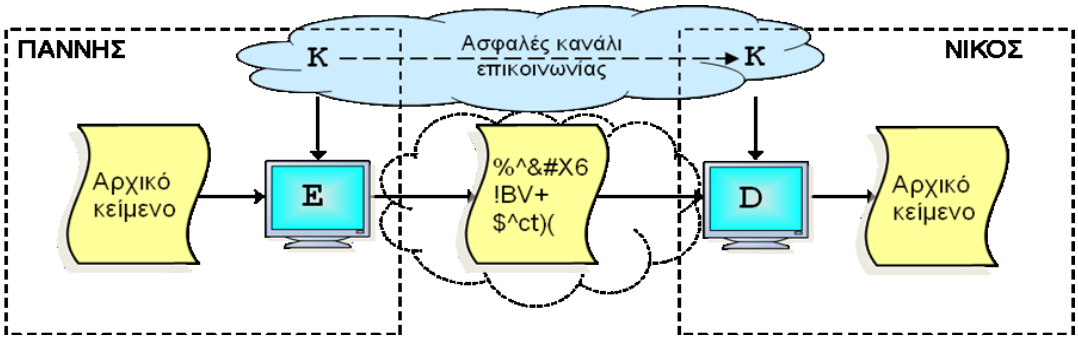
Προστασία και Ασφάλεια Δικτύων (Εργαστήριο)

Ενότητα 2

OpenSSL

Η κρυπτογραφία είναι μια επιστήμη που ασχολείται με τη μελέτη μεθόδων και μηχανισμών που σχετίζονται με την ασφάλεια πληροφοριών. Οι αλγόριθμοι κρυπτογράφησης που έχουν αναπτυχθεί χωρίζονται σε δύο κατηγορίες:

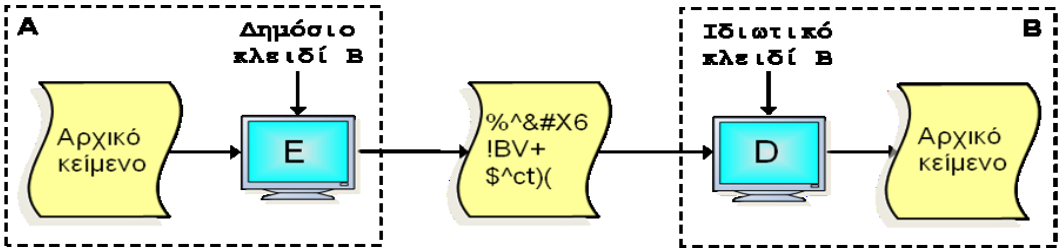
- **Συμμετρικής κρυπτογραφίας**: είναι οι αλγόριθμοι όπου το κλειδί της αποκρυπτογράφησης είναι πολύ εύκολο να υπολογιστεί εάν γνωρίζουμε το κλειδί της κρυπτογράφησης (συνήθως είναι το ίδιο κλειδί). Χρησιμοποιούνται κυρίως για την εξασφάλιση της εμπιστευτικότητας καθώς και για τη δημιουργία άλλων μηχανισμών προστασίας των δεδομένων, όπως είναι οι συναρτήσεις κατακερματισμού.



Σχήμα 1: Συμμετρική κρυπτογραφία

- **Ασύμμετρης κρυπτογραφίας (γνωστή και ως δημοσίου κλειδιού)**: ο κάθε χρήστης έχει ένα ζεύγος κλειδιών: το ιδιωτικό, το οποίο το γνωρίζει μόνο ο ίδιος, και το δημόσιο, το οποίο μπορεί να διαθέσει σε όλους τους ενδιαφερόμενους. Τα δύο κλειδιά συνδέονται με μαθηματικό τρόπο έτσι ώστε ένα μήνυμα κρυπτογραφημένο με το ένα κλειδί του ζεύγους να μπορεί να αποκρυπτογραφηθεί μόνο με το άλλο κλειδί του ίδιου ζεύγους. Τυπικά η κρυπτογράφηση ενός μηνύματος γίνεται με το δημόσιο κλειδί του παραλήπτη ενώ ο παραλήπτης για να αποκρυπτογραφήσει το μήνυμα θα χρησιμοποιήσει το ιδιωτικό του κλειδί.

