στο path που έχουμε τον Apache και αν δεν τον έχουμε εγκαταστήσει τον εγκαθιστούμε με την εντολή **apache -k install ,** στη συνέχεια δίνουμε την εντολή **apache -D SSL** και ξεκινάει ο Apache. Ανοίγοντας ένα browser και δίνοντας https://localhost βλέπουμε το παρακάτω μήνυμα

Το παραπάνω μήνυμα μας ενημερώνει ότι ο web browser δεν αναγνωρίζει την αρχή πιστοποίησης η οποία έκδωσε το πιστοποιητικό. Αυτό το μήνυμα ήταν κάτι αναμενόμενο διότι η αρχή που υπέγραψε το πιστοποιητικό ήταν η δική μας αρχή πιστοποίησης και δεν κάναμε κάτι ώστε να ενημερώσουμε τον browser μας .

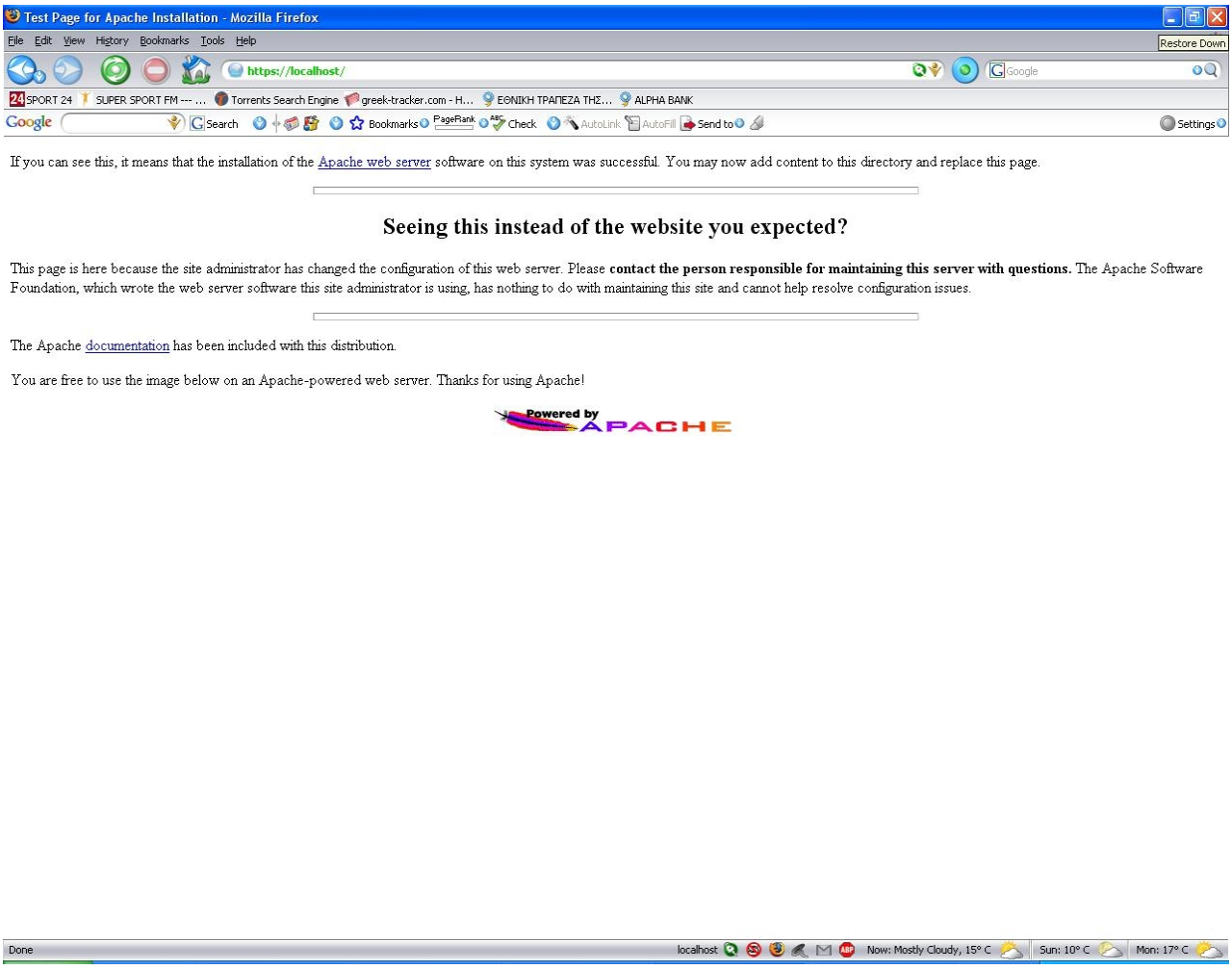
Για να εξετάσουμε το πιστοποιητικό επιλέγουμε "**Examine Certificate**" και εμφανίζεται η παρακάτω οθόνη ενημέρωσης


Certificate Viewer:"MyTestwebserver.teiher.gr"
General
Details
Could not verify this certificate for unknown reasons.
Issued To
Common Name (CN) MyTestwebserver.teiher.gr
Organization (O) TEI CRETE
Organizational Unit (OU) CIT/ATA
Serial Number 05
Issued By
Common Name (CN) TEI CRETE CA
Organization (O) TEI CRETE
Organizational Unit (OU) PKI EVALUATION GROUP
Validity
Issued On 20/3/2007
Expires On 19/3/2008
Fingerprints
SHA1 Fingerprint 61:C2:02:8D:88:F9:AE:78:02:58:47:C8:68:E1:66:CD:42:EF:5F:F8
MD5 Fingerprint 2D:80:7F:71:1B:79:46:5D:83:DB:AA:92:45:8D:38:3E
Close


Certificate Viewer:"MyTestwebserver.teiher.gr"
General
Details
Certificate Hierarchy
MyTestwebserver.teiher.gr
Certificate Fields
MyTestwebserver.teiher.gr
Certificate
Version
Serial Number
Certificate Signature Algorithm
Issuer
Validity
Not Before
Not After
Field Value
E = epp798@epp.teiher.gr
CN = TEI CRETE CA
OU = PKI EVALUATION GROUP
O = TEI CRETE
C = GR
Close

Η οποία μας δείχνει κάποια επιπλέον στοιχεία για το πιστοποιητικό μας. Κλείνοντας το παραπάνω παράθυρο και επιλέγοντας "**Accept this certificate temporarily for this season**", μας εμφανίζεται το παρακάτω μήνυμα

Το οποίο μας ενημερώνει ότι στο πιστοποιητικό είχαμε δηλώσει ότι το όνομα του site στο οποίο θα ανήκει το πιστοποιητικό θα είναι το MyTestWebserver.teiher.gr και εμείς το χρησιμοποιούμε για το https:///localhost. Το μήνυμα ήταν και αυτό αναμενόμενο και δεν μας επηρεάζει για την συνέχεια. Επιλέγοντας "Ok" λέμε στο browser να παραβλέψει αυτό το μήνυμα και αφού του έχουμε πεί προηγουμένος να εμπιστευτεί την αρχή πιστοποίησης, ανοίγουμε ένα ασφαλές κανάλι επικοινωνίας ssl μεταξύ web server και client. Όπως βλέπουμε και στην παρακάτω οθόνη



Συνοψίζοντας, καταφέραμε να δημιουργήσουμε ένα πιστοποιητικό με την βοήθεια του Openssl σε περιβάλλον Linux (kubuntu), στη συνέχεια το περάσαμε στην αρχή πιστοποίησης μας, η αρχή το πιστοποίησε και έτσι το περάσαμε στον web server Apache.

Αυτό που μας μένει είναι να αντιμετωπίσουμε τα παραπάνω μηνύματα λάθους που βλέπουμε όταν συνδεόμαστε στο https:///localhost.

5.1 Αντιμετώπιση Μηνυμάτων Λάθους

Όπως είδαμε υπάρχουν κάποια μειονεκτήματα όταν χρησιμοποιούμε μια αρχή πιστοποίησης η οποία πιστοποιεί (μέσω της υπογραφής της) τον εαυτό της. Σε επαγγελματικό περιβάλλον δεν θα θέλαμε οι χρήστες να δούν μηνύματα τα οποία σχετίζονται με πρόβλημα του ssl πιστοποιητικού πρίν μπούν στο web site μας. Το βασικό ερώτημα είναι πώς θα κάνουμε την αρχή πιστοποίησης γνωστή στον browser. Υπάρχουν δύο τρόποι να το επιτύχουμε ο πρώτος είναι πιο γραφειοκρατικός και

χρονοβόρος ενώ ο δεύτερος είναι πίο γρήγορός και για την περίπτωση μας ο ιδανίκος. Όμως καλύτερα να αναφερθούμε αναλυτικά και στους δύο.

5.1.1 Πρώτη προσέγγιση

Ο κάθε web browser γνωρίζει έναν αριθμό Αρχών Πιστοποίησης. Συνήθως γνωρίζει τις πίο μεγάλες και γνωστές αρχές. Οπότε θα πρέπει μια τέτοια αρχή πιστοποίησης (πχ Verisign) να υπογράψει το πιστοποιητικό του root της δικής μας αρχής. Όταν συμβεί αυτό η αρχή μας θα μπορεί πλέον να αναγνωρισθεί από τον web browser, έτσι δεν θα εμφανισθεί ξανά παρόμιο μήνυμα λάθους. Για να καταφέρουμε λοιπόν να πιστοποιήσουμε την αρχή μας θα πρέπει να αναπτύξουμε μια σχέση εμπιστοσύνης με την "Μεγάλη" αρχή πιστοποίησης. Η σχέση εμπιστοσύνης σημαίνει ότι θα πρέπει να πληροφορίσουμε την "Μεγάλη" αρχή για κάποιες λεπτομέρειες όπως πως λειτουργούμε την δίκη μας αρχή, πως την προστατεύουμε και πως εκδίδουμε πιστοποίητικα. Αυτό φυσικά για να συμβεί χρειάζεται χρόνο, χρήμα και υπευθυνότητα. Λόγω του ότι ο σκοπός μας είναι καθαρά εκπαιδευτικός και όχι επαγγελματικός θα αφήσουμε αυτή την πρώτη προσεγγίση και θα προχωρίσουμε στην δεύτερη.

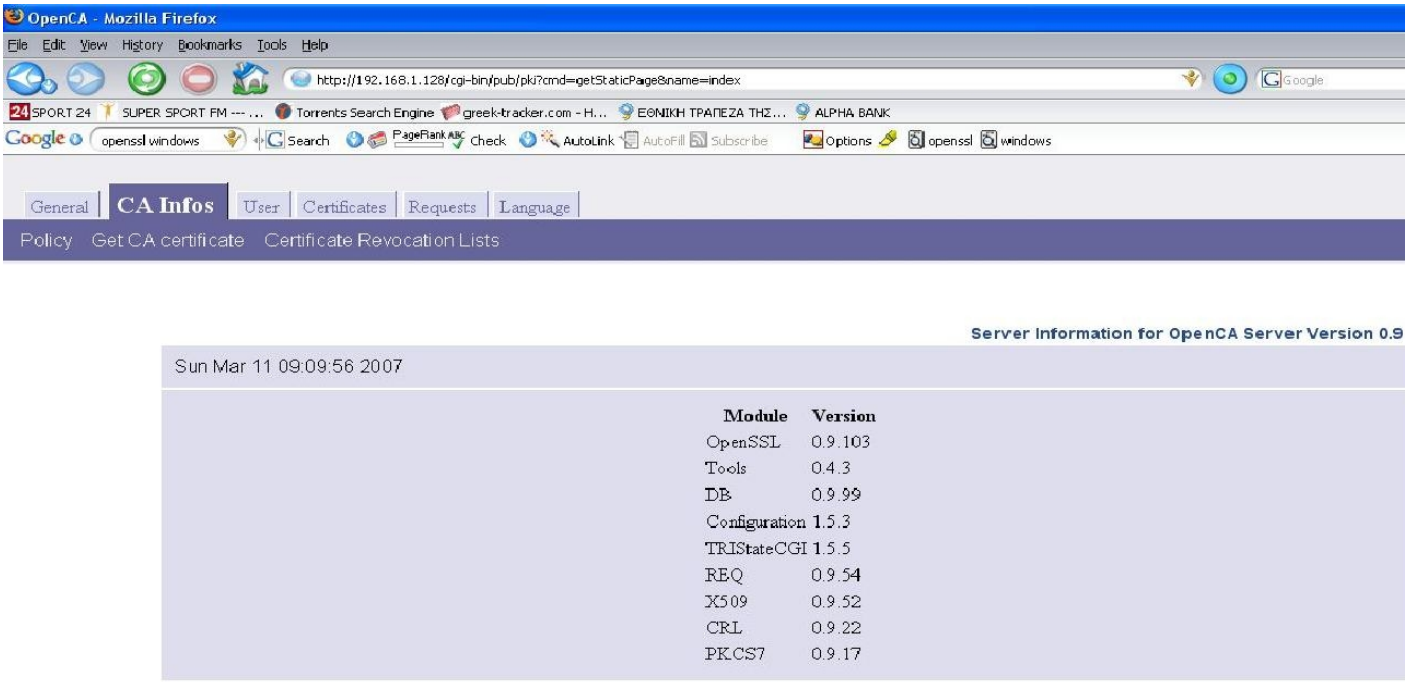
5.1.2 Δεύτερη προσέγγιση

Η δεύτερη προσέγγιση είναι πολύ ευκολότερη και ουσιαστικά η κατάλληλη για τον σκοπό μας. Θα πρέπει να πάρουμε από την αρχή πιστοποίησης μας την οποία έχουμε υλοποίησει με την βοήθεια του OpenCA το πιστοποιητικό του root και να το βάλουμε στον browser μας. Η διαδικασία εξαγωγής του πιστοποιητικού είναι παρόμοια με το κατέβασμα ενός αρχείο από ένα server. Οι περισσότεροι Browsers γνωρίζουν πώς να διαχειριστούν root πιστοποιητικά, έτσι ώστε να μετατρέψουν μια αγνωστή για τον browser αρχή πιστοποίησης (όπως η δική μας) σε μια γνωστή.

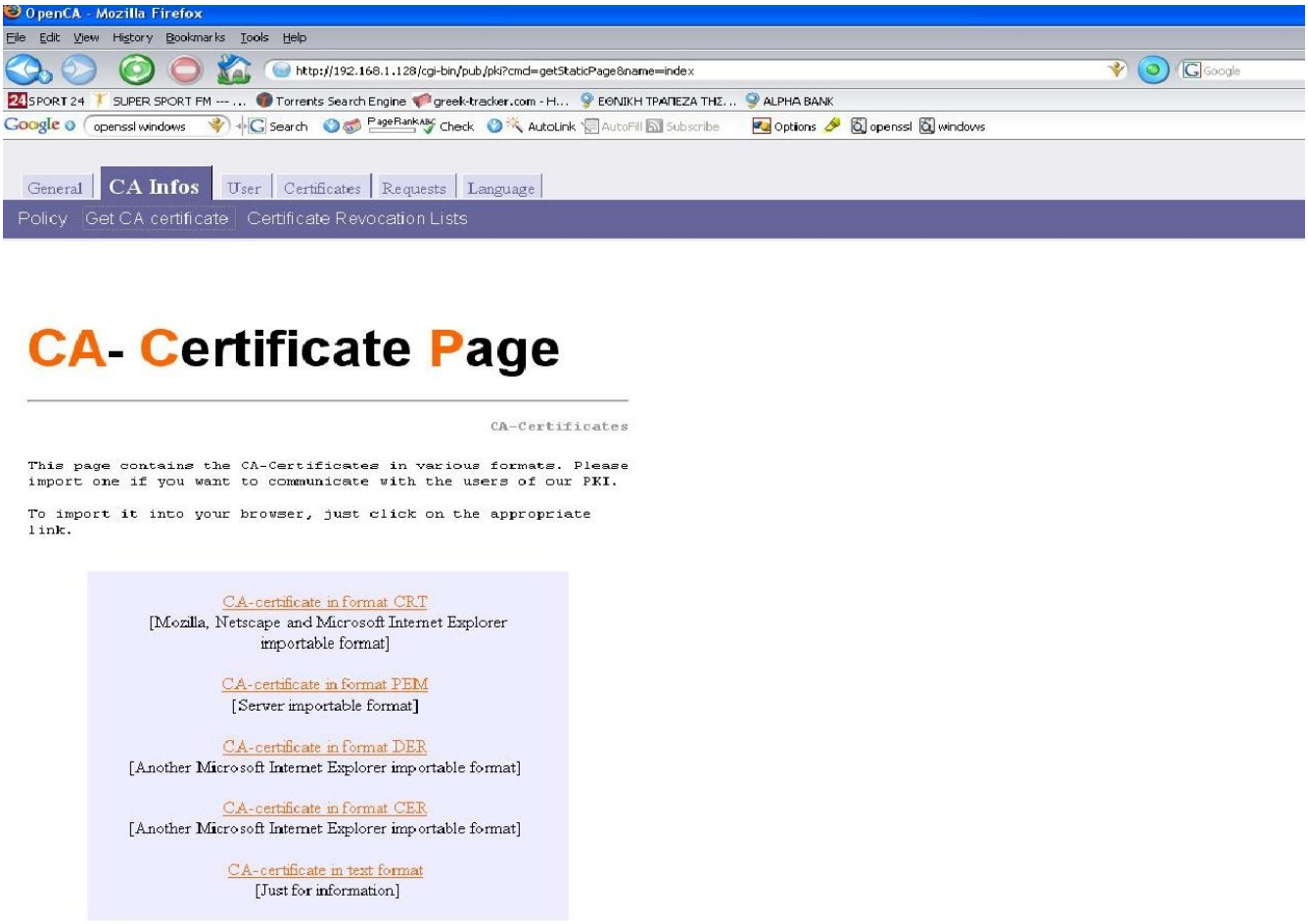
Όπως αναφέραμε παραπάνω η δεύτερη προσέγγιση είναι η πιο εύκολη στην υλοποίηση της και αυτή που ταιρίαζει καλύτερα στην περίπτωση μας. Παρακάτω αναφέρουμε πως μπορούμε να εξάγουμε το πιστοποιητικό του root από την αρχή μας.

5.2 Εξαγωγή του πιστοποιητικού του root

Συνδεόμαστε στην αρχή πιστοποίησης στο /pub, είτε δίνοντας από τον browser ενός υπολογιστή που "βλέπει" την αρχή πιστοποίησης, τη διεύθυνση http://192.168.1.128/pub, είτε μπαίνοντας απο τον browser του Knoppix απο τον υπολογιστή στον οποίο έχουμε εγκαταστήσει την αρχή πιστοποίησης και επιλέγοντας από τα bookmarks το OpenCA:pub. Και οι δύο τρόποι έχουν το ίδιο αποτέλεσμα. Διαλέγουμε τον πρώτο τρόπο δηλαδή την απομακρυσμένη είσοδο και επιλέγουμε την καρτέλα "**CA Infos**"



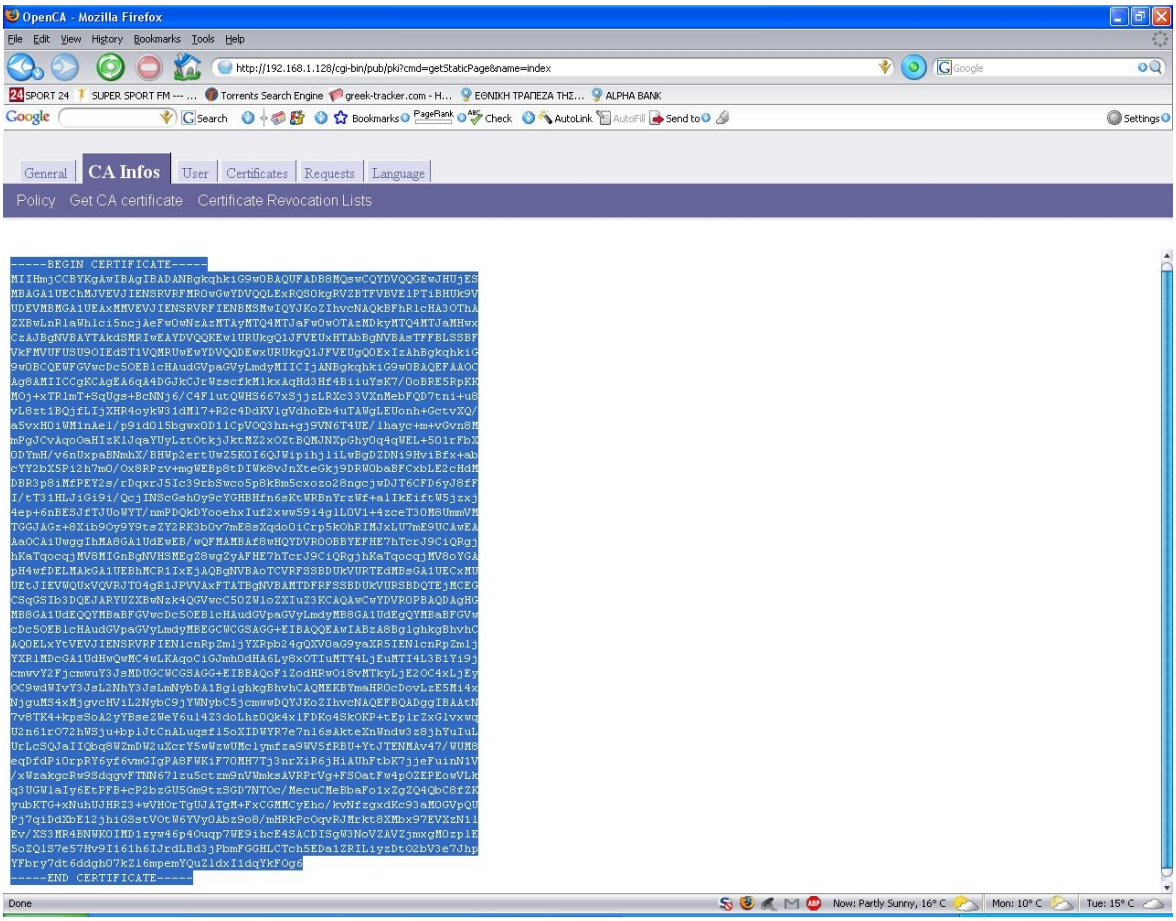
Επιλέγουμε "**Get CA certificate"** και βλέπουμε την παρακάτω οθόνη



Σε αυτό το σημείο έχουμε δύο επιλόγες η πρώτη είναι να επιλέξουμε "**CA – Certificates in Format CRT**". Επιλέγοντας το ο browser μας θα αρχίσει την διαδικασία αποδοχής του πιστοποιητικού του root, έτσι η αρχή μας θα γίνει έμπιστη. Στο κουτί διαλόγου που εμφανίζεται τσεκάρουμε όλες τις επιλογές

Εδώ μπορούμε να εξετάσουμε το πιστοποιητικό μέσω του “View”, πατώντας “OK” . Οπότε καταφέραμε να περάσουμε στον browser το πιστοποιητικό του root administrator της αρχής πιστοποίησης μας.

Η δεύτερη επιλογή χρησιμεύει στην περίπτωση που οι χρήστες του webserver μας δεν έχουν άμεση πρόσβαση στην αρχή πιστοποίησης, οπότε επιλέγουμε “**CA Certificate in PEM format**”θα μας εμφανισθεί ένα μήνυμα όπως το παρακάτω.



Επιλέγουμε όλους τους χαρακτήρες του πιστοποιητικού από το “**Begin Certificate**” έως και το “**End Certificate**” (συμπεριλαμβανομένων των “**Begin Certificate**” “**End Certificate**”) και κάνουμε copy paste σε ένα αρχείο .txt και του δίνουμε το όνομα openca.crt. Απο τη στιγμή που έχουμε το πιστοποίητικου του root μπορούμε πλέον να το εισάγουμε στον Apache. Με αυτή την ενέργεια κερδίζουμε δύο πράγματα. Πρώτον ο web server μας μπορεί να γίνει σημείο διανομής του πιστοποιητικού του root, έτσι όποιος χρήστης μπάινει θα μπορεί να κατεβάσει το openca.crt στον δικό του browser και να του επιτρέψει να εμπιστεύεται την αρχή πιστοποίησης μας. Δεύτερον ο web server μας θα χρειαστεί αυτό το αρχείο έτσι ώστε να πιστοποίει τους χρήστες οι οποίοι έχουν πιστοποιητικά από την δική μας αρχή πιστοποίησης.

5.3 Τελευταίες Ρυθμίσεις του Apache 2.0.59-Openssl_0.9.8d-Win32 Web Server

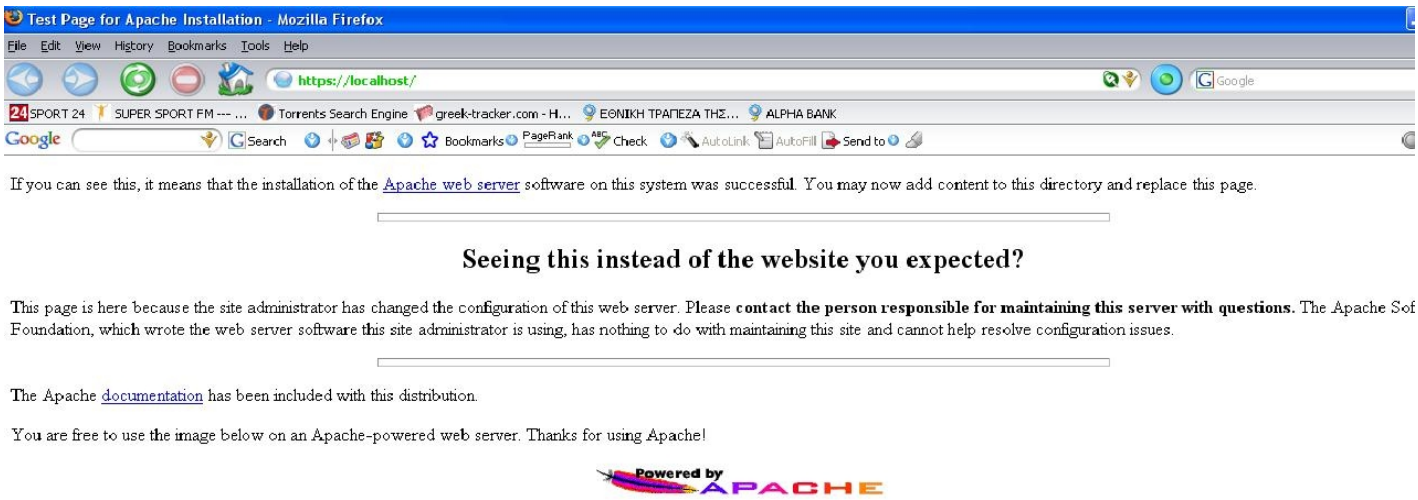
Ο τρόπος εισαγωγής του root πιστοποιητικού διαφέρει λιγο απο προηγουμένος αλλά το αποτέλεσμα είναι το ίδιο, ο web browser αναγνωρίζει την αρχή πιστοποίησης μας και δεν εμφανίζει μήνυμα λάθους.

Για να το περάσουμε λοιπόν στον apache ανοίγουμε το αρχείο ssl.conf το οποίο βρίσκεται στον φάκελο /conf στο path που έχουμε εγκαταστήσει τον apache και βρίσκουμε την γραμμή που αναφέρεται στο SSLCACertificateFile και εκεί δίνουμε το path στο οποίο έχουμε το πιστοποιητικό του root δηλαδή

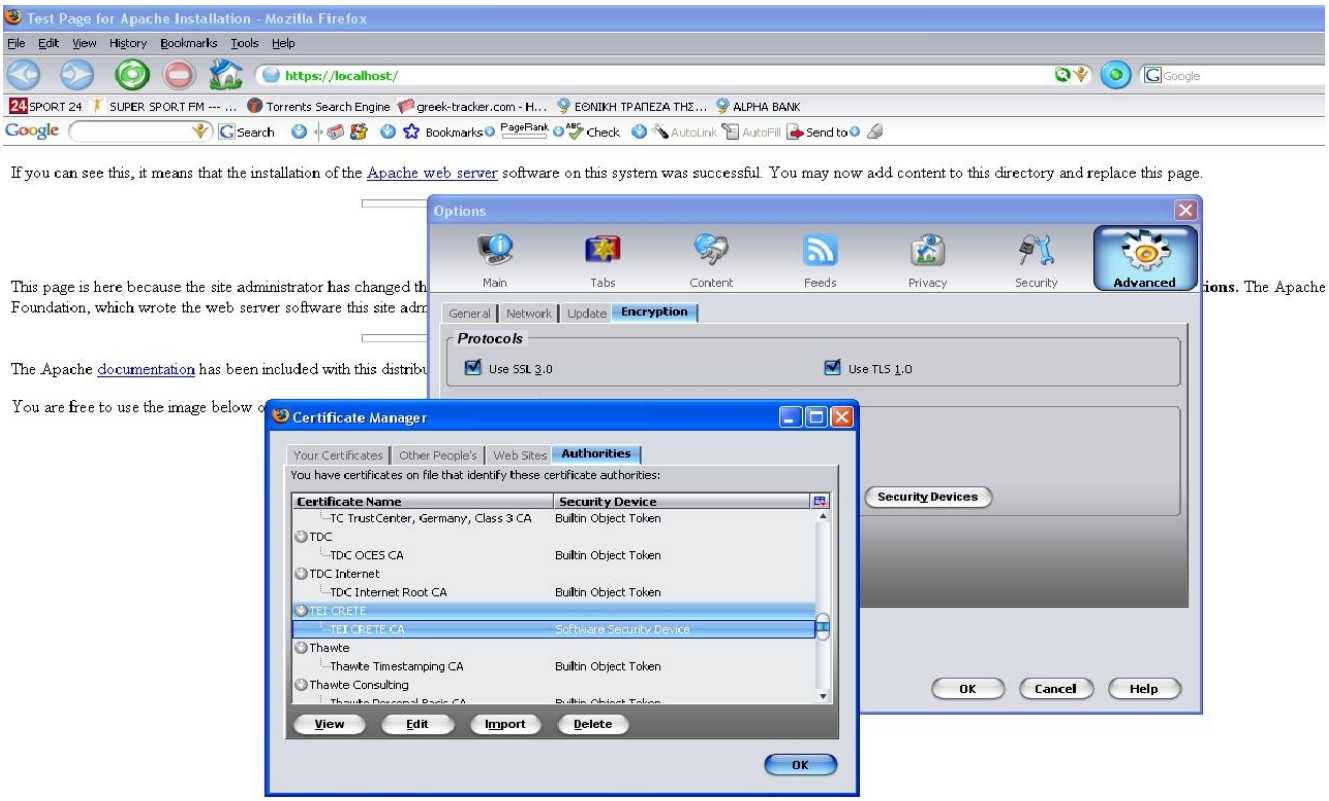
SSLCACertificateFile ***c:/apache/conf/ssl.crt/openca.crt***

Οπότε και με αυτό το δεύτερο τρόπο καταφέραμε να περάσουμε στον browser το πιστοποιητικό του root administrator της αρχής πιστοποίησης μας. Να δούμε όμως τι αποτέλεσμα έχει όλη αυτή η διαδικασία.

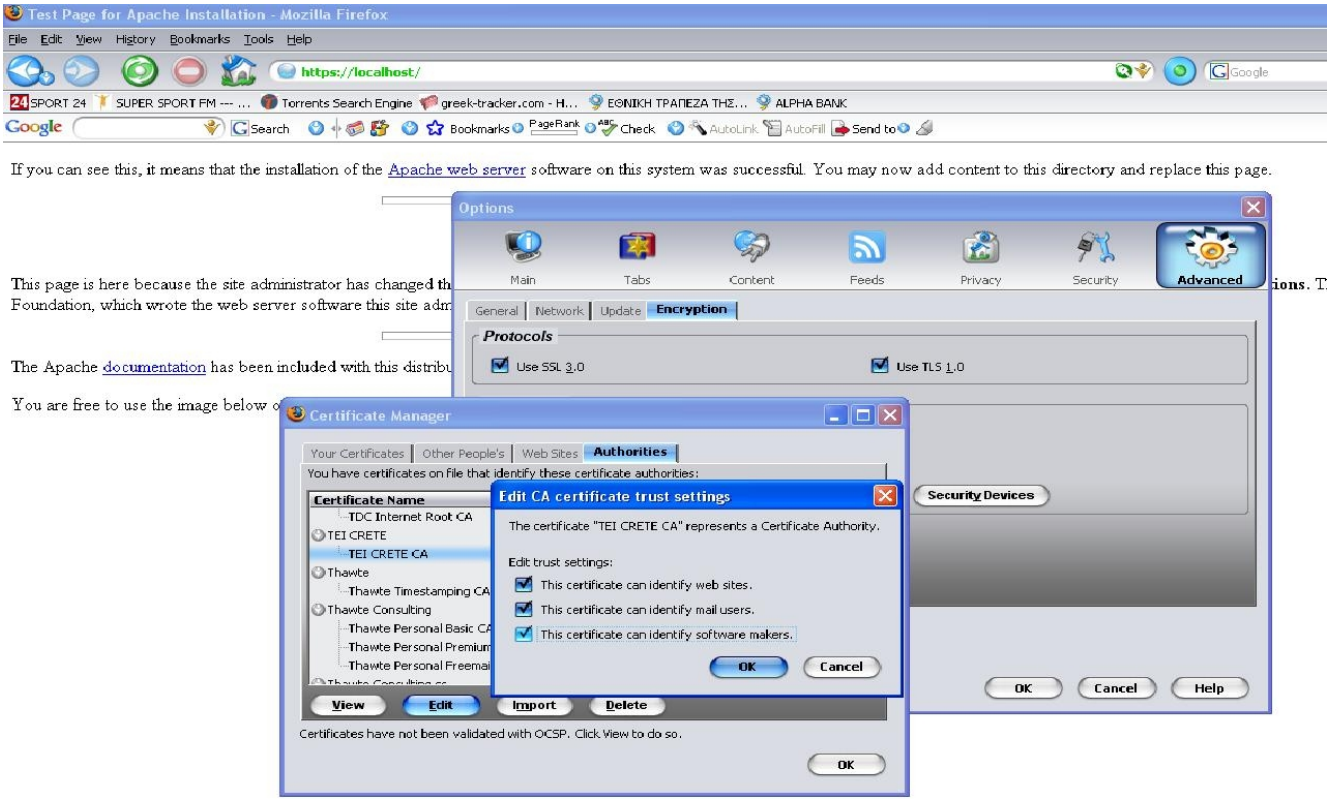
Καταρχάς πια δεν μας εμφανίζεται μήνυμα λάθους για την αναγνώριση της αρχής πιστοποίησης οπότε βλέπουμε κανονικά την αρχική σελίδα του https://localhost



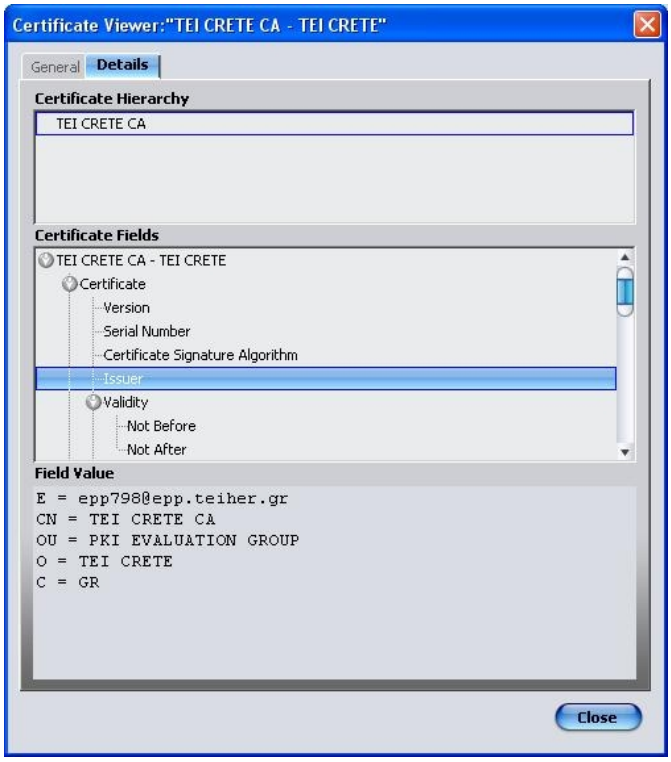
Επίσης αν πάμε Tools->options και μετά στην καρτέλα Encryptions θα δούμε ότι η αρχή πιστοποίησης είναι πια γνωστή στον browser μας ώς TEI CRETE CA



Στην περίπτωση που περάσαμε το πιστοποιητικό του root απο το αρχείο openca.crt τότε μένει να κάνουμε κάτι τελευταίο, να επιλέξουμε Edit και να τσεκάρουμε όλες τις επιλογές



Παρακάτω βλέπουμε τα στοιχεία του πιστοποιητικού του root, επιλέγοντας view απο το παράθυρο certificate manager και έχοντας επιλέξει ήδη TEI CRETE CA.



