Tema 1 - Securitatea Informatiei
grupe semian A

Instalare OpenSSL

Pentru a instala biblioteca OpenSSL pe maşina virtuală Ubuntu (Lubuntu) trebuie efectuaţi următorii paşi:

```
$ wget https://www.openssl.org/source/openssl-1.1.0j.tar.gz
$ tar -xzvf openssl-1.1.0j.tar.gz
```
(*alternativ:* se descarcă din browser arhiva openssl-1.1.0j.tar.gz şi se dezarhivează utilizând utilitarele instalate în maşina virtuală)

se schimbă directorul curent in directorul in care s-a efectuat dezarhivarea
```
$ cd openssl-1.1.0j
```
se citeşte conţinutul fişierului INSTALL din directorul openssl-1.1.0j
```
$ xdg-open INSTALL
```

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

Exemplu fişier Makefile pentru compilarea unui program C ce utilizează funcţii 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
```

Observaţii:

- la adresa: `http://profs.info.uaic.ro/~nica.anca/is/words.txt` se găseşte un fişier text conţinând cuvinte de dicţionar englez, util pentru Exerciţiul 1;
- detalii privind programarea C utilizând funcţiile din API EVP: `https://www.openssl.org/docs/man1.1.0/crypto/crypto.html` sau `https://wiki.openssl.org/index.php/EVP`.

Exerciţiul 1

Biblioteca OpenSSL - Criptare/Decriptare

Punctaj maxim: 5 puncte

Scrieţi un program C/C++ ce utilizează API EVP pentru operaţii de criptare şi decriptare.

Programul va primi ca date de intrare numele a două fişiere, **plaintext**, respectiv **criptotext** şi un mod de operare (ECB sau CBC). Fişierul **criptotext** este rezultatul operaţiei de criptare cu o cheie `k` (şi un vector de iniţializare dat `iv = "\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f"` pentru modul CBC) a fişierului **plaintext**.

Scopul programului este de a găsi cheia `k` utilizată pentru criptarea fişierului **plaintext**, ştiind că:

- cheia `k` are o lungime fixată de 128 biţi (16 octeţi);
- cheia `k` este reprezentarea in hexa a unui cuvânt din dicţionarul englez cu lungime (mai mică) de 16 caractere, (completat cu caracterul spaţiu, cod hexa `\x20`, până la lungimea de 16 caractere);
  de exemplu, pentru cuvântul *cripto*, cheia

  ```
  k = "\x63\x72\x69\x70\x74\x6F\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20"
  ```

La final, programul va afişa cheia `k`, cuvântul care stă la baza ei, precum şi numărul de incercări efectuate până la găsirea cheii `k`.

Exerciţiul 2

Biblioteca OpenSSL - Funcţii hash

Punctaj maxim: 5 puncte

Scrieţi un program C/C++ ce utilizează API EVP pentru implementarea funcţiilor hash.

Programul va primi ca intrare două fişiere text, $\mathbf{file}_1$ şi $\mathbf{file}_2$, ce au acelaşi conţinut, cu excepţia unui singur caracter.

Programul va calcula funcţiile hash asociate celor două fişiere, utilizând algoritmii *MD5* şi *SHA256*, rezultând fişierele **hash1_md5**, **hash2_md5**, respectiv **hash1_sha256**, **hash2_sha256**. De asemenea, programul va compara fişierele rezultate pentru fiecare algoritm in parte (**hash1_md5** va fi comparat cu **hash2_md5**, iar **hash1_sha256** va fi comparat cu **hash2_sha256**) la nivel de octet şi va afişa numărul de octeţi identici in cele două fişiere.

Exerciţiul 3

Moduri operare Criptosisteme Bloc

Punctaj maxim: 10 puncte

Implementaţi o infrastructură de comunicaţie ce foloseşte criptosistemul AES pentru criptarea traficului intre două noduri A şi B cu următoarele caracteristici:

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

K_3 şi va trimite mesajul astfel obţinut ca răspuns pentru nodurile A şi B;

- cele două noduri A şi B vor decripta mesajul primit de la KM şi vor obţine astfel cheia cerută;
- nodul B trimite, după primirea cheii, un mesaj nodului A prin care il anunţă că poate să inceapă comunicarea;
- nodul A criptează conţinutul unui fişier text utilizând AES, cheia primită de la KM şi modul de operare ales. A va transmite nodului B blocurile de criptotext obţinute pe rând, iar nodul B va decripta blocurile primite şi va afişa rezultatul obţinut.

Observaţii:

- se accepta utilizarea oricărui limbaj de programare şi folosirea oricărei librarii criptografice pentru implementare;
- AES poate fi folosit ca algoritm de criptare pus la dispoziţie de orice librarie criptografică.
- se cere ca modul de operare (ECB sau CBC) să fie implementat in cadrul temei.
- Nu se cere rezolvarea de eventuale probleme de sincronizare intre noduri, interfaţă pentru noduri, sau un anumit protocol de comunicare.

Precizări importante

Tema este individuală. Orice tentativă de fraudă este penalizată prin acordarea punctajului 0 tuturor studenţilor implicaţi.

Fiecare student(ă) va trimite prin email la adresa **nica.anca@student.uaic.ro** o arhivă cu numele ***grupa_nume_prenume student***, conţinând câte un director pentru fiecare exerciţiu rezolvat, fiecare director având următoarele fişiere:

- fisierele sursa ce contin implementarea cerinţelor exercitiului respectiv, inclusiv un fisier *makefile* pentru compilare, daca este cazul;
- fisiere de intrare, respectiv de iesire, daca este cazul, pentru exercitiul respectiv;
- un document pentru fiecare exercitiu rezolvat in parte, ce va contine:
  - descrierea mediului de lucru utilizat (alte setări decât cele prezentate in acest document);
  - descrierea modului de rezolvare a cerinţei exerciţiului;
  - testele efectuate pe diverse fişiere de intrare şi observaţiile efectuate.

Termenul de predare a arhivei cu tema este: 20 octombrie 2019, ora 18.00; nu se admit intarzieri decat in cazuri bine justificate, anuntate in prealabil.