

# Criptografie Simetrica. Functii Hash

## Tema 1

Asist. Drd. Ioana Leahu

November 22, 2016

# Instalare OpenSSL

Pentru a instala biblioteca OpenSSL pe mașina virtuală Ubuntu (Lubuntu) trebuie efectuați următorii pași:

```
wget https://www.openssl.org/source/openssl-1.0.2j.tar.gz
tar -xzf openssl-1.0.2j.tar.gz

(alternativ: se descarcă din browser arhiva openssl-1.0.2j.tar.gz și se
dezarhivează utilizând utilitarele instalate în mașina virtuală)

cd openssl-1.0.2j (se schimbă directorul curent în directorul în care s-a
efectuat dezarhivarea)
se citește conținutul fișierului INSTALL din directorul openssl-1.0.2j

se execută următoarele comenzi:
./config
make
make test
sudo make install
```

Exemplu fișier Makefile pentru compilarea unui program C ce utilizează funcții din API EVP:

```
INC=/usr/local/ssl/include/
LIB=/usr/local/ssl/lib/

all:
    gcc -I$(INC) -L$(LIB) -o out source.c -lcrypto -ldl
```

Observații:

- un exemplu de mașină virtuală Lubuntu14.04 având instalate biblioteca

OpenSSL și editorul Bless poate fi descărcată de la adresa: [http://profs.info.uaic.ro/~olgai/si2016/lubuntu1404\\_baza.ova](http://profs.info.uaic.ro/~olgai/si2016/lubuntu1404_baza.ova);

- la adresa: [http://profs.info.uaic.ro/~olgai/si2016/word\\_dict.txt](http://profs.info.uaic.ro/~olgai/si2016/word_dict.txt) se găsește un fișier text conținând cuvinte de dicționar englez, util pentru Exercițiul 1;
- detalii privind programarea C utilizând funcțiile din API EVP: <https://www.openssl.org/docs/man1.0.2/crypto/crypto.html>, sau <https://wiki.openssl.org/index.php/EVP>.

# Exercițiul 1

## Biblioteca OpenSSL - Criptare/Decriptare

Scrieți un program C/C++ ce utilizează API EVP pentru operații de criptare și decriptare.

Programul va primi ca date de intrare numele a două fișiere, *pt*, respectiv *ct* și un mod de operare (*ECB* sau *OFB*). Fișierul *ct* este rezultatul operației de criptare cu o cheie *K* (și un vector de inițializare dat

```
iv = "\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f"
```

pentru modul *OFB*) a fișierului *pt*.

Scopul programului este de a găsi cheia *K* utilizată pentru criptarea fișierului *pt*, știind că:

- cheia *K* are o lungime fixată de 128 biți (16 octeți);
- cheia *K* este reprezentarea în hexa a unui cuvânt din dicționarul englez cu lungime mai mică de 16 caractere, completat cu caracterul spațiu (cod hexa `\x20`) până la lungimea de 16 caractere;  
de exemplu, pentru cuvântul *median*, cheia

```
K = "\x6D\x65\x64\x69\x61\x6E\x20\x20\x20\x20\x20\x20\x20\x20\x20"
```

La final, programul va afișa cheia *K*, cuvântul care stă la baza ei, precum și numărul de încercări efectuate până la găsirea cheii *K*.

## Exercițiul 2

# Biblioteca OpenSSL - Funcții hash

Scrieți un program C/C++ ce utilizează API EVP pentru implementarea funcțiilor hash.

Programul va primi ca intrare două fișiere text,  $f_1$  și  $f_2$ , ce au același conținut, cu excepția unui singur caracter.

Programul va calcula funcțiile hash asociate celor două fișiere, utilizând algoritmi *MD5* și *SHA256*, rezultând fișierele *h1\_md5*, *h2\_md5*, respectiv *h1\_sha256*, *h2\_sha256*. De asemenea, programul va compara fișierele rezultate pentru fiecare algoritm în parte (*h1\_md5* va fi comparat cu *h2\_md5*, iar *h1\_sha256* va fi comparat cu *h2\_sha256*) la nivel de octet și va afișa numărul de octeți identici în cele două fișiere.

