ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Κατανεμημένα Συστήματα

Ενότητα # 7: Θέματα ασφάλειας

Διδάσκων: Γεώργιος Ξυλωμένος

Τμήμα: Πληροφορικής

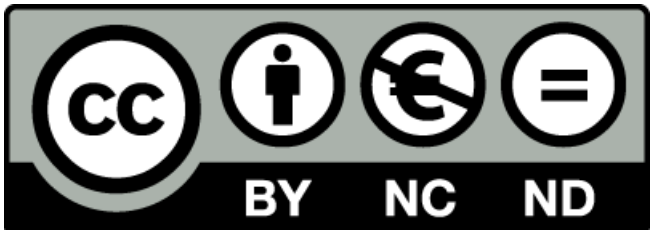
Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Οικονομικό Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Οι εικόνες προέρχονται από το βιβλίο «Κατανεμημένα Συστήματα με Java», Ι. Κάβουρας, Ι. Μήλης, Γ. Ξυλωμένος, Α. Ρουκουνάκη, 3η έκδοση, 2011, Εκδόσεις Κλειδάριθμος.



Σκοποί ενότητας

- Εξοικείωση με την αρχιτεκτονική ασφάλειας της Java και την έννοια των παρόχων ασφάλειας.
- Κατανόηση των βασικών διεπαφών και τάξεων ασφάλειας της Java (λίστες ελέγχου προσπέλασης, συνόψεις μηνυμάτων, κρυπτογραφία, ψηφιακές υπογραφές, ψηφιακά πιστοποιητικά).

Περιεχόμενα ενότητας

- Μηχανισμοί ασφάλειας
- Πάροχοι ασφάλειας
- Λίστες ελέγχου προσπέλασης
- Συνόψεις μηνυμάτων
- Κρυπτογραφία
- Ψηφιακές υπογραφές
- Ψηφιακά πιστοποιητικά

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Μηχανισμοί ασφάλειας

Μάθημα: Κατανεμημένα Συστήματα, **Ενότητα # 7:** Θέματα ασφάλειας

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

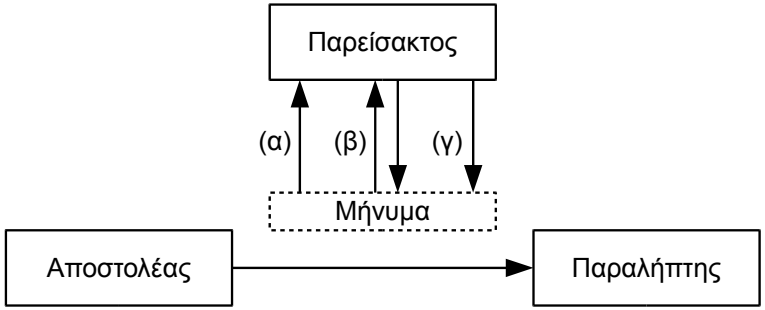
Κατανεμημένη ασφάλεια

- Ασφάλεια σε κατανεμημένα συστήματα
  - Οι μηχανές επικοινωνούν μέσω του δικτύου
    - Κάθε ανοιχτή θύρα κινδυνεύει από παραβίαση
  - Παρείσακτος: κλέβει ή αλλοιώνει πόρους
    - Πληροφορίες, εύρος ζώνης, υπολογιστική ισχύ
  - Η ασφάλεια εξαρτάται από την εφαρμογή
    - Τι επιθέσεις περιμένουμε να δούμε;
    - Πόσο σημαντικό είναι να τις αποτρέψουμε;

Ασφάλεια και Java

- Αρχιτεκτονική κρυπτογραφίας Java (JCA)
- Υπηρεσία πιστοποίησης ταυτότητας και εξουσιοδότησης Java (JAAS)
  - Παρέχουν διεπαφές προγραμματισμού (API)
  - Πάνω τους χτίζονται πιο σύνθετοι μηχανισμοί
    - Παράδειγμα: πιστοποίηση ταυτότητας Kerberos
- Πράκτορας (λογισμικού)
  - Λογισμικό που αντιπροσωπεύει το χρήστη

Μοντέλο παρείσακτου



- Έστω δύο επικοινωνούσες διεργασίες
  - Χρησιμοποιούν ένα ανασφαλές κανάλι
  - Ο παρείσακτος μπορεί να κάνει τρία πράγματα
    - Υποκλοπή μηνυμάτων (παθητικός)
    - Τροποποίηση μηνυμάτων (ενεργητικός)
    - Εισαγωγή νέων μηνυμάτων (ενεργητικός)

Διεπαφές της Java (1 από 2)

- Κρυπτογράφηση δεδομένων
  - Απόκρυψη περιεχομένου μηνυμάτων
    - Αποτρέπει την υποκλοπή μηνυμάτων
- Συνόψεις μηνυμάτων
  - Πιστοποίηση ακεραιότητας μηνύματος
    - Αποτρέπει την τροποποίηση μηνυμάτων
- Ψηφιακές υπογραφές
  - Πιστοποίηση αποστολέα μηνύματα
    - Αποτρέπει την εισαγωγή μηνυμάτων

Διεπαφές της Java (2 από 2)

- Ψηφιακά πιστοποιητικά
  - Πιστοποίηση ψηφιακής υπογραφής χρήστη
    - Δεν επιτρέπουν στον παρείσακτο να υποδύεται άλλους
- Λίστες ελέγχου προσπέλασης
  - Διαχείριση ταυτοτήτων πρακτόρων
  - Έλεγχος προσπέλασης των πόρων
    - Εξουσιοδότηση χρήσης των πόρων