  Η κρυπτογραφία δημοσίου κλειδιού χρησιμοποιείται κυρίως για την κρυπτογράφηση πολύ μικρών μηνυμάτων, όπως είναι ένα κλειδί συμμετρικής κρυπτογράφησης, αλλά και για τις ψηφιακές υπογραφές, με τις οποίες παρέχεται αυθεντικοποίηση, ακεραιότητα και μη αποποίηση αποστολής των δεδομένων.



Σχήμα 2: Κρυπτογραφία δημοσίου κλειδιού

OpenSSL

Το OpenSSL είναι μια συλλογική προσπάθεια που έχει ως αντικείμενο την ανάπτυξη μιας αξιόπιστης, ακόμη και για εμπορικούς σκοπούς, λύσης που περιλαμβάνει την υλοποίηση των πρωτοκόλλων ασφαλείας Secure Sockets Layer (SSL) και Transport Layer Security (TLS). Τα πρωτόκολλα αυτά

χρησιμοποιούνται για την ασφαλή διακίνηση δεδομένων μέσω ανασφαλών καναλιών επικοινωνίας όπως αυτών του διαδικτύου (Internet). Το OpenSSL περιλαμβάνει και μια ανοιχτή βιβλιοθήκη αλγορίθμων κρυπτογράφησης, γενικής χρήσης, η οποία αποτελεί αντικείμενο αυτού του εργαστηρίου.

Το OpenSSL υλοποιεί σχεδόν όλους τους γνωστούς αλγορίθμους κρυπτογράφησης, τόσο συμμετρικής όσο και ασύμμετρης κρυπτογραφίας, μαζί με τις πιο ασφαλείς μεθόδους χρήσης αυτών, καθώς και μηχανισμούς που σχετίζονται άμεσα με την προστασία δεδομένων όπως είναι οι συναρτήσεις κατακερματισμού. Ενδεικτικά, το OpenSSL περιλαμβάνει τους παρακάτω αλγορίθμους συμμετρικής και ασύμμετρης κρυπτογραφίας.

Συμμετρική κρυπτογραφία: DES, AES, IDEA, RC4, RC5, Blowfish.

Ασύμμετρη κρυπτογραφία: RSA

Ο ευκολότερος ίσως τρόπος για να δούμε τις διαθέσιμες εντολές του OpenSSL είναι να δώσουμε μια λάθος παράμετρο, όπως για παράδειγμα να καλέσουμε την `openssl help`:

```
root@opensolaris:~# openssl help
openssl:Error: 'help' is an invalid command.

Standard commands
asn1parse      ca             ciphers        crl            crl2pkcs7
dgst           dh             dhparam        dsa            dsaparam
enc            engine         errstr         gendh          gendsa
genrsa         nseq           ocsp           passwd         pkcs12
pkcs7          pkcs8          prime          rand           req
rsa            rsautl         s_client       s_server       s_time
sess_id        smime          speed          spkac          verify
version        x509

Message Digest commands (see the `dgst' command for more details)
md2            md4            md5            rmd160         sha
sha1

Cipher commands (see the `enc' command for more details)
aes-128-cbc    aes-128-ecb    aes-192-cbc    aes-192-ecb    aes-256-cbc
aes-256-ecb    base64         bf             bf-cbc         bf-cfb
bf-ecb         bf-ofb         cast           cast-cbc       cast5-cbc
cast5-cfb      cast5-ecb      cast5-ofb      des            des-cbc
des-cfb        des-ecb        des-ede        des-ede-cbc    des-ede-cfb
des-ede-ofb    des-ede3       des-ede3-cbc   des-ede3-cfb   des-ede3-ofb
des-ofb        des3           desx           rc2            rc2-40-cbc
rc2-64-cbc     rc2-cbc        rc2-cfb        rc2-ecb        rc2-ofb
rc4            rc4-40
```

Ωστόσο ο ορθός τρόπος για την προβολή των Standard commands, Message Digest commands και Cipher commands είναι μέσω της κλήσης των αντίστοιχων ψευδοεντολών:

```
#Προβολή των standard commands
$ openssl list-standard-commands
#Προβολή των message digest commands
$ openssl list-message-digest-commands
```

και

```
#Προβολή των εντολών κρυπτογράφησης
$ openssl list-cipher-commands
```

Τις διαθέσιμες εντολές μπορούμε να τις καλέσουμε μέσα από το κέλυφος (shell) στο οποίο βρισκόμαστε καλώντας πρώτα την εντολή `openssl` ακολουθούμενη από κάποια από τις standard commands π.χ. `openssl enc` ή αφού μπούμε σε περιβάλλον openssl καλώντας την εντολή `openssl` και μετά την εντολή που θέλουμε, π.χ.

```
$ openssl
OpenSSL> enc
```

