Εξέταση 102 – Μάθημα 20

110.3 Ασφάλιση δεδομένων και κρυπτογράφηση

Η υπηρεσία *SSH* (Secure Shell)

- Παλαιότερα χρησιμοποιείτο η υπηρεσία **TELNET** για πρόσβαση σε κέλυφος άλλων δικτυακών κόμβων. Αυτή η υπηρεσία έχει σοβαρές αδυναμίες γιατί αποστέλλει όλες τις πληροφορίες σε καθαρό κείμενο χωρίς καμία κρυπτογράφηση. Αυτό αποτελεί μια μεγάλη τρύπα στην ασφάλεια των συστημάτων
- Η υπηρεσία **SSH** έχει αντικαταστήσει την TELNET στα σύγχρονα συστήματα γιατί παρέχει Κρυπτογράφηση Δημόσιου Κλειδιού (Public-Key Cryptography) και άρα ασφάλεια στην επικοινωνία. Υπάρχουν δύο εκδόσεις της: version 1 και version 2. Η πρώτη δεν θεωρείται αρκετά ασφαλής και συστήνεται η χρήση της δεύτερης
- Η εντολή **ssh** είναι ο πελάτης που συνδέεται στο δαίμονα **sshd** για πρόσβαση σε κέλυφος
- Υπάρχει και η εντολή **scp** για ασφαλή, δικτυακή μεταφορά αρχείων

Σύνδεση σε άλλους δικτυακούς κόμβους με *ssh*

- Η εντολή **ssh** είναι ο πελάτης του συστήματος **SSH** και χρησιμοποιείται για την σύνδεση σε άλλα συστήματα μέσω δικτύου

- **$ ssh user@example.com # = ssh -l user example.com**

The authenticity of host 'example.com (10.0.1.50)' can't be established.
RSA key fingerprint is 47:e2:fd:2d:62:b8:b4:37:66:b2:c2:d1:59:a5:ab:98.
Are you sure you want to continue connecting (yes/no)?

- Αν απαντήσετε «**yes**» στην πιο πάνω ερώτηση το ποιο πάνω δημόσιο κλειδί θα αποθηκευτεί μόνιμα στο αρχείο **~/.ssh/known_hosts** και δεν θα ερωτηθείτε ξανά. Μετά θα σας ζητήσει το όνομα χρήστη και το συνθηματικό για να ενωθείτε. Αν απαντήσετε «**no**» τότε η σύνδεση θα διακοπεί και δεν θα ενημερωθεί το αρχείο **~/.ssh/known_hosts**

Τα αρχεία ρυθμίσεως του ssh και sshd

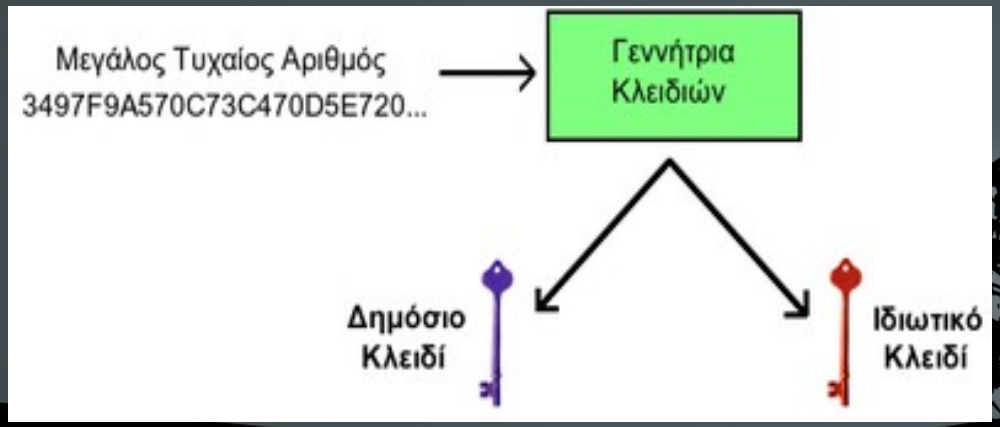
- **/etc/ssh/ssh-config** # αρχείο ρυθμίσεως πελάτη **ssh**
  Port 22 # η προκαθορισμένη θύρα του ssh
  Protocol 2 # σύνδεση σε διακομιστές με SSH version 2 μόνο
- **/etc/ssh/sshd-config** # αρχείο ρυθμίσεως δαίμονα **sshd**
  PermitRootLogin yes # ρυθμίστε το σε **no** για ασφάλεια
  PubkeyAuthentication yes # ενεργοποίηση πιστοποίησης δημόσιου κλειδιού
  Protocol 2 # σύνδεση πελατών με SSH version 2 μόνο
  X11Forwarding yes # Υποστηρίζει την εκτέλεση γραφικών εφαρμογών οι οποίες θα τρέχουν στο διακομιστή SSH αλλά τα παράθυρα τους θα φαίνονται στο διακομιστή X του πελάτη SSH. Απενεργοποιήστε το αν δεν χρειάζεται
- **/etc/ssh_known_hosts ή /etc/ssh/ssh_known_hosts** # καθολικό αρχείο για την εισαγωγή των γνωστών κόμβων με τα δημόσια κλειδιά τους

