Comentarii temă curs I

Silviu-Laurențiu Vasile
Laurențiu-Vasile Crețu

Laborator de finanțe inovative,
13 martie 2025

Cum îmi generez o parolă/certificat digital

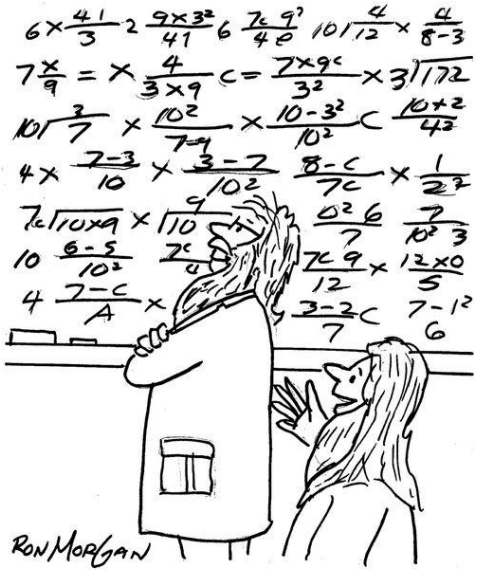
- **Resursă: *https://github.com/laurentiucretu68/ase-cybersecurity***

- ***OpenSSL*** este o librărie *open-source* utilizată pentru criptarea și comunicarea securizată pe rețea. Acronimul SSL vine de la *Secure Socket Layer*, un protocol criptografic care asigură confidențialitatea și integritatea datelor transmise între două părți (un client și un server)

- ***OpenSSL*** oferă o gamă largă de funcționalități, precum:

✓ Criptare simetrică
✓ Criptare asimetrică
✓ Generarea de chei și certificate digitale

- **Temă**: Criptarea/decriptarea simetrică a unui mesaj/fișier



"You've done it - you've come up with the perfect password."

Comentarii tema 1

- **Generare chei:**
  - *openssl genpkey -algorithm RSA -out private_key.pem -pkeyopt rsa_keygen_bits:2048*
  - *openssl rsa -pubout -in private_key.pem -out public_key.pem*
- ® **Observatie:**
  - *openssl genrsa -aes128 -out private_key.pem 2048*
  - *openssl rsa -in private_key.pem -pubout > public_key.pem*

- **Criptare** :
  - *openssl rsautl -encrypt -pubin -inkey public_key.pem -in mesaj.txt -out mesaj_secret.enc*
- ® **Observatie:**
  - *openssl pkeyutl -encrypt -inkey public_key.pem -pubin -in mesaj.txt -out mesaj_secret.enc*

- **Decriptare** :
  - *openssl rsautl -decrypt -inkey private_key.pem -in mesaj_secret.enc -out mesaj_decriptat.txt*
- ® **Observatie:**
  - *openssl pkeyutl -decrypt -inkey private_key.pem -in mesaj_secret.enc > mesaj_decriptat.txt*

"I'm sorry Ms. Tinsdale. My computer got a virus and lost my homework!"

Aplicații: hash fișiere

- Funcțiile hash/semnătură pot fi utilizate pentru verificarea integrității fișierului. De exemplu, pentru a verifica dacă fișierul nu a fost modificat în timpul transferului prin rețea. Acest exemplu scoate în evidență cum poate fi folosit OpenSSL în acest scop:
- Presupunem ca avem fisierul *contract.doc*
- Comanda *openssl dgst* poate fi folosită pentru a efectua diferite operații pentru a genera un hash al fișierului transmis folosind SHA-256, rulați următoarea comandă

```
openssl dgst -sha256 contract.doc
```

- *Rezultat:*

```
SHA2-256(contract.doc)= b5d2cfe30f3c8665d007490bc66f57a8699740e1c014207e178b47e52417b585
```

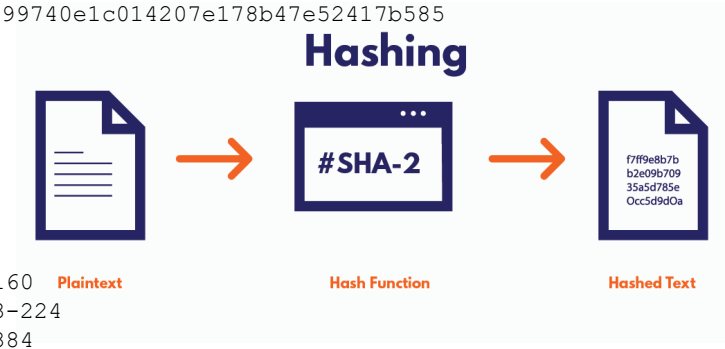
- *Pentru a salva rezultatul într-un fisier:*

```
openssl dgst -sha256 -out contract.sha256 contract.doc
```

- Alte comenzi utilizate pentru a genera hash-uri:

```
openssl list --digest-commands

blake2b512        blake2s256        md5               rmd160
sha1              sha224            sha256            sha3-224
sha3-256          sha3-384          sha3-512          sha384
sha512            sha512-224        sha512-256        shake128
shake256          sm3
```



Aplicații: Certificate digitale

Generarea certificatelor SSL cu OpenSSL implică mai mulți pași:

I. **Generare cheie private**:

```
openssl genpkey -algorithm RSA -out private_key.pem -pkeyopt rsa_keygen_bits:2048
sau varianta protejata prin cheie:
openssl genpkey -algorithm RSA -aes256 -out private_key.pem -pkeyopt rsa_keygen_bits:2048
```

II. **Crearea unei cereri de semnare a certificatului** (CSR)

```
openssl req -new -key private_key.pem -out request.csr
```

Vor fi cerute detalii precum:

```
Country Name (C): RO
State or Province (ST): București
Locality Name (L): București
Organization Name (O): Home SRL
Organizational Unit Name: HR
Common Name (CN): Domeniul certificatului (ex: curs2.ro)
Email Address: webmaster@curs.ro
```

III. **Autosemnarea certificatului** (dacă vrei un certificat self-signed)

```
openssl x509 -req -in request.csr -signkey private_key.pem -out certificate.crt -days 365
```

IV. **Certificat de la o autoritate de certificare** (CA): cel mai probabil vei avea nevoie de un certificat valid, trimite fișierul request.csr unei autorități de certificare (CA) precum Let's Encrypt, DigiCert, GlobalSign...

V. **Verificare certificate**

```
openssl x509 -in certificate.crt -text -noout
```