Μια ενδιαφέρουσα υπηρεσία που παρέχει η εφαρμογή είναι αυτή της μέτρησης του χρόνου εκτέλεσης των διαφόρων κρυπτογραφικών λειτουργιών με τη χρήση της εντολής `speed`. Μπορούμε να ελέγξουμε τους χρόνους για όλους τους αλγορίθμους, ή για συγκεκριμένο αλγόριθμο με τη χρήση του ονόματος του, π.χ. `speed sha1` (για να ελέγξουμε το χρόνο εκτέλεσης της συνάρτησης κατακερματισμού `sha1`).

Συμμετρική κρυπτογράφηση

Το OpenSSL παρέχει τη δυνατότητα κρυπτογράφησης κειμένων και αρχείων χρησιμοποιώντας μια πληθώρα από αλγορίθμους συμμετρικής κρυπτογράφησης και τρόπων κρυπτογράφησης (οι οποίοι προβάλλονται με την εντολή: `openssl list-cipher-commands`) με την εντολή

```
OpenSSL> enc –ciphername
```

(μέσα από το περιβάλλον του OpenSSL: για να μπούμε στο περιβάλλον openssl εκτελούμε την εντολή `openssl`, ενώ για να βγούμε από το περιβάλλον πληκτρολογούμε `exit`)

ή με την εντολή:

```
$ openssl enc -ciphername
```

ή ακόμη και με την εντολή

```
 $ openssl ciphername
```

Οι παράμετροι της εντολής σχετίζονται με τον αλγόριθμο τον οποίο θα επιλέξει ο χρήστης για να κρυπτογραφήσει το αρχείο σε συνδυασμό με τον τρόπο χρήσης αυτού (π.χ. ECB ή CBC).

Η κρυπτογράφηση αρχείου (το οποίο θα πρέπει να δημιουργήσετε) με το όνομα test.txt χρησιμοποιώντας τον αλγόριθμο AES με κλειδί 256-bit σε CBC mode και την αποθήκευση του αποτελέσματος στο αρχείο output.enc γίνεται με την εντολή:

```
$ openssl enc -aes-256-cbc –a -salt -in test.txt -out output.enc
```

Το ίδιο αποτέλεσα επιτυγχάνεται, όπως προαναφέρθηκε, με την κλήση της εντολής:

```
$ openssl aes-256-cbc –a -salt -in test.txt -out output.enc
```

Η παράμετρος **–a** χρησιμοποιείται για την κωδικοποίηση της εξόδου σε base64 η οποία συνιστάται αν θέλουμε να στείλουμε το κρυπτογραφημένο αρχείο με email. Αλλιώς η μορφή του θα είναι binary.

Η αποκρυπτογράφηση γίνεται με τον ίδιο αλγόριθμο με τον οποίο έγινε η κρυπτογράφηση και με τη χρήση της παραμέτρου `–d`. Π.χ.

```
$ openssl enc -d -aes-256-cbc –a -in output.enc -out test1.txt ή
$ openssl aes-256-cbc –d –a -in output.enc -out test1.txt
```

Μετά την αποκρυπτογράφηση ελέγχουμε εάν το αρχείο `test1.txt` είναι ίδιο με το αρχικό αρχείο `test.txt`.

Συναρτήσεις σύνοψης ή κατακερματισμού (hash functions)

Το OpenSSL παρέχει τη δυνατότητα δημιουργίας σύνοψης μηνύματος με τη χρήση της εντολής `dgst` η οποία δέχεται ως παραμέτρους το όνομα της hash function που θέλουμε να χρησιμοποιήσουμε καορηορει το όνομα του αρχείου του οποίου τη σύνοψη θέλουμε να υπολογίσουμε. Για να δούμε τη λίστα με τις διαθέσιμες συναρτήσεις κατακερματισμού χρησιμοποιούμε την επιλογή `list-message-digest-commands`:

```
$ openssl list-message-digest-commands
```

Για τον υπολογισμό της σύνοψης του αρχείου test.txt με τη χρήση της συνάρτησης md5 μέσα από το περιβάλλον της εφαρμογής openssl πληκτρολογούμε `dgst –md5 test.txt`. Το αποτέλεσμα θα πρέπει να είναι το ίδιο με αυτό που θα πάρουμε αν υπολογίσουμε τη σύνοψη με τη χρήση της