Ασφαλής αντιγραφή αρχείων με *scp*

- Η εντολή **scp** μας βοηθά στην ασφαλή αντιγραφή αρχείων από μια μηχανή σε άλλη
- **$ scp mydoc.odt user@example.com:** # αντιγραφή του αρχείου **mydoc.odt** στο προσωπικό κατάλογο του χρήστη **user** στο διακομιστή **example.com**. Το σύμβολο “**:**” είναι πολύ σημαντικό και χωρίς αυτό η **scp** συμπεριφέρεται απλά σαν την **cp**
- **$ scp mydoc.odt user@example.com:Documents** # αντιγραφή του αρχείου **mydoc.odt** στο κατάλογο **Documents** κάτω από τον προσωπικό κατάλογο του χρήστη **user**
- **$ scp user@example.com:Documents/mydoc.odt .** # αντιγραφή του αρχείου **mydoc.odt** από το κατάλογο **Documents** στο διακομιστή **example.com**, στο τρέχον κατάλογο του τοπικού συστήματος

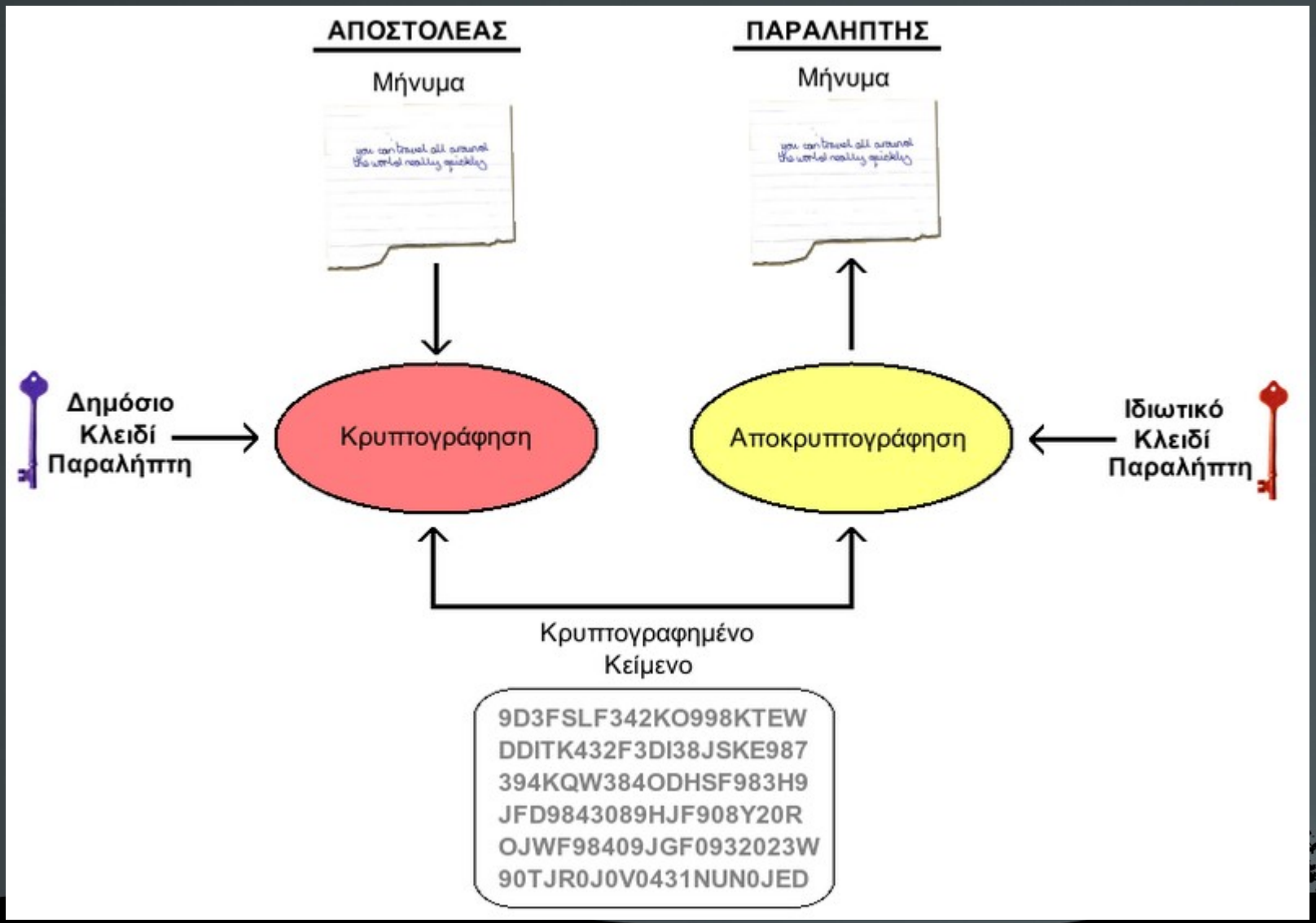
Κρυπτογράφηση Δημόσιου Κλειδιού

- Η Κρυπτογράφηση Δημόσιου Κλειδιού (Public-Key Cryptography) είναι μια τεχνική ασύμμετρης κρυπτογράφησης που χρησιμοποιείται από SSH, SSL, PGP, GPG κτλ
- Χρησιμοποιούνται οι αλγόριθμοι RSA, DSA και ECDSA
- Μια γεννήτρια κλειδιών παράγει δύο κλειδιά όπου η κρυπτογράφηση με το ένα αποκρυπτογραφείται με το άλλο. Το ένα είναι δημόσιο και μπορεί να το δει ο καθένας και το άλλο είναι ιδιωτικό και δεν πρέπει να διαρρεύσει σε μη έμπιστα άτομα