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Πάροχοι ασφάλειας

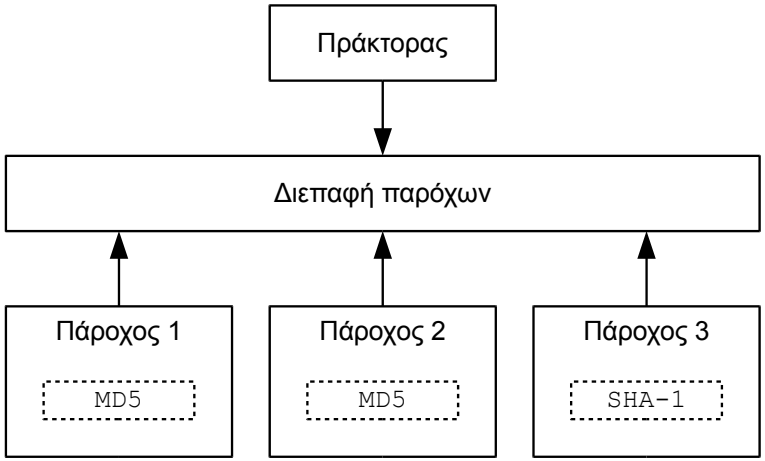
Μάθημα: Κατανεμημένα Συστήματα, **Ενότητα # 7:** Θέματα ασφάλειας

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Πάροχοι ασφάλειας (1 από 2)

- Πάροχος: πακέτο με υλοποίηση υπηρεσίας
  - Αποκρύπτει την υλοποίηση από την εφαρμογή
  - Συμβατότητα με διάφορες υλοποιήσεις
- Διαθέσιμοι πάροχοι ασφάλειας
  - SunJCE: DES, AES, RSA, MD5, SHA-1
  - Sun: DSA
  - SunRsaSign: RSA
  - Εγκατάσταση πρόσθετων παρόχων

Πάροχοι ασφάλειας (2 από 2)



- Ενιαία διεπαφή για τους πράκτορες
  - Επιλογή αλγορίθμου από συγκεκριμένο πάροχο
    - Υλοποίηση του MD5 από τον πάροχο 1
  - Επιλογή αλγορίθμου από οποιονδήποτε πάροχο
    - Οποιαδήποτε υλοποίηση του MD5

Τάξη Provider (1 από 2)

- Αφηρημένη τάξη Provider
  - Πάροχος που υλοποιεί υπηρεσίες ασφάλειας
    - Αλγόριθμοι ασφάλειας (όπως ο RSA)
    - Διαχείριση κλειδιών (παραγωγή, μετατροπή)
  - Ο πραγματικός πάροχος επεκτείνει την τάξη
    - Έχει όνομα και αριθμό έκδοσης
    - Όταν προσαρμόζεται λαμβάνει σειρά προτίμησης
    - Η σειρά ορίζει ποια υλοποίηση θα χρησιμοποιήσουμε

Τάξη Provider (2 από 2)

- Τάξη Provider: πακέτο java.security
  - Provider (String name, double version, String info)
    - Κατασκευάζει έναν πάροχο
      - Το info είναι ένα κείμενο περιγραφής
  - String getName()
  - double getVersion()
  - String getInfo()
    - Επιστρέφουν τις πληροφορίες του παρόχου

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Λίστες ελέγχου προσπέλασης

Μάθημα: Κατανεμημένα Συστήματα, **Ενότητα # 7:** Θέματα ασφάλειας

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Έλεγχος προσπέλασης

- Λίστα ελέγχου προσπέλασης (ACL)
  - Αποτελείται από στοιχεία ή καταχωρίσεις
  - Στοιχείο: ομάδα και δικαίωμα προσπέλασης
    - Η ομάδα μπορεί να είναι ένας πράκτορας
    - Το δικαίωμα δείχνει τι μπορεί να κάνει
  - Κωδικοποιεί κανόνες εξουσιοδότησης
  - java.security.acl: διεπαφές για ACL
  - sun.security.acl: προκαθορισμένες υλοποιήσεις

Διεπαφή Principal

- Εντολέας (principal): χρήστης ή οργανισμός
- Διεπαφή Principal
  - Ακυρώνει equals, hashCode και toString
    - Η υλοποίηση πρέπει να φτιάξει δικές της
  - String getName(): επιστρέφει όνομα εντολέα
- Τάξη PrincipalImpl: υλοποίηση της Principal
  - PrincipalImpl(String user)
    - Δημιουργεί έναν εντολέα με όνομα user

Διεπαφή Group (1 από 2)

- Παριστάνει μια ομάδα εντολέων
  - Επεκτείνει τη διεπαφή Principal
  - boolean addMember(Principal user)
    - Προσθέτει τον εντολέα user στην ομάδα
    - Επιστρέφει false αν ήταν ήδη μέλος
  - boolean removeMember(Principal user)
    - Αφαιρεί τον εντολέα user από την ομάδα
    - Επιστρέφει false αν δεν ήταν μέλος

Διεπαφή Group (2 από 2)

- Μέθοδοι διεπαφής Group
  - boolean isMember(Principal user)
    - Επιστρέφει true αν ο user είναι μέλος της ομάδας
    - Ελέγχει την ομάδα και τυχόν υποομάδες
  - Enumeration<? Extends Principal> members()
    - Επιστρέφει απαρίθμηση χρηστών (εντολείς ή ομάδες)
- Τάξη GroupImpl: υλοποίηση της Group
  - GroupImpl(String groupName)
    - Δημιουργεί μια ομάδα με όνομα groupName