Συνοψίζοντας καταφέραμε να δημιουργήσουμε ένα πιστοποίητικο για web server να το πιστοποιήσουμε με την βοήθεια της δική μας αρχής πιστοποίησης και να το περάσουμε στον server μας. Εκεί αντιμετωπίσαμε κάποια μηνύματα λάθους με βασικό μηνυμα αυτό που αναφερόταν στην μη αναγνώριση της αρχής πιστοποίησης που υπέγραψε το πιστοποιητικό (την δική μας αρχή) το οποίο καταφέραμε να ξεπεράσουμε επιτυχώς και έτσι ο web server μας αναγνωρίζει πια την δική μας αρχή πιστοποίησης TEI CRETE CA.

Μέχρι εδώ έχουμε επιτύχει μισό αποτέλεσμα, δίοτι ναι μέν ο web server είναι ασφαλής αλλά οι χρήστες που μπαίνουν δεν έχουν κάποιο πιστοποιητικό ώστε να εξασφαλισθεί πλήρες ασφάλεια μέσω του καναλίου ssl. Το επόμενο βήμα είναι να δημιουργήσουμε ένα πιστοποιητικό για κάποιον χρήστη μέσω της αρχής μας και να το εισάγουμε στον browser του χρήστη. Έτσι, ο χρήστης θα μπορεί να αποκτήσει ασφαλή πρόσβαση στον ssl web server. Επίσης σαν δεύτερη επιλογή το πιστοποιητικό δεν θα είναι αποθηκευμένο στον browser του χρήστη αλλά θα το περάσουμε σε μία ασφαλή συσκεύη eToken, την οποία θα χρησιμοποιεί ο χρήστης για να μπεί στον web server μας. Όμως πρίν παρουσιάσουμε την παραπάνω διαδικασία θα πρέπει να κάνουμε κάποιες τελευταίες ρυθμίσεις στον web server μας ώστε να είναι έτοιμος να δεχθεί πιστοποίηση ssl από την μερία του χρήστη.

5.4 Ρυθμίζοντας τον SSL Web Server για πιστοποίηση SSL από την πλευρά του χρήστη

Για να επιτύχουμε την πιστοποίηση του χρήστη στον web server μας θα πρέπει, εκτός από το πιστοποιητικό που θα έχει ο χρήστης, να κάνουμε κάποιες ρυθμίσεις στο httpd.conf αρχείο του apache. Οι ρυθμίσεις αυτές θα μας αποδείξουν στο τέλος αν ο χρήστης, που έχει στη διαθεσή του ένα ψηφιακό πιστοποιητικό, μπορεί να αποκτήσει ασφαλή πρόσβαση στον web server μας.

Ο έλεγχος, για το αν ο χρήστης μπορεί να αποκτήσει ασφαλή πρόσβαση στον web server, θα γίνει με τον εξής τρόπο. Θα ορίσουμε ένα path στο οποίο για να μπεί κάποιος θα χρειάζεται ssl πιστοποίηση.

Για να το επιτύχουμε ανοίγουμε το httpd.conf αρχείο του apache και το τμήμα που αναφέρεται στο cgi-bin path το τροποποίουμε ώστε να περιέχει τα εξής.

```
<Directory "c/apache/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
SSLOptions +StdEnvVars
SSLVerifyClient require
SSLVerifyDepth 5
</Directory>
```

Η γραμμή που αναφέρεται στο "**SSLVerifyClient require**" ενημερώνει τον apache ότι η πιστοποίηση του χρήστη θα πρέπει να γίνει πρίν ο χρήστης μπέι σε αυτό το Path. Η γραμμή που αναφέρεται στο "**SSLVerifyDepth**" ενημερώνει τον apache σχετικά με το πόσα επίπεδα εμπιστοσύνης πρέπει υπάρχουν μεταξύ της αρχή πιστοποίησης και του πιστοποιητικού.

Όταν γίνουν αυτές οι αλλαγές σταματάμε και ξαναξεκινάμε τον apache. Θα προσπαθίσουμε να αποκτήσουμε πρόσβαση στο path "/cgi-bin/printenv" το οποίο βρισκέται στο https://localhost/cgi-bin/printenv και ουσιαστικά να δούμε αν στο στάδιο που βρισκόμαστε τώρα μπορούμε να αποκτήσουμε πιστοποιημένη πρόσβαση σαν χρήστες. Δίνοντας λοιπόν το παραπάνω path εμφανίζεται το εξής λάθος

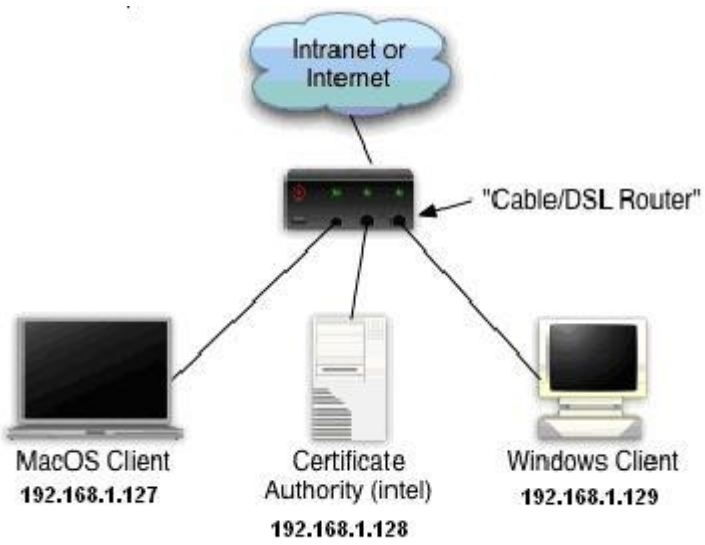
Το οποίο μας παραπέμπει στο ότι δεν μπορεί να επιτυχθεί μια ασφαλή σύνδεση μεταξύ web server και χρήστη. Κάτι απολύτος λογικό διότι δεν έχουμε ακόμα κάποιο πιστοποιητικό για τον χρήστη οπότε δεν μπορεί να γίνει καμία πιστοποίηση και έτσι δεν έχουμε πρόσβαση στο παραπάνω path.

Για να επιτύχουμε λοιπόν την πιστοποίηση από την μερία του χρήστη θα πρέπει να αποκτήσουμε ένα ψηφιακό πιστοποιητικό.

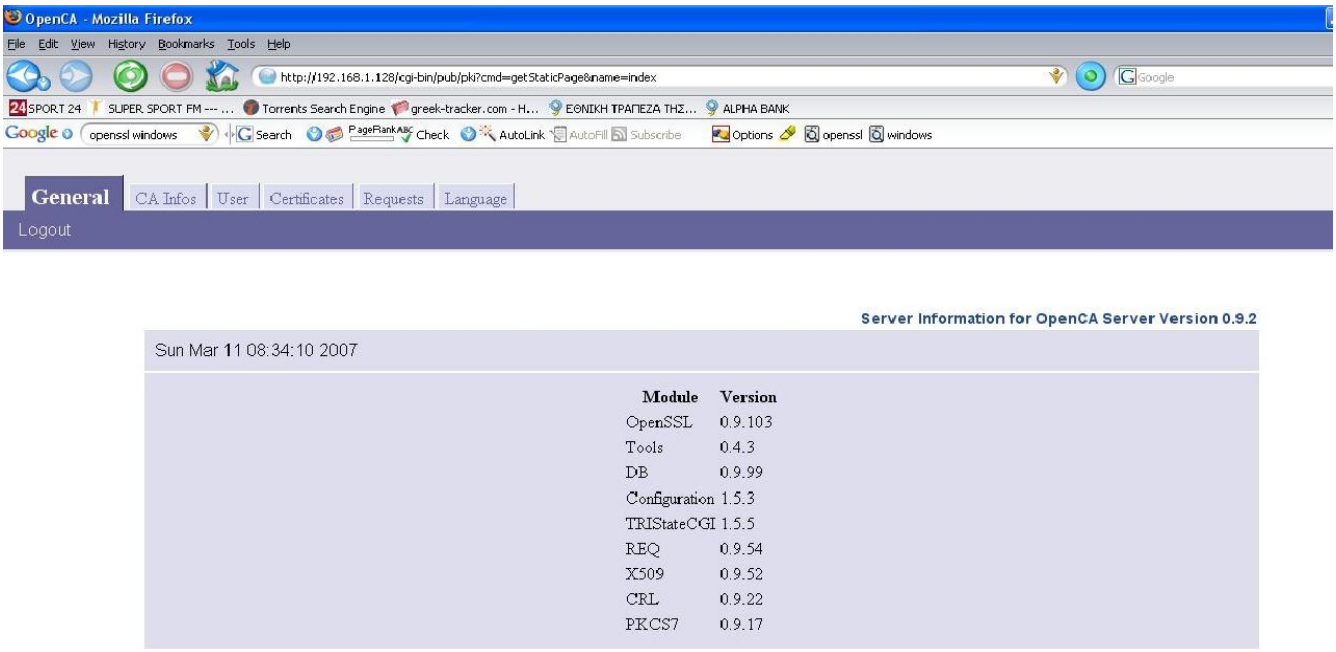
6. Δημιουργία πιστοποιητικού για τον χρήστη

6.1 Αίτηση Πιστοποιητικού χρήστη

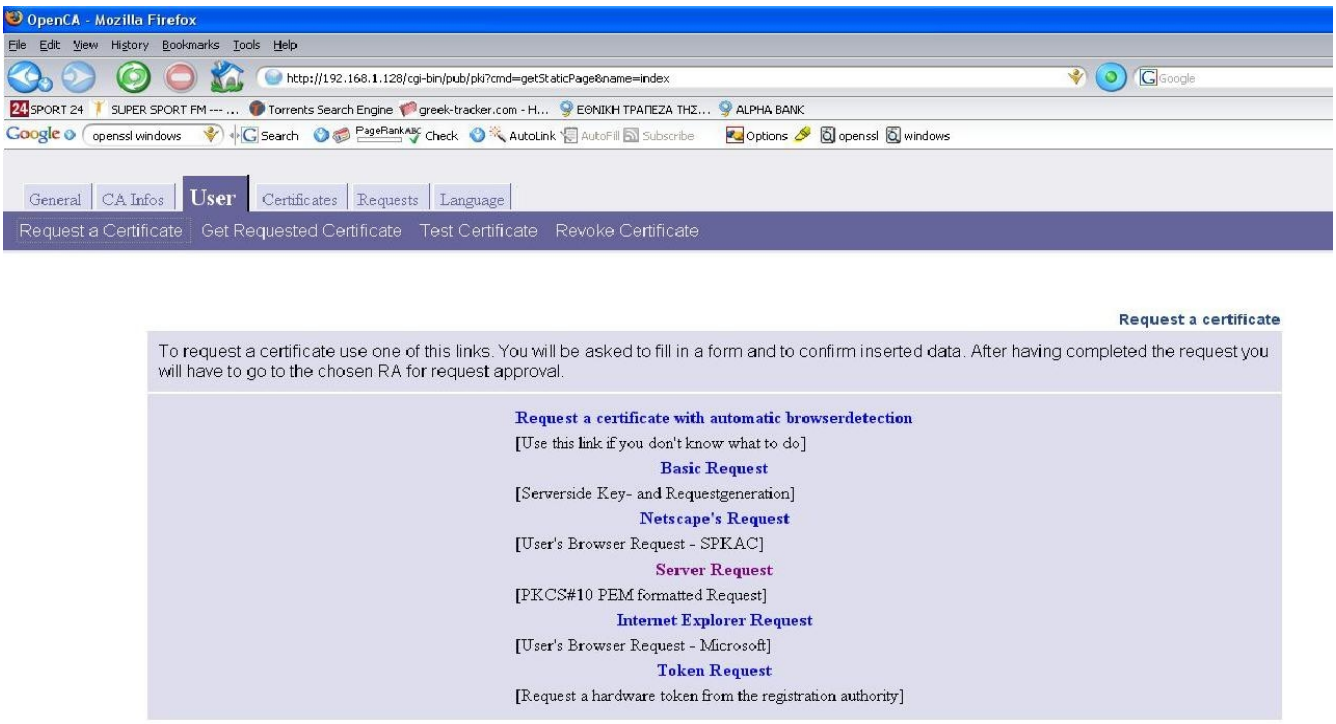
Για να πάρουμε ένα πιστοποιητικό σαν χρήστες θα πρέπει να συνδεθούμε στο σύστημα της αρχής πιστοποίησης. Αύτη η συνδεση μπορεί να επιτευχθεί με δύο τρόπους. Ο πρώτος είναι να μπούμε στο σύστημα που έχουμε την αρχή πιστοποίησης (όπως παρατηρούμε και στο παρακάτω σχήμα) να ανοίξουμε ένα Browser και να δώσουμε http://localhost/pub ή να επιλέξουμε απο την καρτέλα των bookmarks στα αριστερά OpenCa:pub. Ο δεύτερος τρόπος είναι να μπούμε σε ένα υπολόγιστη που ανήκει στο δίκτου που είναι εγκατεστημένο το OpenCa και να δώσουμε την ip του υπολογιστή που φιλοξενή την αρχή πιστοποίησης η οποία στην περίπτωση μας είναι 192.168.1.128, οπότε δίνουμε http://192.168.1.128/pub.



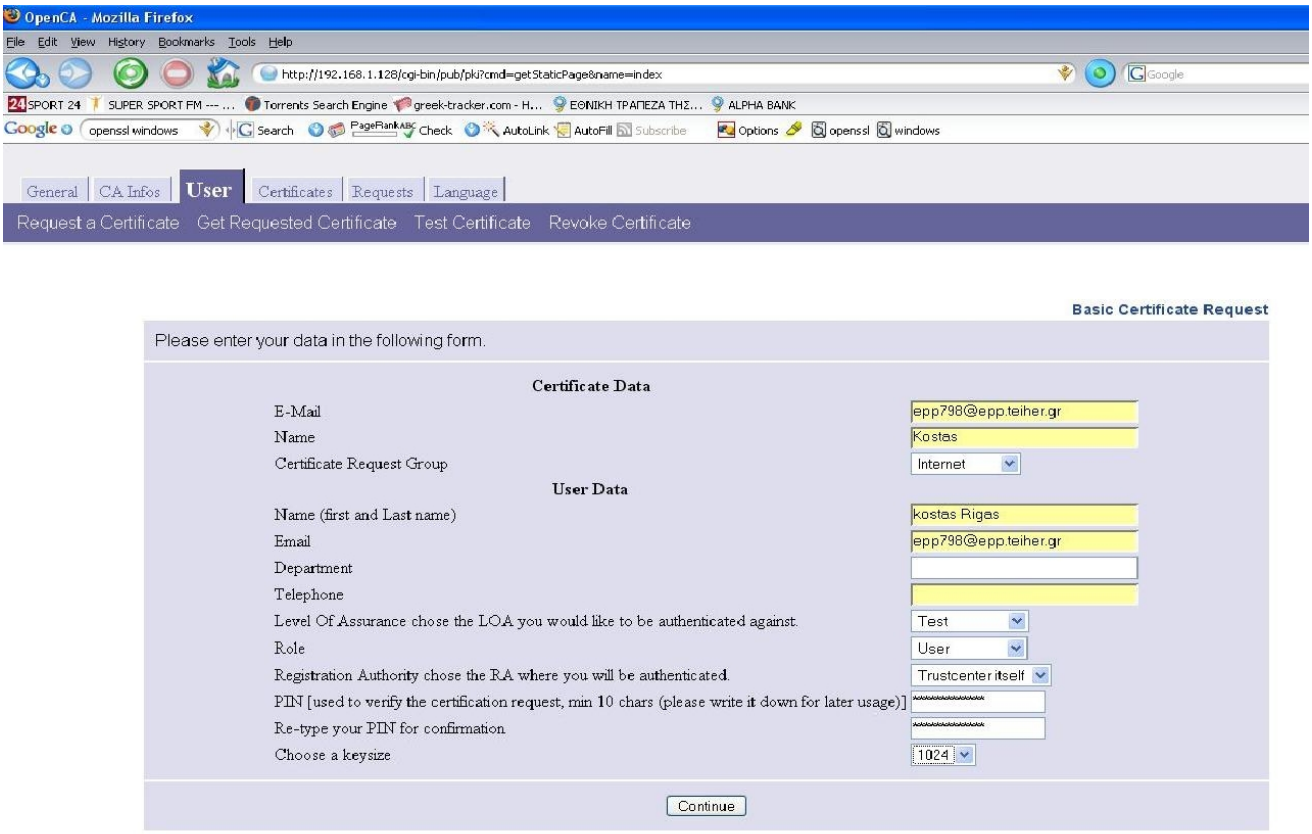
1. Συνδεόμαστε στην αρχή πιστοποίησης στο /pub με ένα από τους δύο παραπάνω τρόπους. Τα παρακάτω screenshoots αναφέρονται στο δέυτερο τρόπο, ανεξαρτήτως τον τρόπο σύνδεσης το αποτέλεσμα που θα πάρουμε θα είναι το ίδιο.



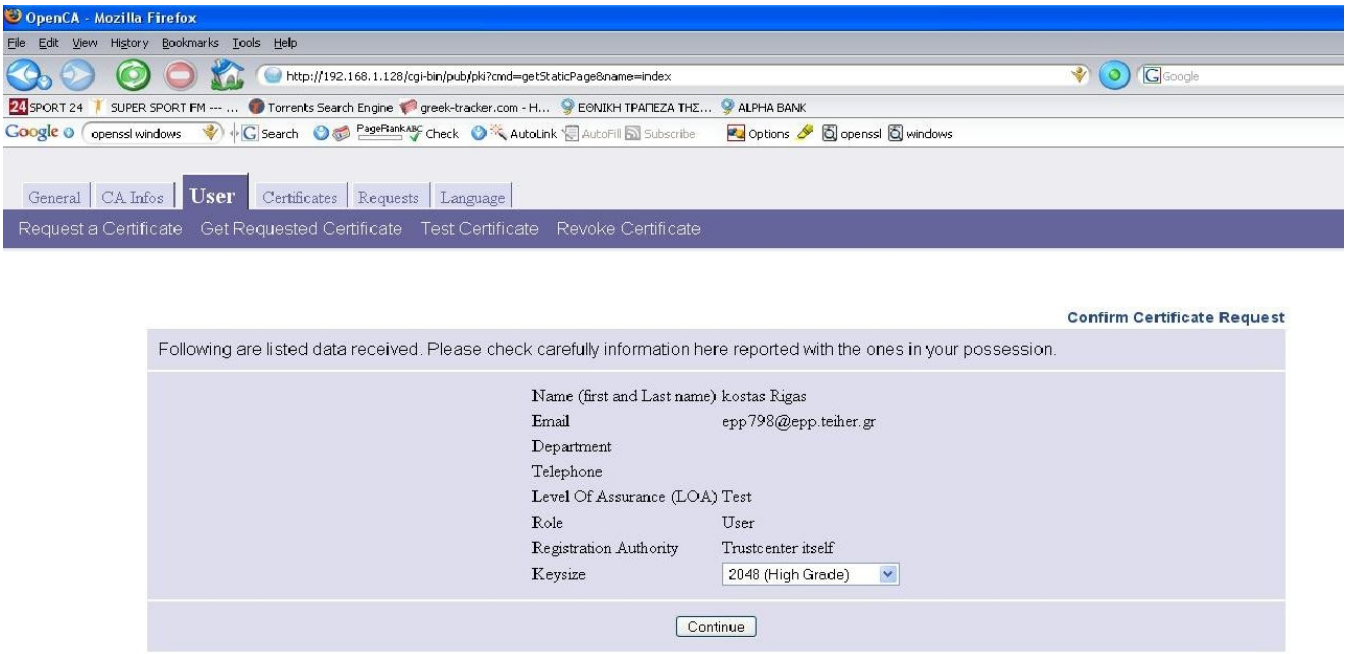
2. Επιλέγουμε την καρτέλα "**User**" και μετά την επιλογή "**Request a certificate**"



3. Επιλέγουμε "**Request a certificate with automatic browser detection**"



4. Συμπληρώνουμε τα απαραίτητα πεδία και επιλέγουμε σαν **Role -> User** επιλέγουμε "**Continue**", επιβεβαίωνουμε την αίτηση.



Η συνέχεια εξαρτάται από τον browser μας, ο mozilla ζητάει ένα password για την προστασία του ιδιωτικού κλειδίου και στη συνέχεια μας εμφανίζει μια οθόνη επιβεβαίωσης.

Certificate Request Confirm

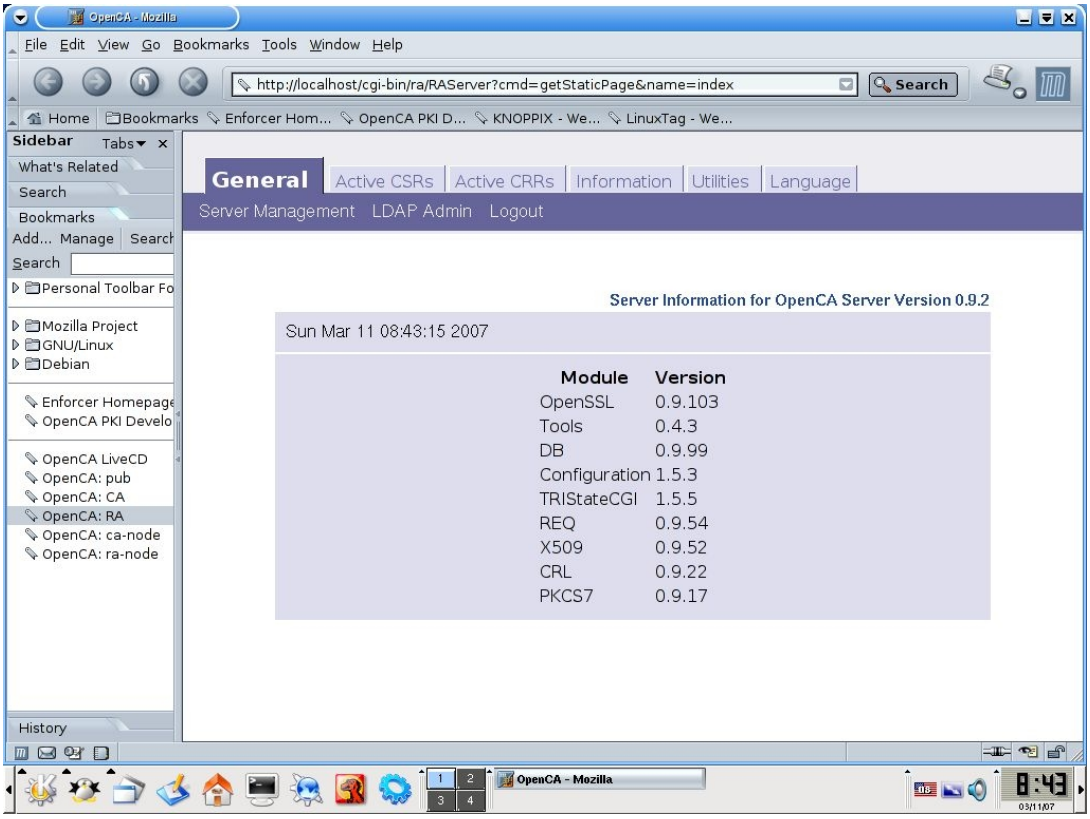
Thank you for requesting your certificate from our organization, your request with the serial 1056 it's been successfully archived and it is now waiting for approval by any of our Registration Authorities (if you are unsure about the receiving of your request by this server, you can check the list of new requests).
To complete the certification process you have to go to one of our Registration Authority office with one of the following documents: o ID card or passport. o Documnetation asserting your role and authorization for requesting a certificate for your organization. If you still have doubts about the issuing process, just use the links provided in the Information section to learn how to complete all the needed steps.

ADDITIONAL_ATTRIBUTE_DEPARTMENT	
ADDITIONAL_ATTRIBUTE_EMAIL	epp798@epp.teiher.gr
ADDITIONAL_ATTRIBUTE_REQUESTERCN	kostas Rigas
ADDITIONAL_ATTRIBUTE_TELEPHONE	
LOA	10
NOTBEFORE	Sun Mar 11 13:40:48 2007 GMT
PIN	b4ea3b3965e53dc76b1888015aa7ba83b1e3278a
RA	Trustcenter itself
ROLE	User
SERIAL	1056
TYPE	SPKAC

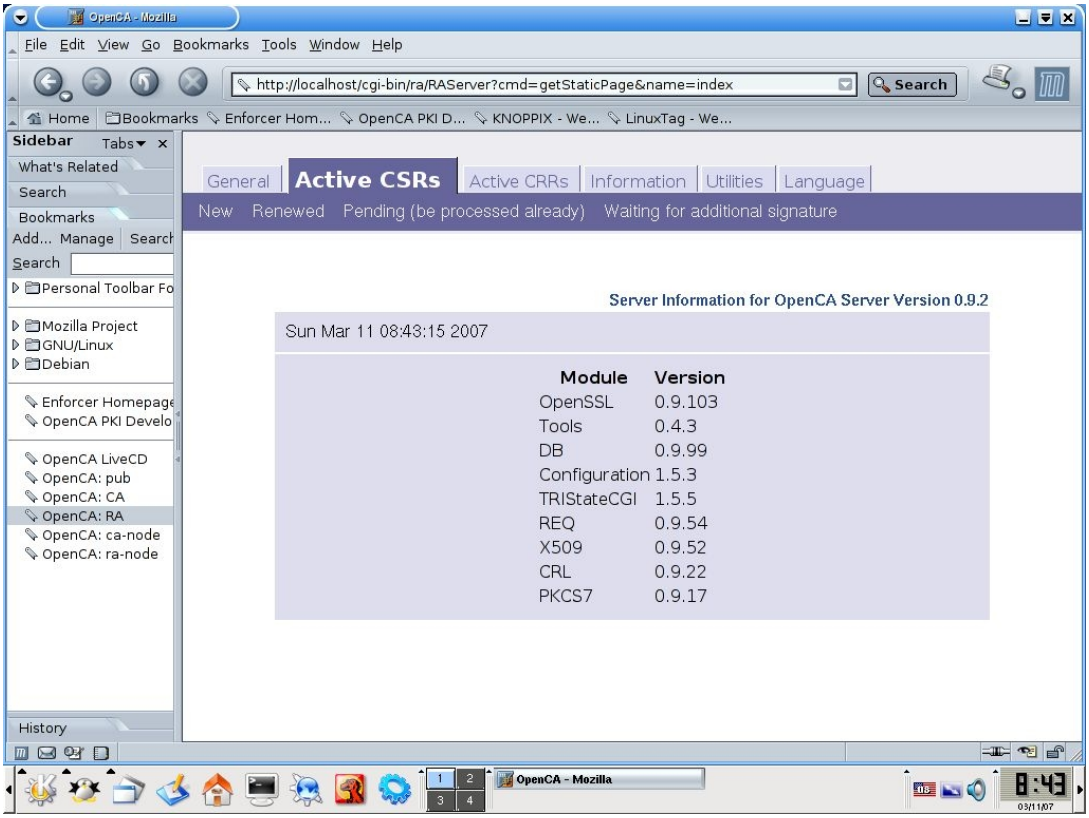
Print

6.2 Αποδοχή του Πιστοποιητικού

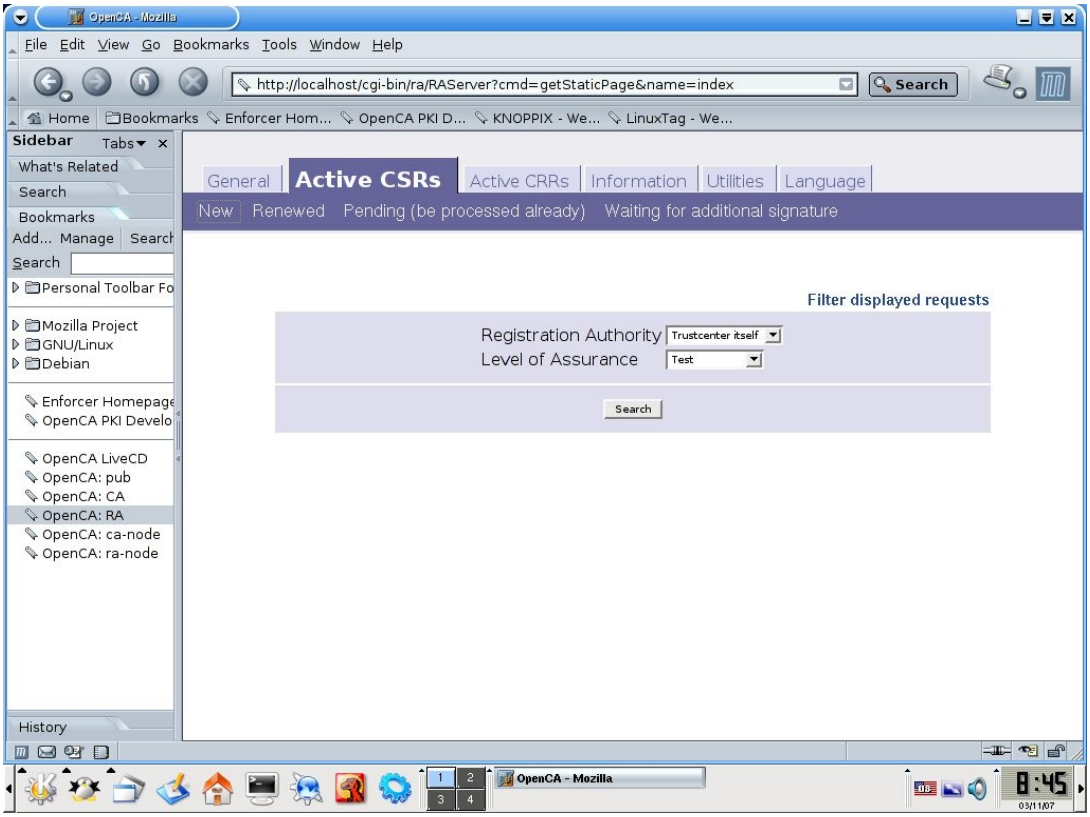
1. Συνδεόμαστε στο http://localhost/ra/ ή επιλεγούμε από την αριστερή στήλη των bookmarks OpenCa:RA (μέσω του υπολογιστή που περιέχει την αρχή πιστοποίησης)



Επιλέγουμε "**Active CSRs** "

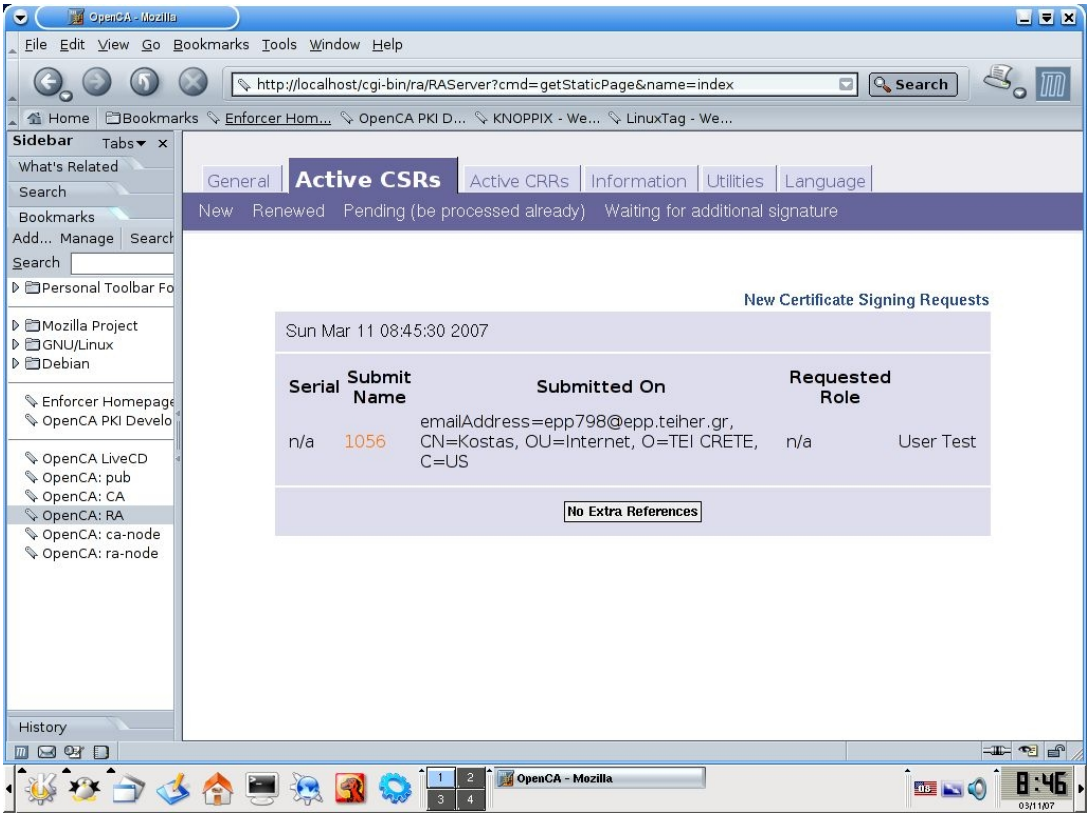


Στη συνέχεια επιλέγουμε "**New**"



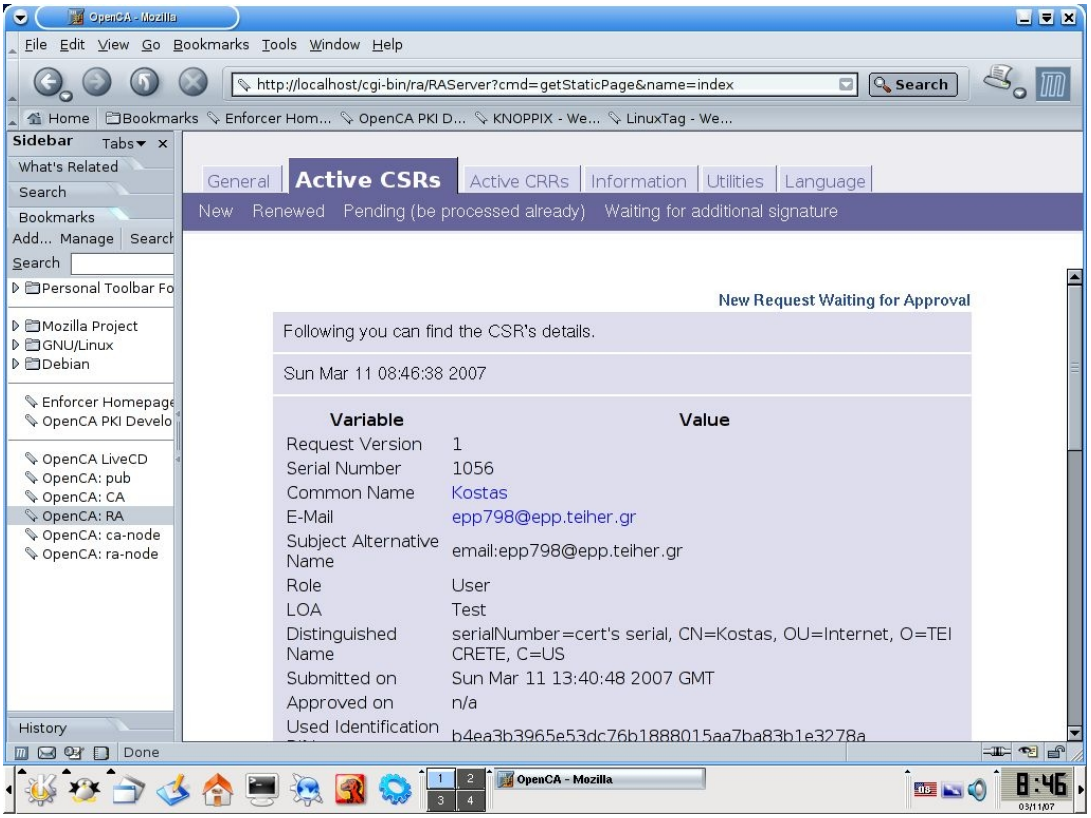
Εμφανίζεται μια σελίδα αναζήτησης, αν δεν έχει αλλαχθεί το "**Trust center**" ή το "**Security level**" της αίτησης μπορούμε να χρησιμοποιήσουμε τις προεπιλεγμένες επιλογές (Registration Authority:Trustcenter itself και Level of Assurance:Test), επιλέγουμε "Search".