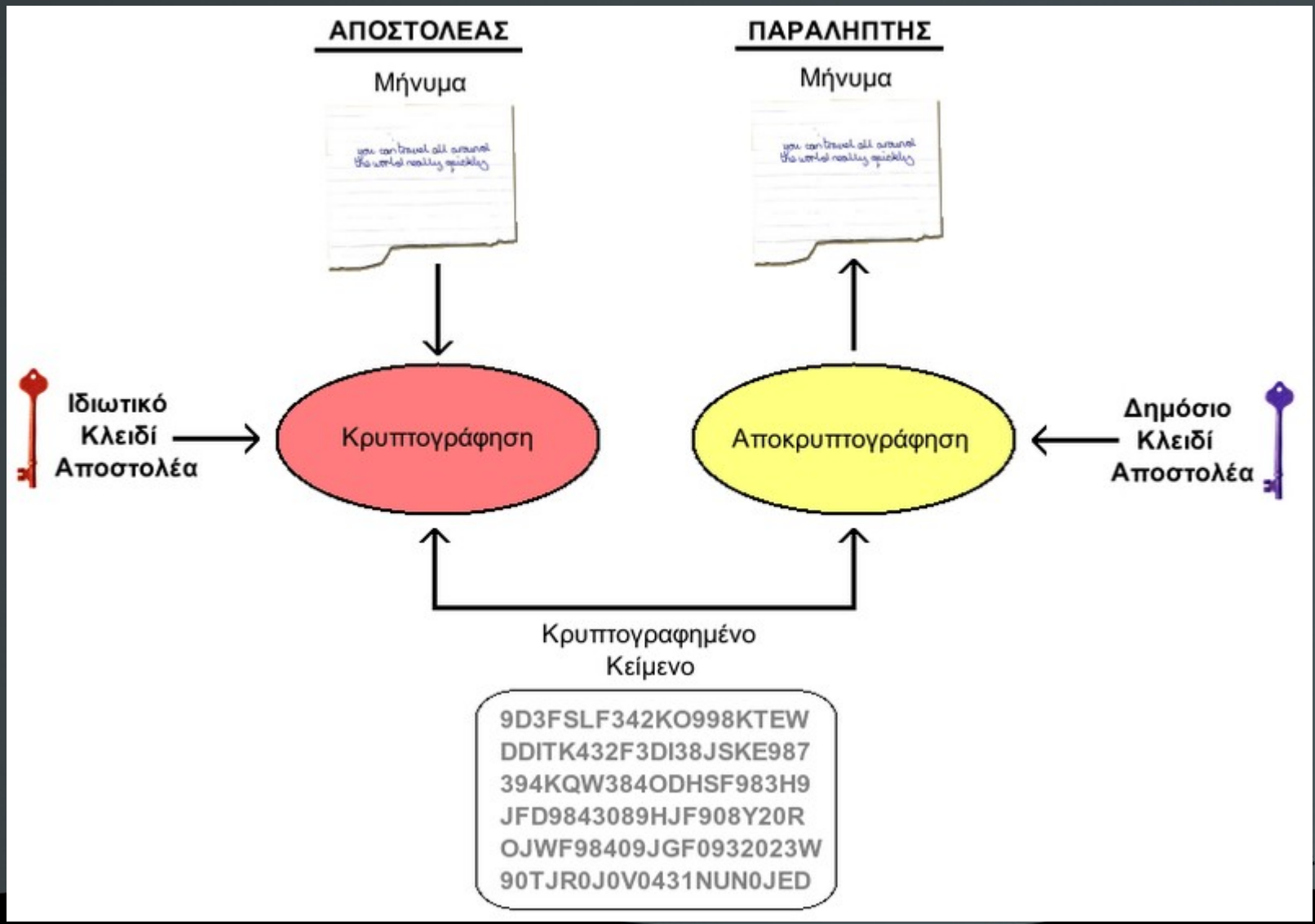
Κρυπτογράφηση Δημόσιου Κλειδιού

Επίτευξη εμπιστευτικότητας αλλά όχι πιστοποίησης



Κρυπτογράφηση Δημόσιου Κλειδιού

Επίτευξη πιστοποίησης αλλά όχι εμπιστευτικότητας



Δημόσια και Ιδιωτικά κλειδιά στο SSH

- Με την πρώτη εκτέλεση του δαίμονα **sshd** αυτός θα δημιουργήσει δημόσια και ιδιωτικά RSA και DSA κλειδιά. Ο προκαθορισμένος αλγόριθμος είναι ο RSA
- `# ls -la /etc/ssh/*key*`

```
-rw------- 1 root root  668 2011-08-28 13:40 /etc/ssh/ssh_host_dsa_key
-rw-r--r-- 1 root root  605 2011-08-28 13:40 /etc/ssh/ssh_host_dsa_key.pub
-rw------- 1 root root 1679 2011-08-28 13:40 /etc/ssh/ssh_host_rsa_key
-rw-r--r-- 1 root root  397 2011-08-28 13:40 /etc/ssh/ssh_host_rsa_key.pub
```

- Όταν συνδεόμαστε για πρώτη φορά σε ένα διακομιστή SSH αυτός μας δίνει το δημόσιο κλειδί του, για να κρυπτογραφήσουμε την επικοινωνία μαζί του.
- Αυτό αποθηκεύεται μόνιμα στο αρχείο **~/.ssh/known_hosts** και μετά επαναχρησιμοποιείται.
- Αν τα κλειδιά αυτά αλλάξουν, το σύστημα θα μας δώσει μια αυστηρή προειδοποίηση και θα αρνηθεί να συνδεθεί