Διεπαφή Permission

- Παριστάνει τα δικαιώματα χρήσης
  - Δεν υλοποιεί συγκεκριμένη συμπεριφορά
    - Δε γνωρίζουμε τι δικαιώματα θέλει ο χρήστης
  - Ακυρώνει τις μεθόδους equals και toString
- Τάξη PermissionImpl: υλοποίηση Permission
  - Χρησιμοποιεί συμβολοσειρές για τα δικαιώματα
  - PermissionImpl(String permission)
    - Κατασκευάζει ένα δικαίωμα με όνομα permission

Διεπαφή AclEntry (1 από 4)

- Παριστάνει ένα στοιχεία μιας ACL
  - Περιέχει ένα σύνολο από δικαιώματα
  - Σχετίζεται με ένα εντολέα ή μία ομάδα
  - Θετική: παραχωρεί δικαιώματα
    - Προεπιλογή αν δεν ορίζεται διαφορετικά
  - Αρνητική: αφαιρεί δικαιώματα
  - Έως μία θετική και μία αρνητική ανά εντολέα
    - Ανεξάρτητα από τα δικαιώματα κάθε μίας

Διεπαφή AclEntry (2 από 4)

- Μέθοδοι διεπαφής AclEntry
  - boolean addPremission(Permission perm)
    - Προσθέτει δικαίωμα perm στο στοιχείο
  - boolean checkPremission(Permission perm)
    - Ελέγχει αν περιέχεται το δικαίωμα perm
  - boolean removePremission(Permission perm)
    - Αφαιρεί το δικαίωμα perm από το στοιχείο
  - Enumeration<Permission> permissions()
    - Επιστρέφει τα δικαιώματα του στοιχείου

Διεπαφή AclEntry (3 από 4)

- Μέθοδοι διεπαφής AclEntry
  - Boolean setPrincipal(Principal user)
    - Καθορίζει τον εντολέα του στοιχείου
  - Principal getPrincipal()
    - Επιστρέφει τον εντολέα του στοιχείου
  - boolean isNegative(): είναι αρνητικό το στοιχείο;
  - void setNegativePermissions()
    - Κάνει το στοιχείο αρνητικό

Διεπαφή AclEntry (4 από 4)

- Τάξη AclEntryImpl
  - Υλοποιεί τη διεπαφή AclEntry
  - AclEntryImpl(Principal user)
    - Δημιουργεί ένα θετικό στοιχείο για τον user
  - AclEntryImpl()
    - Δημιουργεί ένα θετικό στοιχείο χωρίς εντολέα

Διεπαφή Owner (1 από 2)

- Παριστάνει τον ιδιοκτήτη μιας ACL
  - Χρήση και για τη δόμηση των ACL
    - Η διεπαφή Acl επεκτείνει τη διεπαφή Owner
  - Μόνο οι ιδιοκτήτες μιας ACL την αλλάζουν
  - boolean addOwner(Principal c, Principal o)
    - Προσθέτει στους ιδιοκτήτες τον o
    - Καλείται από τον ιδιοκτήτη c
  - boolean deleteOwner(Principal c, Principal o)
    - Διαγράφει από τους ιδιοκτήτες τον o

Διεπαφή Owner (2 από 2)

- Μέθοδοι διεπαφής Owner
  - boolean isOwner(Principal owner)
    - Ελέγχει αν ο owner ανήκει στους ιδιοκτήτες
- Τάξη OwnerImpl
  - Υλοποιεί τη διεπαφή Owner
  - OwnerImpl(Principal owner)
    - Δημιουργεί μια ομάδα ιδιοκτητών
    - Αρχικός ιδιοκτήτης είναι ο owner

Διεπαφή Acl (1 από 5)

- Παριστάνει μία ACL
  - Αποτελείται από στοιχεία (δικαιώματα, εντολείς)
  - Έχει μια ομάδα ιδιοκτητών με διεπαφή Owner
  - Τα θετικά και αρνητικά στοιχεία συνδυάζονται
    - Τα επιμέρους δικαιώματα συνεκτιμώνται
  - Τα στοιχεία εντολέων υπερισχύουν των ομάδων
    - Μια ομάδα μπορεί να έχει ορισμένα δικαιώματα
    - Ένα μέλος της μπορεί να έχει διαφορετικά δικαιώματα

Διεπαφή Acl (2 από 5)

- Μέθοδοι διεπαφής Acl
  - boolean addEntry(Principal c, AclEntry entry)
    - Προσθέτει το στοιχείο entry στη λίστα
    - Ο c είναι ο ιδιοκτήτης της ACL
    - Ο εντολέας του entry περιέχεται στο στοιχείο
  - boolean removeEntry(Principal c, AclEntry entry)
    - Αφαιρεί το στοιχείο entry από τη λίστα

Διεπαφή Acl (3 από 5)

- Μέθοδοι διεπαφής Acl
  - Enumeration<AclEntry> entries()
    - Επιστρέφει όλα τα στοιχεία της λίστας
  - Enumeration<Permission> getPermissions(Principal user)
    - Επιστρέφει τα δικαιώματα του εντολέα user
    - Περιλαμβάνει δικαιώματα ομάδας και του ίδιου
    - Επιστρέφονται όλα τα θετικά δικαιώματα

Διεπαφή Acl (4 από 5)