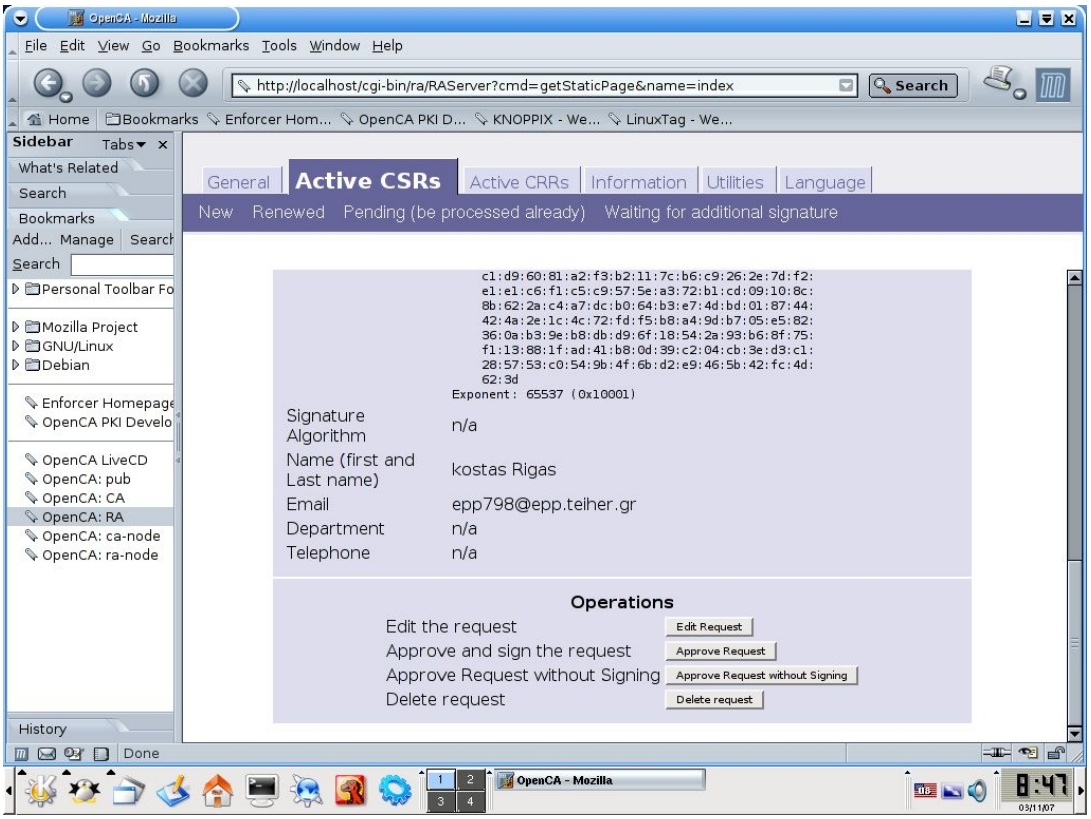
2. Μια λίστα από πιστοποιητικά θα πρέπει να εμφανισθεί.



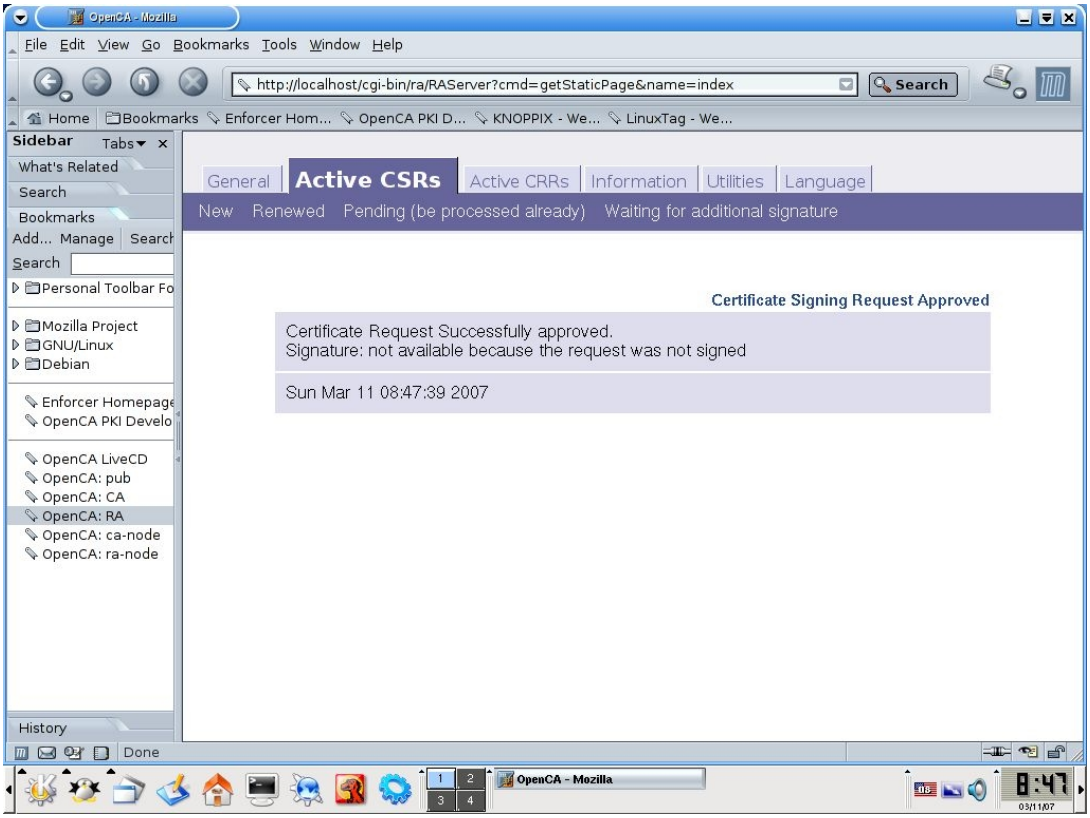
Επιλέγουμε τον αριθμό του πιστοποιητικού με το οποίο θέλουμε να ασχοληθούμε.

3.Ελέγχουμε αν το πιστοποιητικό που έχει εμφανισθεί είναι το επιθυμητό





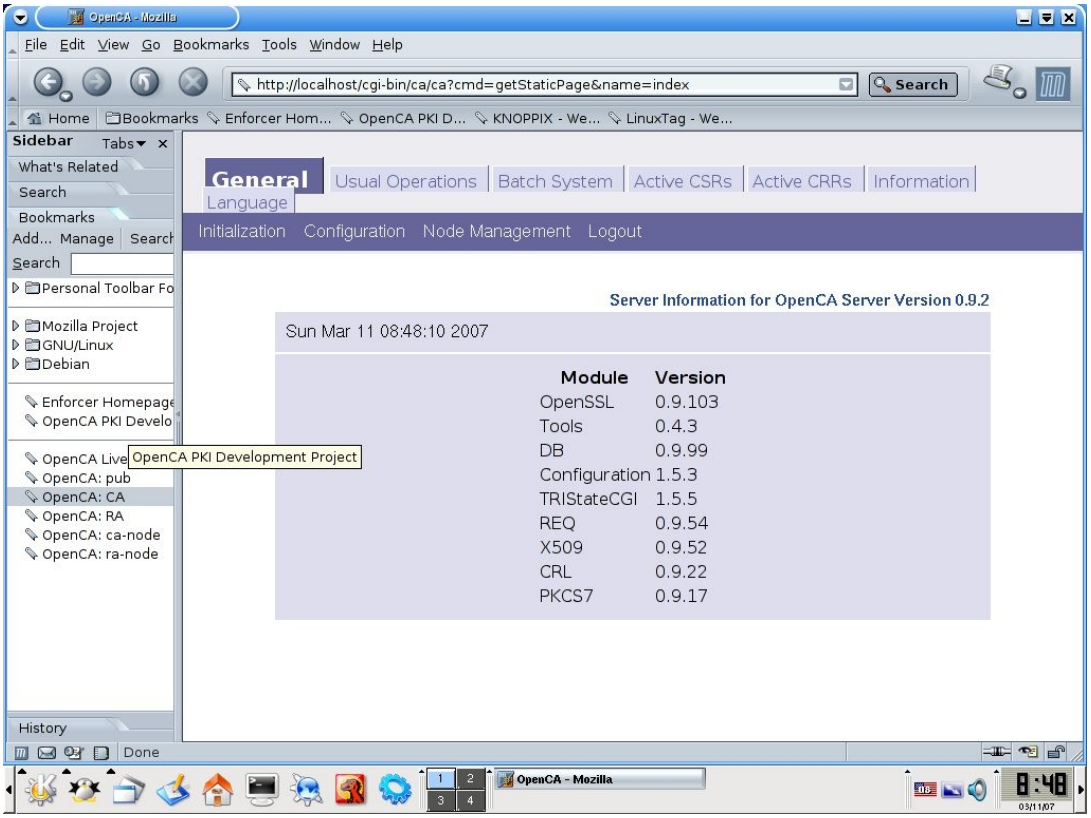
Στη συνέχεια επιλέγουμε "**Approve Request without signing**" (ο mozilla που παρέχει το live cd δεν προσφέρει το SecCLAB plugin, όπως αναφερθήκαμε και παραπάνω, ώστε να επιλέξουμε "**Approve and signing request**"). Εμφανίζεται μια οθόνη επιβεβαίωσης



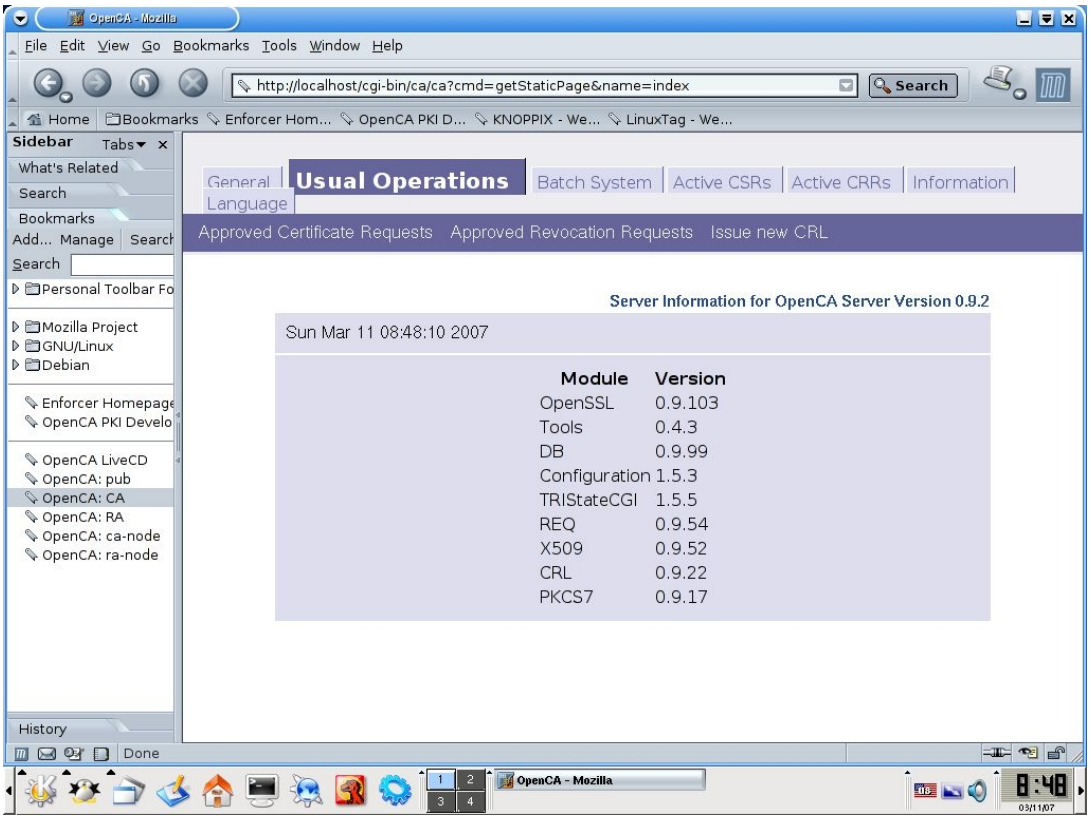
6.3 Επικύρωση του πιστοποιητικού

Αυτός που μπορεί να επικυρώσει πιστοποιητικά είναι ο CA (root) οπότε θα πρέπει να συνδεθούμε (μέσω του μηχανηματος που έχουμε εγκαταστησει την αρχή πιστοποίησης) και να επικυρώσουμε το παραπάνω πιστοποιητικό.

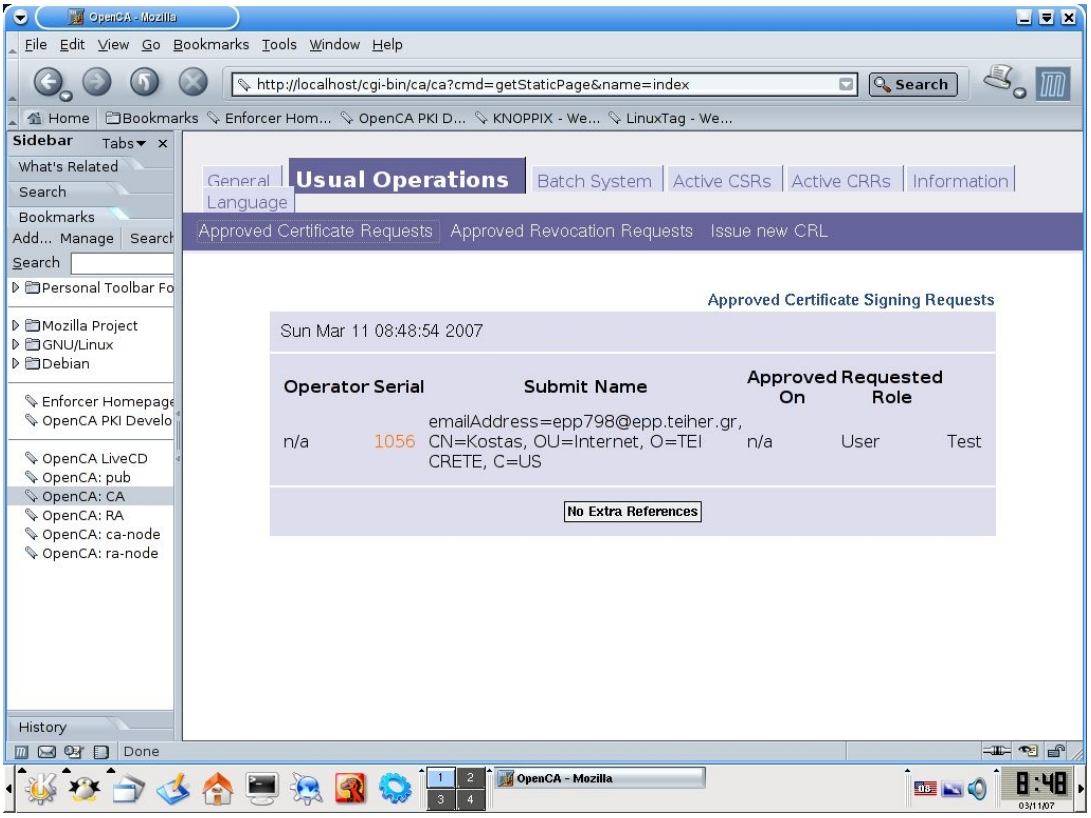
1. Συνδεόμαστε στο http://localhost/ca/ ή μέσω της αριστερής στήλης των bookmarks επιλέγουμε OpenCA:CA



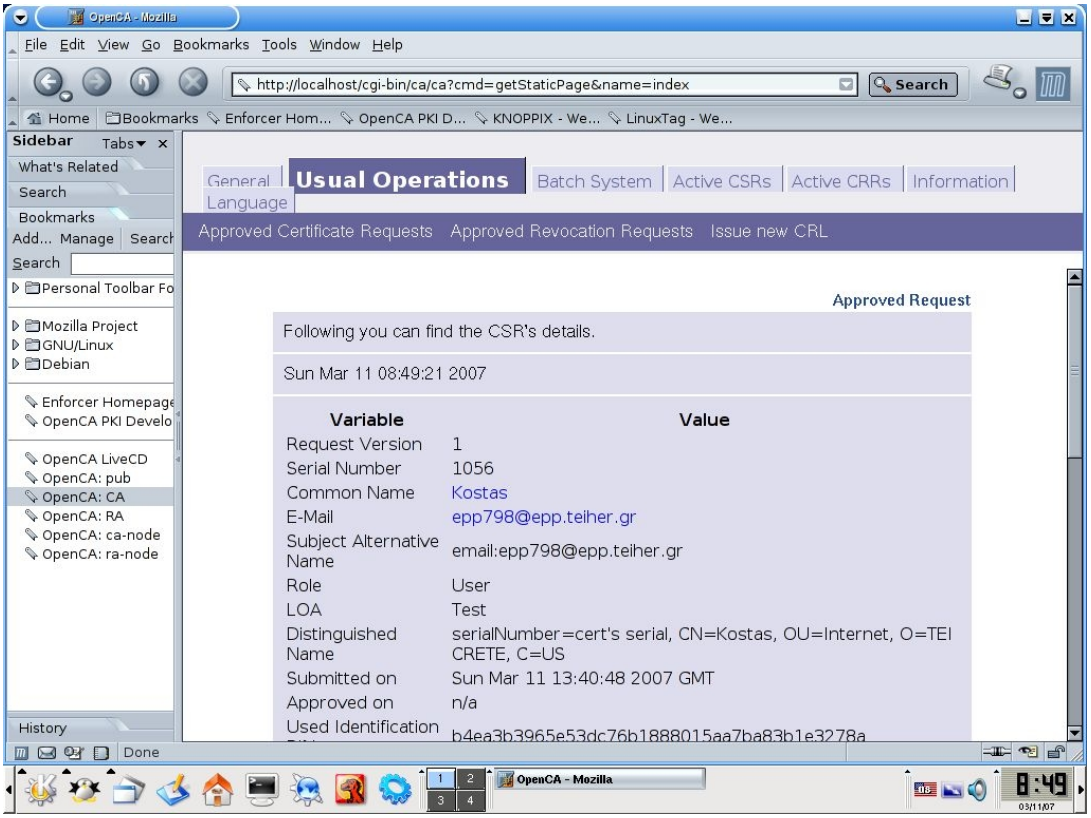
Στη συνέχεια επιλέγουμε την καρτέλα "**Usual Operations**" και μετά επιλέγουμε "**Approved Certificate Requests**"

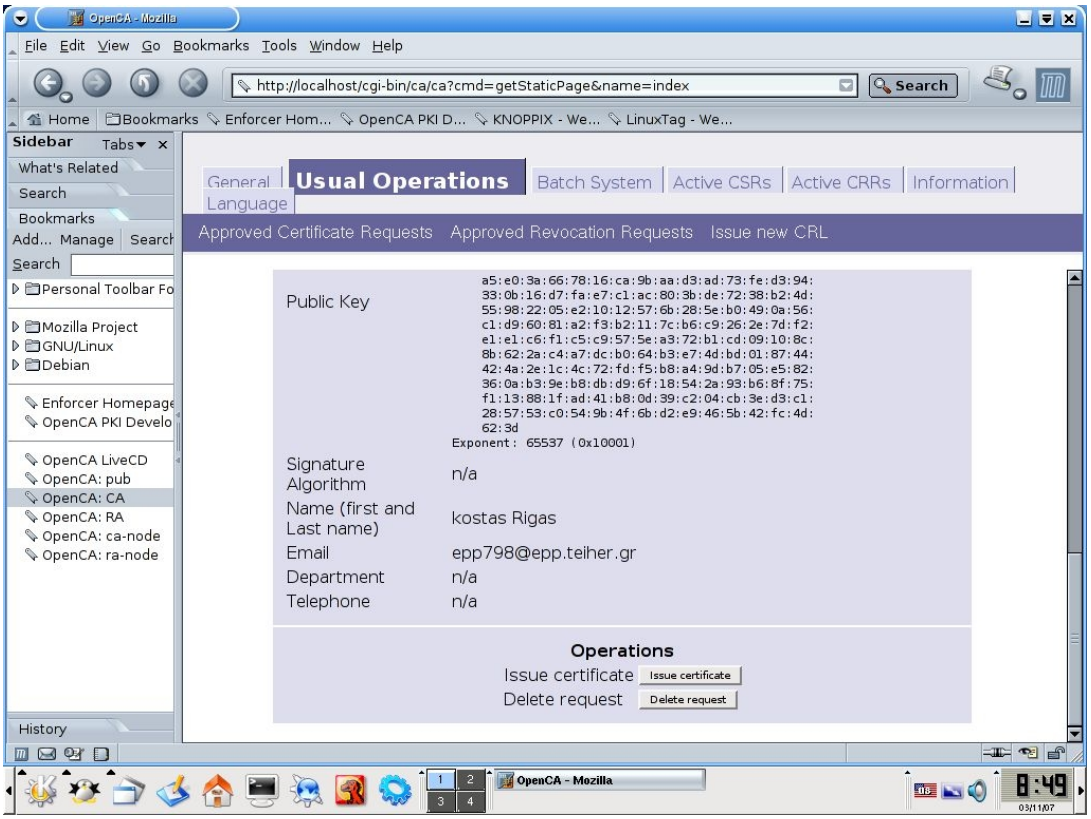


2. Μια λίστα από πιστοποιητικά θα πρέπει να εμφανισθεί , επιλέγουμε τον αριθμό του πιστοποιητικού με το οποίο θέλουμε να ασχοληθούμε



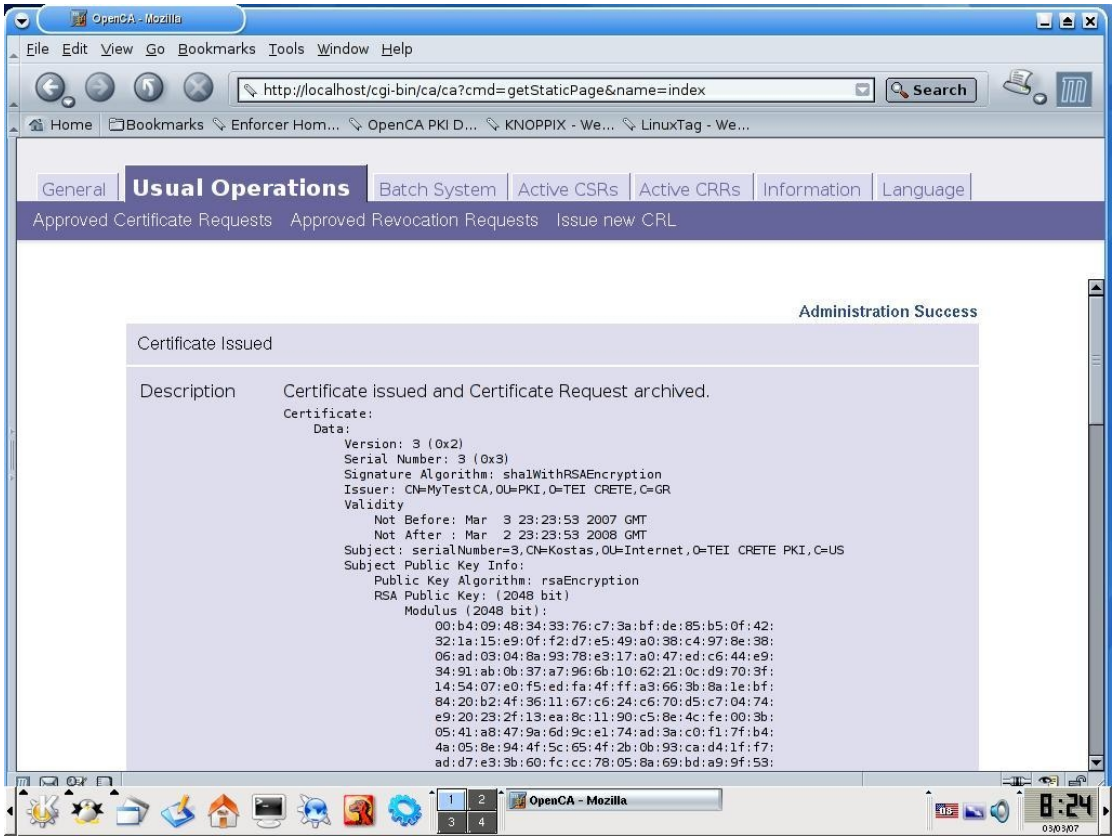
3. Ελέγχουμε αν το πιστοποιητικό που εμφανίσθηκε είναι το επιθυμητό, στη συνέχεια επιλέγουμε "**Issue the certificate**"





Δίνουμε το password του ιδιωτικού κλειδίου του CA ώστε να επιβεβαίωθεί η αίτηση

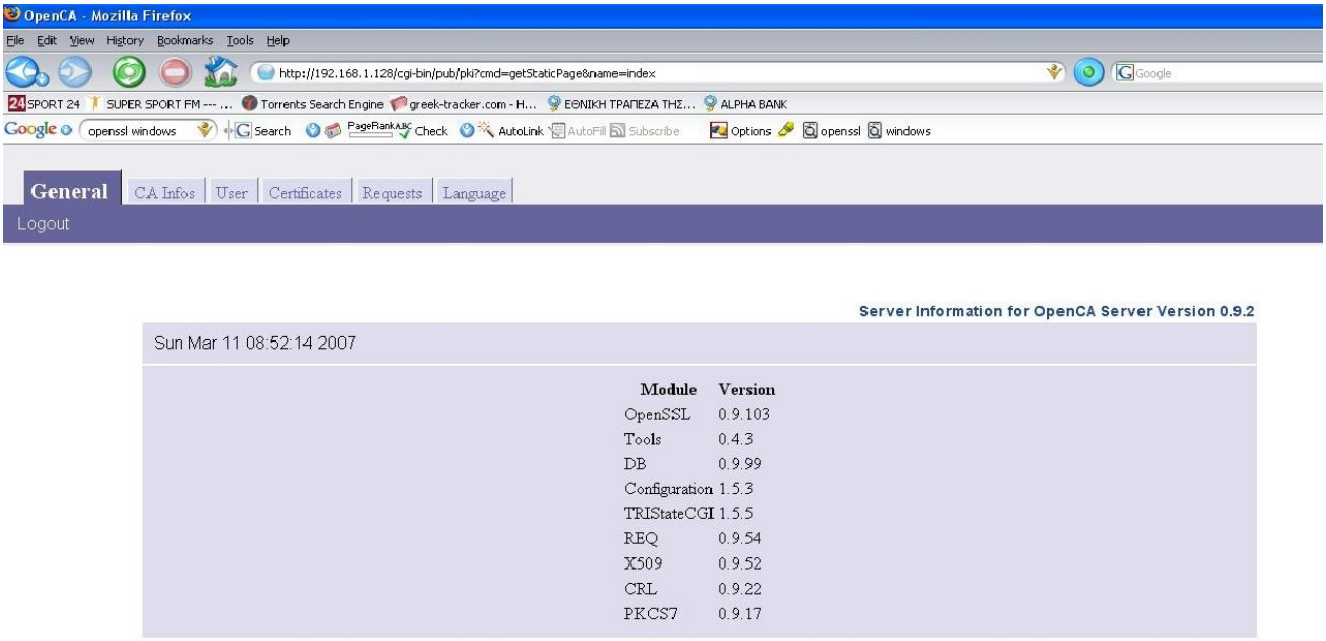
Εμφανίζεται ένα μήνυμα επιτυχίας



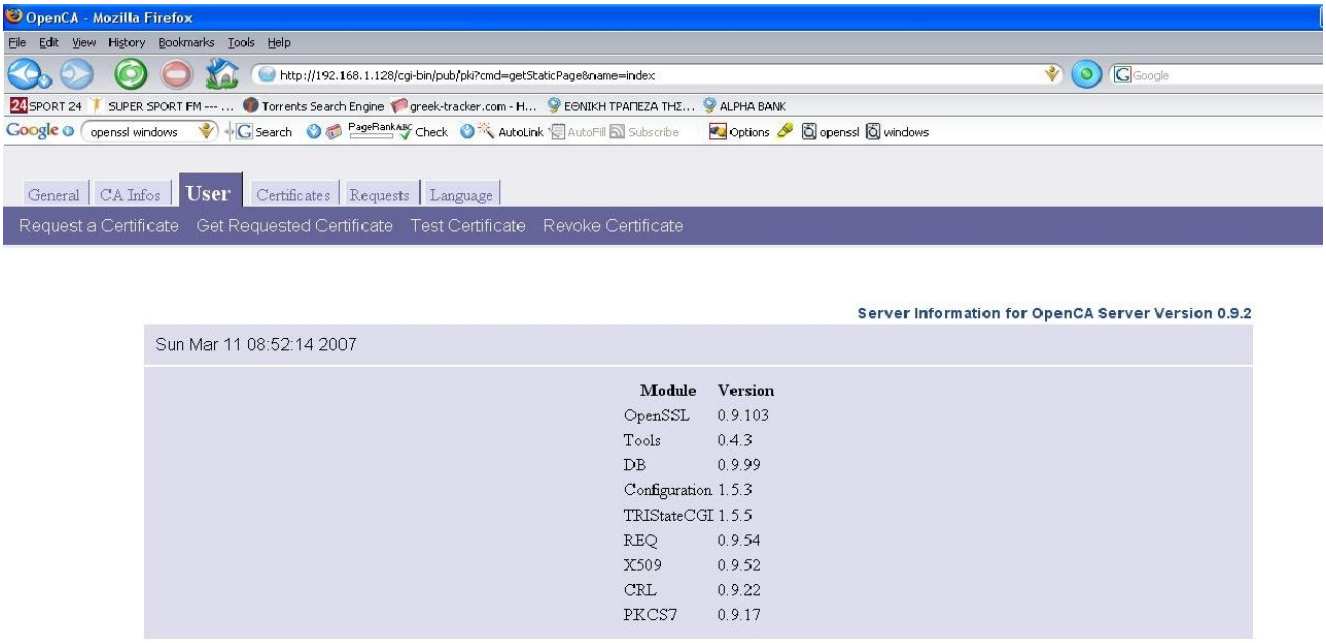
6.4 Εξαγωγή του πιστοποιητικού

1. Συνδεόμαστε στην αρχή πιστοποίησης στο /pub με ένα από τους δύο τρόπους όπως είπαμε και παραπάνω. Ο πρώτος είναι να μπούμε στο σύστημα που έχουμε την αρχή πιστοποίησης να ανοίξουμε ένα Browser και να δώσουμε http://localhost/pub ή να επιλέξουμε απο την καρτέλα των bookmarks στα αριστερά OpenCa:pub. Ο δεύτερος τρόπος είναι να μπούμε σε ένα υπολόγιστη που ανήκει στο δίκτου που είναι

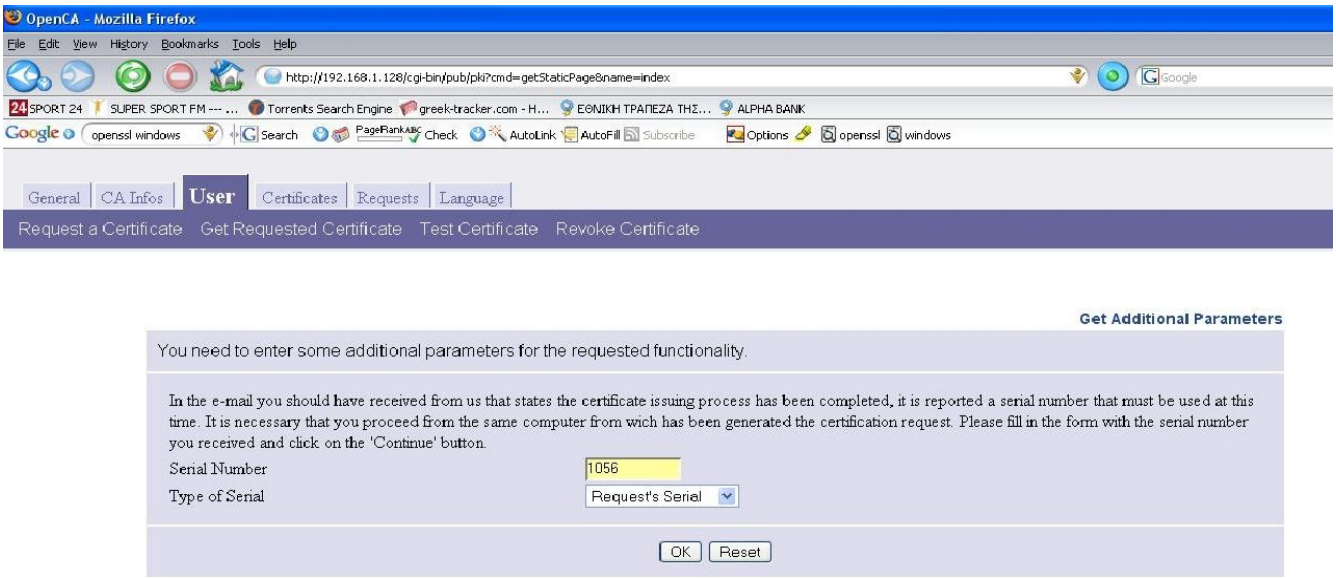
εγκατεστημένο το OpenCa και να δώσουμε την ip του υπολογιστή που φιλοξενή την αρχή πιστοποίησης η οποία στην περίπτωση μας είναι 192.168.1.128, οπότε δίνουμε http://192.168.1.128/pub. Επιλέγουμε τον δεύτερο τρόπο (και οι δύο καταλήγουν στα ίδια αποτελέσματα).



