ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΑΣΦΑΛΕΙΑ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Τεύχος Εργαστηριακών Ασκήσεων

Μαρία Καρύδα

Επίκουρη Καθηγήτρια

Τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

Παναγιώτης Ριζομυλιώτης

Επίκουρος Καθηγητής

Τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

Αναστασία Δούμα

ΕΔΙΠ Σχολής Θετικών Επιστημών

Δημήτρης Παπαμαρτζιβάνος - Ιωάννα Τόπα

Υποψήφιοι Διδάκτορες Τμήματος Μ.Π.Ε.Σ

Μάιος 2017

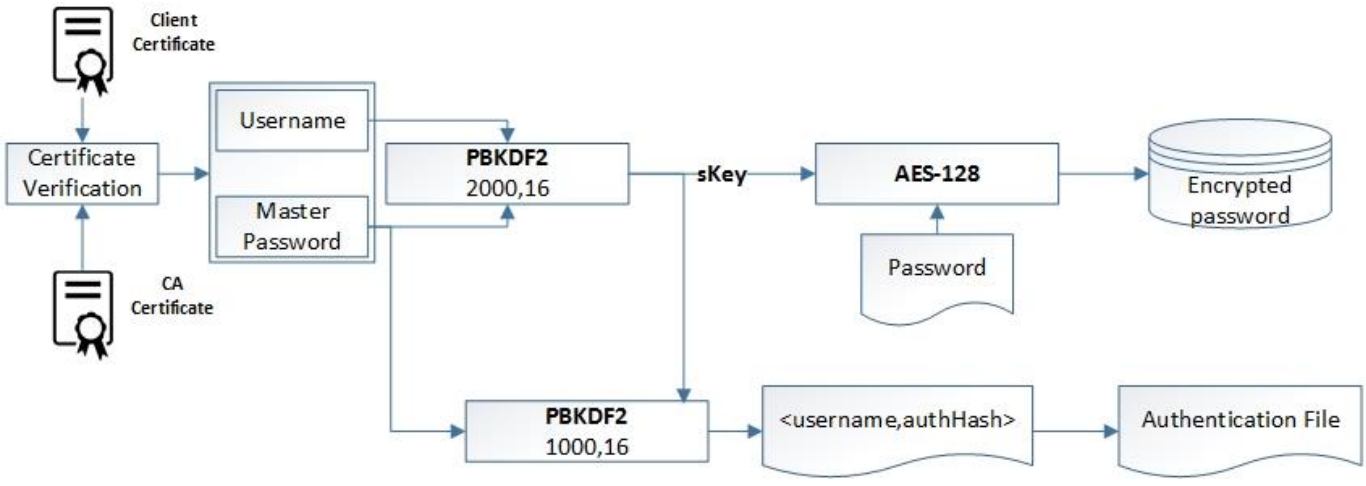
ΑΣΚΗΣΗ 3:
ΥΛΟΠΟΙΗΣΗ ΜΗΧΑΝΙΣΜΩΝ ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗΣ, ΚΡΥΠΤΟΓΡΑΦΗΣΗΣ ΚΑΙ ΣΥΝΟΨΗΣ – ΧΡΗΣΗ ΠΙΣΤΟΠΟΙΗΤΙΚΩΝ – ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΕ JAVA

Περιγραφή

Σκοπός της συγκεκριμένης εργασίας είναι κυρίως η εξοικείωση με το σύνολο των κλάσεων που παρέχονται από το Java API και από την βιβλιοθήκη Bouncy Castle (BC) [1], με σκοπό την υλοποίηση μηχανισμών αυθεντικοποίησης, κρυπτογράφησης, σύνοψης, ψηφιακής υπογραφής και διαχείρισης ψηφιακών πιστοποιητικών. Οι μηχανισμοί αυτοί μπορούν να χρησιμοποιηθούν σε διάφορες εφαρμογές με απαιτήσεις ασφαλείας.