- Υπολογισμός δικαιωμάτων εντολέα
  - Σύνολο θετικών ομαδικών δικαιωμάτων
  - Σύνολο αρνητικών ομαδικών δικαιωμάτων
  - Συνεκτιμώνται τα δύο σύνολα
  - Σύνολο θετικών ατομικών δικαιωμάτων
  - Σύνολο αρνητικών ατομικών δικαιωμάτων
  - Συνεκτιμώνται τα δύο σύνολα
  - Τα ατομικά αναιρούν τα ομαδικά

Διεπαφή Acl (5 από 5)

- Μέθοδοι διεπαφής Acl
  - boolean checkPermission(Principal principal, Permission perm)
    - Επιστρέφει true αν ο principal έχει το δικαίωμα perm
  - void setName(Principal c, String name): ορίζει όνομα
  - String getName(): επιστρέφει το όνομα
- Τάξη AclImpl: υλοποίηση διεπαφής Acl
  - AclImpl(Principal owner, String name)
    - Δημιουργεί μια Acl με ιδιοκτήτη και όνομα

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Συνόψεις μηνυμάτων

Μάθημα: Κατανεμημένα Συστήματα, **Ενότητα # 7:** Θέματα ασφάλειας

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Γιατί συνόψεις;

- Σύνοψη μηνύματος (message digest)
  - Τιμή κατακερματισμού περιεχομένου
  - Ακολουθία δυαδικών ψηφίων σταθερού μήκους
  - Με κρυπτογραφικό αλγόριθμο κατακερματισμού
    - Λέγεται και αλγόριθμος σύνοψης μηνυμάτων
    - Παράδειγμα: MD5 (128 bit), SHA-1 (160 bit)
  - Χαρακτηρίζει μοναδικά το περιεχόμενο

Ιδιότητες συνόψεων (1 από 2)

- Κρυπτογραφική συνάρτηση κατακερματισμού
  - Υπολογιστικά ανέφικτη η αντιστροφή
    - Το μήνυμα δεν μπορεί να βρεθεί από τη σύνοψη
    - Άρα μπορεί να μεταδίδεται χωρίς κρυπτογράφηση
  - Υπολογιστικά ανέφικτη η κατασκευή μηνύματος
    - Αν έχουμε ένα μήνυμα και τον κατακερματισμό
    - Άρα δεν μπορούμε να αντικαταστήσουμε μηνύματα
    - Ακόμη κι αν τα έχουμε υποκλέψει

Ιδιότητες συνόψεων (2 από 2)

- Κρυπτογραφική συνάρτηση κατακερματισμού
  - Υπολογιστικά ανέφικτη η κατασκευή όμοιων
    - Δεν μπορούμε να φτιάξουμε μηνύματα με ίδια τιμή
    - Άρα τα μηνύματα δεν αντικαθίστανται
    - Δεν μπορούμε να αρνηθούμε ένα μήνυμα
  - Λειτουργεί ως ψηφιακό δακτυλικό αποτύπωμα
- Η συνάρτηση μπορεί να έχει συγκρούσεις
  - Απλά δεν μπορούμε να τις αξιοποιήσουμε

Τάξη MessageDigest (1 από 4)

- Αφηρημένη τάξη MessageDigest
  - Διαχειρίζεται συνόψεις μηνυμάτων
  - MessageDigest getInstance(String algorithm)
    - Επιστρέφει υλοποίηση του αλγορίθμου σύνοψης
  - Μπορούμε επίσης να επιλέξουμε τον πάροχο
    - Είτε με συμβολοσειρά είτε με αντικείμενο Provider
  - String getAlgorithm(): όνομα αλγορίθμου
  - Provider getProvider(): πάροχος αλγορίθμου

Τάξη MessageDigest (2 από 4)

- Υπολογισμός σύνοψης
  - Το μήνυμα είναι μια ακολουθία από byte
  - Το στέλνουμε στον αλγόριθμο σε τμήματα
  - void update(byte input): στέλνει ένα byte
  - void update(byte[] input, int offset, int len)
    - Στέλνει len bytes του πίνακα input από το offset
  - void update(byte[] input)
    - Στέλνει όλο τον πίνακα input

Τάξη MessageDigest (3 από 4)

- Υπολογισμός σύνοψης
  - void reset(): αρχικοποιεί τη σύνοψη
  - byte[] digest(): επιστρέφει τη σύνοψη
  - int digest(byte[] buf, int offset, int len)
    - Επιστρέφει τη σύνοψη στον πίνακα buf
    - Ξεκινά από τη θέση offset για έως και len θέσεις
  - byte[] digest(byte[] input)
    - Στέλνει και το input και μετά επιστρέφει τη σύνοψη

Τάξη MessageDigest (4 από 4)

- Μετά τον υπολογισμό γίνεται αρχικοποίηση
  - Δεν χρειάζεται ειδική κλήση αρχικοποίησης
- Έλεγχος σύνοψης
  - Έστω ότι λάβαμε μήνυμα και σύνοψη
  - Υπολογίζουμε μια νέα σύνοψη από το μήνυμα
  - Ελέγχουμε αν ταιριάζει με αυτή που λάβαμε
  - Boolean isEqual(byte[] digesta, byte[] digestb)
    - Συγκρίνει δύο συνόψεις για ισότητα

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Κρυπτογραφία

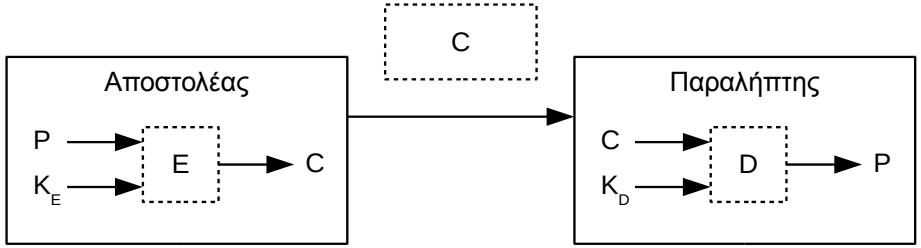
Μάθημα: Κατανεμημένα Συστήματα, **Ενότητα # 7:** Θέματα ασφάλειας