## Exercițiul 3

# Moduri operare Criptosisteme Bloc

### Punctaj maxim: 10 puncte

Implementați o infrastructură de comunicație ce folosește criptosistemul AES pentru criptarea traficului între două noduri  $A$  și  $B$  cu următoarele caracteristici:

- se consideră un nod  $KM$  (key manager) care deține trei chei pe 128 biți:  $K_1$ ,  $K_2$  și  $K_3$ :
  - cheia  $K_1$  este asociată cu modul de operare  $ECB$ ;
  - cheia  $K_2$  este asociată cu modul de operare  $OFB$  (se consideră că vectorul de inițializare are o valoare fixată, cunoscută în prealabil de cele două noduri  $A$  și  $B$ );
  - cheia  $K_3$  este utilizată pentru criptarea cheilor  $K_1$  sau  $K_2$ . Cheia  $K_3$  este deținută din start de nodurile  $A$ ,  $B$  și  $KM$ .
- Pentru a iniția o sesiune de comunicare securizată, nodul  $A$  trimite un mesaj către nodul  $B$  în care comunică modul de operare ( $ECB$  sau  $OFB$ ); de asemenea, nodul  $A$  transmite un mesaj nodului  $KM$  prin care cere cheia corespunzătoare ( $K_1$  pentru modul de operare  $ECB$ , respectiv  $K_2$  pentru modul de operare  $OFB$ ).
- Nodul  $B$ , la primirea mesajului de la nodul  $A$ , cere nodului  $KM$  cheia corespunzătoare ( $K_1$  pentru modul de operare  $ECB$ , respectiv  $K_2$  pentru modul de operare  $OFB$ ).
- nodul  $KM$  va cripta cheia cerută ( $K_1$  sau  $K_2$  în funcție de modul de operare ales) ca un singur bloc, utilizând criptosistemul AES cu cheia

$K_3$  și va trimite mesajul astfel obținut ca răspuns pentru nodurile  $A$  și  $B$ ;

- cele două noduri  $A$  și  $B$  vor decripta mesajul primit de la  $KM$  și vor obține astfel cheia cerută;
- nodul  $B$  trimite, după primirea cheii, un mesaj nodului  $A$  prin care îl anunță că poate să înceapă comunicarea;
- nodul  $A$  criptează conținutul unui fișier text utilizând AES, cheia primită de la  $KM$  și modul de operare ales.  $A$  va transmite nodului  $B$  blocurile de criptotext obținute pe rând, iar nodul  $B$  va decripta blocurile primite și va afișa rezultatul obținut.

Observații:

- se accepta utilizarea oricărui limbaj de programare și folosirea oricărei librării criptografice pentru implementare;
- AES poate fi folosit ca algoritm de criptare pus la dispoziție de orice librerie criptografică.
- se cere ca modul de operare ( $ECB$  sau  $OFB$ ) să fie implementat în cadrul temei.
- Nu se cere rezolvarea de eventuale probleme de sincronizare între noduri, interfață pentru noduri, sau un anumit protocol de comunicare.

# Precizări importante

Tema este individuală. Orice tentativă de fraudă este penalizată prin acordarea punctajului 0 tuturor studenților implicați.

Fiecare student(ă) va trimite prin email la adresa `ioana.leahu@gmail.com` o arhivă cu numele *prenume\_nume\_student\_grupa*, conținând câte un director pentru fiecare exercițiu rezolvat, fiecare director având următoarele fișiere:

- fișierele sursă ce conțin implementarea cerințelor exercițiului respectiv, inclusiv un fișier Makefile pentru compilare, dacă este cazul;
- fișiere de intrare, respectiv (dacă este cazul) de ieșire pentru exercițiul respectiv;
- un document pentru fiecare exercițiu rezolvat în parte, ce va conține:
  - descrierea mediului de lucru utilizat (alte setări decât cele prezentate în acest document);
  - descrierea modului de rezolvare al cerinței exercițiului;
  - testele efectuate pe diverse fișiere de intrare și observațiile efectuate.

**Termenul de predare al referatului** este fix, nu se admit întârzieri decât în cazuri bine justificate, anunțate în prealabil.