ΠΡΟΣΤΑΣΙΑ ΚΑΙ ΑΣΦΑΛΕΙΑ ΣΥΣΤΗΜΑΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ (Εργαστήριο)

Ενότητα 6
OpenSSL

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

Τις διαθέσιμες εντολές τις καλούμε μέσα από το περιβάλλον openssl, στο οποίο μπαίνουμε καλώντας την εντολή `openssl`. Στην συνέχεις πληκτρολογούμε την εντολή που θέλουμε, π.χ.

```
OpenSSL> enc
```

Για να δούμε τις διαθέσιμες εντολές του OpenSSL χρησιμοποιούμε την εντολή `help` μέσα από το περιβάλλον του OpenSSL.

Για να δούμε τους υποστηριζόμενους αλγορίθμους και εντολές για κάθε μία κατηγορία αλγορίθμων: digest, cipher, και public-key χρησιμοποιούμε την εντολή `list`. Τις παραμέτρους που υποστηρίζει η εντολή τις βλέπουμε με την παράμετρο `-help`. Π.χ.

```
OpenSSL> list -digest-commands
```

Χρόνος εκτέλεσης κρυπτογραφικών λειτουργιών

Μια ενδιαφέρουσα υπηρεσία που παρέχει η σουίτα OpenSSL 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`).

```
OpenSSL> speed aes-128-cbc
```

Άσκηση

Συγκρίνετε τις ταχύτητες των αλγορίθμων des-ede-cbc και aes-128-cbc που χρησιμοποιούν παρόμοιου μεγέθους κλειδιά.

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: `enc -help`) με την εντολή

```
OpenSSL> enc –«ciphername»
```

όπου «ciphername» το όνομα του αλγορίθμου που θέλουμε να χρησιμοποιήσουμε ή ακόμη και χρησιμοποιώντας απ'ευθείας το όνομα του αλγορίθμου ως 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`.

Άσκηση

Κρυπτογραφήστε κείμενο με αλγόριθμο της επιλογής σας, διαφορετικό του παραδείγματος, και στείλτε το με email ώστε ο παραλήπτης να το αποκρυπτογραφήσει.

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 και το

όνομα του αρχείου του οποίου τη σύνοψη θέλουμε να υπολογίσουμε. Για να δούμε τη λίστα με τις διαθέσιμες συναρτήσεις κατακερματισμού χρησιμοποιούμε την επιλογή `-help`:

```
OpenSSL> dgst -help
```

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
```

Άσκηση

Χρησιμοποιώντας τη συνάρτηση sha256 υπολογίστε τη σύνοψη κειμένου που υπάρχει σε αρχείο και αφού αλλάξτε κατά ένα γράμμα το κείμενο στο αρχείο επανυπολογίστε τη σύνοψη. Συγκρίνετε τα δύο αποτελέσματα. Τι παρατηρείτε;

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 κάποιον 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
```

Άσκηση

Δημιουργείστε ζεύγος κλειδιών RSA μεγέθους 2048bits τα οποία θα προστατεύετε με κωδικό.

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.

Δημιουργία κλειδιού συμμετρικής κρυπτογράφησης

Η δημιουργία ενός ισχυρού κλειδιού συμμετρικής κρυπτογραφησης μπορεί να γίνει με τη χρήση μιας γεννήτριας ψευδοτυχαίων αριθμών. Στο παρακάτω παράδειγμα το κλειδί που δημιουργείται είναι μεγέθους 256bits.

```
OpenSSL> rand -base64 32 –out key.bin
```

Κρυπτογράφηση με τη χρήση δημοσίου κλειδιού

Η κρυπτογραφία δημοσίου κλειδιού δε χρησιμοποιείται για την κρυπτογράφηση μεγάλου όγκου δεδομένων. Ωστόσο χρησιμοποιείται εκτενώς για την κρυπτογράφηση δεδομένων περιορισμένου όγκου, όπως είναι τα κλειδιά της συμμετρικής κρυπτογράφησης. Στο ακόλουθο παράδειγμα χρησιμοποιούμε το δημόσιο κλειδί που δημιουργήσαμε προηγουμένως για να κρυπτογραφήσουμε το κλειδί που αποθηκεύσαμε στο key.bin.

```
OpenSSL> rsautl –encrypt -inkey pubkey.pem -pubin -in key.bin -out
keybin.enc
```

Άσκηση

Ως συνέχεια των προηγούμενων ενεργειών, βρείτε και καταγράψτε τις εντολές ώστε 1.
Να κρυπτογραφήσετε ένα μεγάλο αρχείο
2. Να αποκρυπτογραφήσετε το αρχείο.