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Κρυπτογραφία (1 από 2)

- Διασφάλιση εμπιστευτικότητας μηνυμάτων
  - Η σύνοψη διασφαλίζει την ακεραιότητα
  - Η εμπιστευτικότητα απαιτεί κρυπτογραφία
- Κρυπτογράφηση
  - Παραγωγή κρυπτογραφημένου κειμένου
  - Από απλό (ή καθαρό) κείμενο
- Αποκρυπτογράφηση: αντίστροφη διαδικασία
  - Παράγει το απλό από το κρυπτογραφημένο

Κρυπτογραφία (2 από 2)



- Κρυπτογραφικός αλγόριθμος
  - Μαθηματική συνάρτηση (από)κρυπτογράφησης
  - Κλειδί: παραμετροποιεί τις συναρτήσεις
  - Έστω P το απλό και C το κρυπτογραφημένο
  - K_E: κλειδί κρυπτογράφησης, K_D: αποκρυπτογράφησης
  - Κρυπτογράφηση: $E(K_E,P)=C$
  - Αποκρυπτογράφηση: $D(K_D,C)=P$

Διεπαφή Key

- Παριστάνει οποιοδήποτε είδος κλειδιού
  - Κάθε κλειδί έχει τρία χαρακτηριστικά
  - String getAlgorithm(): αλγόριθμος κατασκευής
  - Byte[] getEncoded(): πρότυπη αναπαράσταση
    - Επιτρέπει ανταλλαγή κλειδιών έξω από τη JVM
  - String getFormat(): μορφότυπο κλειδιού
    - Για την πρότυπη αναπαράσταση
  - Επεκτείνεται από Public/Private/SecretKey

Μυστικά κλειδιά (1 από 3)

- Κρυπτογραφία μυστικού κλειδιού
  - Λέγεται και συμμετρική κρυπτογραφία
  - Ίδιο κλειδί κρυπτο-αποκρυπτογράφησης
    - Μυστικό, καταμεριζόμενο ή συμμετρικό κλειδί
  - Απλή και γρήγορη μέθοδος
    - Σε σχέση με κρυπτογραφία δημόσιου κλειδιού
  - Αν διαρρεύσει το κλειδί είναι άχρηστη
    - Το κλειδί όμως πρέπει να ανταλλαγεί!

Μυστικά κλειδιά (2 από 3)

- Τάξη KeyGenerator
  - Γεννήτρια συμμετρικών κλειδιών
  - Κατάλληλα για συγκεκριμένο αλγόριθμο
  - KeyGenerator getInstance(String algorithm)
    - Παράγει μια γεννήτρια κλειδιών για τον αλγόριθμο
  - Μπορούμε επίσης να επιλέξουμε τον πάροχο
    - Είτε με συμβολοσειρά είτε με αντικείμενο Provider
  - String getAlgorithm(): όνομα αλγορίθμου
  - Provider getProvider(): πάροχος αλγορίθμου

Μυστικά κλειδιά (3 από 3)

- Τάξη KeyGenerator
  - void init(AlgorithmParameterSpec params)
    - Αρχικοποιεί τη γεννήτρια κλειδιών
    - Χρήση συνόλου παραμέτρων params
  - SecretKey generateKey()
    - Παράγει ένα μυστικό κλειδί
    - Χωρίς init, χρήση προκαθορισμένων παραμέτρων
- Διεπαφή AlgorithmParameterSpec
  - Προδιαγραφή κρυπτογραφικών παραμέτρων

Δημόσια κλειδιά (1 από 5)

- Κρυπτογραφία δημόσιου κλειδιού
  - Λέγεται και ασύμμετρη κρυπτογραφία
  - Ζεύγος κλειδιών
    - Ιδιωτικό κλειδί: γνωστό στον ιδιοκτήτη
    - Δημόσιο κλειδί: γνωστό σε όλους
    - Δεν μπορεί να υπολογιστεί το ένα από το άλλο
  - Ο αποστολέας κωδικοποιεί με δημόσιο κλειδί
  - Ο παραλήπτης αποκωδικοποιεί με ιδιωτικό κλειδί

Δημόσια κλειδιά (2 από 5)

- Τα κλειδιά λειτουργούν και αντίστροφα
  - Κρυπτογράφηση με ιδιωτικό
  - Αποκρυπτογράφηση με δημόσιο
- Δημόσια ή μυστικά κλειδιά;
  - Τα δημόσια είναι πιο εύκολα στη διανομή
  - Είναι όμως πολύ πιο ακριβά υπολογιστικά
  - Χρήση δημόσιων για ανταλλαγή μυστικών
    - Συνδυασμός ευκολίας και ταχύτητας

Δημόσια κλειδιά (3 από 5)

- Τάξη KeyPair
  - Αντιπροσωπεύει ένα ζεύγος κλειδιών
  - KeyPair(PublicKey pubKey, PrivateKey privKey)
    - Κατασκευάζει ζεύγος κλειδιών pubKey/privKey
  - PublicKey getPublic()
    - Επιστρέφει το δημόσιο κλειδί
  - PrivateKey getPrivate()
    - Επιστρέφει το ιδιωτικό κλειδί

Δημόσια κλειδιά (4 από 5)