Σε πρώτη φάση θα πρέπει να αναπτύξετε μία εφαρμογή σε Java, όπου μέσω μίας απλής διεπαφής χρήστη (GUI) θα δίνεται η δυνατότητα στους χρήστες να εγγράφονται σε μια υπηρεσία ασφαλούς διαχείρισης και αποθήκευσης κωδικών. Καλείστε ουσιαστικά να δημιουργήσετε ένα Password Manager (PM), με τη χρήση του οποίου ένας χρήστης μπορεί να διατηρεί τους διάφορους κωδικούς του αποθηκευμένους σε κρυπτογραφημένη μορφή. Ένας PM δίνει την δυνατότητα στο χρήστη να διατηρεί συγκεντρωτικά τους διάφορους κωδικούς που χρησιμοποιεί στην καθημερινότητά του και απαιτεί από αυτόν να θυμάται έναν και μοναδικό κωδικό, αυτόν δηλαδή που χρησιμοποιείται για την είσοδο στον PM (Master password). Η εφαρμογή σας θα πρέπει να είναι σε θέση να αυθεντικοποιεί τους χρήστες και να παρέχει μηχανισμούς διασφάλισης της εμπιστευτικότητας και της ακεραιότητας των αποθηκευμένων κωδικών.

Η εφαρμογή θα δίνει τη δυνατότητα σε ένα χρήστη (μέσω κατάλληλης διεπαφής) να δημιουργήσει νέο λογαριασμό στην υπηρεσία ή να αυθεντικοποιηθεί σε περίπτωση που έχει ήδη λογαριασμό. Οι διαδικασίες της αυθεντικοποίησης και της κρυπτογράφησης απεικονίζονται παρακάτω.



Α. Μηχανισμός αυθεντικοποίησης

Ο μηχανισμός αυθεντικοποίησης του PM βασίζεται στη χρήση ψηφιακών πιστοποιητικών X.509 και στη χρήση username/Master password. Η εφαρμογή θα πρέπει να λειτουργεί σαν μία Αρχή Πιστοποίησης (CA) η οποία εκδίδει ψηφιακά υπογεγραμμένα πιστοποιητικά στον εκάστοτε χρήστη που επιθυμεί να εγγραφεί (Register) στην εφαρμογή. Έτσι, κατά την είσοδο (Login) ενός χρήστη στην εφαρμογή, εκτός από τα username/ Master password, του ζητείται και το αρχείο του ψηφιακού πιστοποιητικού του.

Α.1. Δημιουργία λογαριασμού – Registration

Κατά την διάρκεια της δημιουργίας ενός νέου λογαριασμού χρήστη, ο τελευταίος δίνει τα προσωπικά του στοιχεία (ονοματεπώνυμο, email) και τα επιθυμητά username/Master password για την χρήση της εφαρμογής. Στη συνέχεια, βάσει των προσωπικών στοιχείων, η εφαρμογή θα εκδώσει ένα ψηφιακό πιστοποιητικό για τον χρήστη το οποίο είναι απαραίτητο για την ταυτοποίηση του κάθε φορά που ο χρήστης επιθυμεί να συνδεθεί (login) στην εφαρμογή.

Α.1.1. Διαδικασία διαχείρισης ψηφιακών πιστοποιητικών.

Κάνοντας χρήση των κατάλληλων κλάσεων και μεθόδων που παρέχονται από τη βιβλιοθήκη Bouncy Castle (BC) [1], καλείστε να διαχειριστείτε τις διαδικασίες έκδοσης των απαιτούμενων ψηφιακών πιστοποιητικών.

Αρχικά θα πρέπει να δημιουργήσετε ένα αυθυπόγραφο πιστοποιητικό (self-signed certificate) που θα ανήκει στην εφαρμογή. Η εφαρμογή (που έχει και το ρόλο της CA) θα μπορεί να εκδίδει πιστοποιητικά για κάθε νέο χρήστη. Για το σκοπό αυτό αφού ο χρήστης δώσει τα προσωπικά του στοιχεία και τα username/Master password, η εφαρμογή θα δημιουργεί ένα ζεύγος ιδιωτικού/δημόσιου κλειδιού για τον χρήστη. Στη συνέχεια, θα πρέπει να δημιουργείται ένα αίτημα υπογραφής πιστοποιητικού (certificate signing request - CSR) για τον χρήστη. Αφού γίνει επιβεβαίωση της ορθότητας της αίτησης, η εφαρμογή εκδίδει το σχετικό πιστοποιητικό για τα στοιχεία και το ζεύγος κλειδιών του χρήστη. Στο τέλος αυτής της διαδικασίας ο χρήστης θα έχει στην κατοχή του 1) ένα ζεύγος ιδιωτικού/δημόσιου κλειδιού και 2) το ψηφιακά υπογεγραμμένο πιστοποιητικό του. Για το σκοπό αυτό θα πρέπει να κάνετε export τα παραπάνω σε μορφή αρχείων.