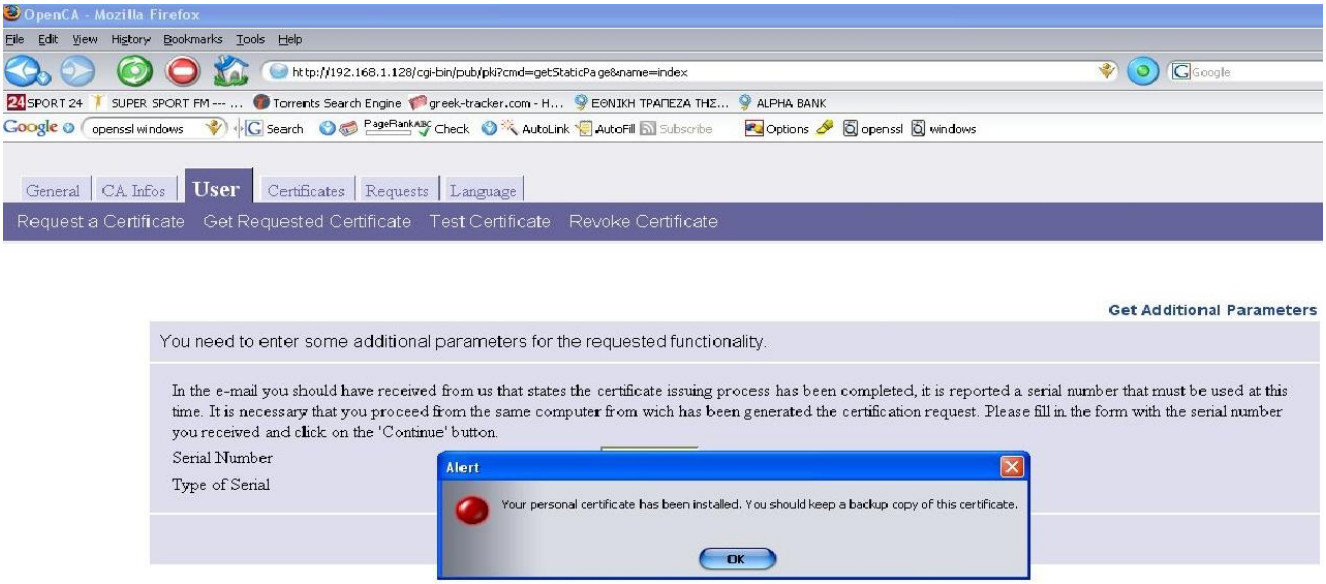
2. Επιλέγουμε την καρτέλα "**User**" και μετά το "**Get Request certificate**"



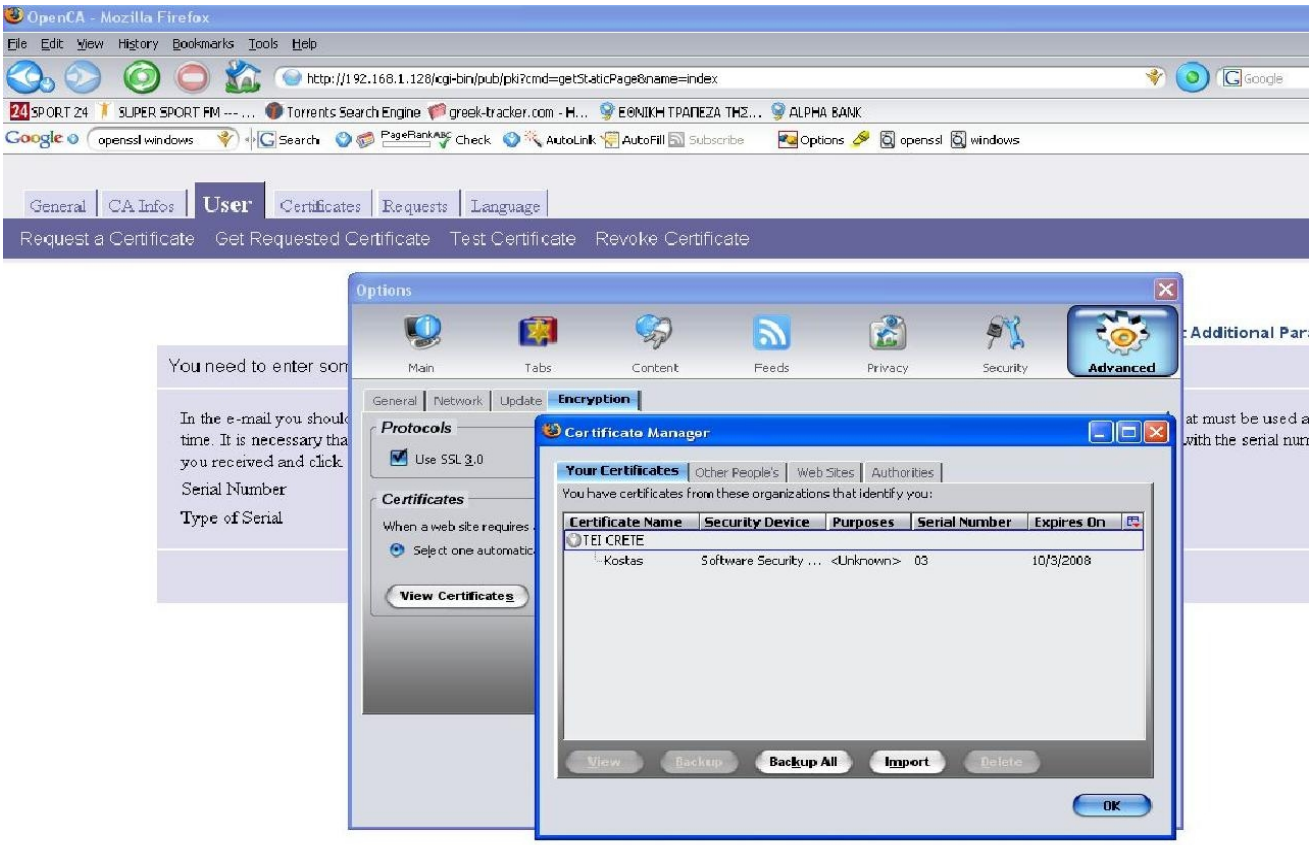
Δίνουμε το **serial number** της αίτησης και επιλέγουμε στο "**Type of Serial**" την επιλογη **Request serial**



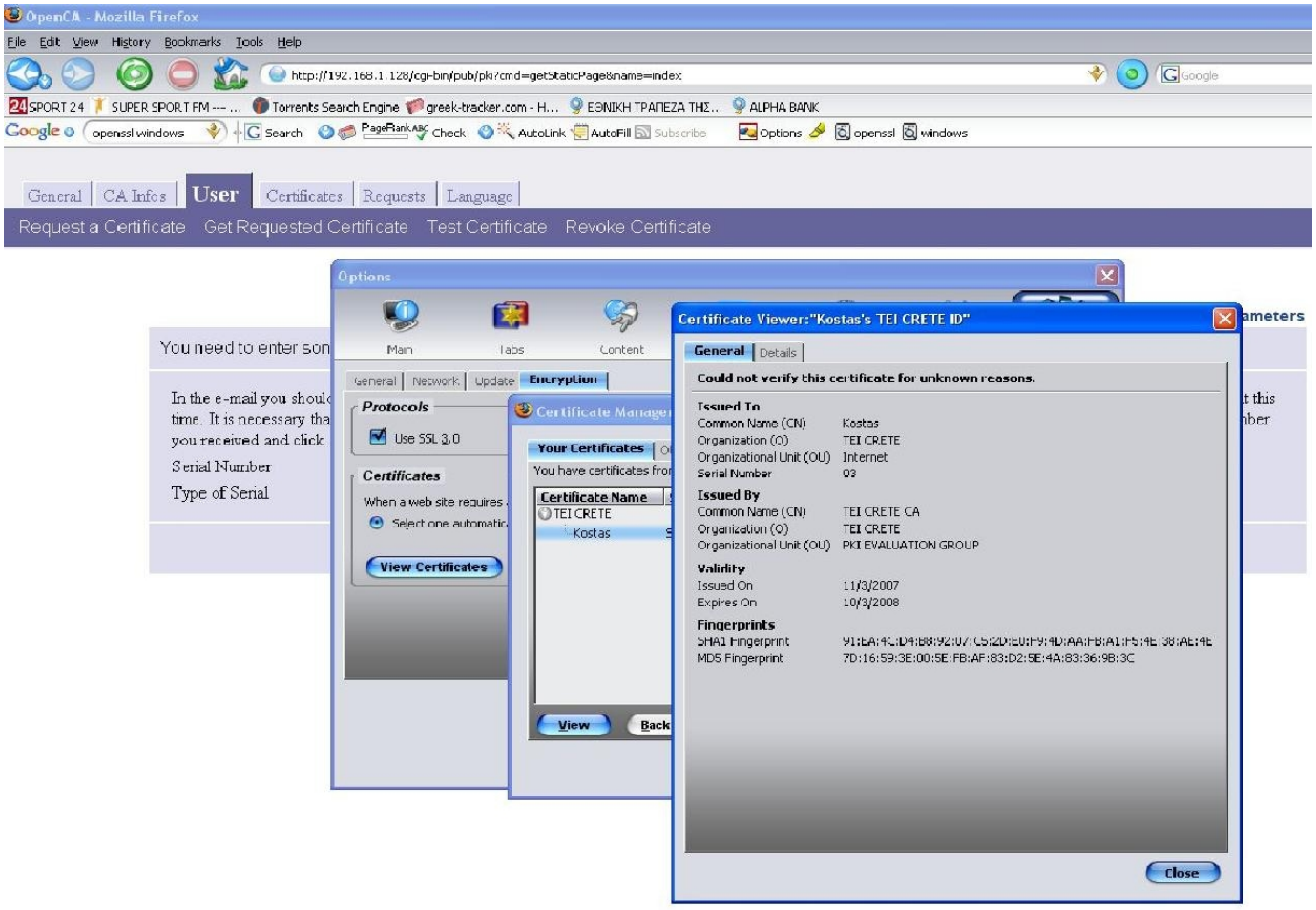
Εμφανίζεται μια οθόνη επιβεβαίωσης



3. Επιβεβαιώνουμε αν το πιστοποιητικό υπάρχει στον mozilla για την έκδοση 2.0.0.2 επιλέγουμε **Tools-> Options-> Advanced-> καρτέλα Encryption**



Επιλέγοντας το πιστοποιητικό και πατώντας “View” εμφανίζονται επιμέρους πληροφορίες για το πιστοποιητικό.



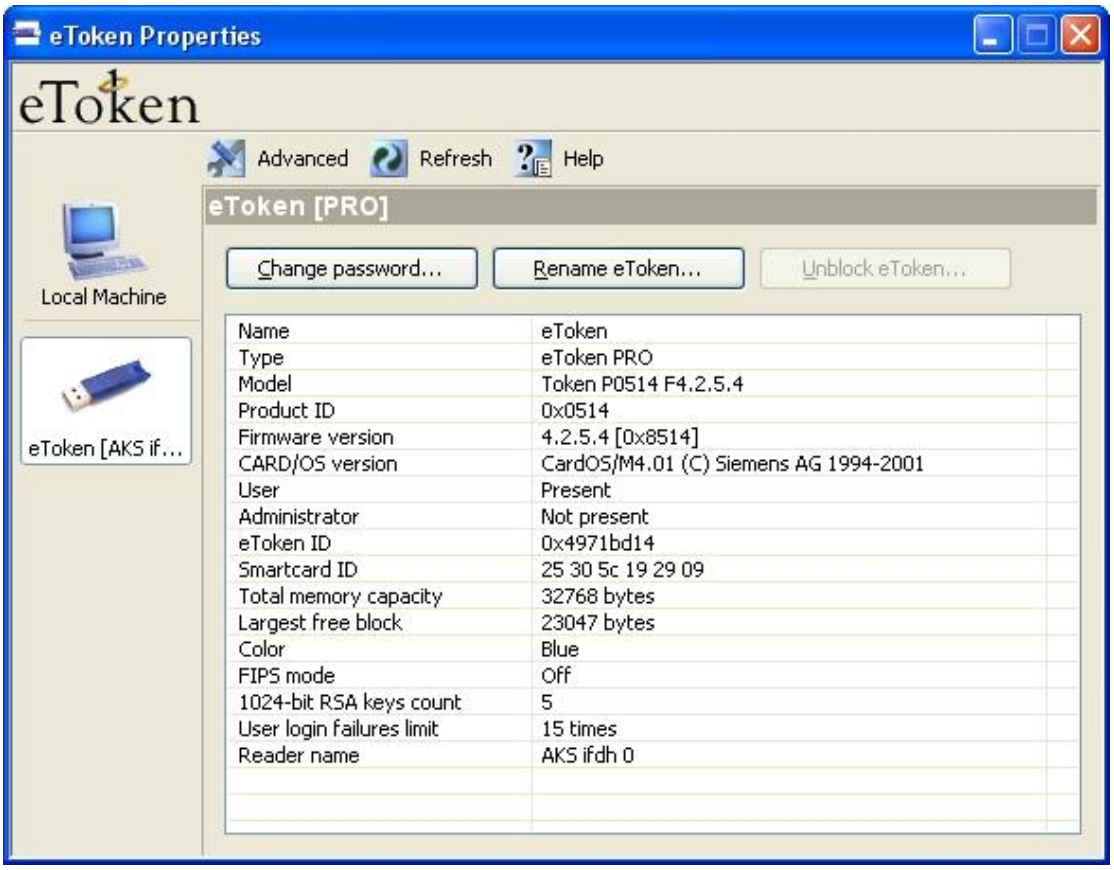
7. Δημιουργία πιστοποιητικού για τον χρήστη με την χρήση του εξοπλισμού eToken Pro 32k της Aladdin

7.1 Σχετικά με την συσκευή eToken Pro 32k της Aladdin

Πρίν αναφερθούμε στη διαδικασία της δημιουργίας και επικυρωσης ενός ψηφιακού πιστοποιητικού με την χρήση μίας συσκευής ασφαλούς αποθήκευσης, θα ασχοληθούμε λίγο με το πώς λειτουργεί και τι τελικά μας προσφέρει η συσκεύη eToken της εταιρίας Aladdin.

Η συσκευή αυτή όπως παρατηρούμε και παραπάνω μοίαζει με ένα usb flash drive (σε μικρότερο μέγεθος) όμως διαφέρει στην χωρητικότητα η οποία στην περιπτωσή μας είναι της τάξης των 32 kilobyte. Παρόλα αυτά μπορεί να δημιουργήσει κρυπτογραφημένα ζευγάρια κλειδιών. Όταν ζητήσουμε ένα ψηφιακό πιστοποιητικό από μια αρχή πιστοποιήσης με την βοήθεια κάποιου browser σε λειτουργίκο σύστημα Windows Xp, η συσκευή χρησιμοποιείται έκτος από την αποθήκευση του πιστοποιητικού και στη δημιουργία του δημόσιου/ιδιωτικού ζευγαρίου κλειδίων.

Για την ομαλή λειτουργίας της συσκευής σε περιβάλλον Windows Xp πρέπει να χρησιμοποιήσουμε το software της εταιρίας το οποίο ονομάζεται RTE (Run Time Environment). Μέσω αυτού του software έχουμε τη δυνατότητα να διαχειριστούμε πλήρως την συσκεύη etoken, δηλαδή μπορούμε να βάλουμε password για την διασφάλιση των πιστοποιητικών μας, να διαγράψουμε πιστοποιητικά και γενικότερα μας παρέχει ότι χρειαζόμαστε για να αξιοποιήσουμε στο μέγιστο την συσκευή, παρακάτω βλέπουμε μια εικόνα με την εφαρμογή.



Σε αυτό το περιβάλλον τα πιστοποιητικά καθώς και τα ιδιωτικά κλειδία αποθηκεύονται με ένα ιδιοκτησιακό τρόπο, αυτό σημαίνει ότι μόνο ο υπολογιστής που έχει το RTE software μπορεί να έχει πρόσβαση και διαχείριση των περιεχόμενων της συσκεύης. Η εφαρμογή RTE είναι απαραίτητη για την χρήση του etoken και με την βοήθεια της θα μπορέσουμε να αποθηκεύσουμε και να διαχειρίστουμε τα ψηφιακά μας πιστοποιητικά, τα οποία θα έχουμε πάρει από την δική μας αρχή πιστοποίησης.

Για την ασφάλεια της συσκευής, όπως αναφέραμε και προηγουμένος, θα πρέπει να δώσουμε ένα password. Έτσι ώστε όποιος θέλει να αποκτήσει πρόσβαση στην συσκευή θα πρέπει να γνωρίζει αυτό το password. Εργοστασιακά η συσκευή περιέχει ένα password το 1234567890. Μόλις ο χρήστης ανοίξει την εφαρμογή του eToken τότε θα του ζητηθεί να αλλάξει το εργοστασιακό password, έτσι ώστε να αποκτήσει πρόσβαση σε όλα τα επιμέρους τμήματα της εφαρμογής που διαχειρίζεται το eToken.

Κατά την διαδικασία εισαγωγής του password μας εμφανίζεται ένα ποσοστό επί τοίς εκατό το οποίο μας ενημερώνει (με άριστα το 100 %) πόσο ασφαλές και δύσκολο στο να παραβιαστεί είναι το password το οποίο δίνουμε. Επίσης η εφαρμογή, μας προτείνει να χρησιμοποιήσουμε τουλάχιστον οκτώ χαρακτήρες, οι οποίοι να αποτελούνται από ένα συνδιασμό κεφαλάιων και μικρών γραμμάτων καθώς και

αριθμών και συμβόλων, έτσι ώστε το password της συσκευής να είναι όσο το δυνατών πιο ασφαλές

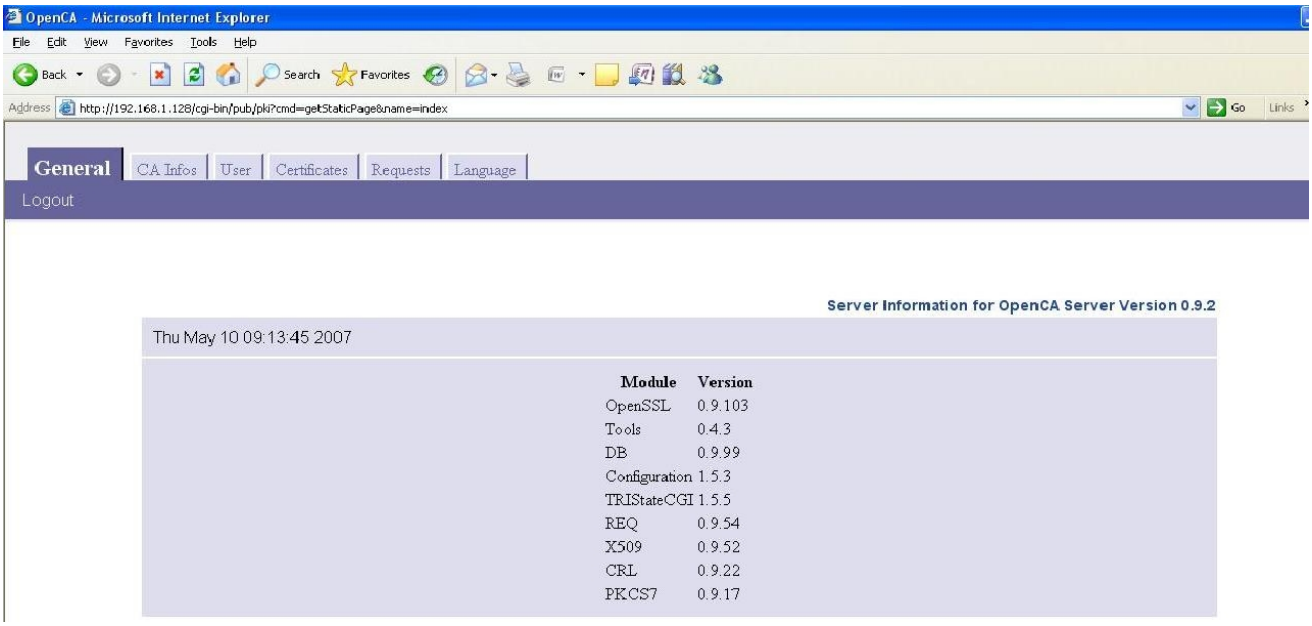
Παρακάτω παρουσιάζεται η διαδικασία δημιουργίας και αποθήκευσης ψηφιακού πιστοποιητικού στην συσκευή eToken Pro 32k. Η διαδικασία περιλαμβάνει όπως και προηγουμένος τα εξής βήματα, αίτηση πιστοποιητικού, αποδοχή του πιστοποιητικού, επικύρωση του πιστοποιητικού και τέλος εξαγωγή του πιστοποιητικού.

7.2 Αίτηση πιστοποιητικού για τον χρήστη

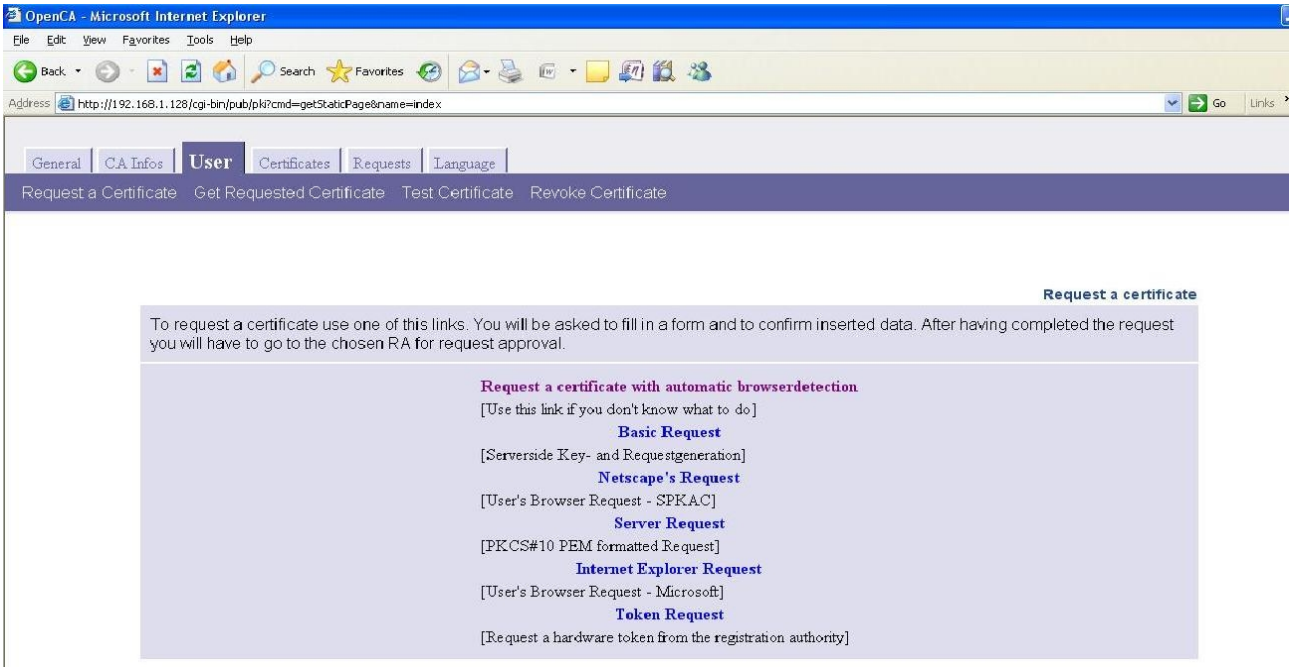
Θα ακολουθήσουμε τα ίδια βήματα όπως κάναμε και προηγουμένος με την μόνη διαφορά ότι τώρα έχουμε συνδέση στον υπολογιστή μας την συσκευή eToken Pro 32k και έχει αναγνωρισθεί από την εφαρμογή RTE.

Προηγουμένος χρησιμοποιήσαμε τον Browser mozilla firefox, τώρα θα χρησιμοποιήσουμε τον browser Internet explorer θέλοντας να δείξουμε ότι η διαδικασία είναι η ίδια και στους δύο browsers.

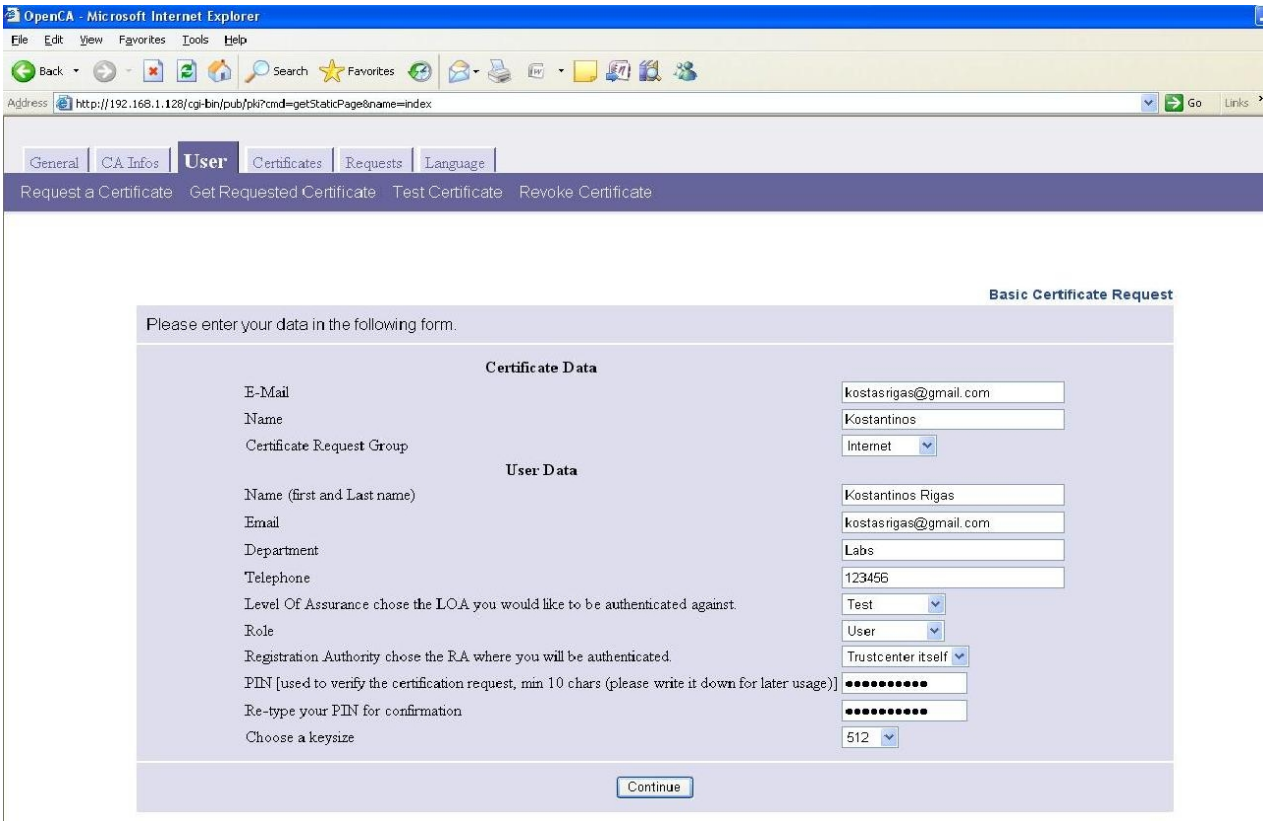
Θα πρέπει λοιπόν ο υπολογιστής μας να "βλέπει" την αρχή πιστοποίησης μέσω του δικτύου μας. Οπότε συνδεόμαστε μέσω του IE στην ip της αρχής πιστοποίησης http://192.168.1.128/pub



Στην συνέχεια επιλέγουμε την καρτέλα **"User"** και μετα την επιλογή **"Request a certificate"** και βλέπουμε την παρακάτω οθόνη.

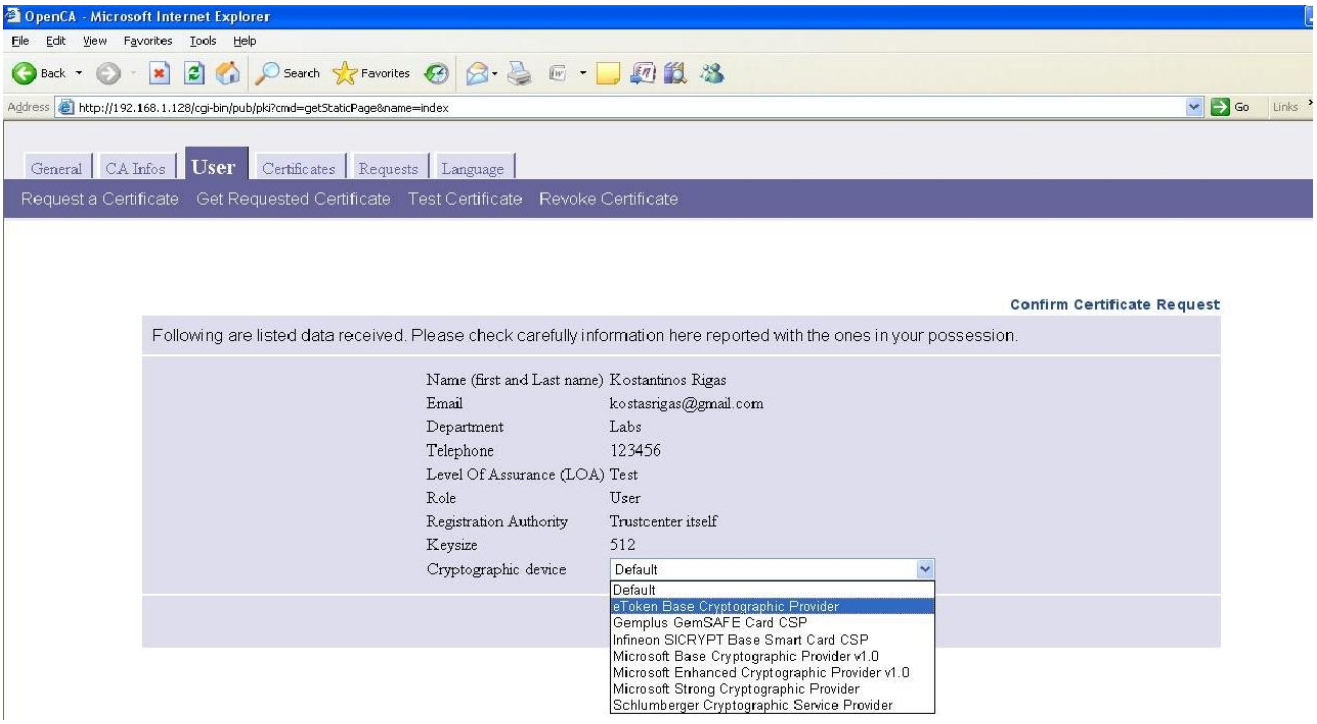


Επιλέγουμε την πρώτη επιλογή "**Request a certificate with automatic browser detection**" στη συνέχεια εμφανίζεται η κύρια φόρμα των στοιχείων που θα έχει το πιστοποιητικό μας την συμπληρώνουμε και στο τέλος στο πεδίο "**Choose a Key Size**" επιλέγουμε την τιμή 512 και αυτό δίοτι το etoken Pro 32K δεν υποστηρίζει μεγαλύτερα κλείδια από 512 σε αντίθεση με το etoken Pro 64K.



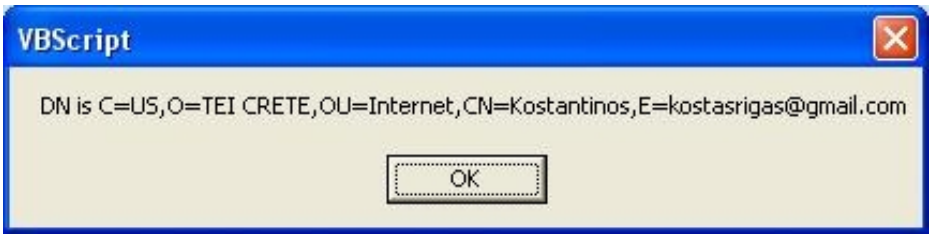
Μέχρι εδώ η διαδικασία είναι ίδια με προηγουμένος, όταν δεν είχαμε δηλαδή την συσκευή etoken επιλέγοντας όμως "**Continue**" μας εμφανίζεται το εξής μήνυμα

Το οποίο μας ενημερώνει ότι ανιχνεύθηκε η συσκευή etoken, επίλεγουμε "**Ok**" και εμφανίζεται η παρακάτω οθόνη στην οποία και επιλέγουμε το όνομα της συσκευής μας.

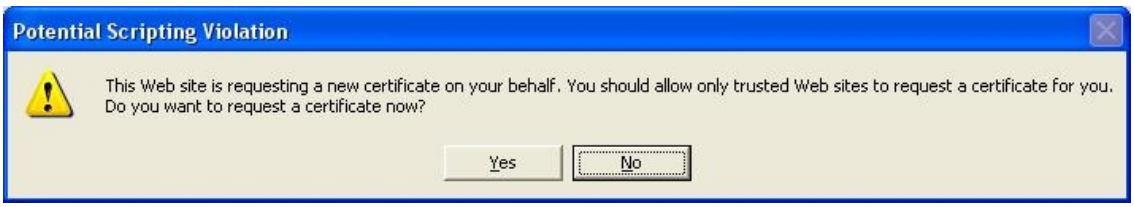


Στην συνέχεια μας εμφανίζονται μηνύματα τα οποία μας ενημερώνουν για την επιτυχία της διαδικασίας.

Επιλέγουμε "**Ok**"



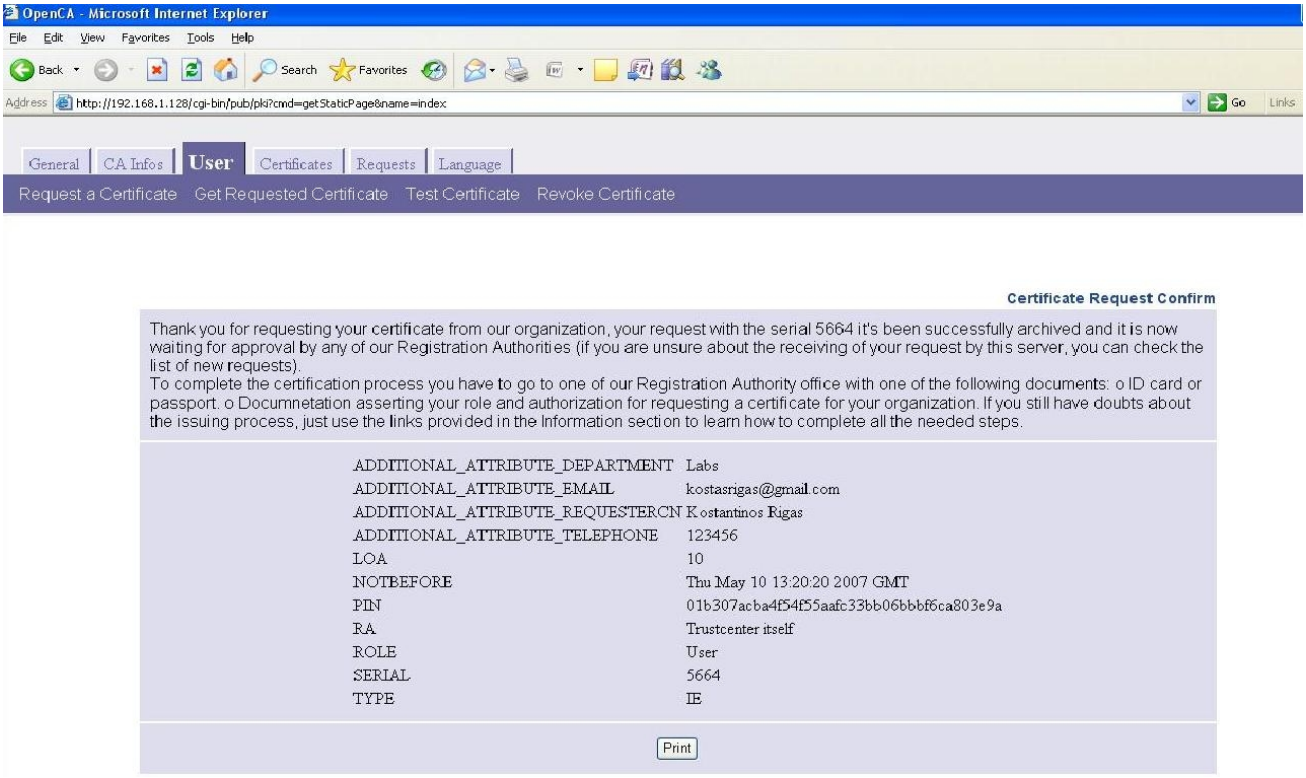
Επιλέγουμε "**Ok**"



Επιλέγουμε "**Yes**" και δίνουμε το Password το οποίο ασφαλίζει την συσκευή μας

Τέλος ενημερωνόμαστε ότι η διαδικασία ήταν επιτυχής

Παραπάνω η συσκευή etoken δημιούργησε το ζέυγος κλειδίων δημόσιο/ιδιωτικό με επιτυχία. Επιλέγοντας "**Ok**" μας εμφανίζεται το τελευταίο μήνυμα αυτού του σταδίου της διαδικασίας το οποίο μας ενημερώνει για την επιτυχή κατάληξη της αίτησης μας.