συνάρτησης `md5sum` η οποία υπολογίζει τη σύνοψη ενός αρχείου αποκλειστικά με τη χρήση της συνάρτησης md5. (Προσοχή: η εντολή md5sum εκτελείται εκτός περιβάλλοντος OpenSSL). Π.χ.

```
$ openssl dgst -md5 test.txt
```

Κρυπτογραφία δημοσίου κλειδιού

Η κρυπτογραφία δημοσίου κλειδιού όπως προαναφέρθηκε μπορεί να χρησιμοποιηθεί για την κρυπτογράφηση πολύ μικρών μηνυμάτων ή για την δημιουργία και επαλήθευση υπογραφών. Ο αλγόριθμος που θα χρησιμοποιήσουμε για αυτό το σκοπό στο πλαίσιο αυτού του εργαστηρίου είναι ο RSA. Για να μπορέσουμε να χρησιμοποιήσουμε τις συναρτήσεις που σχετίζονται με αλγόριθμο RSA, θα πρέπει πρώτα να δημιουργήσουμε ένα ζεύγος κλειδιών, με την εντολή genrsa.

```
# Δημιουργία κλειδιού 1024-bit. Το αποθηκεύουμε στο αρχείο
# privkey.pem
$ openssl genrsa -out privkey.pem 1024
# Εάν θέλουμε το κλειδί να προστατεύεται με passphrase θα πρέπει
# να χρησιμοποιήσουμε και την αντίστοιχη παράμετρο. Το νέο
# κρυπτογραφημένο κλειδί αποθηκεύεται στο αρχείο privkeyenc.pem
$ openssl genrsa -des3 -out privkeyenc.pem 1024
```

Για να δούμε σε μορφή κειμένου τα περιεχόμενα του αρχείου του κλειδιού που έχουμε δημιουργήσει, χρησιμοποιούμε την εντολή rsa.

```
# Προβολή περιεχομένων του αρχείου κλειδιού privkey.pem
$ openssl rsa -in privkey.pem –text -noout
```

Προκειμένου να μπορέσουμε να χρησιμοποιήσουμε το δημόσιο κλειδί (π.χ. να το δώσουμε σε κάποιο ν τρίτο για να μπορέσει να επαληθεύσει την υπογραφή σε ένα αρχείο) θα πρέπει πρώτα να το εξάγουμε και να το αποθηκεύσουμε σε ένα ξεχωριστό αρχείο.

```
# Αποθήκευση του δημοσίου κλειδιού στο αρχείο pubkey.pem
$ openssl rsa -in privkey.pem –pubout –out pubkey.pem
```

Δημιουργία και επαλήθευση υπογραφής

Το OpenSSL παρέχει στο χρήστη τη δυνατότητα της υπογραφής μιας σύνοψης ενός μηνύματος και της επαλήθευσης αυτής με τη χρήση του ιδιωτικού και δημόσιου κλειδιού του χρήστη αντίστοιχα.

```
# Υπογραφή της SHA1 σύνοψης του αρχείου test.txt και αποθήκευση
# στο αρχείο test.txt.sign
$ openssl dgst –sha1 –sign privkey.pem –out test.txt.sign test.txt
```

Για την επαλήθευση της υπογραφής ο χρήστης χρειάζεται το αρχικό αρχείο για το οποίο δημιουργήθηκε η υπογραφή μαζί με το αρχείο που περιλαμβάνει την υπογραφή.

```
# Επαλήθευση της υπογραφής που βρίσκεται στο αρχείο test.txt.sign
# και έχει δημιουργηθεί για το αρχείο test.txt με τη χρήση του
# κλειδιού που βρίσκεται στο αρχείο pubkey.pem
$ openssl dgst –sha1 –verify pubkey.pem –signature test.txt.sign
test.txt
```

Άσκηση

Τροποποιήστε το αρχείο που υπογράψατε αρχικά αφού πρώτα κάνετε μια μικρή αλλαγή.

1. Ξαναυπογράψτε το, αποθηκεύοντας την υπογραφή σε ένα νέο αρχείο, και συγκρίνετε την αρχική υπογραφή με τη νέα.
2. Προσπαθήστε να επαληθεύσετε την αρχική υπογραφή στο νέο αρχείο και δείτε το αποτέλεσμα.