Α1.2. Διαδικασία διαχείρισης username/Master password

Ο χρήστης χρειάζεται να θυμάται μόνο το username του και ένα μοναδικό μυστικό κωδικό Master Password για να αυθεντικοποιηθεί στην εφαρμογή. Τα username/Master password του χρήστη χρησιμοποιούνται για την παραγωγή του συμμετρικού κλειδιού (sKey) με το οποίο εξασφαλίζεται η εμπιστευτικότητα των διάφορων κωδικών που αποθηκεύονται στον PM. Η παραγωγή του συμμετρικού κλειδιού κρυπτογράφησης (sKey) θα γίνει ακολουθώντας το πρότυπο PBKDF2 (RFC2898) [2]. Η απλούστερη μορφή της PBFKD2 για την παραγωγή ενός κρυπτογραφικού κλειδιού είναι η εξής:

$$\boldsymbol{DK} = PBKDF2\ (P, S, c, dkLen)$$

Όπου:

- DK: derived key
- P: password
- S: salt
- c: iteration count (positive integer)
- dkLen: intended length of derived key (bytes)

Το συμμετρικό κλειδί κρυπτογράφησης (sKey) με το οποίο ο χρήστης κρυπτογραφεί τα αρχεία του παράγεται από το συνδυασμό του Master password και του username. Στη συνέχεια η ίδια διαδικασία ακολουθείται για την παραγωγή του authHash το οποίο όμως προέρχεται από το sKey και το Master Password του χρήστη. Το ζεύγος <Username, authHash> χρησιμοποιείται από την εφαρμογή για την αυθεντικοποίηση των χρηστών.

***sKey** = PBKDF2 (Master Password, Username, 2000, 16)*
***authHash** = PBKDF2 (sKey, Master Password, 1000, 16)*

Το συμμετρικό κλειδί κρυπτογράφησης δεν πρέπει σε καμία περίπτωση να αποθηκευτεί σε κάποιο αρχείο (υπάρχει μόνο στη μνήμη). Δημιουργείται κατά την είσοδο του χρήστη στην εφαρμογή και κατά την έξοδο η εφαρμογή θα πρέπει να διαγράφει κάθε αναφορά στο κλειδί του χρήστη.

Α.2 Είσοδος ήδη εγγεγραμμένου χρήστη στο σύστημα – Login

Κατά την είσοδο του χρήστη η εφαρμογή θα ζητάει το ψηφιακό του πιστοποιητικό (που εκδόθηκε κατά το registration) καθώς και τα username/Master password. Στη συνέχεια, αφού γίνει η επιβεβαίωση ότι το πιστοποιητικό έχει εκδοθεί από την ίδια την εφαρμογή, θα γίνεται αναζήτηση του ζεύγους <Username, authHash> του χρήστη και θα γίνεται σύγκριση της παραγόμενης με την αποθηκευμένη σύνοψη. Αν ταιριάζουν, τότε ο χρήστης θα αυθεντικοποιείται επιτυχώς στο σύστημα και μπορεί να συνεχίσει στις λειτουργίες της εφαρμογής.

Σε περίπτωση αποτυχημένης αυθεντικοποίησης η εφαρμογή θα ενημερώνει τον χρήστη με κατάλληλο μήνυμα. Συνολικά ο χρήστης θα έχει 10 προσπάθειες για να αυθεντικοποιηθεί. Ανάμεσα στις προσπάθειες αυθεντικοποίησης θα πρέπει να υπάρχει χρονοκαθυστέρηση που θα αυξάνεται όσο αυξάνεται ο αριθμός αποτυχημένων προσπαθειών. Το χρονικό διάστημα μεταξύ των προσπαθειών αφήνεται στην κρίση σας.