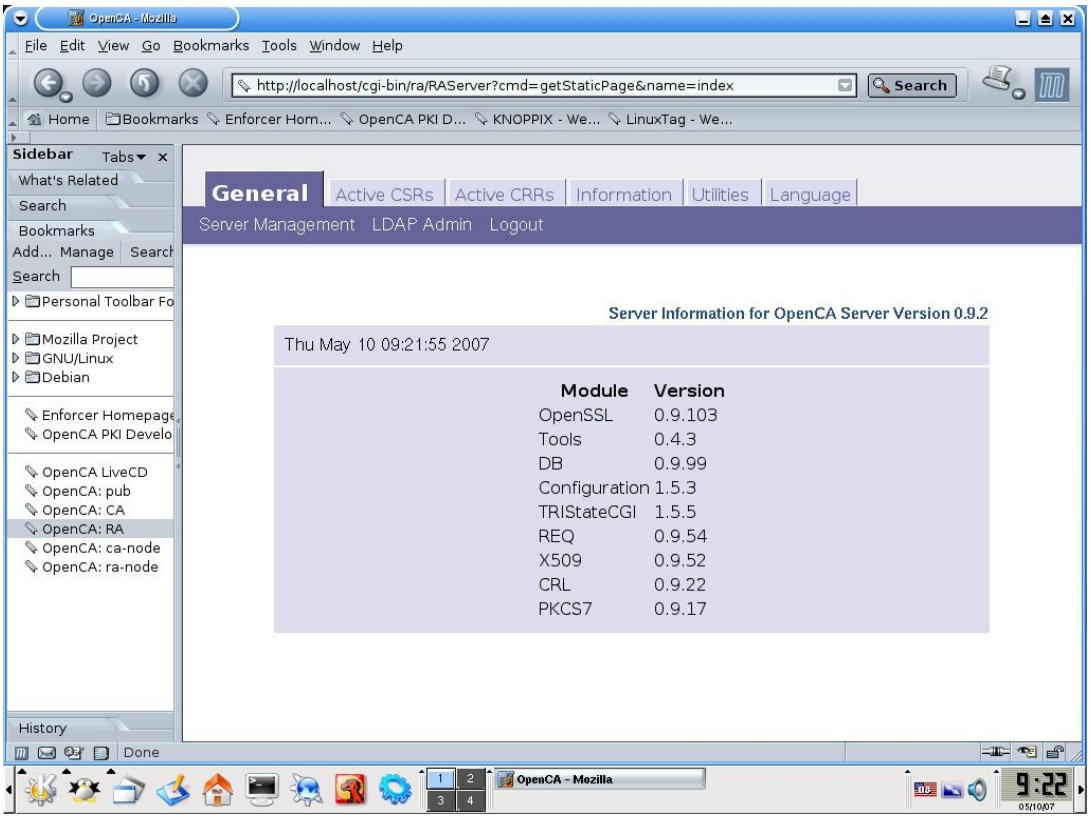


Το πρώτο βήμα τελείωσε οπότε περνάμε στο δεύτερο το οποίο έχει να κάνει με την αποδοχή του πιστοποιητικού από τον RA.

7.3 Αποδοχή του Πιστοποιητικού

Μετά την αίτηση για το πιστοποιητικό η αρχή πιστοποίησης θα πρέπει να αποδεχτεί την αίτηση και να προχωρίσει στην δημιουργία του πιστοποιητικού. Οπότε θα μπούμε στην αρχή πιστοποίησης και μέσω του Ra θα αποδεχτούμε την αίτηση του πιστοποιητικού.

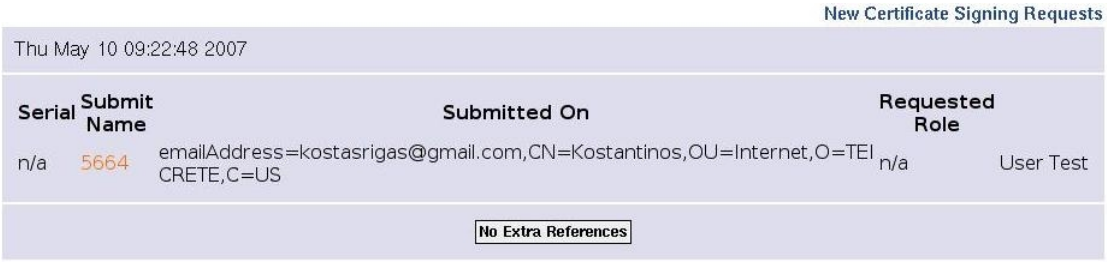
Συνδεόμαστε στο http://localhost/ra/ ή επιλεγούμε από την αριστερή στήλη των bookmarks OpenCa:RA (μέσω του υπολογιστή που περιέχει την αρχή πιστοποίησης)



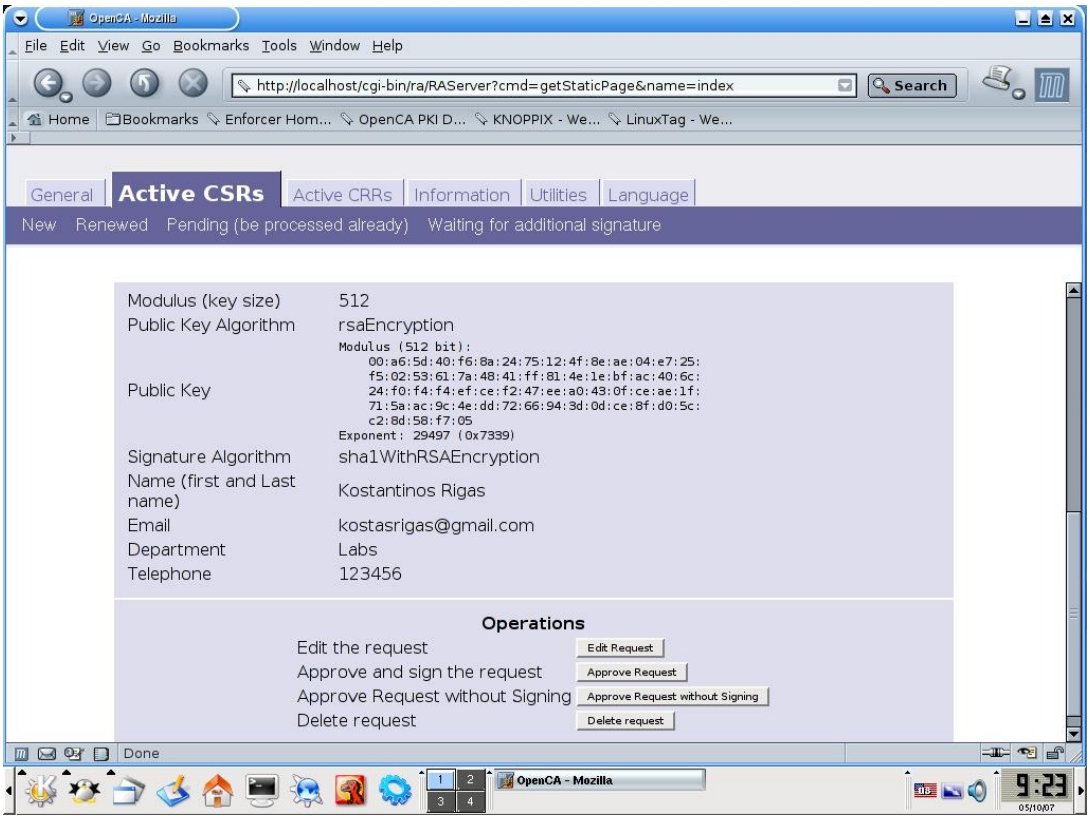
Επιλέγουμε "**Active CSRs**" και στη συνέχεια "**New**", εμφανίζεται η παρακάτω οθόνη

Αφήνουμε τις επιλογές ως έχουν και επιλέγουμε "Search"

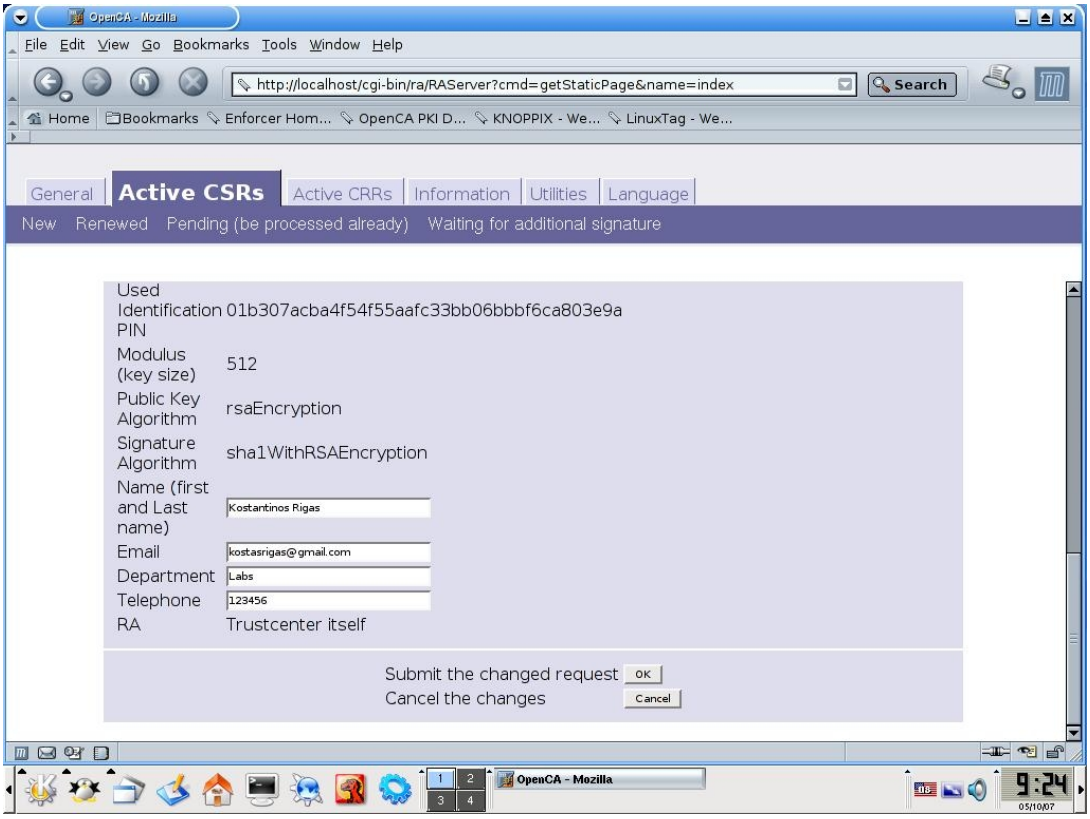
Εμφανίζεται η λίστα με την αίτηση του πιστοποιητικού μας



Για να προχωρίσουμε επιλέγουμε τον σειριακό αριθμό της αίτησης (5664) και περνάμε στην παρακάτω οθόνη η οποία μας δείχνει αναλυτικά τα στοιχεία της αίτησης



Επιλέγουμε “**Edit Request**”, και εμφανίζεται η παρακάτω οθόνη



Επιλέγουμε "**Ok**" και γυρνάμε στην προηγούμενη οθόνη όπου επιλέγουμε "**Approve Request without Signing**" (ο mozilla που παρέχει το live cd δεν προσφέρει το SecCLAB plugin, όπως αναφερθήκαμε και παραπάνω, ώστε να επιλέξουμε "**Approve and signing request**")

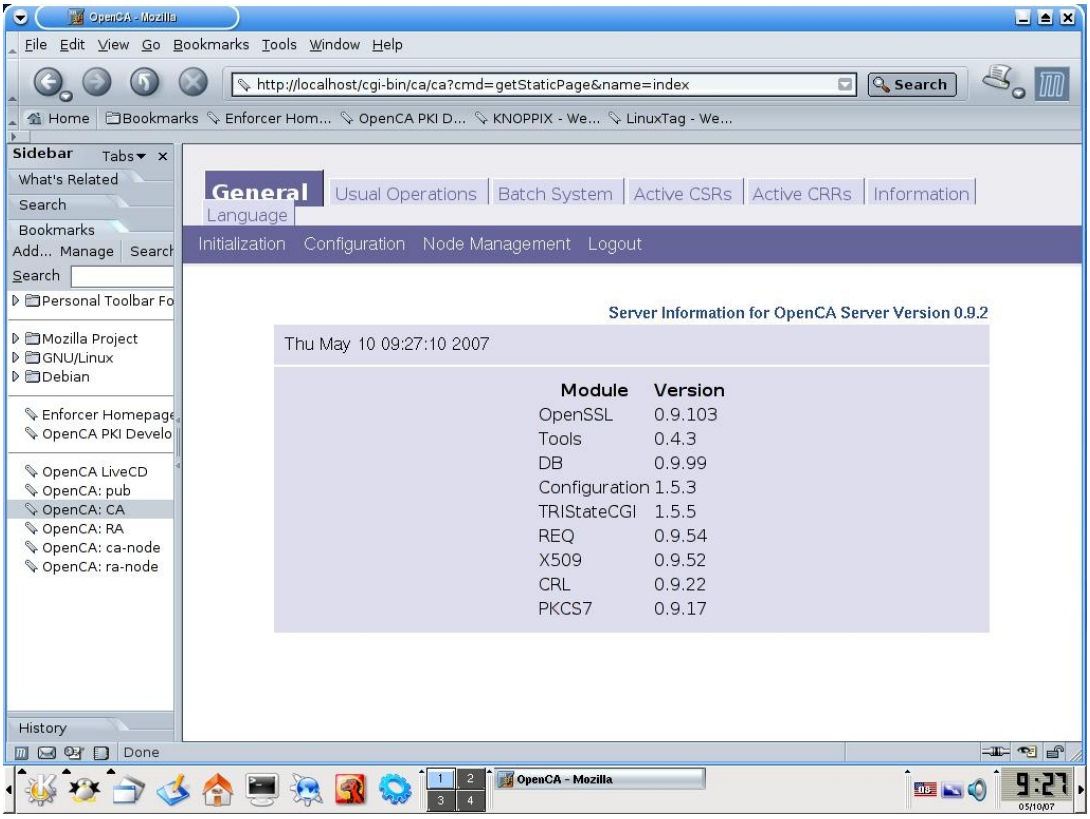
Αν όλα πάνε καλά και δεν υπάρξει κάποιο πρόβλημα με την αίτηση ή το etoken εμφανίζεται το παρακάτω μήνυμα

Η αίτηση για το πιστοποιητικό εγινε αποδεχτεί και το μόνο που μένει είναι η επικύρωση του πιστοποιητικού

7.4 Επικύρωση του πιστοποιητικού

Αυτός που μπορεί να επικυρώσει πιστοποιητικά είναι ο CA (root) οπότε θα πρέπει να συνδεθούμε (μέσω του μηχανηματος που έχουμε εγκαταστησει την αρχή πιστοποίησης) και να επικυρώσουμε το παραπάνω πιστοποιητικό.

Συνδεόμαστε στο http://localhost/ca/ ή μέσω της αριστερής στήλης των bookmarks επιλέγουμε OpenCA:CA

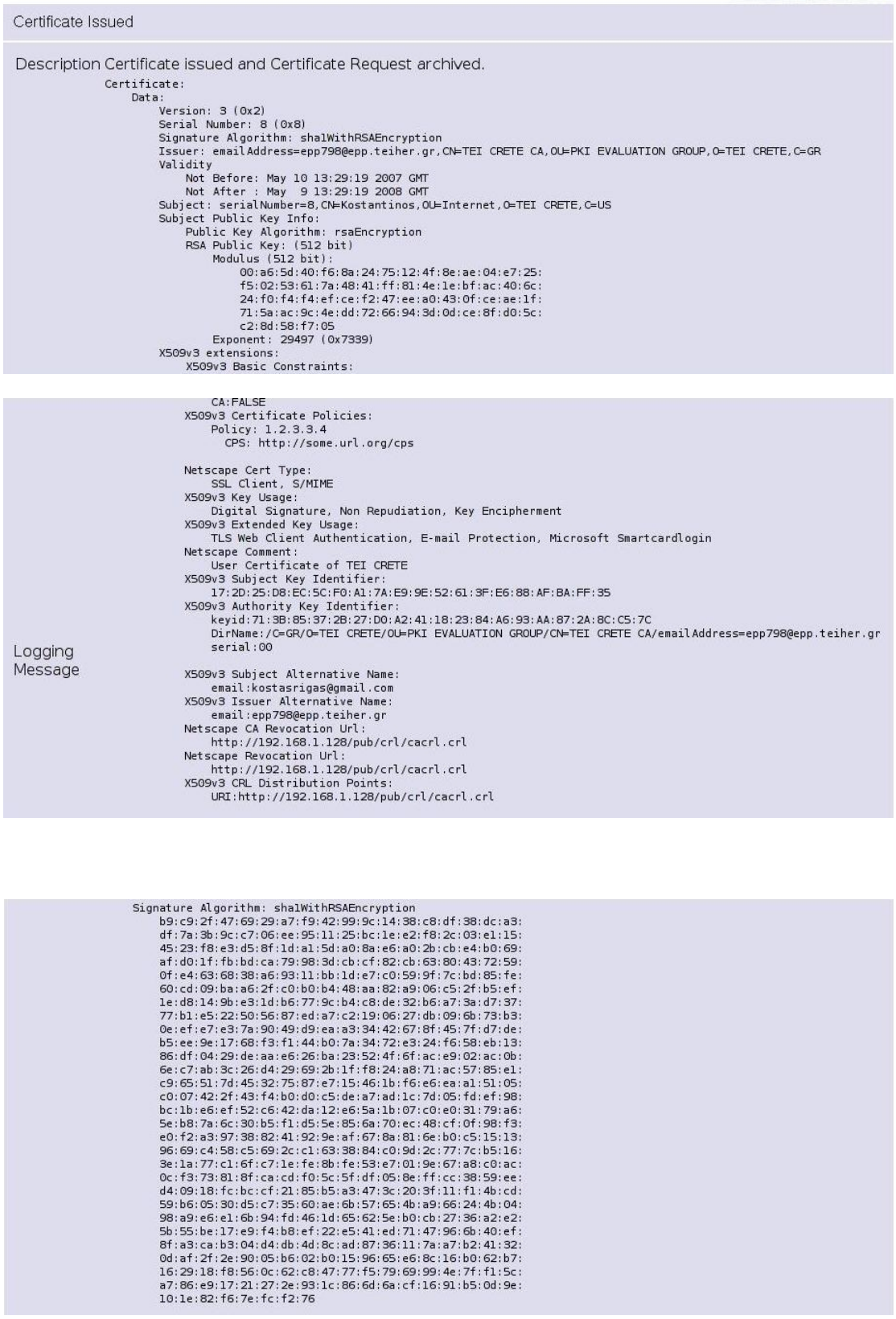


Επιλέγουμε την καρτέλα "**Active CSRs**" και στη συνέχεια την επιλογή "**Approved**", θα μας εμφανισθεί μια λίστα με όσες αιτήσεις έχουν εγκριθεί από τον Ra

Επιλέγοντας τον σειριακό αριθμό της αίτησης (5664) μας εμφανίζονται πληροφορίες σχετικά με τα στοιχεία του πιστοποιητικού

Επιλέγουμε "**Issue Certificate**", στη συνέχεια θα πρέπει να δώσουμε το password που προστατεύει το ιδιωτικό κλειδί του CA

Τέλος εμφανίζεται ένα μήνυμα επιτυχίας με διάφορες πληροφορίες για το πιστοποιητικό

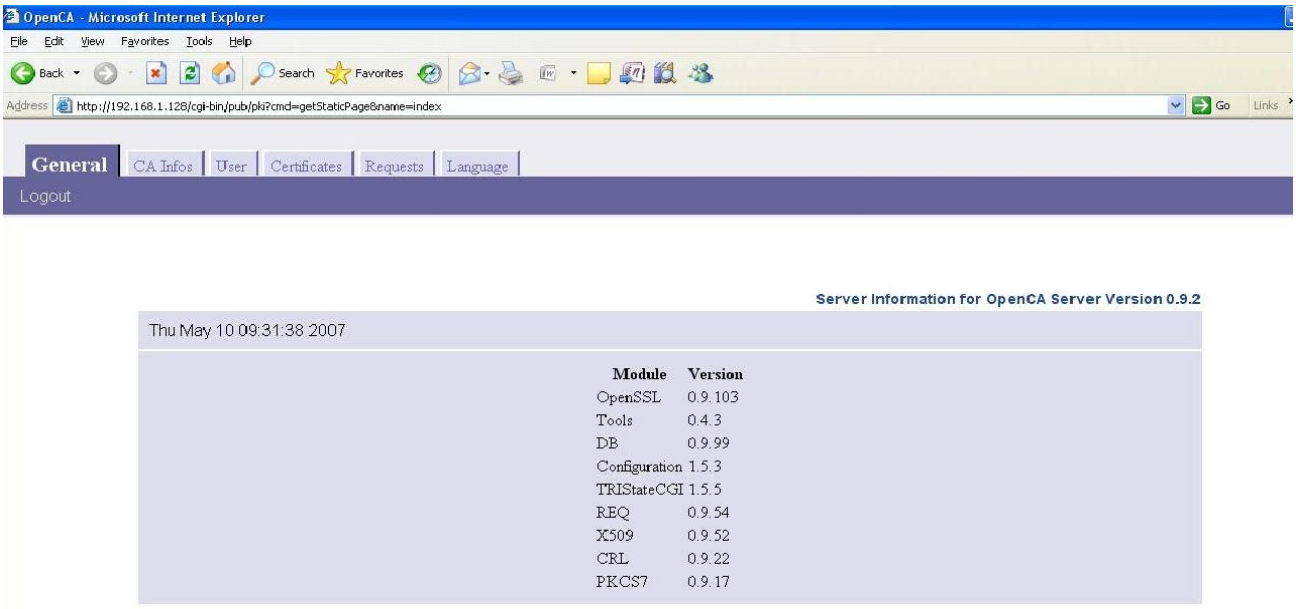


Η διαδικασία της επικύρωσης του πιστοποιητικού τελείωσε με επιτυχία και αυτό που μένει είναι να μπουμε στην αρχή πιστοποίησης ως χρήστες πια και να ανακτήσουμε

το πιστοποιητικό μας. Πρέπει να προσέξουμε όμως να μπούμε από τον ίδιο υπολογιστή που κάναμε προηγουμένος την αίτηση.

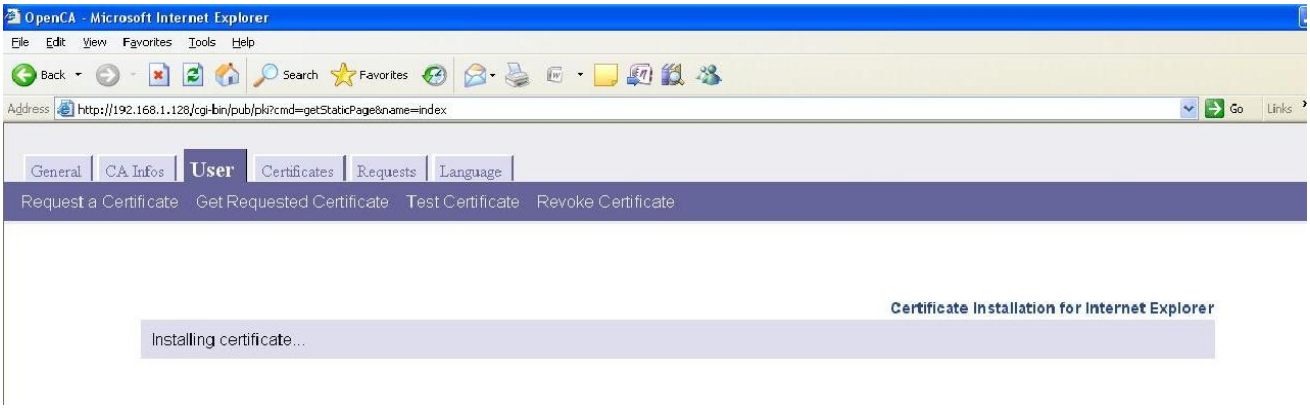
7.5 Εξαγωγή του πιστοποιητικού

1. Συνδεόμαστε στην αρχή πιστοποίησης στο /pub με ένα από τους δύο τρόπους όπως είπαμε και παραπάνω. Ο πρώτος είναι να μπούμε στο σύστημα που έχουμε την αρχή πιστοποίησης να ανοίξουμε ένα Browser και να δώσουμε http://localhost/pub ή να επιλέξουμε απο την καρτέλα των bookmarks στα αριστερά OpenCa:pub. Ο δεύτερος τρόπος είναι να μπούμε σε ένα υπολόγιστη που ανήκει στο δίκτου που είναι εγκατεστημένο το OpenCa και να δώσουμε την ip του υπολογιστή που φιλοξενή την αρχή πιστοποίησης η οποία στην περίπτωση μας είναι 192.168.1.128, οπότε δίνουμε http://192.168.1.128/pub. Επιλέγουμε τον δεύτερο τρόπο (και οι δύο καταλήγουν στα ίδια αποτελέσματα).

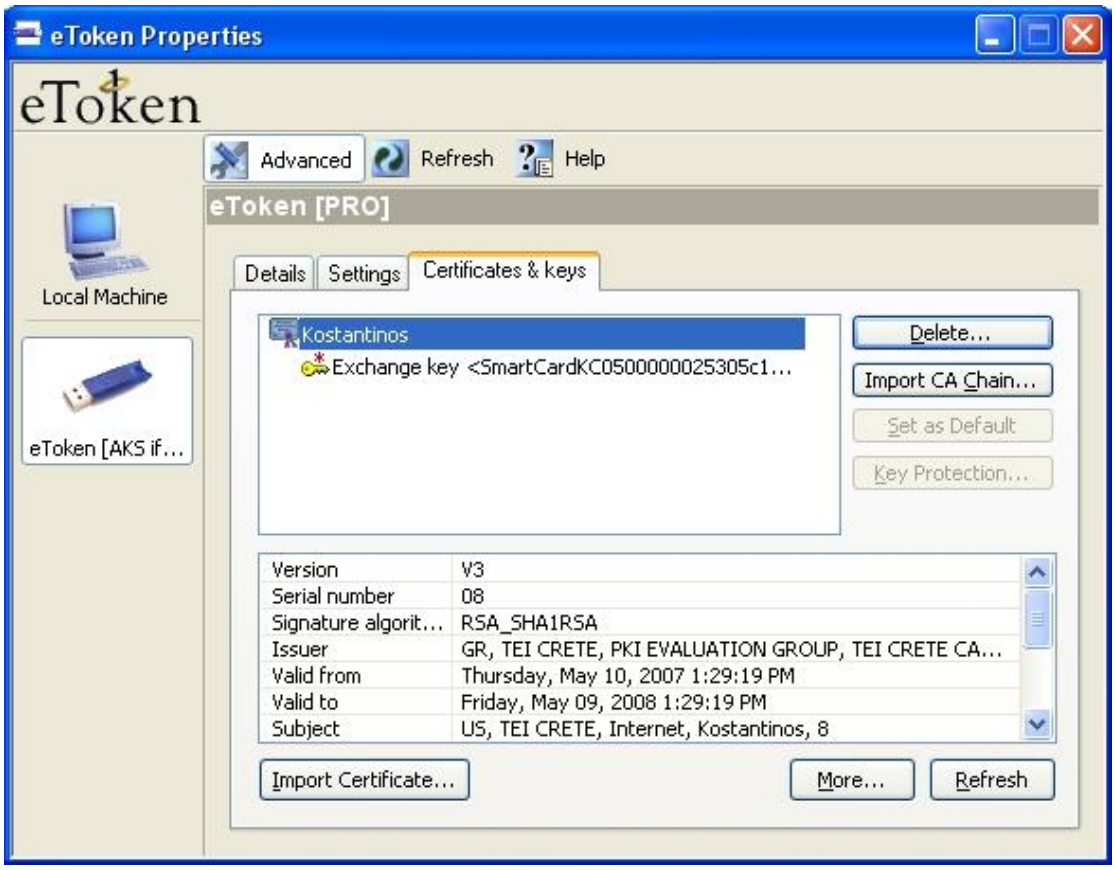


Επιλέγουμε "**User**" και μετά "**Get Requested certificate**", δίνουμε τον σειριακό αριθμό της αίτησης μας (5664)

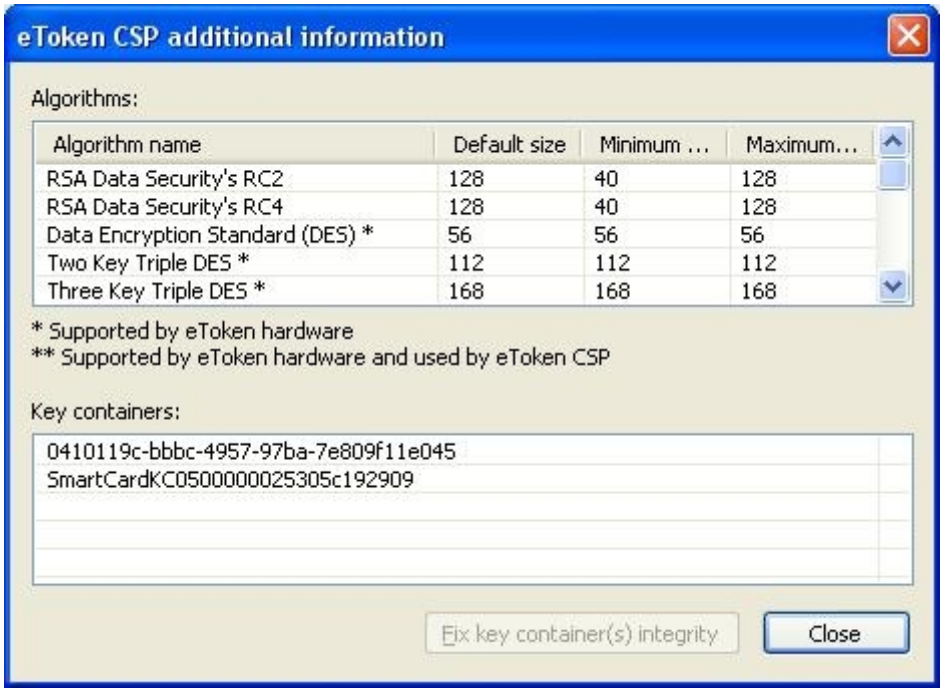
Επιλέγουμε "**Ok**" και αρχίζει το κατέβασμα του πιστοποιητικού και η εγκατάσταση του στο eToken μας.



Μόλις τελειώσει η διαδικασία ανοίγουμε την εφαρμογή RTE του etoken μας και παρατηρούμε ότι έχει ολοκληρωθεί η διαδικασία με το πιστοποιητικό και το ιδιωτικό του κλείδι να βρίσκονται αποθηκευμένα στο eToken

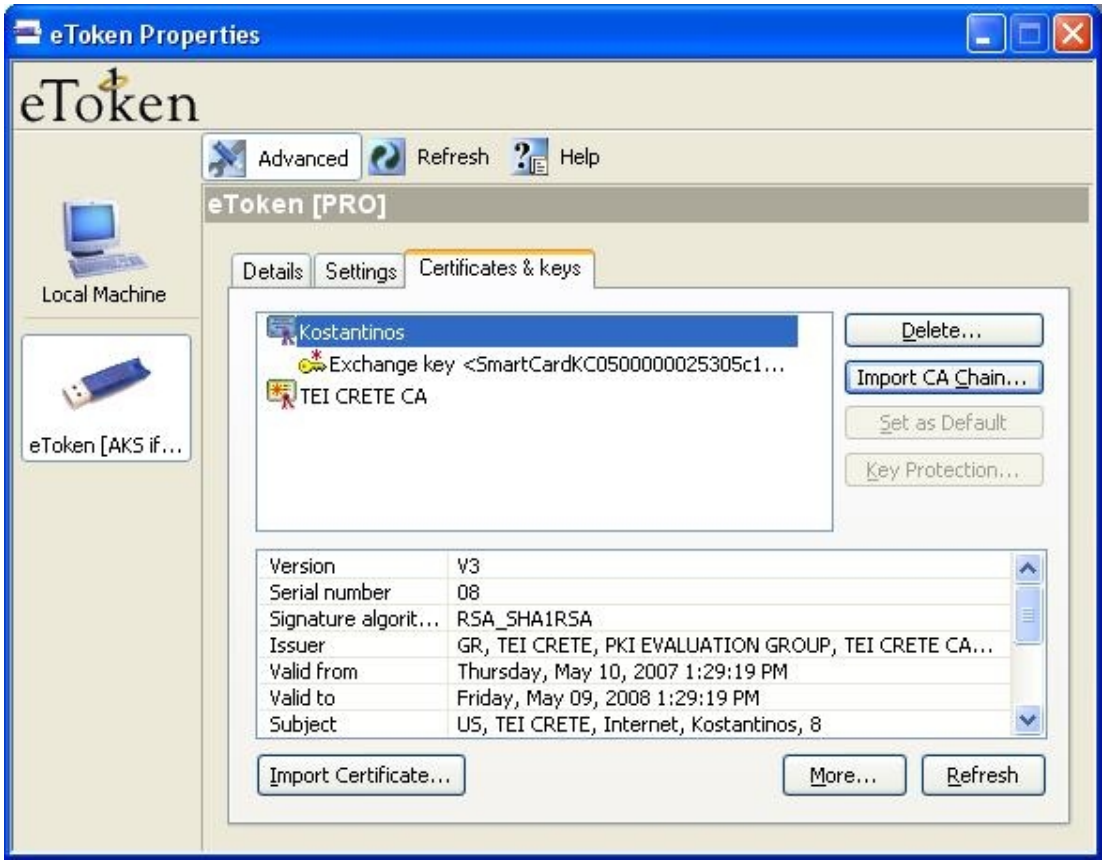


Επιλέγοντας "**More**" εμφανίζονται περισσότερες πληροφορίες για το πιστοποιητικό



Όμως θα πρέπει στο etoken να εισάγουμε και το πιστοποιητικό της Αρχής Πιστοποίησης το οποίο το έχουμε κατεβάσει από την αρχή πιστοποίησης και το έχουμε αποθηκεύσει κάπου στον υπολογιστή μας.

Η εισαγωγή του πιστοποιητικού της αρχής μας στο etoken μπορεί να γίνει με δύο τρόπους ο πρώτος είναι να επιλέξουμε "**Import CA Chain**" απο το αρχικό menu του etoken και ο δεύτερος είναι να επιλέξουμε "**Import Certificate**" και να δώσουμε το path στο οποίο βρίσκεται το πιστοποιητικό. Το αποτέλεσμα θα είναι κάπως έτσι



Με την παραπάνω διαδικασία καταφέραμε να δημιουργήσουμε ένα ψηφιακό πιστοποιητικό και να το εισάγουμε στην ασφαλή συσκευή etoken της εταιρίας Aladdin. Απο εδώ και πέρα θα χρησιμοποιήσουμε τη συσκευή ώστε να πιστοποιήσουμε την ταυτότητα μας για διάφορες ενέργειες, όπως να αποκτήσουμε πιστοποιημένη πρόσβαση, από την πλευρά του χρήστη, σε έναν ασφαλή server (https), να στείλουμε ένα email με την ψηφιακή μας υπογραφή ή να υπογράψουμε ένα αρχείο pdf. Όλα αυτά θα τα δούμε αναλυτικά παρακάτω.

8. Πρόσβαση σε Web Server και πιστοποίηση SSL από την πλευρά του χρήστη

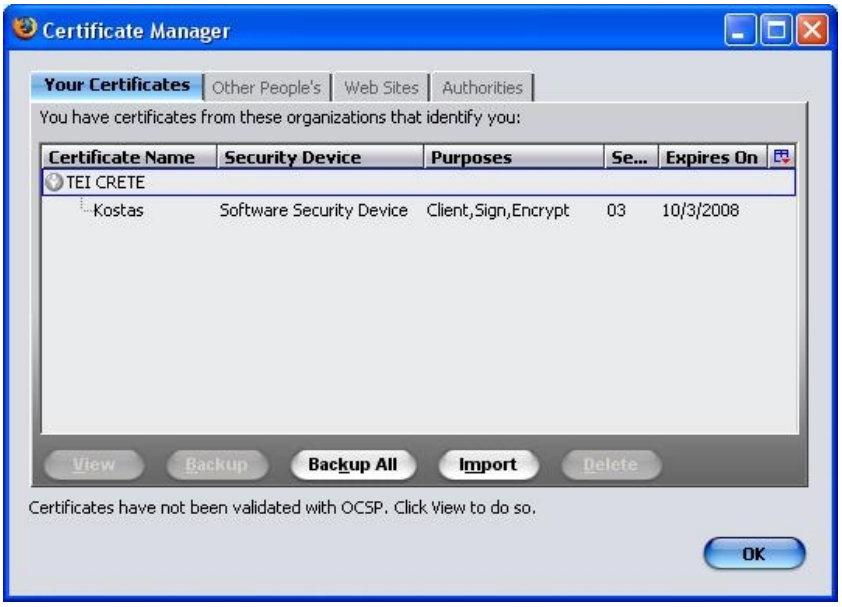
8.1 Εισαγωγή

Σε προηγούμενο βήμα, όταν είχαμε ρυθμίσει και σηκώσει τον web server, προσπαθήσαμε να αποκτήσουμε πρόσβαση στο path https://localhost/cgi-bin/printenv το οποίο και είχαμε ρυθμίσει έτσι, ώστε για να αποκτήσει κάποιος χρήστης πρόσβαση θα πρέπει να έχει ένα ψηφικό πιστοποιητικό, ώστε να γίνει η επικοινωνία μέσω του καναλίου ssl. Δεν καταφέραμε να αποκτήσουμε πρόσβαση διότι δεν είχαμε ακόμα δημιουργήσει ψηφιακό πιστοποιητικό για τον χρήστη. Τώρα λοιπόν που το έχουμε θα δοκιμάσουμε την είσοδο μας στο παραπάνω path.

Η δοκιμή αυτή θα γίνει δύο φορές την πρώτη φορά το πιστοποιητικό του χρήστη θα είναι αποθηκευμένο μονάχα στο browser και την δεύτερη φορά θα είναι αποθηκευμένο μονάχα στη συσκευή eToken.

8.2 Το πιστοποιητικού του χρήστη είναι αποθηκευμένο στον browser

Σε παραπάνω βήμα, όταν δημιουργήσαμε το πρώτο μας ψηφιακό πιστοποιητικό για χρήστη, το πιστοποιητικό είχε αποθηκευθεί στον browser. Ανοίγοντας λοιπόν τον browser μας ο οποίος είναι ο mozilla, επιλέγουμε από την μπάρα "**Tools**" στη συνέχεια "**Options**" και μετά στην καρτέλα που εμφανίζεται "**Advanced**". Στο πεδίο "**Encryption**" επιλέγουμε "**View Certificates**" εμφανίζεται το παρακάτω παράθυρο στο οποίο βλέπουμε τα πιστοποιητικά που είναι αποθηκευμένα στον web browser. Στην περιπτωση μας έχουμε ένα, με όνομα "Kostas" το οποιο μας ενημερώνει ότι είναι αποθηκευμένο στην συσκευή "**Software Security Device**" που δεν είναι άλλη από τον ίδιο τον browser.



Επιλέγοντας το πιστοποιητικό και μετά την επιλογή "View" εμφανίζεται το παρακάτω παράθυρο

Το οποίο μας ενημερώνει για τις δυνατότητες του πιστοποιητικού. Αναφέρει ότι το πιστοποιητικό μας μπορεί να χρησιμοποιηθεί για χρήσεις όπως "**SSL Client Certificate**", "**Email Signer Certificate**" και "**Email Recipient Certificate**". Αν δεν υπάρχει ψηφιακό πιστοποιητικό τότε θα πρέπει να εισάγουμε ένα.

Ήρθε η στιγμή να δοκιμάσουμε για δεύτερη φορά, μετά την πρώτη αποτυχημένη προσπάθεια, την προσβαση στο path https://localhost/cgi-bin/printenv . Δίνουμε λοιπόν στον browser το παραπάνω path. Ουσιαστικά το μέσω του path αυτού θα φορτώθει ένα script που θα μας ενημερώνει για τα στοιχεία του πιστοποιητικόυ.

Παραπάνω παρατηρούμε καταρχάς ένα ενημερωτικό μήνυμα το οποίο μας λέει ότι το Site (δηλαδή το συγκεκριμένο path) έχει ζητήσει να πιστοποίησουμε την ταυτότητα μας μέσω ενός πιστοποιητικού. Παρακάτω υπάρχουν κάποιες λεπτομέρειες, όπως τα στοιχεία στα οποία έχει βγεί το πιστοποιητικό, ποτέ λήγει κλπ. Σημαντικό είναι να προσέξουμε που είναι αποθηκευμένο, και όπως αναφέρεται στην τελευτάια γραμμή βρίσκεται αποθηκευμένο στη "Software Security Device" , η οποία όπως είπαμε δεν είναι πραγματική συσκευή, είναι μια ονομασία την οποία δίνει ο browser για να αναφερθεί στην τοπική αποθήκευση του πιστοποιητικού (σκλήρο δίσκο).

Άρα επιτύχαμε την πρόσβαση στο https://localhost/cgi-bin/printenv και αυτό γιατί αυτή τη φορά είχαμε ένα ψηφιακο πιστοποιητικό και καταφέραμε να πιστοποιήσουμε την ταυτότητα μας.

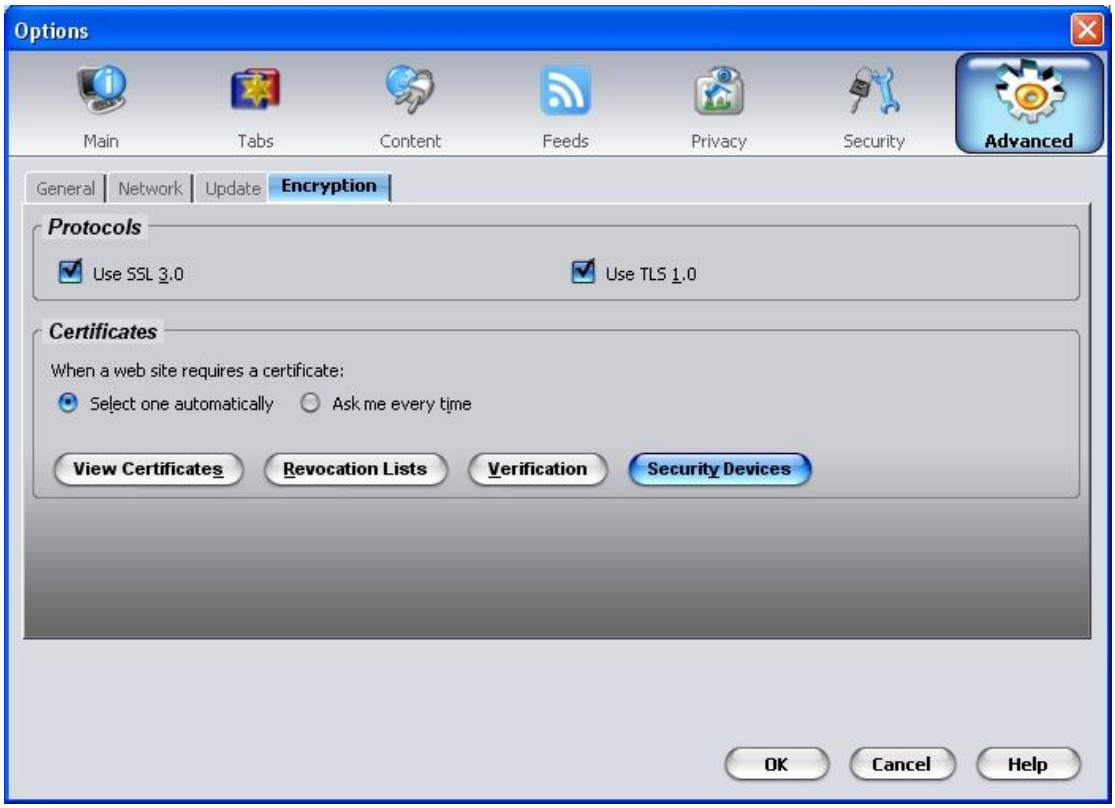
Τώρα μένει να κάνουμε ακριβώς το ίδιο μόνο που αυτή τη φορά το πιστοποιητικό δεν θα είναι αποθηκευμένο στον browser αλλά θα βρίσκεται σε μια συσκευή eToken της Aladdin.

8.3 Το πιστοποιητικού του χρήστη είναι αποθηκευμένο στην συσκευή eToken 32 k της Aladdin

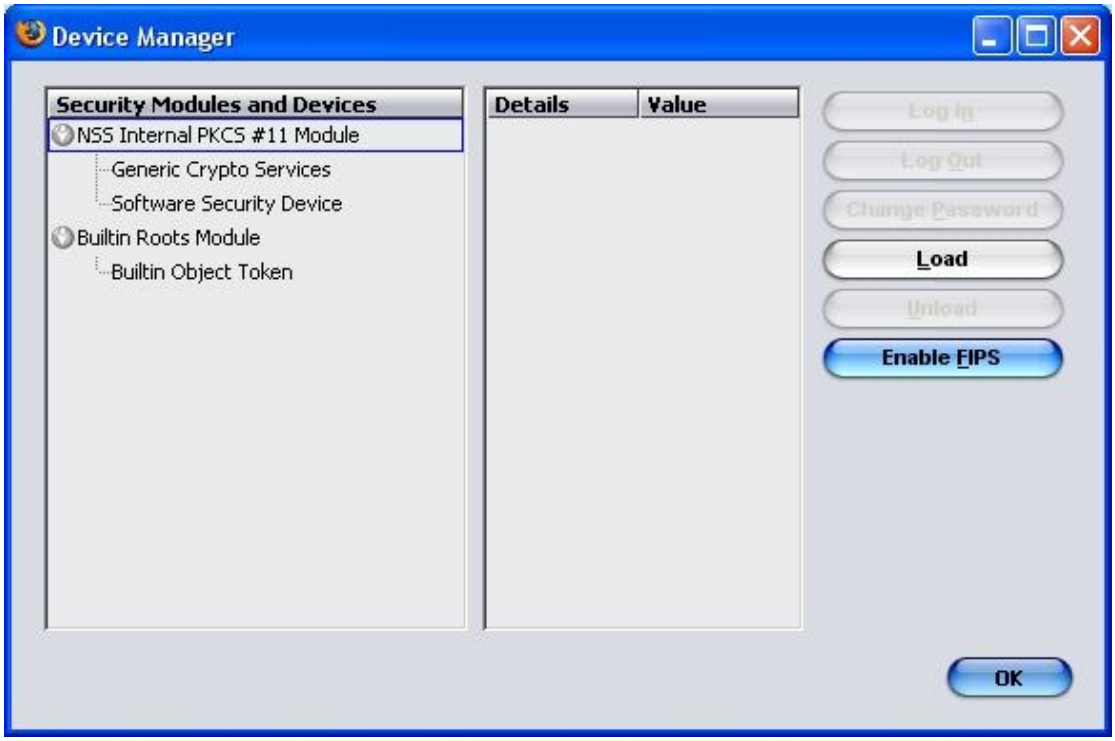
Αρχικά θα πρέπει να υπάρχει αποθηκευμένο στη συσκευή eToken το ψηφιακό μας πιστοποιητικό. Στη συνέχεια θα πρέπει να κάνουμε κάποιες ρυθμίσεις στον browser ώστε να μπορέσει να αναγνωρίσει την συσκευή μας.

8.3.1 Ρυθμίζοντας τον mozilla για την συσκευή eToken 32 k της Aladdin

Ανοίγουμε τον browser και από την αρχική μπάρα επιλέγουμε "**Tools**" και "**Options**", στη συνέχεια επιλέγουμε το εικονίδιο με τίτλο "**Advanced**" και στη συνέχεια την τελευταία καρτέλα με τίτλο "**Encryption**".



Επιλέγουμε "**Security Devices**" και βλέπουμε ποίες συσκευές είναι ενεργοποιημένες



Για να προσθέσουμε την δική μας και αφού έχουμε το software RTE (Run Time Enviroment) για windows της Aladdin, το οποίο σχετίζεται με την επικοινωνία της συσκευής etoken με το λειτουργικό σύστημα, επιλέγουμε "**Load**" και θα πρέπει να δώσουμε ένα όνομα για την συσκευή, στο παράδειγμα μας αφήνουμε το "New PKCS#11 Module"

Στο πεδίο "**Module Filename**" θα πρέπει να δώσουμε το εξής path C:\WINDOWS\system32\eTpkcs11.dll και επιλέγουμε "**Ok**". Η βιβλιοθήκη που αντισοιχεί στην συσκευή μας εγκαταστάθηκε επιτυχώς όπως μας ενημερώνει και το παρακάτω μήνυμα

Τώρα συνδέουμε την συσκευή eToken στον υπολογιστή μας και αφού έχουμε σηκώσει τον web server δίνουμε το path path https://localhost/cgi-bin/printenv. Θα πρέπει λοιπόν να εμφανισθούν τα στοιχεία του πιστοποιητικού. Πρώτα όμως μας εμφανίζεται ένα μήνυμα για να δώσουμε τον κωδικό στης συσκευής.

Αφού τον πληκτρολογίσουμε εμφανίζονται τα στοιχεία του πιστοποιητικού όπως και προηγουμένος με τη μόνη διαφορά ότι τώρα το πιστοποιητικό είναι αποθηκευμένο σε μια εξωτερικη συσκευή eToken. Όπως βλέπουμε και από την τελευταία γραμμή του παρακάτω ενημερωτικού παραθύρου.

Σε αυτή την παράγραφο δείξαμε πως επιτυγχάνεται η πρόσβαση και η πιστοποίηση ένος χρήστη σε έναν ssl web server.

9. Αποστολή email με ψηφιακή υπογραφή με την χρήση της συσκευής eToken Pro 32k της Aladdin

9.1 Εισαγωγή

Για να στείλουμε ένα email με ψηφιακή υπογραφή με την χρήση της συσκευής etoken θα πρέπει καταρχάς να έχουμε στην συσκευή μας ένα ψηφιακό πιστοποιητικό. Στη συνέχεια θα πρέπει να επιλέξουμε έναν email client και να τον ρυθμίσουμε έτσι ώστε να αναγνωρίσει την συσκέυη μας και κατεπέκταση το ψηφιακό μας πιστοποιητικό. Στο παράδειγμα μας θα χρησιμοποίησουμε τον email client Thunderbird version 2.0.0.0, ο οποίος είναι απόλυτα συμβατός με το etoken pro 32k της Aladdin.

9.2 Χρησιμοποιόντας τη λειτουργία Αποκρυψης (Encryption)

Χρησιμοποιόντας τον email client Thunderbird (version 2.0.0.0) για να στείλουμε ένα email με την λειτουργία της αποκρύψης θα πρέπει ο client να γνωρίζει το δημόσιο κλείδι του παραλήπτη του μηνύματος μας. Σε ένα επαγγελματικό περιβάλλον, για παράδειγμα μια μεγάλη επιχείρηση, όλα τα δημόσια κλειδία των υπαλλήλων μπορούν να είναι αποθηκευμένα σε μία βάση δεδομένων στην οποία έχει πρόσβαση ο email client. Έτσι όταν χρειάζεται κάποιος υπάλληλος να στείλει ένα μήνυμα με απόκρυψη σε κάποιον άλλο υπάλληλο ο email client θα ανατρέχει στη βάση δεδομένων με τα δημόσια κλείδια και θα επιλέγει το δημόσιο κλειδί του παραλήπτη.

Στην περίπτωση όμως που θέλουμε να επικοινωνίσουμε με κάποιον ο οποίος δεν ανήκει στη εταιρία και δεν υπάρχει το δημόσιο κλειδί του στη βάση δεδομένων θα

πρέπει να μας στείλει ένα μήνυμα στο οποίο θα συμπεριλαμβάνεται το δημόσιο του κλείδι, έτσι ώστε να επιτευχθεί η λειτουργία της απόκρυψης.

Η διαδικασία της αποκρύψης ενός μηνύματος email σχηματικά έχει ως εξής

1. Αρχικά παίρνουμε το περιεχόμενο του μηνύματος.

2. Στη συνέχεια ανακτώνται οι πληροφορίες του αποστολέα.

3. Αρχίζει η διαδικασία της απόκρυψης του μηνύματος, χρησιμοποιόντας τις μοναδικές πληροφορίες του αποστολέα (ψηφιακό πιστοποιητικό) έτσι ώστε να δημιουργηθεί ένα κρυπογραφημένο μήνυμα.

4. Το κρυπτογραφημένο μήνυμα αντικαθιστά το αρχικό μήνυμα.

5. Το μήνυμα αποστέλλεται.

Η παραπάνω διαδικασία χρειάζεται μοναδικές πληροφορίες από τον αποστολέα (ψηφιακό πιστοποιητικό) έτσι η απόκρυψή του μηνύματος προωθεί την εμπιστευτικότητα. Μονάχα ο εξουσιοδοτημένος παραλήπτης μπορεί να δεί το μήνυμα ακολουθόντας την παρακάτω διαδικασία.

1. Παραλαμβάνεται το μήνυμα

2. Ανακτάται το περιεχόμενο του μηνύματος

3. Ανακτόνται οι μοναδικές πληροφορίες (ψηφιακό πιστοποιητικό) από το μήνυμα.

4. Ξεκινάει η διαδικασία της αποκρυπτογράφιση του μηνύματος με τη χρήση των μοναδικών πληροφορίων του παραλήπτη του μηνύματος.

5. Το αποκρυπογραφημένο μήνυμα επιστρέφει στον παραλήπτη και είναι έτοιμο για ανάγνωση.

9.3 Χρησιμοποιόντας την ψηφιακή υπογραφή

Η ψηφιακή υπογραφή σε ένα email δίνει την δυνατότητα στον παραλήπτη του email να πιστοποιήσει τον αποστολέα του email. Με αυτό τον τρόπο ο παραλήπτης είναι σίγουρος για τον αποστολέα δίοτι έχει στα διαθεσή του την ψηφιακή υπογραφή του αποστολέα και κατεπέκταση το ψηφιακό του πιστοποιητικό. Το οποίο έχει εκδωθεί από μια αρχή πιστοποίησης η οποία με την σειρά της έχει ελέγξει τα στοίχεια του αποστολέα και τα έχει εγκρίνει, οπότε ο παραλήπτης μπορεί να τα εμπιστευθεί.

Σχηματικά η διαδικασία της ψηφιακής υπογραφής ένος μηνύματος έχει ως εξής

1. Αρχικά παίρνουμε το περιεχόμενο του μηνύματος.

2. Στη συνέχεια ανακτώνται οι πληροφορίες του αποστολέα.

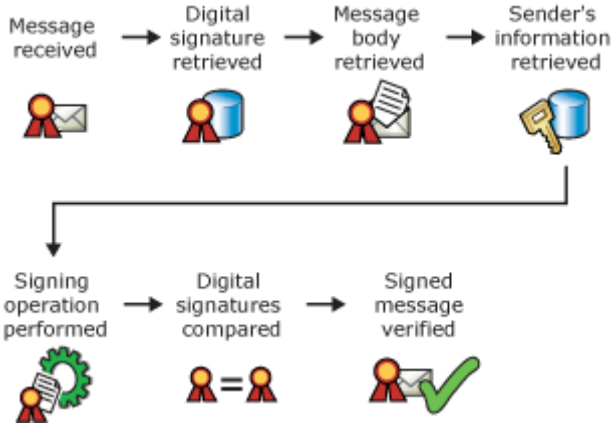
3. Αρχίζει η διαδικασία της υπογραφής του μηνύματος, χρησιμοποιόντας τις μοναδικές πληροφορίες του αποστολέα (ψηφιακό πιστοποιητικό) έτσι ώστε να δημιουργηθεί η ψηφιακή υπογραφή.

4. Η ψηφιακή υπογραφή προσθέτεται στο μήνυμα.

5. Το μήνυμα αποστέλλεται.

Η παραπάνω διαδικασία χρειάζεται μοναδικές πληροφορίες από τον αποστολέα (ψηφιακό πιστοποιητικό) έτσι μπορεί να αποδειχθεί ότι το μήνυμα μπορεί να σταλεί μονάχα από τον αποστολέα.

Στη συνέχεια ακολουθεί η διαδικασία της πιστοποίησης από τον παραλήπτη η οποία έχει ως εξής.



1. Παραλαμβάνεται το μήνυμα

2. Ανακτάται η ψηφιακή υπογραφή από το μήνυμα

3. Ανακτάται το μήνυμα.

4. Λαμβάνονται οι πληροφορίες του αποστολέα (ψηφιακό πιστοποιητικό).

5. Ξεκινάει η διαδικασία της ψηφιακής υπογραφής του μηνύματος.

6. Η ψηφιακή υπογραφή που περιλαμβανόταν στο εισερχόμενο μήνυμα συγκρίνεται με την ψηφιακή υπογραφή που παράχθηκε κατά την παραλαβή.

7. Αν οι δύο ψηφιακές υπογραφές ταιρίζουν τότε το μήνυμα είναι έγκυρο.

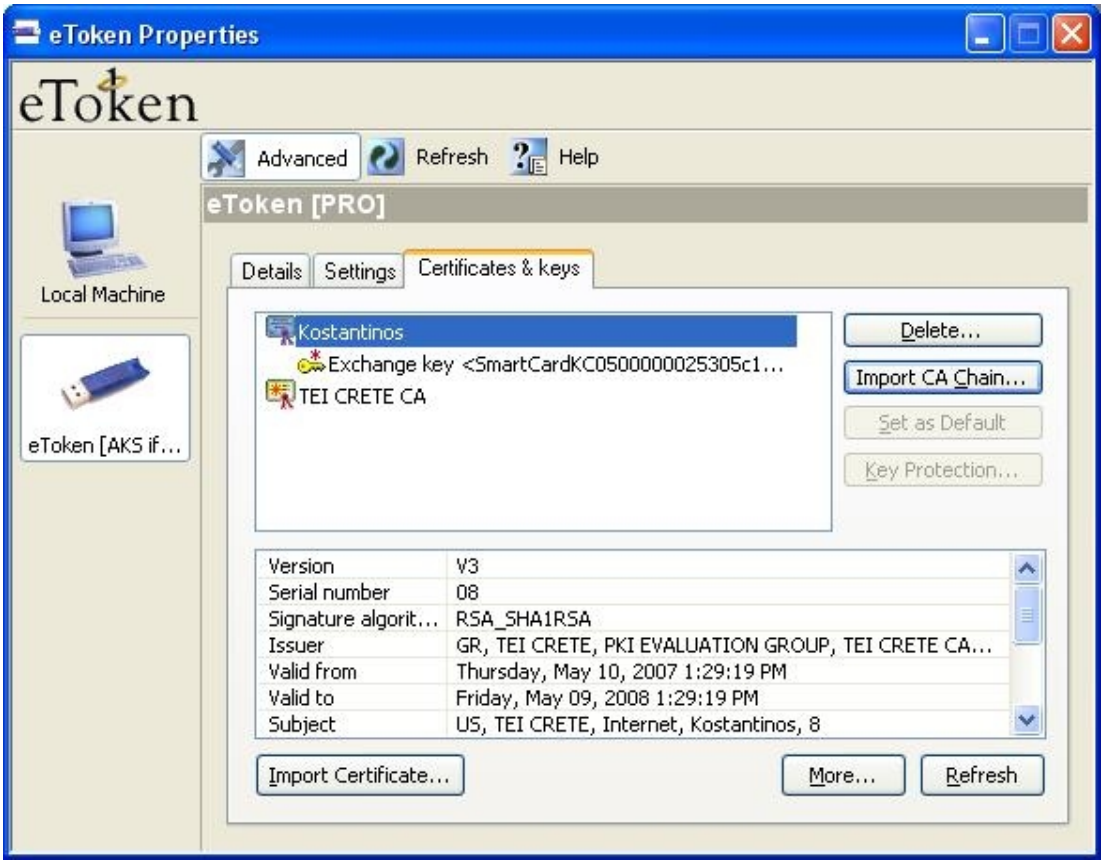
Εμείς στο παράδειγμα μας θα στείλουμε ένα email, υπογεγραμένο με την ψηφιακή μας υπογραφή, στον εαυτό μας. Αυτό για να το επιτύχουμε θα πρέπει πρώτα να κάνουμε κάποιες ρυθμίσεις στον email client Thunderbird (version 2.0.0.0).

Τέλος για λόγους ασφαλείας καλό θα ήταν να χρησιμοποιούμε διαφορετικό κλειδί κρυπτογράφησης για τις παραπάνω λειτουργίες.

9.4 Ρυθμίσεις του email client Thunderbird *(version 2.0.0.0)*

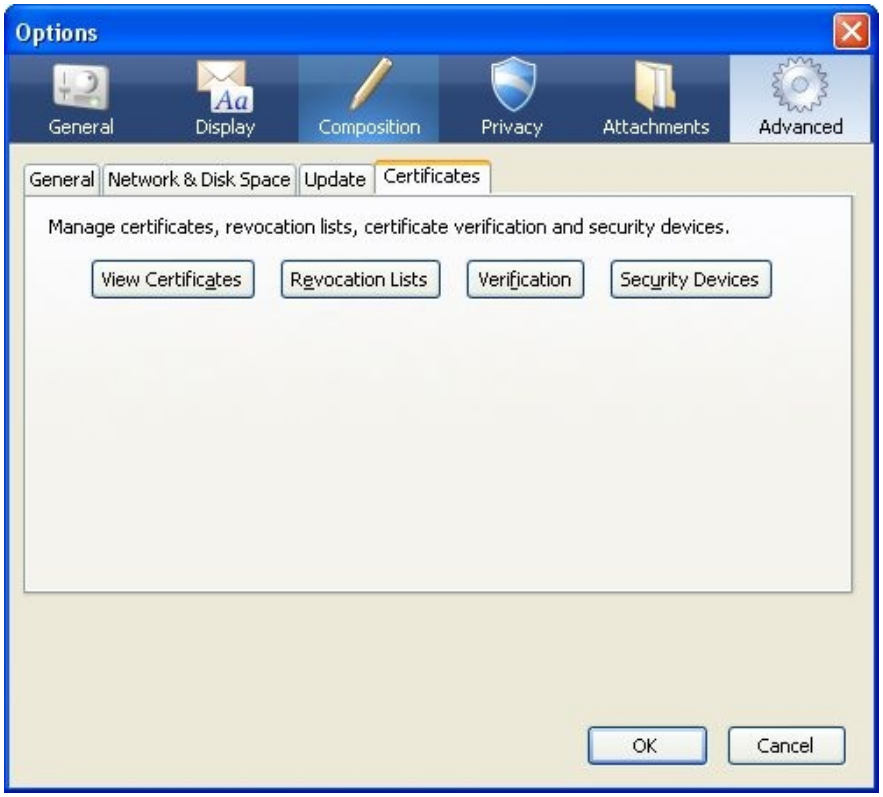
Ξεκινώντας θα πρέπει να σιγουρευτούμε ότι έχουμε εγκατεστημένο στο σύστημα μας το software RTE (Run Time Enviroment) για windows xp, το οποίο σχετίζεται με την επικοινωνία της συσκευής etoken με το λειτουργικό σύστημα. Αν δεν το έχουμε μπορούμε να το βρούμε από το http://www.aladdin.com.

Συνεχίζοντας θα πρέπει, αν δεν το έχουμε κάνει ήδη, να περάσουμε στο etoken το ψηφιακό μας πιστοποιητικό καθώς και το πιστοποιητικό της αρχής πιστοποίησης (root CA), έτσι ώστε να μπορεί να ταυτοποιηθεί από οποιαδήποτε εφαρμογή η γνησιότητα του πιστοποιητικού μας. Παρακάτω βλέπουμε το περιεχόμενο της συσκεύης etoken, το οποίο είναι, το ψηφιακό πιστοποιητικό (konstantinos) και το πιστοποιητικό της αρχής πιστοποίησης (TEI CRETE CA) .

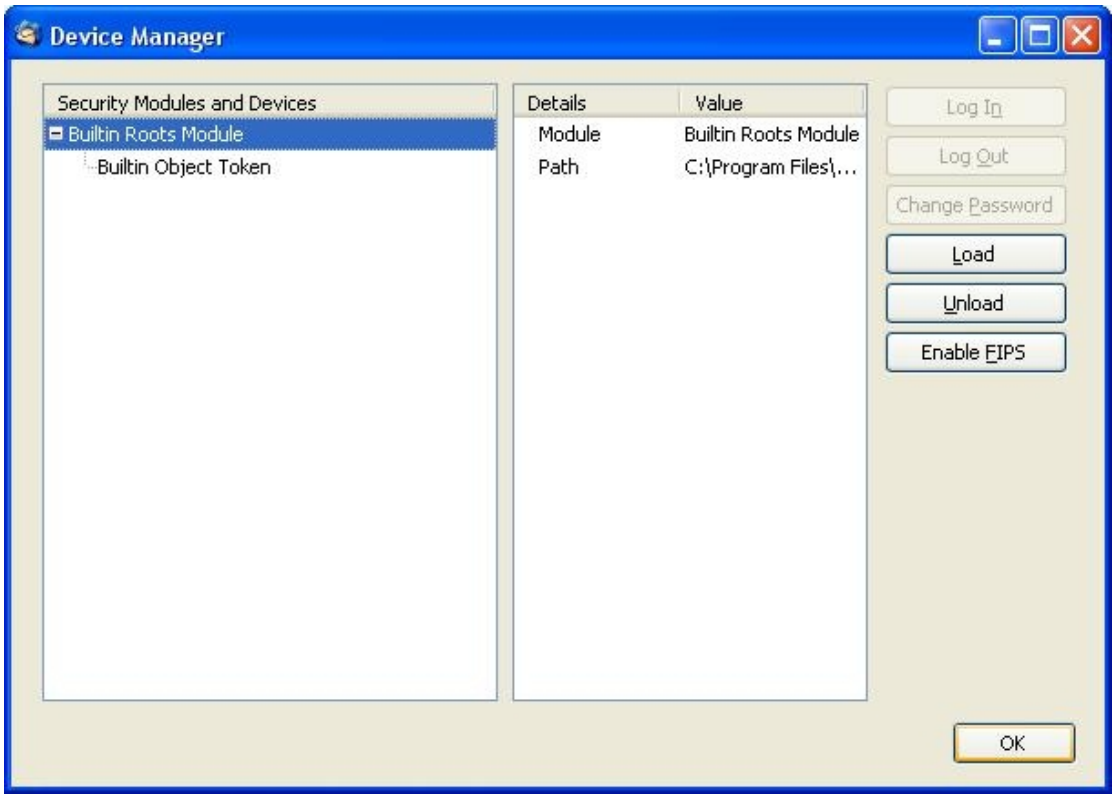


Αφού λοιπόν είναι όλα έτοιμα με το etoken θα ρυθμίσουμε τον email client ώστε να αναγνωρίσει αρχίκα την συσκευή μας και το περιεχόμενο της και στη συνέχεια θα στείλουμε ένα email με την ψηφιακή μας υπογραφή.

Ανοίγουμε τον Thunderbird (version 2.0.0.0) και επιλέγουμε από την μπάρα "**Tools**" και μετά "**Options**" μας εμφανίζεται το παρακάτω παράθυρο.



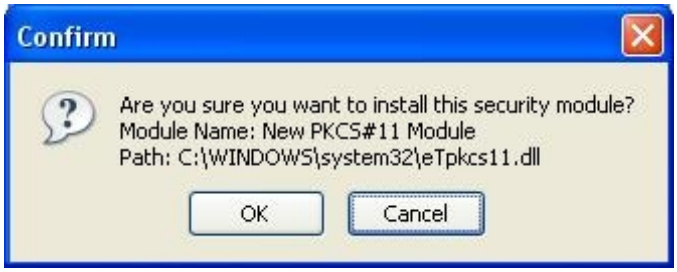
Επιλέγουμε "**Advanced**", "**Certificates**" και τέλος "**Security Devices**", βλέπουμε την παρακάτω καρτέλα



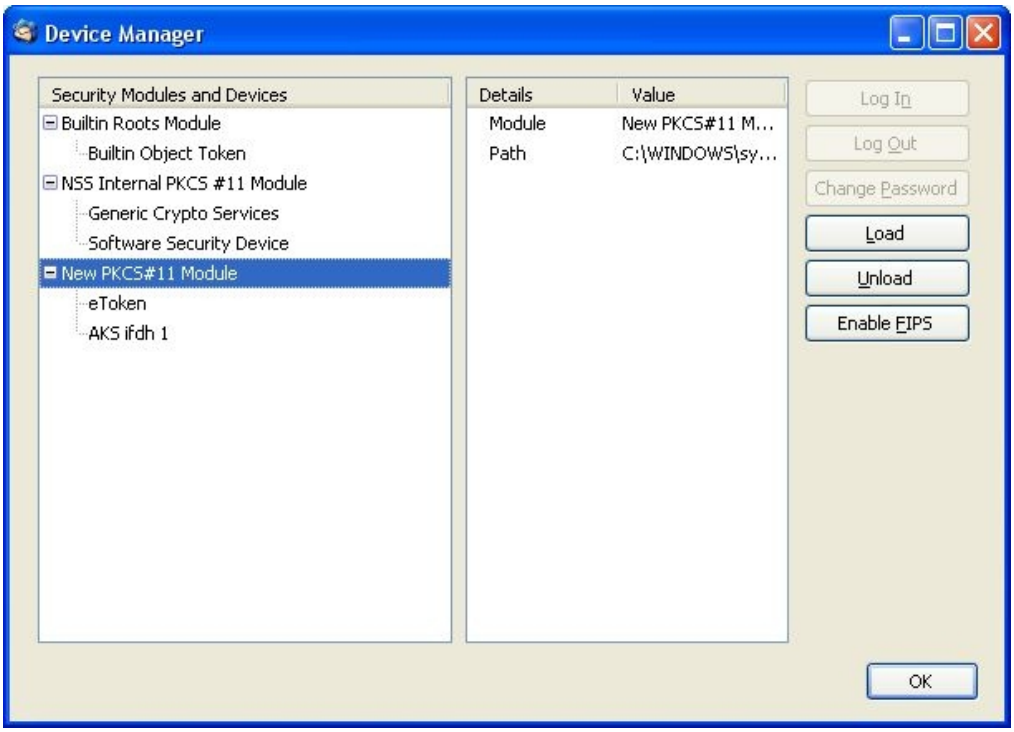
Παρατηρούμε ότι η συσκεύη μας δεν φαίνεται πουθενά οπότε θα πρέπει να ενημερώσουμε τον Thundebird για την ύπαρξη της. Επιλέγουμε "**Load**", θα πρέπει να δώσουμε ένα όνομα για την συσκευή, στο παράδειγμα μας αφήνουμε το "New PKCS#11 Module"

Στο πεδίο "Module Filename" θα πρέπει να δώσουμε το εξής path C:\WINDOWS\system32\eTpkcs11.dll

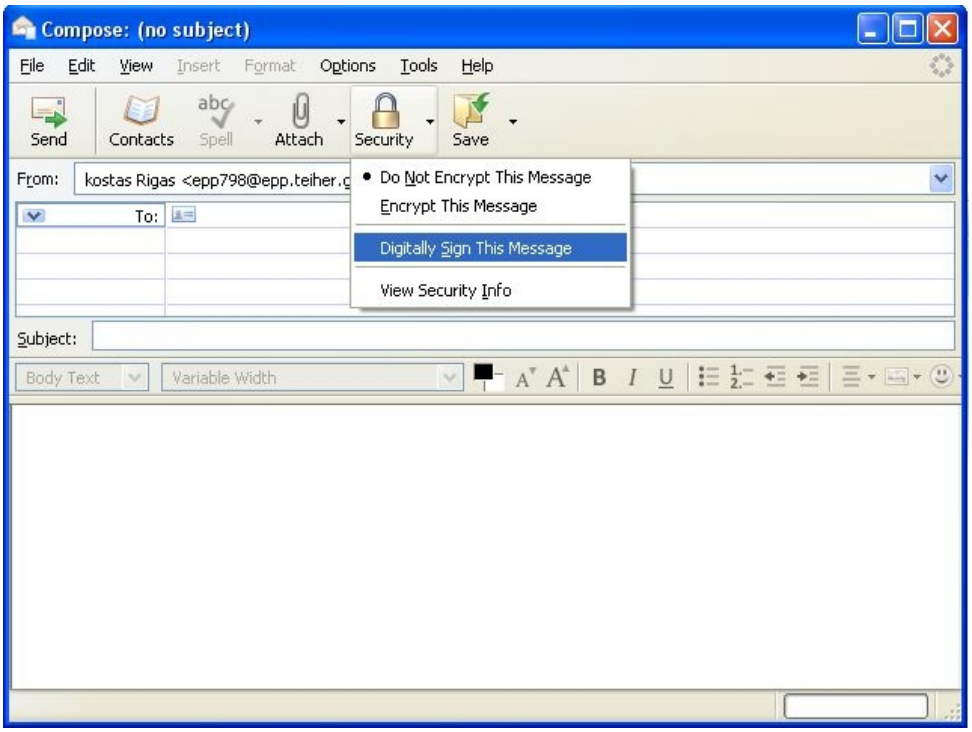
Επιλέγουμε "**Ok**" και εμφανίζεται το παρακάτω μήνυμα



Επιλέγουμε "**Ok**" και παρατηρούμε ότι η συσκευή μας αναγνωρίσθηκε όπως βλέπουμε και παρακάτω, τώρα μπορύμε να χρησιμοποίησουμε την συσκεύη μας σε συνεργασία με τον Thunderbird

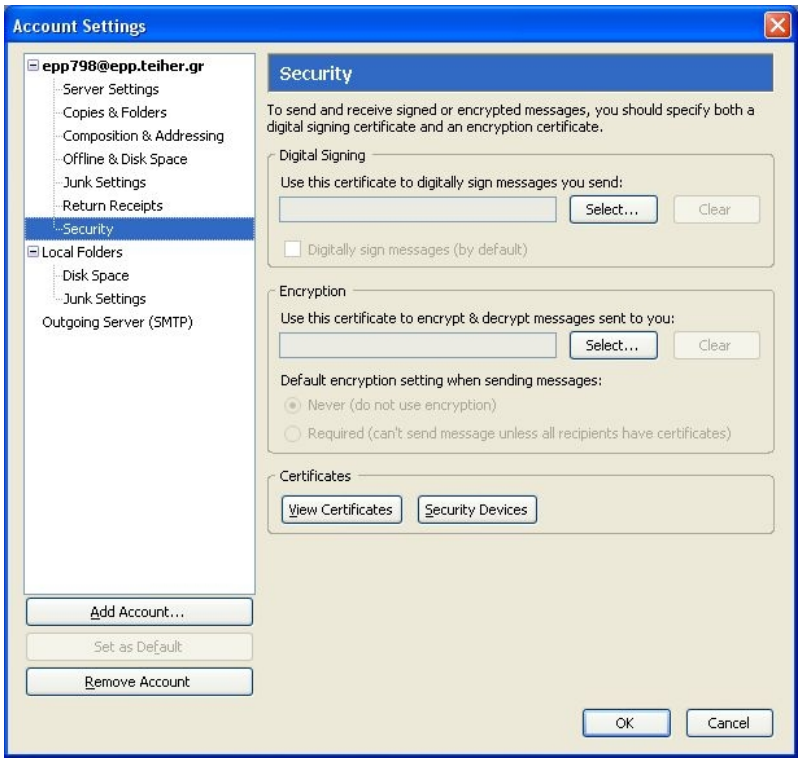


Επιλέγουμε από το αρχικό menu “File” και “New Message” και εμφανίζεται το παράθυρο του νέου μηνύματος, εκεί επιλέγουμε από την μπάρα “Security” και στη συνέχεια “Digital Sign The Message ”

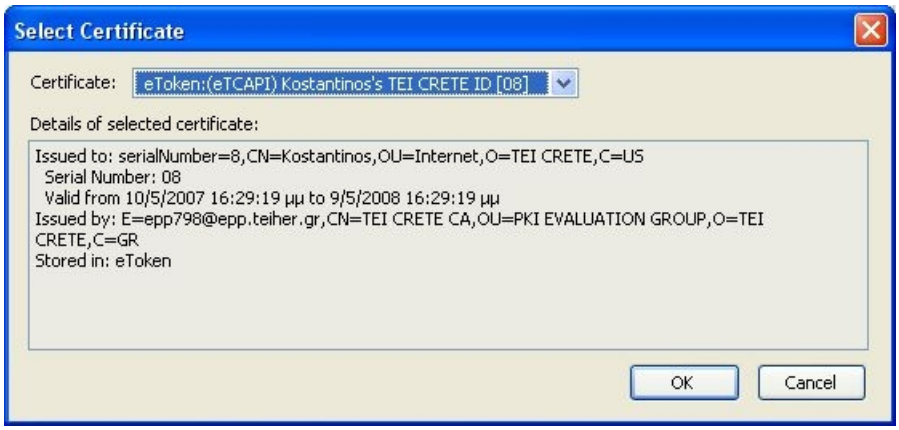


Εμφανίζεται το παρακάτω μήνυμα το οποίο μας ενημερώνει ότι για να προχωρίσουμε την διαδικασία και να υπογράψουμε το email θα πρέπει να χρησιμοποίησουμε ένα πιστοποιητικό.

Επιλέγουμε “**Ok**” και εμφανίζεται η παρακάτω οθόνη

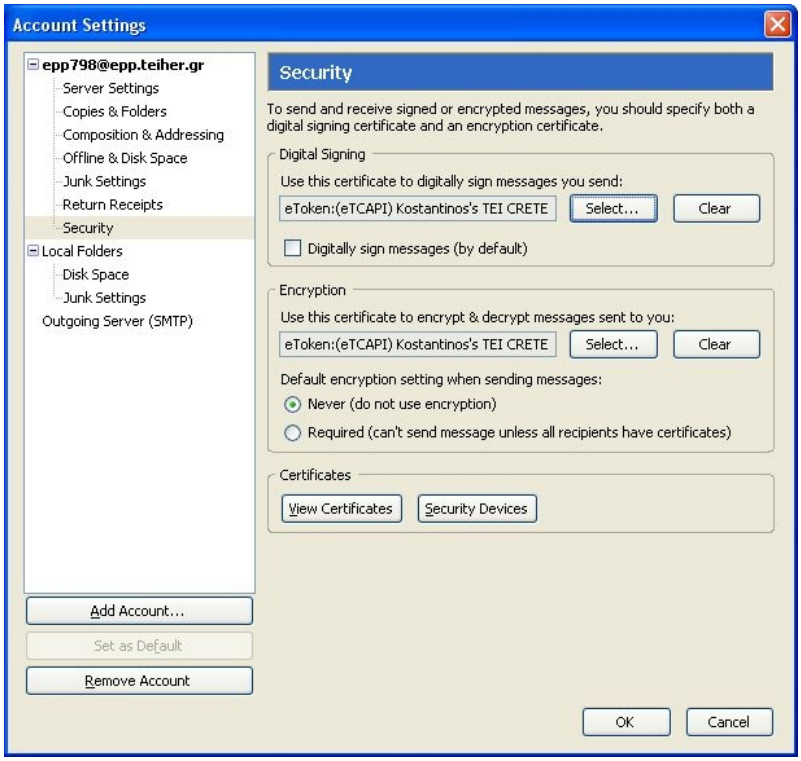


Στο τμήμα "**Digital Signing**" επιλέγουμε "**Select**" και αφού δίνουμε τον password του etoken εμφανίζονται οι παρακάτω πληροφορίες που σχετίζονται με το περιεχόμενο της συσκευής μας



Αφού λοιπόν μας παρουσιάστηκαν οι πληροφορίες των δύο πιστοποιητικών τα οποία περιλαμβάνονται στο etoken επιλέγουμε "**Ok**".

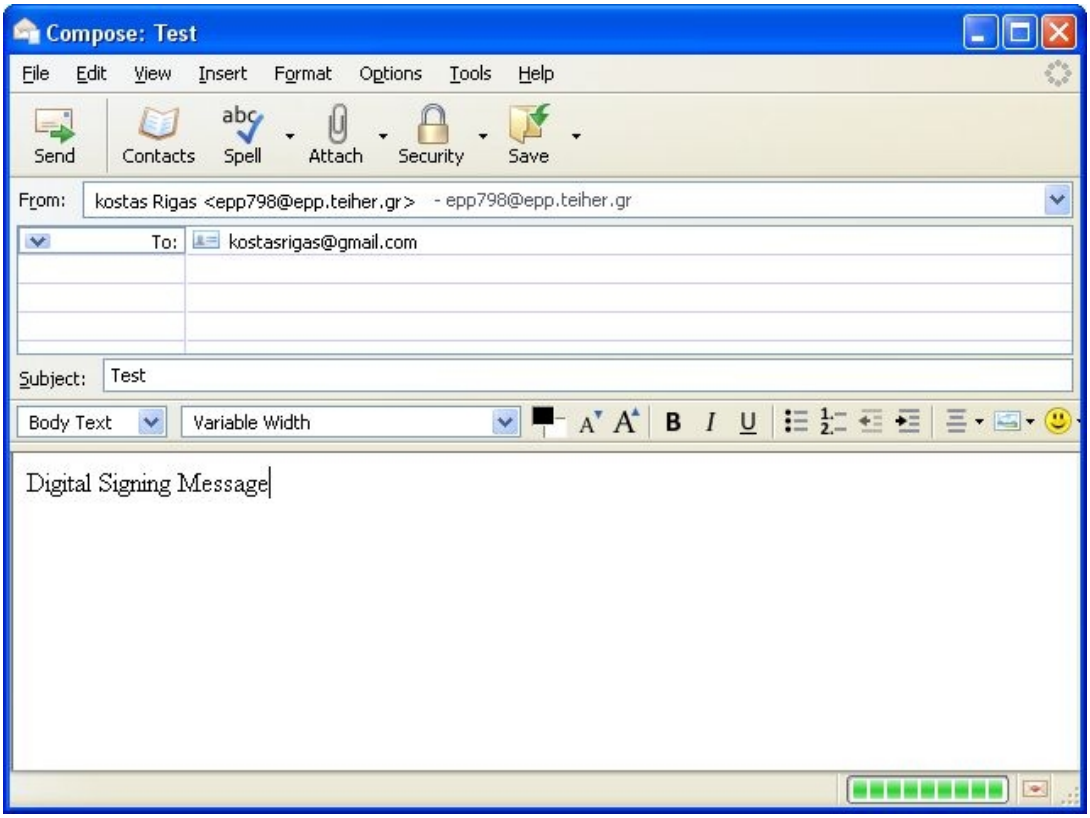
Το παραπάνω παράθυρο μας ενημερώνει αν θέλουμε να χρησιμοποιούμε το ίδιο πιστοποιητικό για την λειτουργία της ψηφιακής υπογραφής (Digital Signing) και την λειτουργία της απόκρυψης (Encryption), επιλέγουμε "**Ok**". Εμφανίζεται το προηγούμενο παράθυρο με συμπλήρωμενα τα πεδία που μας ενδιαφέρουν.



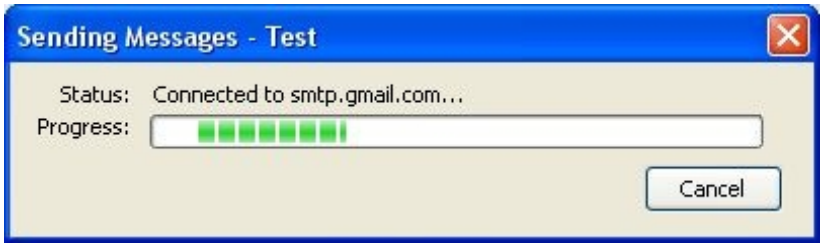
Τώρα είμαστε έτοιμοι να στείλουμε ένα email με την ψηφιακή μας υπογραφή.

9.5 Αποστολή email με ψηφιακή υπογραφή

Συμπληρώνουμε το μήνυμα μας καθώς και την διεύθυνση του αποστολέα και το στέλνουμε



Το μήνυμα αποστέλεται όπως βλέπουμε και παρακάτω



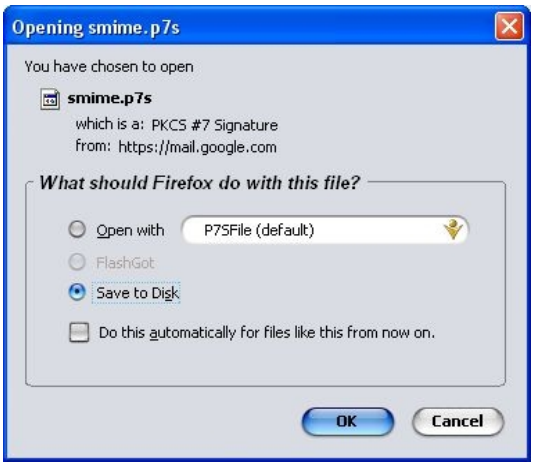
Τώρα για να επαληθεύσουμε την προσπάθεια μας θα μπούμε στο web email client του gmail (στον οποίο και στείλαμε το email μας) και θα δούμε αν όντως περιέχει την ψηφιακή μας υπογραφή.

Παρακάτω παρατηρούμε ότι το email με τίτλο ίδιο με αυτό που συμπληρώσαμε προηγουμένος (Test) ήρθε

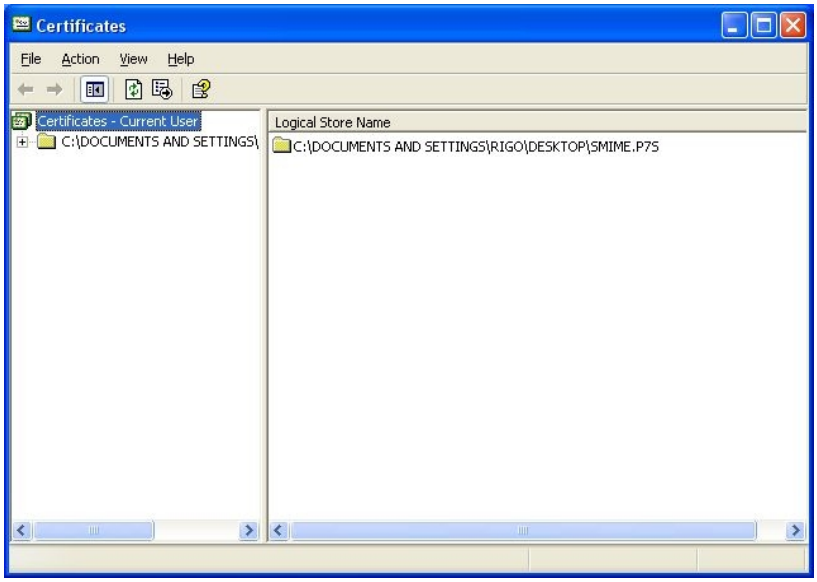
Το ανοίγουμε και βλέπουμε τα εξής

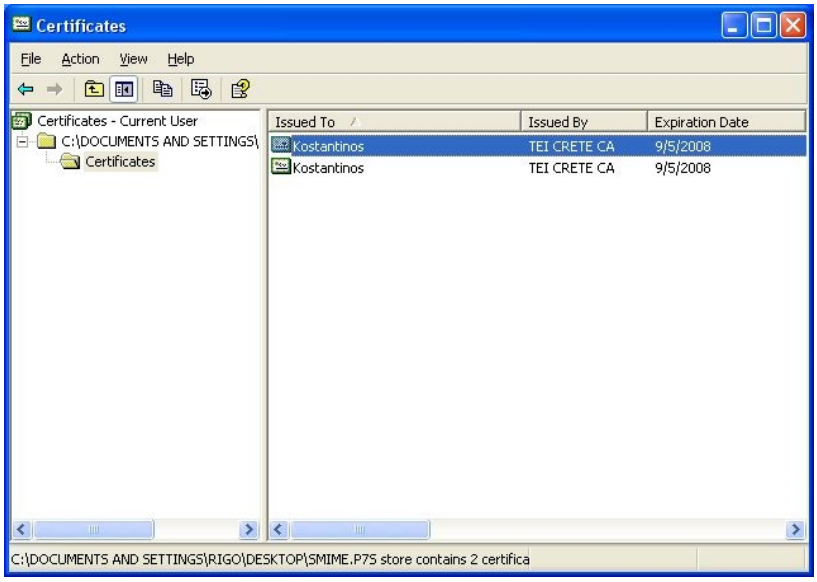
Εκτός από τον τίτλο και το μήνυμα παρατηρούμε ότι έχει έρθει μαζί και ένα αρχείο με το όνομα smime.p7s. Παρακάτω θα αναφερθούμε αναλυτικότερα στο πρότυπο smime.

Επιλέγοντας "**Download**" θα κατεβάσουμε στον υπολογιστή μας αυτό το αρχείο, όπως βλέπουμε παρακάτω



Εκτελώντας το αρχείο smime.p7s μας εμφανίζονται οι πληροφορίες του πιστοποιητικού μας, έτσι ο παραλήπτης του μηνύματος μπορεί να επαληθεύσει τα στοιχεία και να είναι σίγουρος για τον αποστολέα.





Με το παραπάνω παράδειγμα δείξαμε πως με την χρήση ενός ψηφιακού πιστοποιητικού, και την διασφάλιση του μέσω μιας συσκευής etoken, μπορούμε να στείλουμε ένα email με την ψηφιακή μας υπογραφή. Πιστοποιώντας με αυτό τον τρόπο την ταυτοτητά μας στο παραλήπτη του μηνύματος.

9.6 Το πρότυπο S/MIME (Secure / Multipurpose Internet Mail Extensions)

9.6.1 Εισαγωγή

Πρίν το S/MIME (Secure / Multipurpose Internet Mail Extensions) οι διαχειριστές συστημάτων χρησιμοποιούσαν ένα ευρέως διαδεδομένο πρωτόκολλο για αποστολή email, αυτό το πρωτόκολλο ονομαζόταν Simple Mail Transfer Protocol (smtp). Το smtp πρωτόκολλο δεν ήταν ασφαλές, έτσι υπήρχε η ανάγκη για την δημιουργία ένος καινούργιου πρωτόκολλου για την επικοινωνία μέσω email το οποίο θα προσέφερε περισσότερη ασφάλεια και συνδεσιμότητα από τον προκάτοχο του. Έτσι δημιουργήθηκε το S/MIME το οποίο προχώρισε ένα βήμα παραπέρα από το smtp πρωτόκολλο και έκανε πραγματικότητα την χρήση του email χωρίς παραχωρίσεις στο τομέα της ασφάλειας.

9.6.2 Τι προσφέρει το S/MIME (Secure / Multipurpose Internet Mail Extensions)

Το πρωτοκολλό S/MIME προσφέρει δύο υπηρεσίες ασφάλειας, την ψηφιακή υπογραφή (Digital Signature) και την κρυπτογράφηση μηνύματος (Message Encryption). Αυτές οι δύο υπηρεσίες είναι ο πυρήνας του πρωτοκόλλου. Πάνω σε αυτές τις υπηρεσίες έχουν αναπτυχθεί πολλά σενάρια ασφαλείας σχετικά με την υπηρεσία του email.

Οι mail clients που υποστηρίζουν το πρωτόκολλο είναι οι

- Microsoft Outlook (Windows)
- Microsoft Outlook Express (Windows)
- Mozilla Thunderbird (Linux, Mac OS X, Windows, Solaris)
- Netscape Communicator (Windows, Solaris)
- Mail (Mac OS X)

Συνοψίζοντας λοιπόν αυτή τη μικρή αναφορά μας στο πρωτόκολλο S/MIME, μπορούμε να πούμε ότι το πρωτόκολλο αυτό είναι η ασφαλής έκδοση του MIME

(Multipurpose Internet Mail Extensions) πρωτοκόλλου, η οποία υποστηρίζει την κρυπτογράφηση μηνυμάτων και βασίζεται στην τεχολογία RSA's public key encryption. Με την βοήθεια του επιτυγχάνεται η ασφαλής επικοινωνία μέσω email.

10. Υπογράφοντας αρχεία pdfs με την χρήση της συσκευή eToken 32K της Aladdin

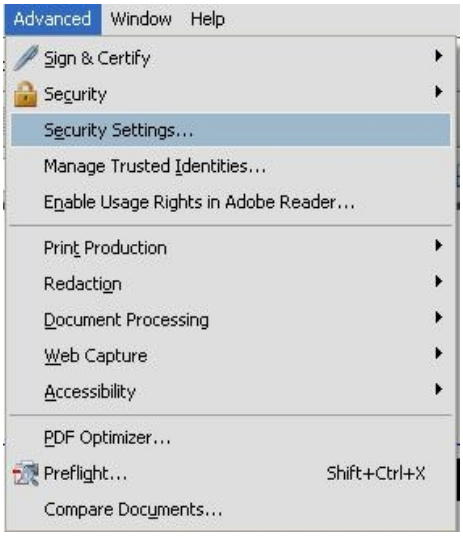
10.1 Εισαγωγή

Σε αυτή την παράγραφο θα ασχοληθούμε με την υπογραφή σε αρχεία pdfs. Υπογράφοντας ένα αρχείο pdf με το ψηφιακό μας πιστοποιητικό ουσιαστικά πιστοποιούμε ποίος δημιουργήσε αυτό το αρχείο. Επιπλέον όμως μπορούμε να το διασφαλίσουμε έτσι ώστε να μην μπορεί κανείς να αλλάξει κάτι ούτε να το αντιγράψει, να μπορεί μονάχα να το διαβάσει.

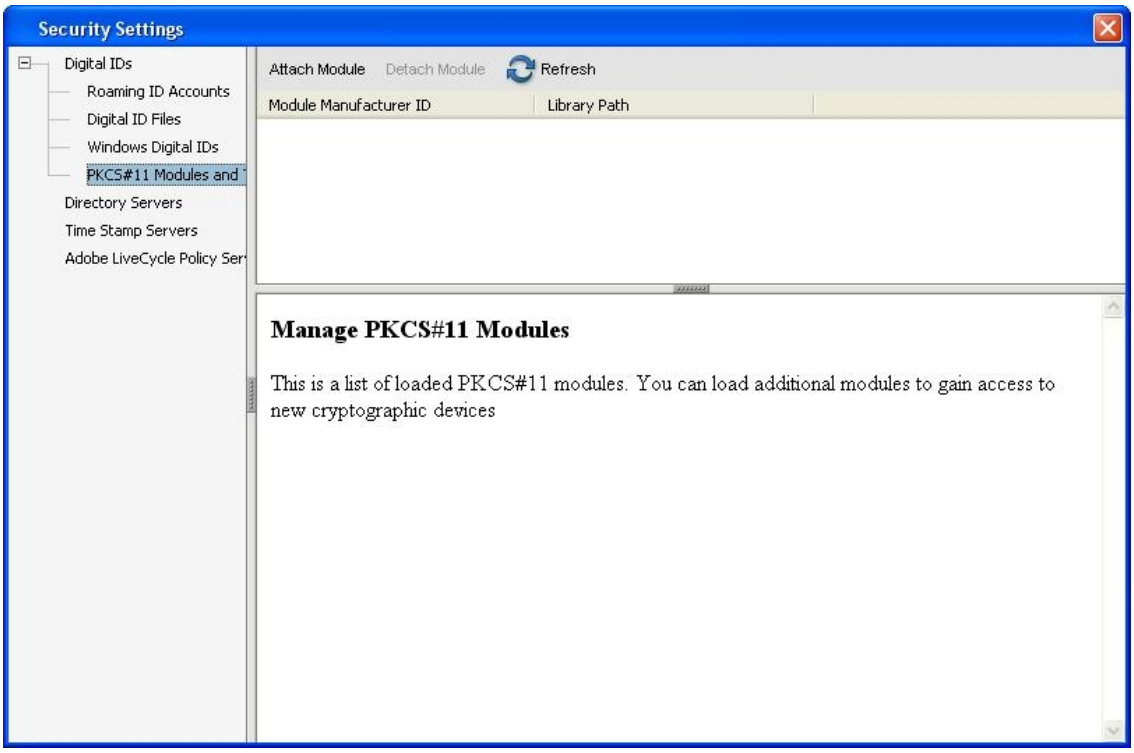
Στο παράδειγμα μας θα χρησιμοποιήσουμε την έκδοση Adobe Acrobat professional 8 καθώς και το eToken 32k της Aladdin. Η σύνδεση της συσκευής etoken και της εφαρμογής Adobe Acrobat professional 8 θα επιτευχθεί με την βοήθεια της βιβλιοθήκης PKCS#11 η οποία παρέχεται από την Aladdin. Τέλος το λειτουργικό σύστημα που θα χρησιμοποιήσουμε είναι τα Windows Xp Pro.

10.2 Ρυθμίσεις του Adobe Acrobat professional 8

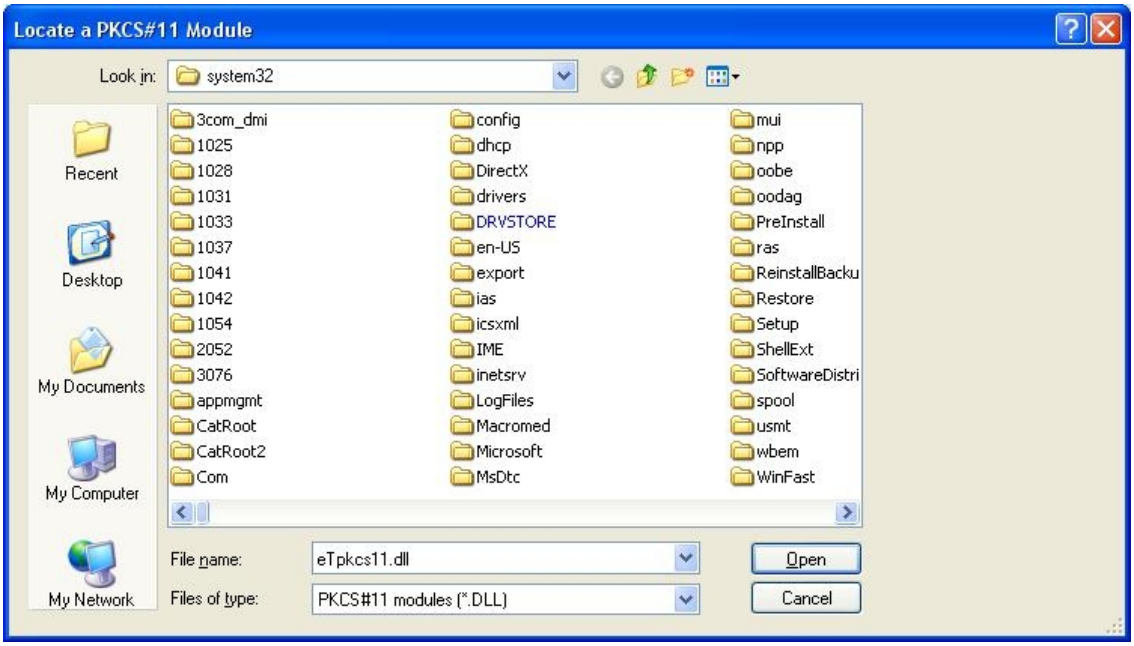
Ανοίγοντας την εφαρμογή επιλέγουμε "**Advanced**" και "**Security Settings**"



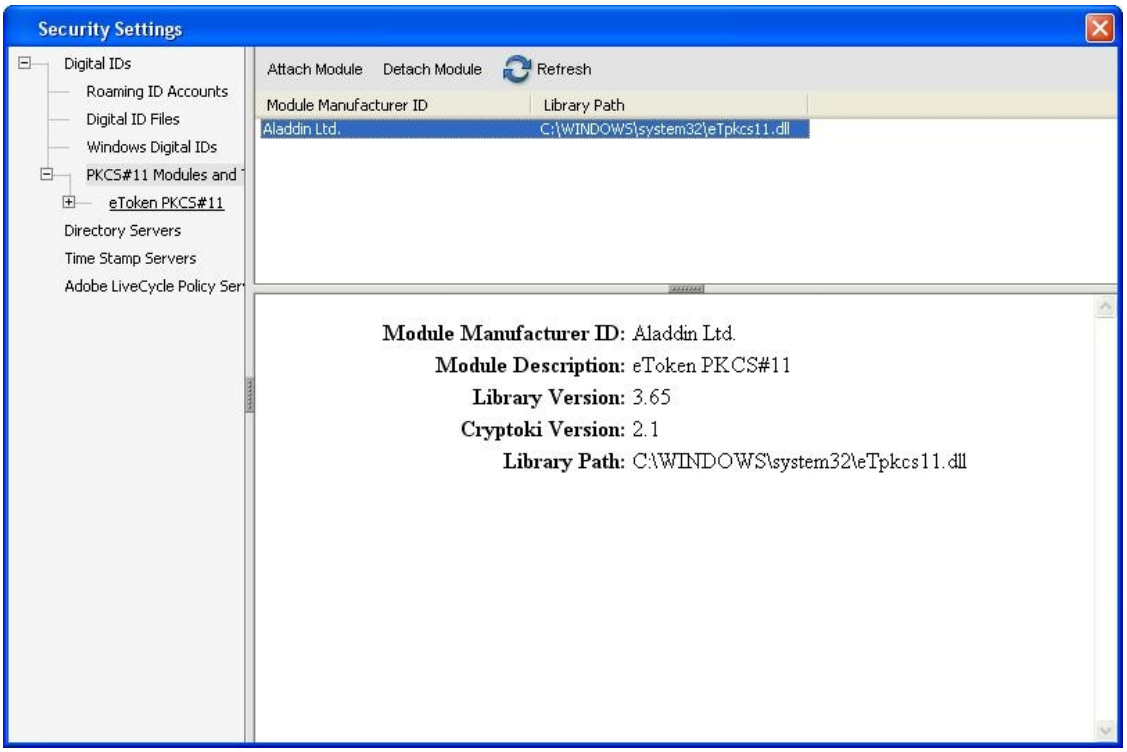
Θα μας εμφανισθεί η παρακάτω οθόνη στην οποία θα πρέπει να κάνουμε τις απαραίτητες ρυθμίσεις ώστε το Acrobat να αναγνωρίσει την συσκευή μας



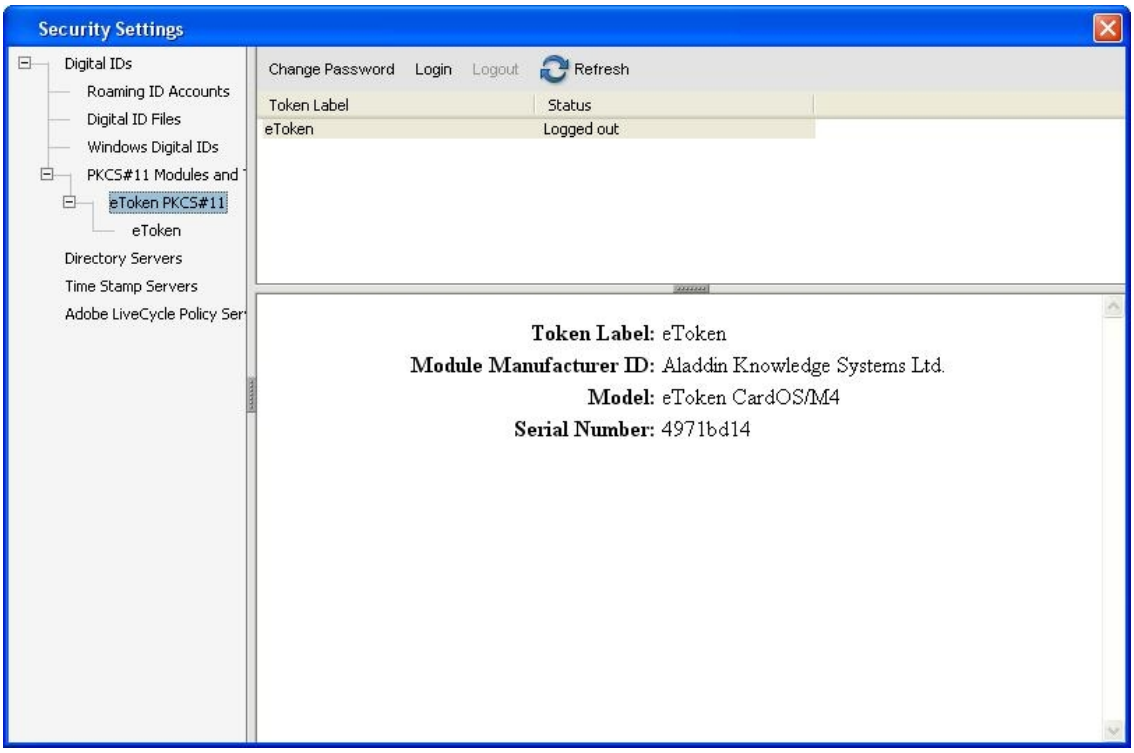
Επιλέγουμε το PKCS#11 Module… από το Digital IDs στο αριστερό τμήμα της οθόνης και στην συνέχεια την επιλογή "**Attach Module** ". Στο παρακάτω παράθυρο θα πρέπει να δώσουμε το Path στο οποίο βρίσκεται το PKCS#11 Module.



Το path είναι το εξής \windows\system32\eTpkcs11.dll (η βιβλιοθήκη αυτή είναι διαθέσιμη μονάχα αν έχει γίνει η εγκατάσταση του Run Time Enviroment της Aladdin). Επιλέγοντας "**Open**" εμφανίζεται η παρακάτω οθόνη



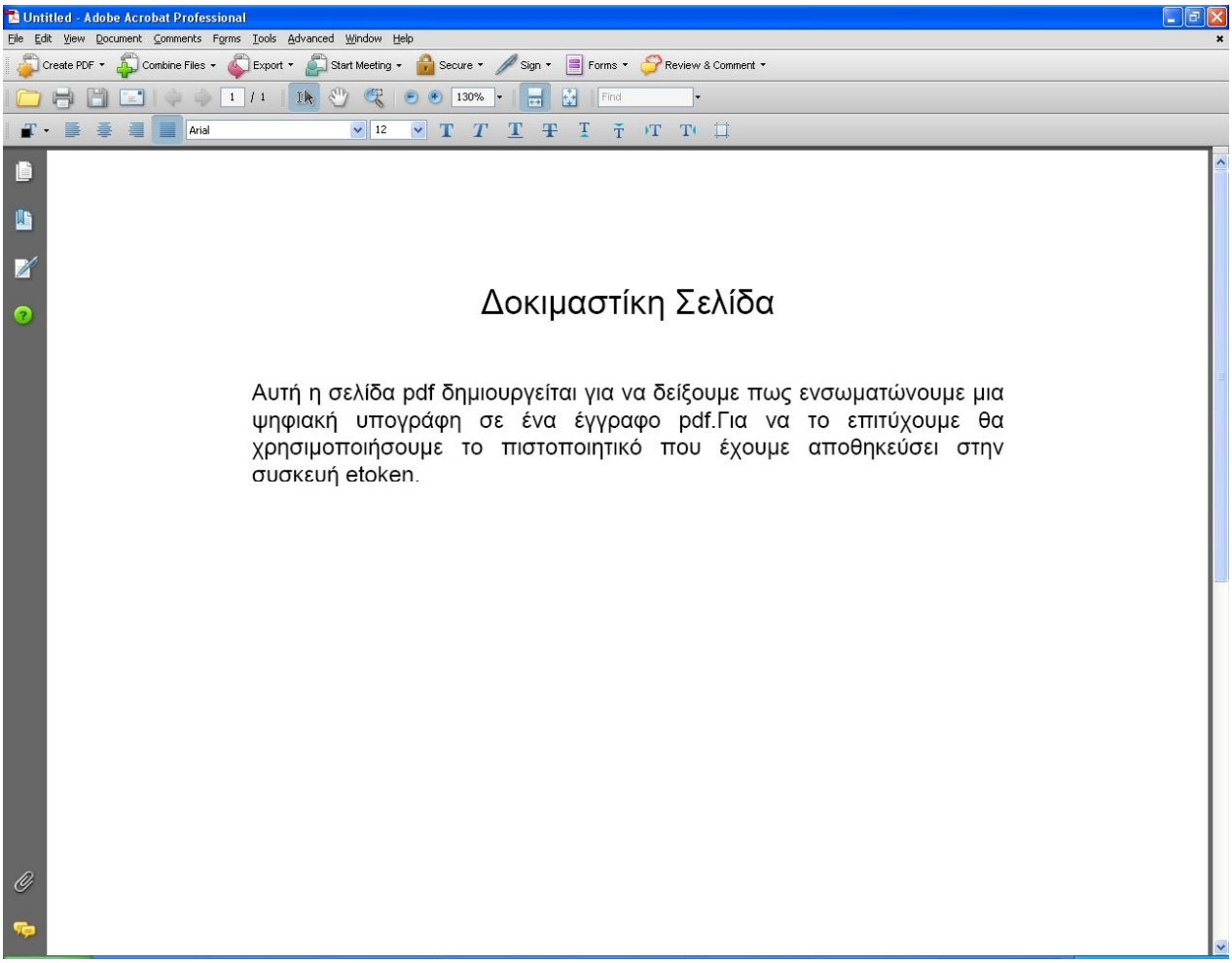
Παρατηρούμε ότι η επιλογή στο αριστερό τμήμα της οθόνης PKCS#11 Module… μπορεί να επεκταθεί, πατώντας + βλέπουμε την επιλογή **eTokenPKCS#11**. Επιλέγουμε το **eTokenPKCS#11** και αν δεν έχουμε συνδέση τη συσκευή eToken με τον υπολογιστή μας το κάνουμε τώρα. Θα εμφανισθεί η παρακάτω οθόνη



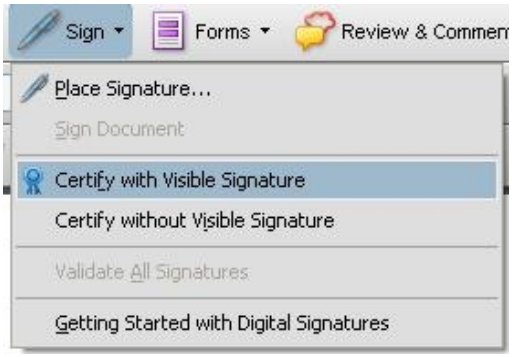
Στην δεξία πλευρά παρατηρούμε στο Token Label το όνομα που έχουμε δώσει στη συσκευή μας και στο Status αν η συσκευή μας είναι προσβάσιμη (για να γίνει προσβάσιμη θα πρέπει να εισάγουμε το κωδικό ασφαλείας στης συσκευής). Επίσης στο κάτω μέρος δεξία βλέπουμε κάποιες επιπλέον πληροφορίες σχετικά με την συσκευή μας.

10.3 Υπογράφοντας ένα έγγραφο

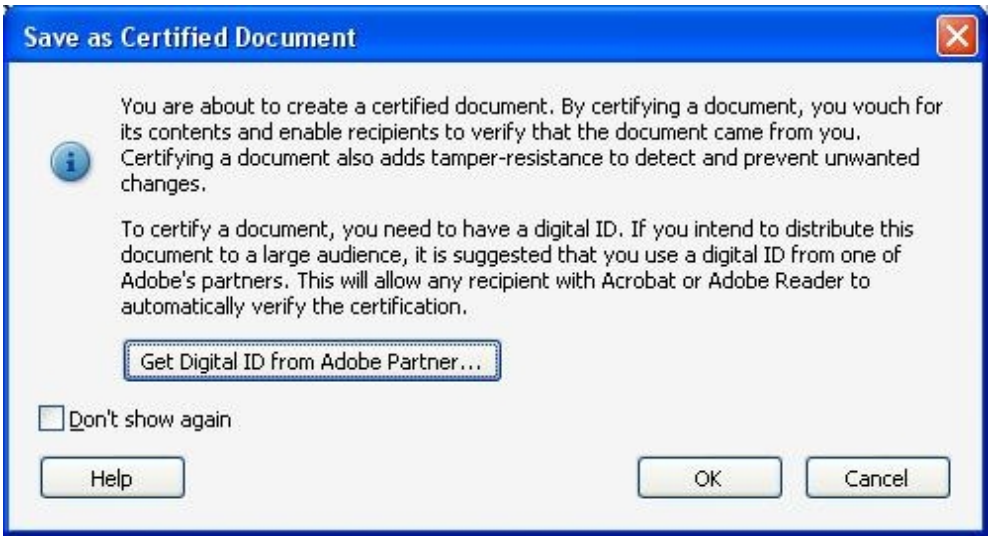
Δημιουργούμε ένα έγγραφο με τη βοήθεια του Acrobat, όπως βλέπουμε παρακάτω



Πρός το τέλος της μπάρας εργαλείων υπάρχει ένα εικονίδιο με σχήμα πένας το οποίο ονομάζεται "Sign" το επιλέγουμε και στη συνέχεια επιλέγουμε "**Certify with Visible Signature**" όπως βλέπουμε και στην παρακάτω μπάρα επιλογής

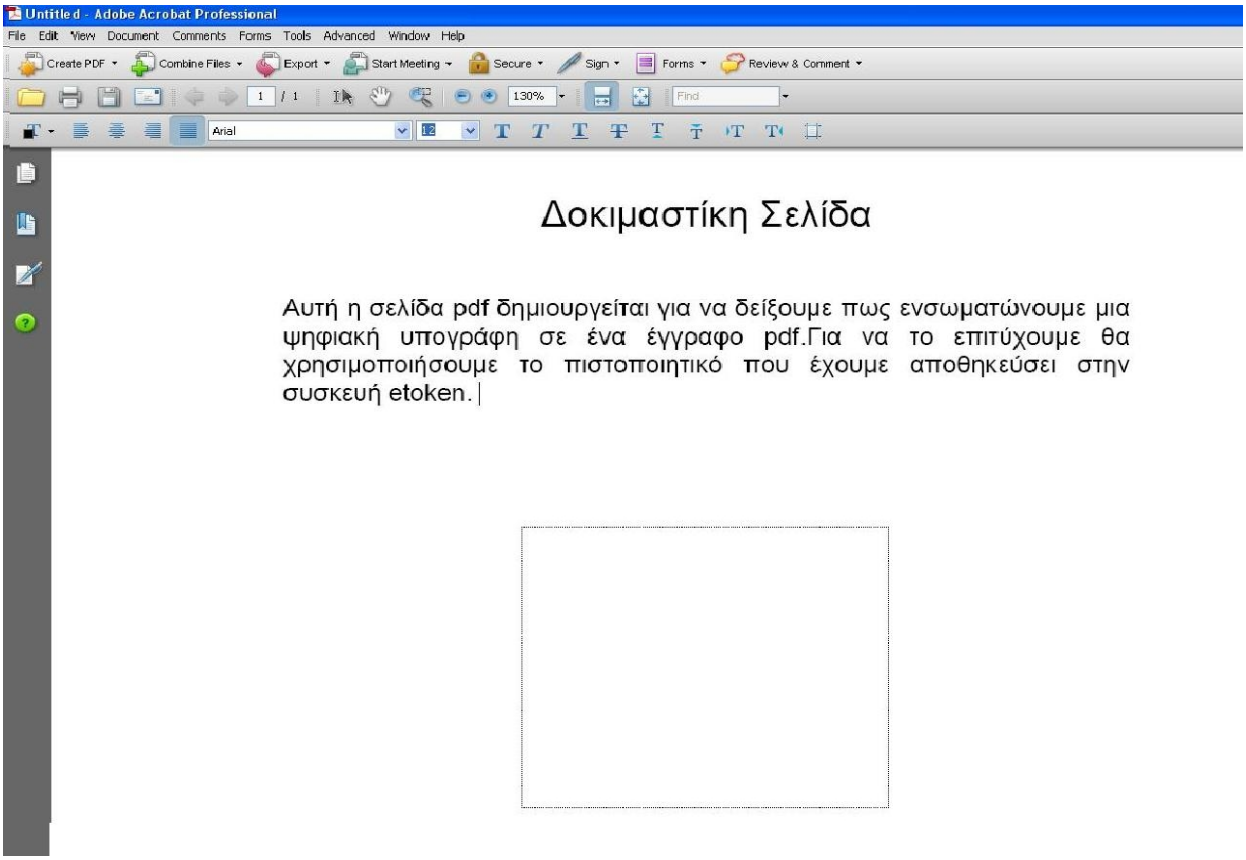


Θα εμφανισθεί ο παρακάτω διάλογος



Ο διάλογος αυτός μας ενημερώνει για τις ιδιότητες που θα έχει το εγγραφό μας όταν υπογραφεί με το ψηφιακό πιστοποιητικό και μας προτρέπει αν δεν έχουμε ένα ψηφιακό πιστοποιητικό να το αποκτήσουμε από κάποιο συνεργάτη της Adobe. Εμείς έχουμε οπότε επιλέγουμε "**Ok**". Εμφανίζεται ο παρακάτω διάλογος

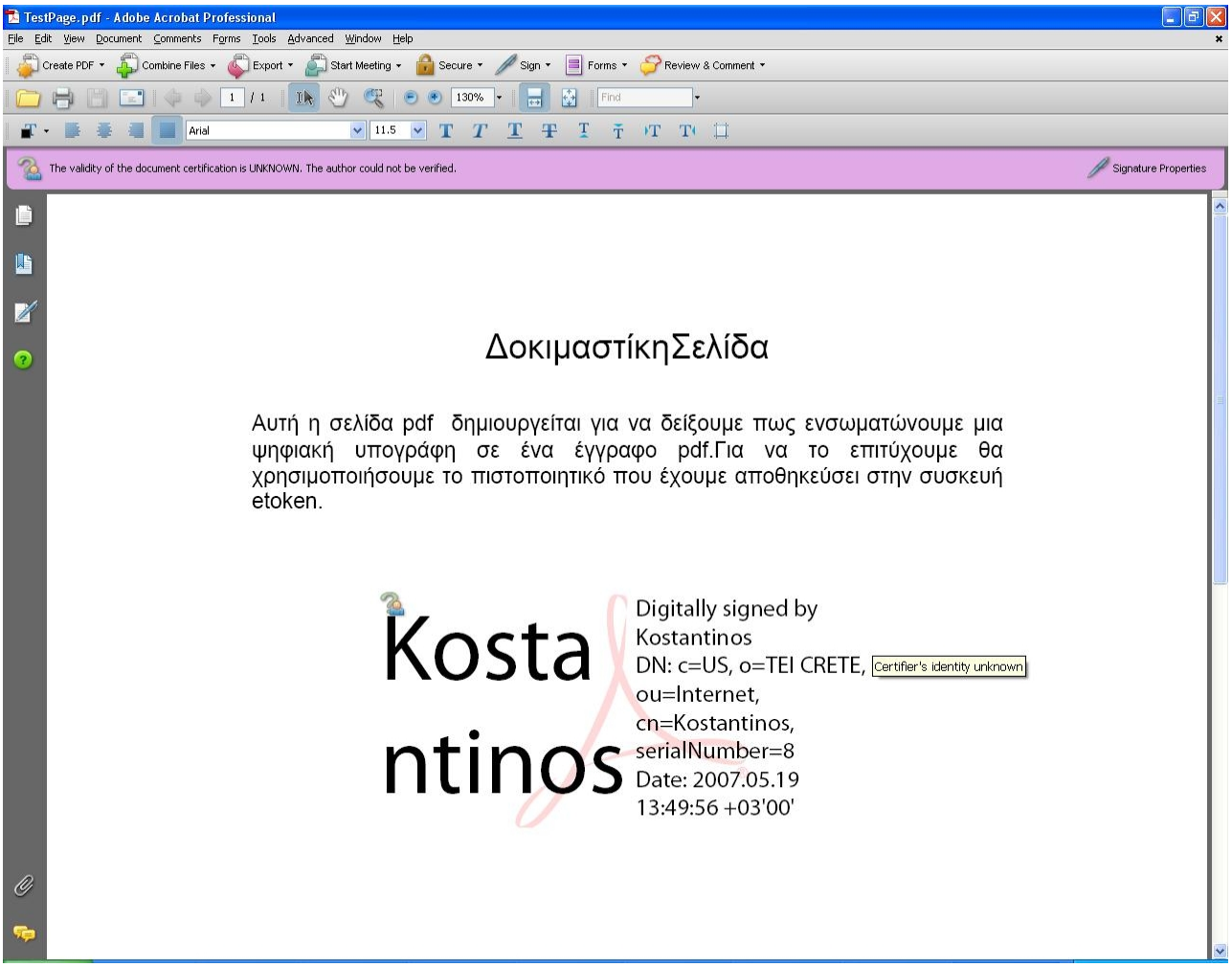
Παραπάνω ενημερωνόμαστε ότι πατώντας "**Ok**" θα πρέπει να επιλέξουμε ένα χώρο στο έγγραφο μας ώστε να τοποθετηθεί η υπογράφη μας. Επιλέγουμε λοιπόν το χώρο όπως βλέπουμε και παρακάτω



Στη συνέχεια εμφανίζονται τα επιμέρους στοιχεία της υπογραφής μας

Επιλέγοντας "**Sign**" θα μας ζητηθεί ένα path για αν αποθηκεύσουμε το αρχείο μας και στη συνέχεια θα μας ζητηθεί ο κωδικός ασφαλείας της συσκευής eToken όπως βλέπουμε και παρακάτω (αν δεν την έχουμε δώσει προηγουμένος κάνοντας Log in).

Η ψηφιακή μας υπογραφή εμφανίζεται στο εγγραφό μας

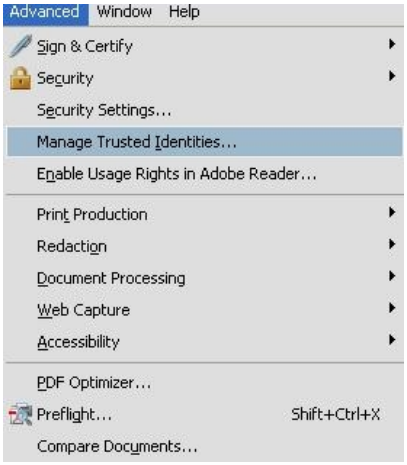


Όμως παρατηρούμε στο επάνω μέρος του εγγράφου ένα μήνυμα το οποίο λέει ότι δεν μπορεί να πιστοποιηθεί η εγγυρότητα του εγγράφου λόγω του ότι η αρχή πιστοποίηση είναι άγνωστη στην εφαρμογή της Adobe. Αυτό που πρέπει να κάνουμε είναι να εισάγουμε το πιστοποιητικό της Αρχής Πιστοποίησης, root CA, (η οποία έχει δημιουργήσει το πιστοποιητικό μας) στην εφαρμογή της Adobe.

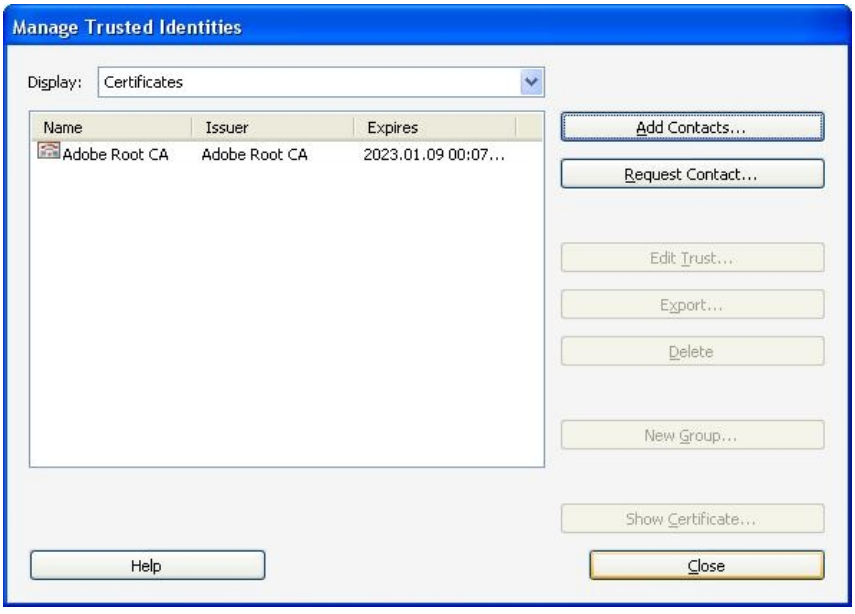
10.4 Ορίζοντας την Αρχή Πιστοποίησης

Η εφαρμογή της Adobe αρχικά γνωρίζει μονάχα το πιστοποιητικό της αρχής πιστοποίησης της Adobe. Στην δική μας περίπτωση, η αρχή πιστοποίηση δεν είναι η Adobe, οπότε θα πρέπει να εισάγουμε το πιστοποιητικό της δικής μας Αρχής Πιστοποίησης στην εφαρμογή ώστε να είναι σε θέση να αναγνωρίσει το ψηφιακό πιστοποιητικό του χρήστη.

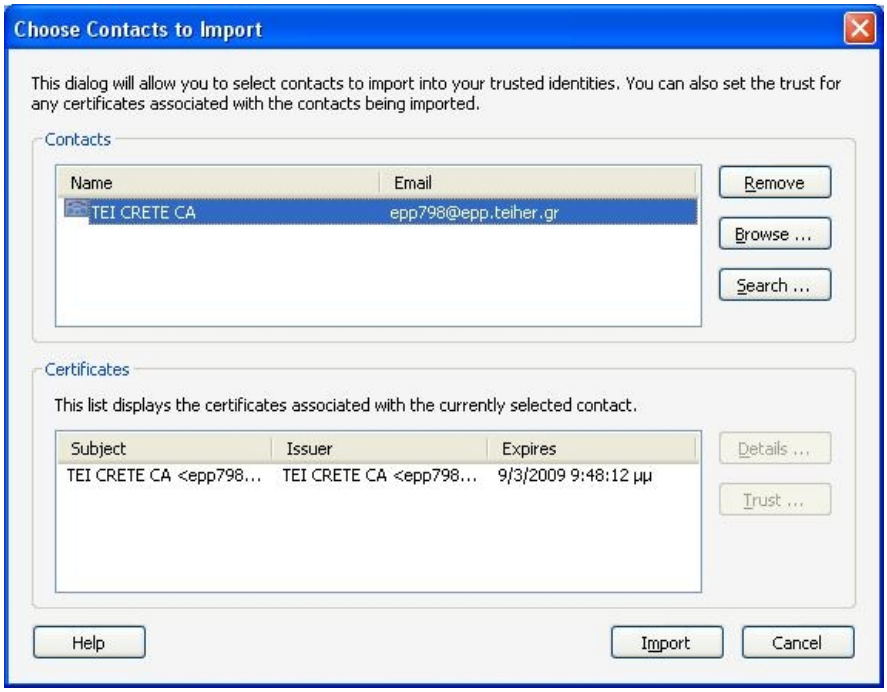
Για να επιτύχουμε το παραπάνω επιλέγουμε "**Advanced**" από την μπάρα και στη συνέχεια "Manage Trusted Identities… "



Εμφανίζεται το παρακάτω παράθυρο

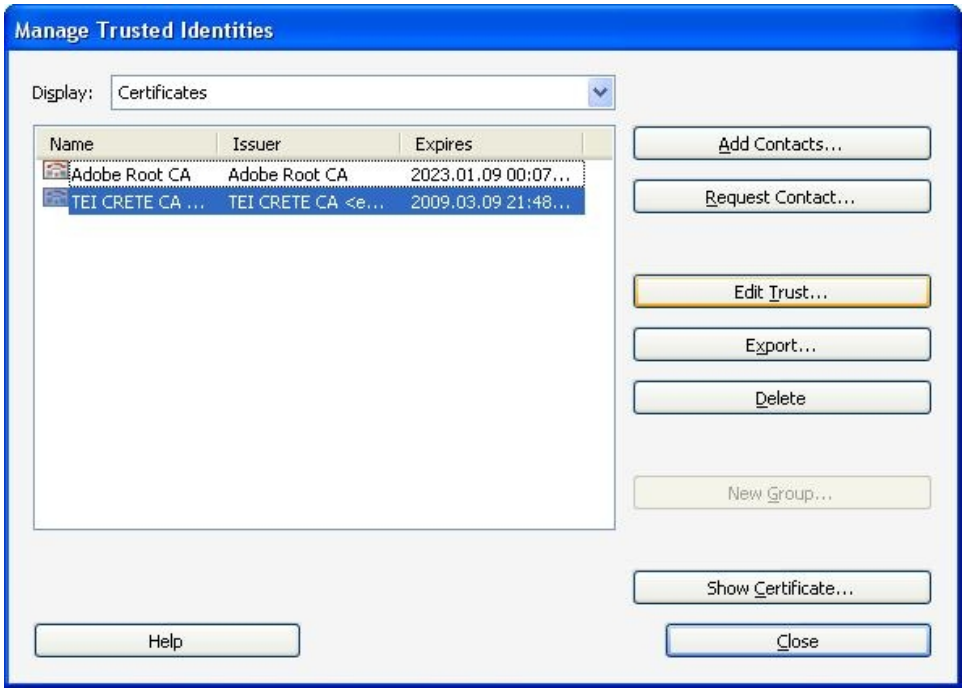


Παρατηρούμε ότι η μόνη αρχή πιστοποίησης όπως είπαμε και προηγουμένος είναι μονάχα η Adobe, για να είσάγουμε την δική μας επιλέγουμε "Add Contacts…" και εμφανίζεται το παρακάτω παράθυρο

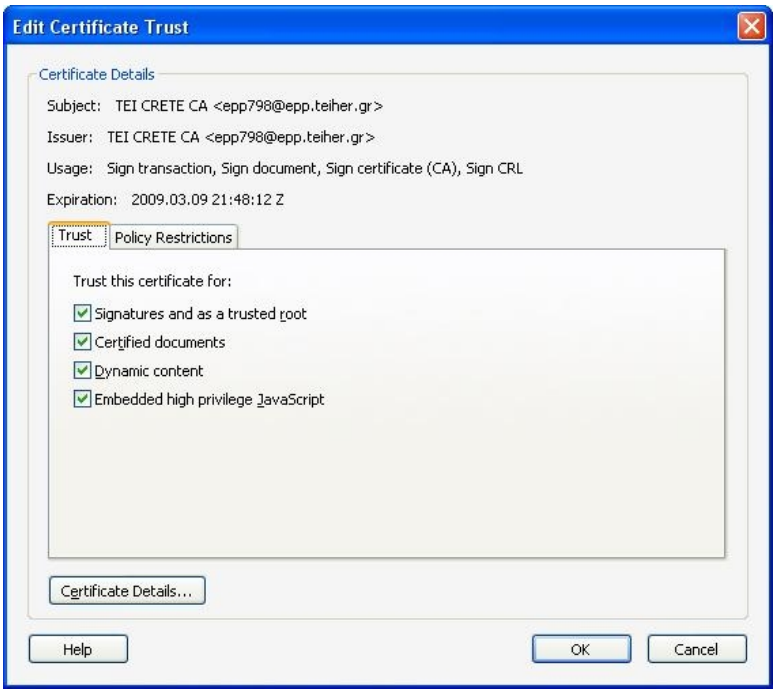


Το οποίο περιλαμβάνει στο τμήμα Contacts μια λίστα με γνωστά πιστοποιητικά (στην περίπτωση μας έχουμε ένα μόνο). Επιλέγουμε "**Browse**" βρίσκουμε το πιστοποιητικό της αρχής πιστοποίησης το οποίο βρίσκεται μέσα στην συσκευή μας και επιλέγουμε "**Import**", εμφανίζεται το παρακάτω μήνυμα

Επιλέγουμε "**Ok**" και βλέπουμε παρακάτω ότι το πιστοποιητικό της αρχή μας με όνομα **TEI CRETE CA** αναγνωρίσθηκε από την εφαρμογή



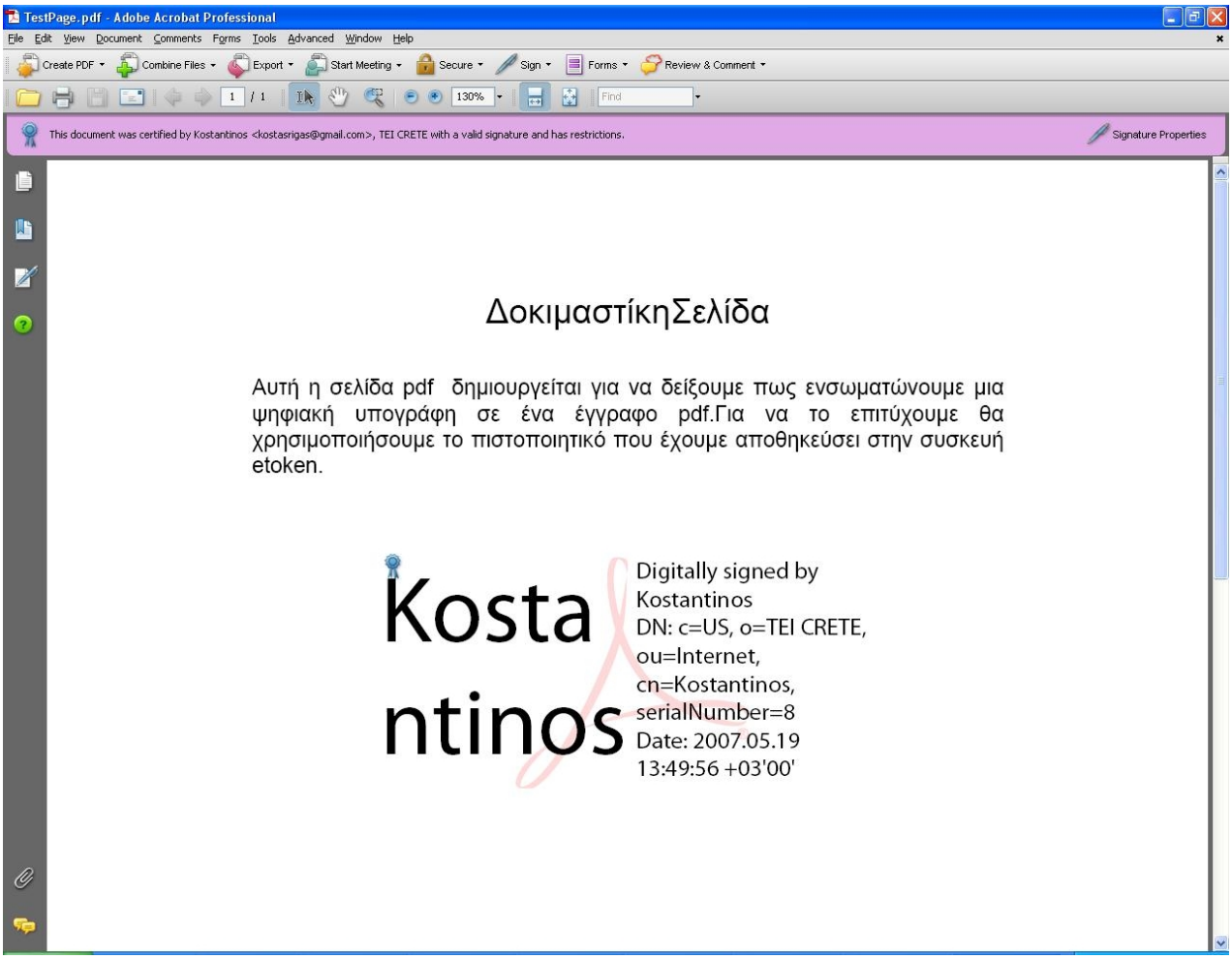
Επιλέγουμε από την λίστα το πιστοποιητικό **TEI CRETE CA** και μετά την επιλογή "**Edit Trust**"



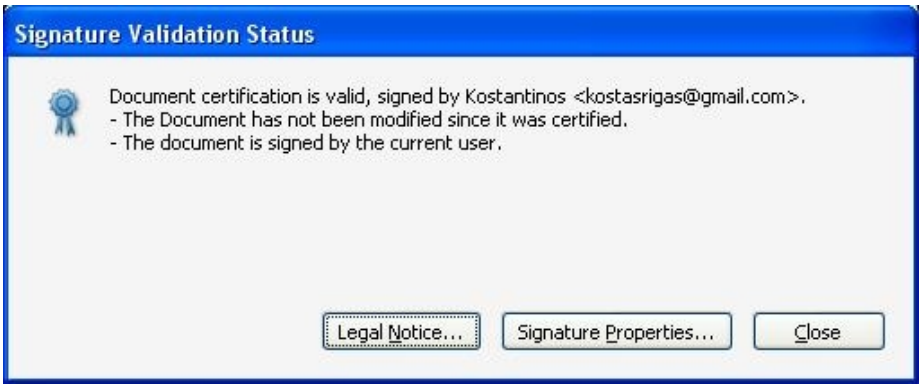
Τσεκάρουμε όλες τις επιλογές.

Τώρα είμαστε έτοιμοι να υπογράψουμε έγγραφα με πιστοποιητικά τα οποία έχουν παραχθεί από την αρχή πιστοποίηση **TEI CRETE** και θα μπορούν να αναγνωρισθούν κανονικά από το Adobe.

Ανοίγουμε ξανά το προηγούμενο αρχείο που είχαμε υπογράψει και παρατηρούμε ότι δεν εμφανίζεται πλέον το προειδοποιητικό μήνυμα για την αρχή πιστοποίησης



Εμφανίζεται ένα μήνυμα που μας ενημερώνει ότι το έγγραφο διαθέτει μια ψηφική υπογραφή από την Αρχή Πιστοποίησης TEI CRETE, αν επιλέξουμε "**Signature Properties**" εμφανίζεται το παρακάτω παράθυρο



Σε αυτή την παράγραφο δείξαμε πώς γίνεται με τη χρήση της συσκευής eToken 32K της Aladdin, η οποία περιέχει ένα πιστοποιητικό για τον χρήστη και ένα πιστοποιητικό για την Αρχή Πιστοποίησης, να υπογράψουμε ένα έγγραφο pdf από το Adobe Acrobat 8. Έτσι διασφαλίσαμε αυτό το έγγραφο από περαιτέρο αλλάγες των δεδομένων του και το πιστοποίησαμε με την υπογραφή μας έτσι ώστε να μπορεί να δεί ο κάθε χρήστης ότι το αρχείο αυτό έχει δημιουργηθεί από εμάς.

11. Συμπεράσματα

Τελειώνοντας αυτή τη μελέτη μπορούμε να αναφερθούμε σε κάποια βασικά συμπεράσματα που προέκυψαν.

Πρώτον, η χρήση των ασφαλών συσκευών δεν είναι κάτι το δύσκολο και δυσνόητο για τον χρήστη ο οποίος εύκολα μπορεί να τις διαχειριστεί και να τις χρησιμοποιήσει. Δεν χρειάζεται κάποια ιδιαίτερη τεχνογνωσία για την συσκευή που χρησιμοποιήσαμε (eToken 32 Pro της Aladdin) , συνδέεται με την usb θύρα του υπολογιστή και διαχειρίζετε εύκολα από το software RTE.

Δεύτερων, η χρησιμότητα των ασφαλών συσκευών είναι ουσιαστική και έχει σαν στόχο την διασφάλιση της ταυτότητας του χρήστη αλλά και την περαιτέρω ασφάλεια του στις διάφορες συναλλαγές. Έτσι οι χρήστες δεν χρειάζετε πια να ανησυχούν για τα ευαίσθητα δεδομένα τους όταν χρειάζετε να πιστοποιηθεί η ταυτότητα τους.

Τρίτων, η χρήση των ψηφιακών πιστοποιητικών σε συνδυασμό με μια ασφαλή συσκευή είναι η πιο ασφαλής λύση για την φύλαξη αλλά και την χρησιμοποίηση των ψηφιακών πιστοποιητικών. Διότι τα ψηφικά πιστοποιητικά διασφαλίζουν την ταυτότητα μας αλλά θα πρέπει να διασφαλίσουμε την αποθήκευση τους έτσι ώστε να μην υπάρξη παραβίαση των προσωπικών δεδομένων από τρίτους.

Βιβλιογραφία

Papers

CA in a Box paper by Mark Franklin, Kevin Mitcham, Sean Smith, Joshua Stabiner, and Omen Wild Dartmouth PKI Lab and Department of Computer Science

Using Dartmouth's "CA in a Box" by Ron DiNapoli Cornell Univercity

Links από το Internet

Openca

http://www.dartmouth.edu/~deploypki/

http://www.dartmouth.edu/~pkilab/

http://www.dartmouth.edu/~deploypki/CA/InstallOpenCALiveCD.html

http://www.dartmouth.edu/~deploypki/deploying/index.html

https://www.openca.org/projects/openca/

Web Server

http://www.apache.org/

http://en.wikipedia.org/wiki/Apache_HTTP_Server

http://en.wikipedia.org/wiki/Transport_Layer_Security

http://www.howtoforge.com/linux_apache2_ssl_php5_zendoptimizer_ioncubeloader

http://www.vanemery.com/Linux/Apache/apache-SSL.html

http://www.nslu2-linux.org/wiki/HowTo/EnableHTTPSforApache

http://www.apachefriends.org/en/xampp-linux.html

http://www.ubuntuforums.org/showthread.php?t=51753

http://ubuntuforums.org/showthread.php?t=4466

Certificates

http://en.wikipedia.org/wiki/Digital_certificates

http://en.wikipedia.org/wiki/S/MIME

Tokens

www.aladdin.com

http://en.wikipedia.org/wiki/Security_token