- Τάξη KeyPairGenerator
  - Γεννήτρια ζευγών κλειδιών
  - Κατάλληλα για συγκεκριμένο αλγόριθμο
  - KeyPairGenerator getInstance(string Algorithm)
    - Παράγει μια γεννήτρια κλειδιών για τον αλγόριθμο
  - Μπορούμε επίσης να επιλέξουμε τον πάροχο
    - Είτε με συμβολοσειρά είτε με αντικείμενο Provider
  - String getAlgorithm(): όνομα αλγορίθμου
  - Provider getProvider(): πάροχος αλγορίθμου

Δημόσια κλειδιά (5 από 5)

- Τάξη KeyPairGenerator
  - void initialize(AlgorithmParameterSpec params)
    - Αρχικοποιεί τη γεννήτρια κλειδιών
    - Χρήση συνόλου παραμέτρων params
  - KeyPair generateKeyPair()
    - Παράγει ένα ζεύγος κλειδιών
    - Χωρίς initialize, χρήση προκαθορισμένων παραμέτρων
    - Πρόσβαση στα κλειδιά με τις μεθόδους της KeyPair

Τάξη Cipher (1 από 5)

- Υλοποιεί κρυπτο-αποκρυπτογράφηση
  - Ανεξάρτητα από το είδος συστήματος
  - Ανεξάρτητα από τον συγκεκριμένο αλγόριθμο
  - Cipher getInstance(String transformation)
    - Παράγει αντικείμενο που υλοποιεί το transformation
  - Μπορούμε επίσης να επιλέξουμε τον πάροχο
    - Είτε με συμβολοσειρά είτε με αντικείμενο Provider
  - String getAlgorithm(): όνομα αλγορίθμου
  - Provider getProvider(): πάροχος αλγορίθμου

Τάξη Cipher (2 από 5)

- Ομάδες κρυπτογραφήματος
  - Ακολουθίες bit σταθερού μήκους
  - Η είσοδος είναι ακολουθία τέτοιων ομάδων
  - Οι ομάδες μπορεί να εξαρτώνται μεταξύ τους
    - Ανάλογα με τον τρόπο ανάδρασης
    - Ξεκινάμε με ένα διάνυσμα αρχικοποίησης
  - Σχήμα συμπλήρωσης για τελευταία ομάδα
    - Πρέπει όλες να έχουν το ίδιο μέγεθος

Τάξη Cipher (3 από 5)

- Μετασχηματισμός: συμβολοσειρά
  - Περιγράφει τις λειτουργίες που εκτελούνται
  - Τρία στοιχεία που χωρίζονται με /
    - Αλγόριθμος κρυπτογράφησης
    - Τρόπος ανάδρασης (προαιρετικά)
    - Σχήμα συμπλήρωσης (προαιρετικά)
    - Παράδειγμα: DES/ECB/PKCS5Padding
    - Προεπιλεγμένες τιμές για ό,τι δεν ορίζεται

Τάξη Cipher (4 από 5)

- Μέθοδοι τάξης Cipher
  - void init(int opmode, Key key, AlgorithmParametersSpec params)
    - Αρχικοποιεί κωδικοποιητή με κλειδί key
    - Λειτουργία ανάλογα με το init
    - ENCRYPT_MODE: κρυπτογράφηση
    - DECRYPT_MODE: αποκρυπτογράφηση
  - void init(int opmode, Key key)
    - Αρχικοποίηση με προκαθορισμένες παραμέτρους

Τάξη Cipher (5 από 5)

- Μέθοδοι τάξης Cipher
  - byte[] update(byte[] input)
    - Ξεκινά ή συνεχίζει την κωδικοποίηση
    - Παρέχεται το επόμενο τμήμα δεδομένων
    - Επιστρέφει την κωδικοποίηση του τμήματος
  - byte[] doFinal(): ολοκληρώνει κωδικοποίηση
    - Συμπληρώνει είσοδο και επιστρέφει αποτέλεσμα
    - Μετά επαναφέρει τον κωδικοποιητή στην αρχή
  - byte[] doFinal(byte[] input): παρέχει τελικά δεδομένα

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Ψηφιακές υπογραφές

Μάθημα: Κατανεμημένα Συστήματα, **Ενότητα # 7:** Θέματα ασφάλειας

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Ψηφιακές υπογραφές (1 από 4)

- Ψηφιακή υπογραφή
  - Υποκαθιστά τη φυσική υπογραφή
  - Συνδέεται με ένα μήνυμα
  - Επαληθεύεται στον παραλήπτη
  - Χρήση κρυπτογραφίας δημόσιου κλειδιού
- Υπολογισμός ψηφιακής υπογραφής
  - Υπολογισμός σύνοψης μηνύματος
  - Κωδικοποίηση σύνοψης με ιδιωτικό κλειδί

Ψηφιακές υπογραφές (2 από 4)

- Επαλήθευση ψηφιακής υπογραφής
  - Υπολογισμός νέας σύνοψης μηνύματος
  - Αποκρυπτογράφησης ψηφιακής υπογραφής
    - Χρήση δημόσιο κλειδιού του αποστολέα
  - Σύγκριση νέας και αρχικής σύνοψης
    - Αν είναι ίδιες, τότε έχουμε επαλήθευση
    - Τι σημαίνει ακριβώς επαλήθευση ή μη;

Ψηφιακές υπογραφές (3 από 4)