Σύνδεση με χρήση πιστοποίησης δημόσιου κλειδιού

- Το SSH παρέχει διάφορους τρόπους πιστοποίησης (authentication). Εκτός από την παραδοσιακή πιστοποίηση με όνομα χρήστη και συνθηματικό υπάρχει και η πιστοποίηση δημόσιου κλειδιού
- **yioryos@local:~$ whoami** # ο τοπικός χρήστης είναι ο **yioryos**
  yioryos
- **yioryos@local:~$ ssh-keygen -t rsa -b 2048** # δημιουργία ζεύγους κλειδιών

Generating public/private rsa key pair.
Enter file in which to save the key (/home/yioryos/.ssh/id_rsa):
Created directory '/home/yioryos/.ssh'.
Enter passphrase (empty for no passphrase): # χωρίς φράση κλειδί!
Enter same passphrase again: # χωρίς φράση κλειδί!
Your identification has been saved in /home/yioryos/.ssh/id_rsa.
Your public key has been saved in /home/yioryos/.ssh/id_rsa.pub.
The key fingerprint is:
83:40:c4:05:39:d8:58:c0:ed:d4:a0:40:6d:87:6c:a4 yioryos@local

Σύνδεση με χρήση πιστοποίησης δημόσιου κλειδιού

- **yioryos@local:~$ ls -l .ssh/** # ιδιωτικό και δημόσιο κλειδί

-rw------- 1 yioryos yioryos 1679 Apr 4 22:35 id_rsa

-rw-r--r-- 1 yioryos yioryos 400 Apr 4 22:35 id_rsa.pub

- **yioryos@local:~$ cat .ssh/id_rsa.pub | ssh user@remote.dom **
  "xargs -I {} echo {} >> .ssh/authorized_keys" # επισύναψη δημόσιου κλειδιού χρήστη **yioryos@local.dom** στο αρχείο **.ssh/authorized_keys** (σε κάποια συστήματα είναι **authorized_keys2**) του χρήστη **user@remote.dom**

user@remote.dom's password: # συνθηματικό χρήστη **user**

- **ssh user@remote.dom** # είσοδος χωρίς κωδικό!

Last login: Wed Apr 4 22:58:35 2012 from local.dom

user@remote:~$

- **user@remote:~$ whoami**

  user

Σύνδεση με χρήση πιστοποίησης δημόσιου κλειδιού

- **user@remote:~$ grep yioryos .ssh/authorized_keys**
- **ssh-rsa**
  AAAAB3NzaC1yc2EAAAADAQABAAABAQDBAfRMHpzJ6
  NfnOCcBOjE5Xxip03eHbIIDGnrpmyc8fWjGqwN3mZZ3HzJ
  2fNJZJA6rdMEtlcGt1MgcPcLUqLx93jZr/ZL3Me3d9e9Jretiv
  jcicFV4gU/2m3pQHy1aKvyioGqytmtUKwEZzJzC+nZNK/Fd
  +glMUu6q8Py3QFspPBb1NEw7cKgYKn5kOcV3Th4KAvYwz
  o+VlHuHIS0MlffGDxId4m7C+DqMX1utdUJ7reYAGNWFlS
  Ah7ajqVNDtOGAQC743BOBuyTdPrnLtFc1A45mR2l2P9PU
  4iqYhiYpKLOxqK6oeQIcomq/KcCz1+HPoYWPDq2EB5CW
  AmaKLNQcb yioryos@local
- Η πιο πάνω μέθοδος είναι χρήσιμη για την εκτέλεση σεναρίων από ένα διακομιστή σε άλλο χωρίς την χρήση συνθηματικού (πχ backup), σε συστοιχίες διακομιστών ή απλά για ευκολία

Η εντολή *ssh-keygen*

- Η εντολή **ssh-keygen** χρησιμοποιείται για την δημιουργία ζεύγους ιδιωτικών/δημόσιων κλειδιών τόσο για τον κάθε χρήστη ξεχωριστά όσο και για τον δαίμονα **sshd**
- **$ ssh-keygen -l -f .ssh/id_rsa** # προβολή πληροφοριών για κλειδιά
  2048 a5:c6:87:06:ea:d6:09:f6:4d:a9:25:31:e4:a0:fb:df
  .ssh/id_rsa.pub (RSA)