Β. Λειτουργίες Εφαρμογής

Όσοι χρήστες έχουν ακολουθήσει την παραπάνω διαδικασία εγγραφής θα έχουν τη δυνατότητα να χρησιμοποιήσουν την υπηρεσία του Password Manager. Για κάθε χρήστη η εφαρμογή θα δημιουργεί ένα κατάλογο (directory) με το username του στον οποίο θα φυλάσσονται τα διάφορα αρχεία που μπορεί να προκύψουν από τις διαδικασίες της εφαρμογής. Θα πρέπει να υλοποιήσετε κατάλληλη διεπαφή, όπου ο χρήστης θα έχει τις παρακάτω δυνατότητες:

- **Προσθήκη ενός νέου κωδικού:** Ο χρήστης θα πρέπει να έχει τη δυνατότητα να εισάγει μία νέα εγγραφή κωδικού. Με την ολοκλήρωση της εισαγωγής των απαραίτητων στοιχείων της εγγραφής θα πρέπει να γίνεται απευθείας και η κρυπτογράφησή της. Μία εγγραφή θα μπορούσε να έχει την παρακάτω μορφή. Μπορείτε να επεκτείνετε τη μορφή όπως επιθυμείτε.

<<Domain, Username, Password, Comment>>
<<dropbox.com, icsd13300, mypass123, Pass for academic login in dropbox>>
<<www.icsd.aegean.gr, icsd13300, IalwaysForgetThisPass!$^, university login>>

- **Εμφάνιση/Απόκρυψη (Κρυπτογράφηση/Αποκρυπτογράφηση):** Χρησιμοποιώντας το συμμετρικό κλειδί κρυπτογράφησης που παράγεται κατά την είσοδο του χρήστη

στην εφαρμογή, θα δίνεται η δυνατότητα στο χρήστη να αποκαλύπτει/αποκρυπτογραφεί ή να αποκρύπτει/κρυπτογραφεί κάποια αποθηκευμένη εγγραφή κωδικού.

- **Τροποποίηση κωδικού:** Θα πρέπει να δίνεται η δυνατότητα στο χρήστη να τροποποιήσει μία υπάρχουσα εγγραφή. Η λειτουργία της τροποποίησης θα πρέπει αυτόματα να πραγματοποιεί και την λειτουργία της αποκρυπτογράφησης ώστε να μπορεί ο χρήστης να δει το περιεχόμενο της εγγραφής που επέλεξε να τροποποιήσει.
- **Διαγραφή κωδικού:** Θα πρέπει να δίνεται η επιλογή ο χρήστης να διαγράφει μία εγγραφή.

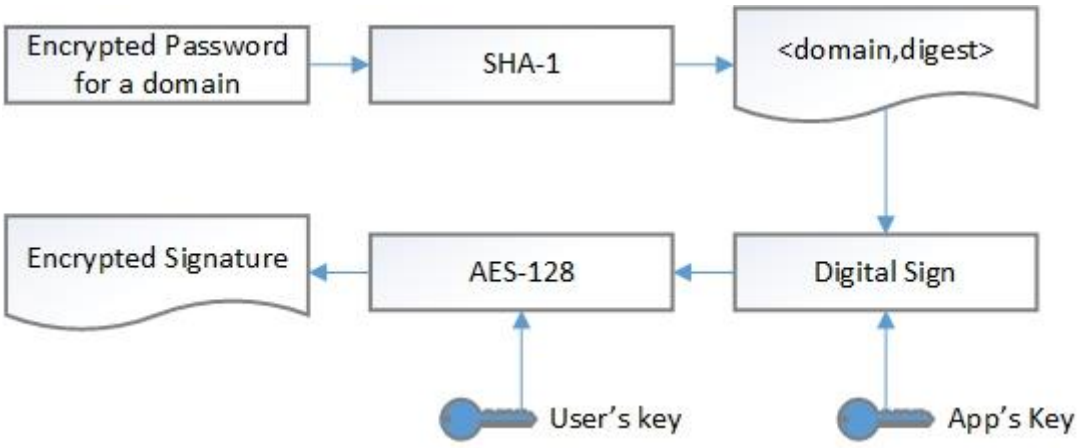
Για να μπορεί ο χρήστης να γνωρίζει ποια είναι η εγγραφή που πρέπει να αποκρυπτογραφήσει, θα πρέπει να αναγνωρίζει την σχετική εγγραφή από κάποιο χαρακτηριστικό γνώρισμα. Επιλέξτε ως χαρακτηριστικό το domain της εγγραφής.

Μπορείτε να προσθέσετε επιπλέον λειτουργίες σε περίπτωση που η υλοποίηση σας το απαιτεί. Σε κάθε περίπτωση όμως θα πρέπει να εξασφαλίζεται η λειτουργικότητα της εφαρμογής και να ικανοποιούνται οι απαιτήσεις της εργασίας.