- Επιτυχής επαλήθευση
  - Το μήνυμα δεν έχει αλλοιωθεί
    - Αλλιώς θα άλλαζε η σύνοψη
  - Έχει υπογραφεί με συγκεκριμένο μυστικό κλειδί
    - Αυτό που ταιριάζει με το δημόσιο κλειδί
  - Πώς συνδέουμε πρόσωπο με κλειδί;
    - Πρέπει να γνωρίζουμε το δημόσιο κλειδί του
    - Πρέπει να μην έχει διαρρεύσει το ιδιωτικό κλειδί

Ψηφιακές υπογραφές (4 από 4)

- Ημερομηνία και ώρα υπογραφής
  - Συνήθως προστίθεται στο μήνυμα
  - Άρα περιλαμβάνεται στη σύνοψη
  - Αν λήξει το κλειδί, δεν ισχύουν οι υπογραφές
- Τι ακριβώς κρυπτογραφούμε;
  - Αν θέλουμε εμπιστευτικότητα, το μήνυμα
  - Αν θέλουμε ταυτότητα, μόνο την υπογραφή
    - Μεγάλο κέρδος σε χρόνο λόγο μικρότερου μεγέθους

Τάξη Signature (1 από 3)

- Αφηρημένη τάξη Signature
  - Παριστάνει αλγόριθμο ψηφιακών υπογραφών
    - Αλγόριθμοι κρυπτογράφησης και σύνοψης
  - Signature getInstance(String Algorithm)
    - Παράγει αντικείμενο που υλοποιεί τον αλγόριθμο
  - Μπορούμε επίσης να επιλέξουμε τον πάροχο
    - Είτε με συμβολοσειρά είτε με αντικείμενο Provider
  - String getAlgorithm(): όνομα αλγορίθμου
  - Provider getProvider(): πάροχος αλγορίθμου

Τάξη Signature (2 από 3)

- Μέθοδοι τάξης Signature
  - void initSign(PrivateKey privKey)
    - Αρχικοποιεί αλγόριθμο για υπογραφή
    - Περνάμε το ιδιωτικό κλειδί
  - void initVerify(PublicKey pubKey)
    - Αρχικοποιεί αλγόριθμο για επαλήθευση
    - Περνάμε το δημόσιο κλειδί

Τάξη Signature (3 από 3)

- Μέθοδοι τάξης Signature
  - void update(byte[] data): προσθήκη δεδομένων
    - Για υπογραφή ή επαλήθευση
    - Επαναλαμβάνουμε μέχρι να τελειώσουν
  - byte[] sign(): ολοκληρώνει υπογραφή
    - Επιστρέφει την ψηφιακή υπογραφή
  - boolean verify(byte[] sig): επαληθεύει υπογραφή
    - Περνάμε την υπογραφή προς επαλήθευση

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Ψηφιακά πιστοποιητικά

Μάθημα: Κατανεμημένα Συστήματα, **Ενότητα # 7:** Θέματα ασφάλειας

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Ψηφιακά πιστοποιητικά (1 από 5)

- Επίθεση του ενδιάμεσου
  - Έστω εισβολέας ανάμεσα σε δύο πράκτορες
    - Τα άκρα ανταλλάσσουν τα δημόσια κλειδιά τους
    - Ο εισβολέας τα αλλάζει με τα δικά του
  - Ο εισβολέας παρεμβάλλεται μεταξύ τους
    - Αποκρυπτογραφεί και κρυπτογραφεί τα πάντα
  - Πώς ξέρουμε αν έχουμε το σωστό δημόσιο κλειδί;
    - Δεν είναι μυστικό, αλλά μπορεί να έχει πειραχτεί

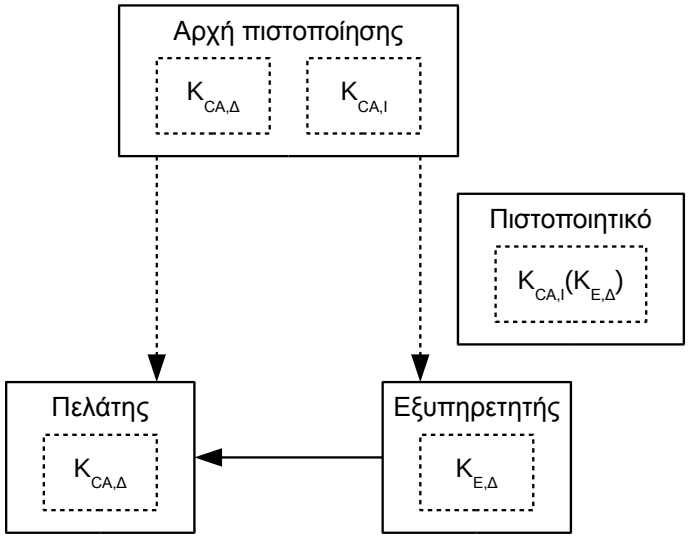
Ψηφιακά πιστοποιητικά (2 από 5)

- Ψηφιακό πιστοποιητικό
  - Εκδίδεται από κάποια αρχή πιστοποίησης (CA)
  - Η CA επιβεβαιώνει τον κάτοχο ενός κλειδιού
    - Συνήθως, ενός δημόσιου κλειδιού
  - Περιλαμβάνει κλειδί, όνομα και διεύθυνση
    - Επιπλέον όνομα CA και ημερομηνία λήξης κλειδιού
  - Υπογράφεται από τον εκδότη
    - Αρχή πιστοποίησης ή απλό χρήστη

Ψηφιακά πιστοποιητικά (3 από 5)