Επιλογές:

- **-b** # αριθμός μπιτ. Προκαθορισμένο 1024, συστήνεται 2048 και άνω
- **-p** # αλλαγή φράσης κλειδιού (passphrase)
- **-f file** # καθορισμός αρχείου εξόδου
- **-C 'some comments'** # καθορισμός σχολίων για κλειδί
- **-N 'somelongpassphrase'** # καθορισμός φράσεως κλειδιού

Περισσότερη ασφάλεια με *ssh-agent*

- Το πρόβλημα με την μέθοδο που είδαμε προηγουμένως είναι ότι αν κάποιος παραβιάσει τον λογαριασμό σας ή καταλάβει το κέλυφος σας θα μπορεί να μπει σε άλλους διακομιστές χωρίς κωδικό!
- Το **ssh-agent** μας δίνει την δυνατότητα να δίνουμε μια φορά την φράση κλειδί που υπάρχει στα κλειδιά μας και μετά να συνδεόμαστε όσες φορές θέλουμε
- Για να γίνει αυτό θα πρέπει να δηλώσουμε μια φράση κλειδί, η οποία συστήνεται να είναι αρκετά μεγάλη, κατά την δημιουργία του ζεύγους κλειδιών μας με **ssh-keygen**

Περισσότερη ασφάλεια με *ssh-agent*

- **$ ssh-keygen -t rsa -b 2048** # δημιουργία ζεύγους κλειδιών

Generating public/private rsa key pair.
Enter file in which to save the key (/home/yioryos/.ssh/id_rsa):
Created directory '/home/yioryos/.ssh'.
Enter passphrase (empty for no passphrase): I**taneMiaForaMatiaMoy**
Enter same passphrase again: I**taneMiaForaMatiaMoy**
Your identification has been saved in /home/yioryos/.ssh/id_rsa.
Your public key has been saved in /home/yioryos/.ssh/id_rsa.pub.
The key fingerprint is:
83:40:c4:05:39:d8:58:c0:ed:d4:a0:40:6d:87:6c:a4 yioryos@mypc

- **$ ssh-agent /bin/bash** # ενεργοποίηση **ssh-agent** σε καινούργιο κέλυφος. Για να το ενεργοποιήσετε στο τρέχον κέλυφος δοκιμάστε την εντολή '**eval ssh-agent**'
- **$ ssh-add ~/.ssh/id_rsa** # προσθήκη κλειδιού σε **ssh-agent**. Αν τρέξετε απλά την '**ssh-add**' θα προστεθούν όλα τα κλειδιά. Αυτό χρειάζεται να γίνει μόνο την πρώτη φορά

SSH Port channels (Tunneling)

- **$ ssh -X user@10.0.1.50** # δυνατότητα εκτέλεσης γραφικών εφαρμογών από διακομιστή 10.0.1.50 με παράθυρα στο τοπικό διακομιστή X (X11Forwarding yes).
- **$ ssh -N -f -L 2525:smtp.example.com:25 bob@gate.example.com** # Προώθηση της τοπικής θύρας **2525** στη απομακρυσμένη θύρα **25** του διακομιστή **smtp.example.com** μέσω του ενδιάμεσου διακομιστή **gate.example.com**
- **$ telnet localhost 2525** # θα με οδηγήσει στην ουσία στη θύρα **25** του διακομιστή **smtp.example.com**
- **$ ssh -L 3306:localhost:3306 bob@mysql.example.com** # σύνδεση της τοπικής θύρας 3306 με την θύρα 3306 στο διακομιστή **mysql.example.com**
- Η δυνατότητα αυτή μπορεί να απενεργοποιηθεί με την επιλογή **AllowTcpForwarding no**

Επιλογές της ssh

- **-l** # καθορισμός ονόματος χρήστη
- **-X** # δυνατότητα εκτέλεσης γραφικών προγραμμάτων στο τοπικό διακομιστή X
- **-L** # σύνδεση τοπικής θύρας με απομακρυσμένη
- **-R** # σύνδεση απομακρυσμένης θύρας με τοπική
- **-N** # Να μην εκτελεστεί απομακρυσμένη εντολή (πχ bash)
- **-f** # αποστολή διεργασίας ssh στο παρασκήνιο
- **-v** # αναλυτική προβολή (χρήσιμο για επίλυση προβλημάτων)

Εργαλείο κρυπτογράφησης και πιστοποίησης *GPG*

- Το εργαλείο **GPG** (GNU Privacy Guard) είναι ένα εργαλείο κρυπτογράφησης και πιστοποίησης αρχείων και μηνυμάτων ηλεκτρονικής αλληλογραφίας.
- Χρησιμοποιεί κυρίως Κρυπτογραφία Δημόσιου Κλειδιού και σχεδιάστηκε σαν εναλλακτική λύση στο PGP (Pretty Good Privacy)
- Μπορεί να χρησιμοποιηθεί αυτόνομα αλλά και από άλλες εφαρμογές όπως πελάτες ηλεκτρονικής αλληλογραφίας

Εργαλείο κρυπτογράφησης και πιστοποίησης *GPG*

- **$ gpg –gen-key** # παραγωγή ζεύγους κλειδιών **GPG**

Please select what kind of key you want:
(1) RSA and RSA (default)
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
Your selection? **4**
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (1024) **2048**
Requested keysize is 2048 bits
Please specify how long the key should be valid.
0 = key does not expire
<n> = key expires in n days
<n>w = key expires in n weeks
<n>m = key expires in n months
<n>y = key expires in n years
Key is valid for? (0) **5y**
Key expires at Tue 04 Apr 2017 12:52:36 AM EEST
Is this correct? (y/N) **y**

Εργαλείο κρυπτογράφησης και πιστοποίησης *GPG*

You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: **Bob Crypt**
Email address: **bob.crypt@example.com**
Comment: **Bob the one**
You selected this USER-ID:
    "Bob Crypt (Bob the one) <bob.crypt@example.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? **o**
You need a Passphrase to protect your secret key.

gpg: gpg-agent is not available in this session

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.

Εργαλείο κρυπτογράφησης και πιστοποίησης *GPG*

...+++++
....+++++
gpg: /home/yioryos/.gnupg/trustdb.gpg: trustdb created
gpg: key 1C877AA9 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2017-04-03
pub   2048R/1C877AA9 2012-04-04 [expires: 2017-04-03]
      Key fingerprint = 537D E04B 6852 4F7E 5880 AFAC E49A 1815 1C87 7AA9
uid   Bob Crypt (Bob the one) <bob.crypt@example.com>

Note that this key cannot be used for encryption. You may want to use
the command "--edit-key" to generate a subkey for this purpose.

Εργαλείο κρυπτογράφησης και πιστοποίησης *GPG*

- **$ ls -l .gnupg**

total 32

-rw------- 1 yioryos yioryos 9398 2012-04-05 00:46 gpg.conf

-rw------- 1 yioryos yioryos 646 2012-04-05 00:55 pubring.gpg

-rw------- 1 yioryos yioryos 646 2012-04-05 00:55 pubring.gpg~

-rw------- 1 yioryos yioryos 600 2012-04-05 00:55 random_seed

-rw------- 1 yioryos yioryos 1335 2012-04-05 00:55 secring.gpg

-rw------- 1 yioryos yioryos 1280 2012-04-05 00:55 trustdb.gpg

Άλλες λειτουργίες της gpg

- **$ gpg --import user_test_example.asc** # εισαγωγή δημόσιου κλειδιού άλλου χρήστη από αρχείο **.asc**
- **$ gpg --edit-key "User.test"** # υπογραφή (sign) εισηγμένου κλειδιού με το δικό μας κλειδί. Θα πρέπει να ξέρουμε το όνομα χρήστη (User.test) που σχετίζεται με αυτό
- **$ gpg –list-keys** # λίστα προσωπικών και εισηγμένων κλειδιών
- **$ gpg --export my_gpg_key_backup** # εξαγωγή κλειδιών για φύλαξη
- **$ gpg -e -u "Bob Crypt" -r "User Test" mydoc.odt** # κρυπτογράφηση αρχείου ώστε μόνο ο **User Test** να μπορεί να το ανοίξει
- **$ gpg -d mydoc.odt** # αποκρυπτογράφηση αρχείου από **User Test**