Για να δει ο χρήστης μία εγγραφή κωδικού θα πρέπει να προηγηθεί η αποκρυπτογράφηση του, ενώ κατά το κλείσιμο της εφαρμογής θα πρέπει να κρυπτογραφούνται αυτόματα όσες εγγραφές έχει αποκρυπτογραφήσει ο χρήστης.

Γ. Μηχανισμός ακεραιότητας αρχείου κωδικών

Η εφαρμογή θα πρέπει να εξασφαλίζει την ακεραιότητα των κωδικών. Για το λόγο αυτό θα πρέπει να αναπτύξετε ένα μηχανισμό διασφάλισης των κωδικών από μη-εξουσιοδοτημένη τροποποίηση. Ο μηχανισμός αυτός ακολουθεί το παρακάτω σχήμα:



Κατά το κλείσιμο της εφαρμογής θα πρέπει να υπολογίζονται οι συνόψεις όλων των κρυπτογραφημένων κωδικών ξεχωριστά χρησιμοποιώντας τη μονόδρομη συνάρτηση κατακερματισμού SHA-1. Παράγωγο αυτής της διαδικασίας είναι τα ζεύγη <domain,digest>. Κάθε ζεύγος <domain,digest> θα υπογράφεται ψηφιακά από την εφαρμογή. Τέλος, η ψηφιακή υπογραφή θα κρυπτογραφείται με το συμμετρικό κλειδί του χρήστη (sKey) και θα αποθηκεύεται σε συγκεκριμένο αρχείο. Η αντίστροφη διαδικασία της επιβεβαίωσης θα πρέπει να ακολουθείται κατά την εκκίνηση της εφαρμογής. Ο χρήστης θα πρέπει να ενημερώνεται εάν έχει γίνει κάποια μη-εξουσιοδοτημένη τροποποίηση των κωδικών του είτε όχι με σχετικό μήνυμα.

Τροποποιείστε εσκεμμένα κάποιο κωδικό και επιβεβαιώστε ότι ο μηχανισμός ακεραιότητας που υλοποιήσατε λειτουργεί σωστά.

Τεχνικές λεπτομέρειες

- Η εφαρμογή θα πρέπει να έχει κατάλληλο keystore για την αποθήκευση του ιδιωτικού και δημόσιου κλειδιού της καθώς και του αυθυπόγραφου πιστοποιητικού της. Για την δημιουργία και διαχείριση ενός keystore μπορείτε να χρησιμοποιήσετε το εργαλείο keytool που παρέχεται από το Java JDK. Ωστόσο, προτείνεται η χρήση των κατάλληλων κλάσεων και μεθόδων του Java για την δημιουργία και διαχείριση των εγγραφών ενός keystore. Αν κάποια ομάδα επιλέξει τη χρήση keytool θα πρέπει στην αναφορά της να περιγράψει τον τρόπο λειτουργίας του εργαλείου και τις εντολές που χρησιμοποιήθηκαν.

- Η εφαρμογή θα πρέπει να είναι σε θέση να διαχειρίζεται πολλαπλούς χρήστες. Κάθε χρήστης θα έχει και το δικό του directory. Τα στοιχεία (subject, email) που θα δώσετε κατά την δημιουργία των ψηφιακών πιστοποιητικών θα πρέπει να είναι τα στοιχεία των μελών της ομάδας. Το ψηφιακό πιστοποιητικό της εφαρμογής (CA) θα πρέπει να έχει διάρκεια ισχύος $1^{ος}$ χρόνου, ενώ τα πιστοποιητικά των χρηστών 6 μήνες.

- Το συμμετρικό κλειδί κρυπτογράφησης (skey) δεν πρέπει σε καμία περίπτωση να αποθηκευτεί σε κάποιο αρχείο (υπάρχει μόνο στη μνήμη). Δημιουργείται κατά την είσοδο του χρήστη στην εφαρμογή και κατά την έξοδο η εφαρμογή θα πρέπει να διαγράφει κάθε αναφορά στο κλειδί του χρήστη.

- Θα πρέπει να παρουσιάσετε στιγμιότυπα των κρυπτογραφικών κλειδιών και των πιστοποιητικών στο log του περιβάλλοντος που προγραμματίζετε (π.χ. NetBeans).

- Για την κρυπτογράφηση των κωδικών (με το sKey) θα χρησιμοποιήσετε AES-128, ενώ το ζεύγος Δημόσιου-Ιδιωτικού κλειδιού της εφαρμογής θα είναι RSA-2048.

- Οι διαδικασίες δημιουργίας και υπογραφής πιστοποιητικών θα γίνουν με χρήση κώδικα Java αξιοποιώντας το API της βιβλιοθήκης Bouncy Castle. Στα παρακάτω Links μπορείτε να βρείτε βοηθητικό υλικό και παραδείγματα χρήσης της βιβλιοθήκης:
  - Java Wiki: https://www.bouncycastle.org/wiki/display/JA1/Java+APIs+1.X
  - Examples: http://media.wiley.com/product_ancillary/30/07645963/DOWNLOAD/beg_crypto_examples.zip
  - Crypto Workshop Guide: https://www.cryptoworkshop.com/guide/cwguide-070313.pdf
  - More examples: https://www.cryptoworkshop.com/guide/cwcode-070313.zip

- Για την υποστήριξη της υλοποίησης σας αρκεί να κατεβάσετε και να προσθέσετε στο java project σας το provider jar (bcprov-jdk15on-156.jar) που προσφέρετε στο site της Bouncy Castle, στην ενότητα latest java releases.

- Τα πιστοποιητικά και τα ζεύγη δημόσιων και ιδιωτικών κλειδιών, τόσο της εφαρμογής όσο και των χρηστών, αποτελούν μέρος των παραδοτέων της εργασίας. Επομένως θα πρέπει να αναπτύξετε κατάλληλο κώδικα για την αποθήκευση (export) των παραπάνω σε κατάλληλα αρχεία.

- Τα πιστοποιητικά που θα δημιουργηθούν πρέπει να ακολουθούν το πρότυπο X.509.

- Για την ανάπτυξη του γραφικού περιβάλλοντος της εφαρμογής μπορείτε να χρησιμοποιήσετε GUI Builder, αν το επιθυμείτε.

- Ο πηγαίος κώδικας θα αξιολογηθεί ως προς το αν υλοποιεί τα βασικά ζητούμενα της εκφώνησης, εκτελείται χωρίς να προκύπτουν σφάλματα λογισμικού (bugs), ακολουθεί «καλές αρχές» προγραμματισμού (π.χ. σχολιασμό, στοίχιση, εύγλωττη ονοματοδοσία μεταβλητών, επαναχρησιμοποίηση κώδικα, κλπ).

Παραδοτέα

- **Πηγαίος κώδικας** της εφαρμογής που αναπτύξατε με αναλυτικά σχόλια.
- **Αναφορά** που θα περιέχει οθόνες εκτέλεσης **ΟΛΩΝ** των λειτουργιών της εφαρμογής και **σχολιασμός των αποτελεσμάτων**. Οι οθόνες εκτέλεσης θα πρέπει να είναι **ευκρινείς**!
- Στην αναφορά σας θα πρέπει να περιγράψετε τα βήματα που ακολουθήσατε για τη δημιουργία και υπογραφή των ψηφιακών πιστοποιητικών.
- Τα πιστοποιητικά και τα ζεύγη δημόσιων και ιδιωτικών κλειδιών, τόσο της εφαρμογής όσο και των χρηστών, αποτελούν μέρος των παραδοτέων της εργασίας.
- Στο τέλος της εργασίας θα πρέπει να αναφέρετε όλες τις πηγές που έχετε χρησιμοποιήσει (και σχετικούς συνδέσμους σε όσες αναφορές υπάρχουν).

Οδηγίες Παράδοσης

Για την 3η εργασία θα πρέπει να παραδοθεί ένα συμπιεσμένο αρχείο που θα περιέχει ένα pdf αρχείο με όλα τα ζητούμενα της εργασίας (π.χ οθόνες εκτέλεσης, σχολιασμός των αποτελεσμάτων κλπ.), τα ψηφιακά πιστοποιητικά και ζεύγη κλειδιών καθώς και τα αρχεία με τον πηγαίο κώδικα των εργαλείων σας.

Το όνομα του αρχείου θα πρέπει να είναι της μορφής AM1_Lab03.zip (π.χ. icsd14001_Lab03.zip – αριθμός μητρώου του αρχηγού της ομάδας) και να μην περιέχει όλα τα μέλη της ομάδας.

Θα διορθωθούν μόνο οι ασκήσεις που πληρούν την παραπάνω περιγραφή.

Αναφορές

[1] https://www.bouncycastle.org/
[2] https://www.ietf.org/rfc/rfc2898.txt