- Αυτοϋπογεγραμμένο κλειδί
  - Ο κάτοχος υπογράφει το κλειδί του
    - Χρήση του εργαλείου keytool στη Java
- Κλειδί υπογεγραμμένο από CA
  - Υπογράφεται με το ιδιωτικό κλειδί της CA
  - Το δημόσιο κλειδί της CA είναι ευρέως γνωστό
    - Προεγκατεστημένο σε λειτουργικό ή φυλλομετρητή
    - Καθένας μπορεί να ελέγξει το πιστοποιητικό

Ψηφιακά πιστοποιητικά (4 από 5)



- Παράδειγμα: αρχή πιστοποίησης CA
  - Όλοι γνωρίζουν το δημόσιο κλειδί της $K_{CA,\Delta}$
  - Υπογράφει το κλειδί $K_{E,\Delta}$ του εξυπηρετητή
  - Ο πελάτης επαληθεύει το κλειδί του εξυπηρετητή

Ψηφιακά πιστοποιητικά (5 από 5)

- Αλυσίδες πιστοποιητικών
  - Κάθε πιστοποιητικό βασίζεται σε προηγούμενο
  - Μήκος 1: αυτοϋπογεγραμμένο πιστοποιητικό
    - Μπορεί να είναι το πιστοποιητικό μιας CA
  - Μήκος 2: πιστοποιητικό χρήστη
    - Η CA πιστοποιεί την ταυτότητα του χρήστη
  - Μήκος 3: πιστοποιητικό άλλου χρήστη
    - Ο πιστοποιημένος χρήστης πιστοποιεί τον άλλο

Τάξη Certificate

- Αφηρημένη τάξη Certificate
  - Παριστάνει ένα ψηφιακό πιστοποιητικό
  - Τάξη X509Certificate: πιστοποιητικά X.509
  - void verify(PublicKey key)
    - Επαληθεύει το πιστοποιητικό ή ρίχνει εξαίρεση
    - Χρήση του δημόσιου κλειδιού key για επαλήθευση
  - PublicKey getPublicKey(): επιστρέφει το κλειδί
  - String toString(): επιστρέφει το πιστοποιητικό

Εργαλείο keytool (1 από 2)

- Διαχείριση κλειδιών/πιστοποιητικών X.509
  - Δημιουργία ζευγών κλειδιών
  - Παραγωγή αυτοϋπογεγραμμένων πιστοποιητικών
  - Έκδοση αιτήσεων για πιστοποιητικά προς CA
  - Εισαγωγή πιστοποιητικών σε αποθήκη
  - Χαρακτηρισμός δημόσιων κλειδιών ως έμπιστα
  - Διαχείριση αποθηκών κλειδιών
  - Χρήση του αρχείου .keystore

Εργαλείο keytool (2 από 2)

- Αποθήκη κλειδιών
  - Προστατευμένη βάση δεδομένων
  - Προστατεύεται από συνθηματικό
    - Πρόσθετο συνθηματικό για κάθε ιδιωτικό κλειδί
  - Καταχώριση κλειδιού
    - Μυστικό ή ιδιωτικό κλειδί
    - Αλυσίδα πιστοποιητικών για αντίστοιχο δημόσιο κλειδί
  - Καταχώριση έμπιστου πιστοποιητικού
    - Πιστοποιητικό δημόσιου κλειδιού που εμπιστευόμαστε

Τάξη KeyStore (1 από 4)

- Προσπέλαση και διαχείριση αποθήκης
  - KeyStore getInstance(String type)
    - Παράγει αποθήκη τύπου type
  - Java Key Store (JKS): υλοποίηση της Sun
    - Χρήση κωδικών για προστασία αποθήκης και κλειδιών
  - Μπορούμε επίσης να επιλέξουμε τον πάροχο
    - Είτε με συμβολοσειρά είτε με αντικείμενο Provider
  - Provider getProvider(): πάροχος αποθήκης

Τάξη KeyStore (2 από 4)

- Μέθοδοι τάξης KeyStore
  - void load(InputStream stream, char[] password)
    - Φορτώνει αποθήκη από το ρεύμα stream
    - Αν το stream είναι null, δημιουργία νέας
    - Χρήση κωδικού password για έλεγχο πρόσβασης
  - void store(OutputStream stream, char[] password)
    - Γράφει αποθήκη στο ρεύμα stream
    - Προστατεύει την αποθήκη με κωδικό password

Τάξη KeyStore (3 από 4)

- Μέθοδοι τάξης KeyStore
  - void setKeyEntry(String alias, Key key, char[] password, Certificate[] chain)
    - Συσχετίζει κλειδί key με ψευδώνυμο alias
    - Αλυσίδα πιστοποίησης chain για δημόσιο κλειδί
    - Προστασία με τον κωδικό password
  - void setCertificateEntry(String alias, Certificate cert)
    - Συσχετίζει πιστοποιητικό cert με ψευδώνυμο alias

Τάξη KeyStore (4 από 4)

- Μέθοδοι τάξης KeyStore
  - Key getKey(String alias, char[] password)
    - Επιστρέφει κλειδί με ψευδώνυμο alias
  - Certificate getCertificate(String alias)
    - Επιστρέφει το πιστοποιητικό με όνομα alias
    - Επιστρέφει πρώτο στοιχείο αλυσίδας για κλειδί alias
  - Certificate[] getCertificateChain(String alias)
    - Επιστρέφει αλυσίδα πιστοποιητικών για κλειδί alias
  - void deleteEntry(String alias)
    - Διαγράφει την καταχώριση με ψευδώνυμο alias

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Τέλος Ενότητας # 7

Μάθημα: Κατανεμημένα Συστήματα, **Ενότητα # 7:** Θέματα ασφάλειας

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής