Univ. Babeș-Bolyai,

Facultatea de Matematică și Informatică

Lect. dr. Darius Bufnea

# Notițe de curs: Crearea unei autorități de certificare proof of concept, eliberarea unei certificat digital, semnarea unui document

## (material în cadrul cursului Protocoale de Securitate în Comunicați – nivel master)

Prezentul material își propune să fixeze cunoștințele legate de autoritățile de certificare, certificatele digitale, semnături digitale și semnarea unui document.

Instrumente necesare: Linux (orice distribuție) și OpenSSL. Teoretic exemplele din prezentul material funcționează și folosind OpenSSL for Windows (not tested). OpenSSL este o librărie open source ce implementează o serie de algoritmi criptografici (simetrici și asimetrici), folosită la scară largă cu precădere de software open source dar nu numai. Este însoțită și de un utilitar numit chiar openssl care permite folosirea directă a acestei librării in diverse scopuri specifice.

Problemă: Se dorește crearea în cadrul facultății a unui infrastructuri de Securitate bazata pe chei publice/chei private. Această infrastructură va fi folosită pentru comunicare între "actorii" din cadrul facultății (studenți, cadre didactice, secretariat, etc.) și pentru semnarea documentelor din cadrul fluxului de documente al organizației (facultății). Spre exemplu, un student ar putea primi nota în format digital (în cadrul unui document) care să fie semnat digital de către cadrul didactic. Nota în format digital semnată de cadrul didactic va fi păstrata de către student (fără a putea fi alterată/modificată de acesta) și va putea fi verificată ca validă de oricine dorește acest lucru: părinți studentului :), un angajator la angajarea acestuia, etc.

Autoritatea de certificare (CA) în cazul de față este Decanatul care va emite certificate digitale tuturor celor interesați (și implicați în prezentul scenariu). Este nevoie în primul rând ca autoritatea de certificare să își genereze o cheie privată (cea mai „sensibilă" resursă în această infrastructură din punct de vedere al securității) și să își emită un certificat self-signed (certificatul conține cheia publică a autorității de certificare = Decanatul și este semnat cu cheia privată a Decanatului = cheia privată corespunzătoare cheii publice din cadrul certificatului).

Observație: Pentru ca un certificat a unei autorități de certificare (Decanatul în cazul nostru) să fie recunoscut de toți actorii implicați (părinții studentului, secretariat, potențiali angajatori, etc.) trebuie ca:

- Certificatul să nu fie self-signed ci să fie semnat de o altă autoritate de certificare (acestea sunt organizate ierarhic, în vârful ierarhiei aflându-se Trusted Root Certification Authorities) care să fie recunoscută de browser-ul / sistemul de operare al tuturor celor implicați;
- Dacă certificatul Decanatului este self-signed, acesta trebuie diseminat spre toți cei implicați printr-un canal securizat (spre exemplu prin intermediul protocolului HTTPS sau înmânare personală pe un stick USB) și instalat ca și Trusted în cadrul browserului / sistemului de operare

ai celor ce vor să verifice actele emise (semnate) de Decanat dar și de profesorii cărora Decanatul le eliberează certificate (precum notele/cataloagele/carnetele digitale semnate de profesori).

```
$ # generare cheie privata a autoritatii de certificare si a
$ # certificatul aferent acesteia ce contine cheia publica
$ openssl req -new -x509 -keyout CA.private.encrypted.key -out CA.crt -days
60 -newkey rsa:4096
```

Descrierea parametrilor

- `-x509` este formatul in care va fi memorat certificatul digital;
- `-keyout CA.private.encrypted.key` specifică fișierul în care va fi scrisă cheia privată. Poate fi orice nume de fișier, numele ales `CA.private.encrypted.key` este doar o convenție pentru face înțelegerea numelui fișierului mai facilă;
- `-out CA.crt` numele fișierului în care va fi scris certificatul digital al autorității de certificare;
- `-days 60` specifică cât timp este valid certificatul (acesta conține două câmpuri, de când este valabil - de obicei data emiterii - și până când este valabil);
- `-newkey rsa:4096` - algoritmul și lungimea cheii (RSA, cheie pe 4096 de biți).

Nota: în toate exemplele din prezentul material `$` reprezintă prompterul Linux la care urmează să fie introdusă o comandă, iar # reprezintă un comentariu.

La generarea certificatului se cer câteva informații de identificare a celui căruia i se emite certificatul. Odată certificatul emis, aceste informații pot fi previzualizate în Windows în cadrul proprietăților certificatului (dublu click pe certificat, tab-ul Details):

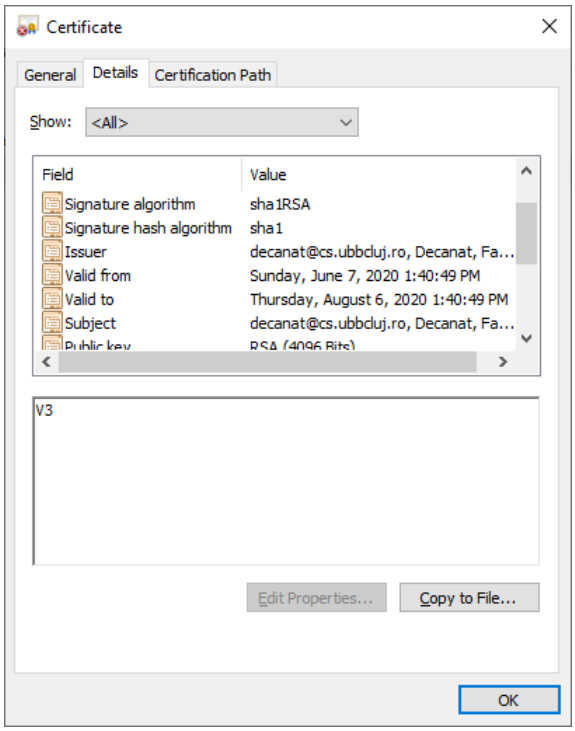



Figura 1: Proprietățile unui certificat digital

Observații:

- Pentru a se asigura confidențialitatea cheii private, aceasta se memorează criptat în sistemul de fișiere folosind un algoritm de criptare simetric (aceeași cheie/parolă este folosită atât la criptare cât și la decriptare). La generarea cheii private, se va cere parola de criptare a cheii private, parola ce va trebui să fie introdusă ori de câte ori aceasta este necesară (spre exemplu în cazul în care Decanatul trebuie să semneze ceva, sau să emită un certificat digital nou unui cadru didactic).
- Cheia privată este recomandat a fi memorată în sistemul de fișiere cu drepturi de citire/scriere doar pentru utilizator, nu și pentru grup sau alții (others):

```
-rw------- 1 bufny wheel 3394 Jun  7 12:12 CA.private.encrypted.key
```

- Doar cheia privată reprezintă informația „sensibilă”, nu și certificatul digital al autorității de certificare care trebuie diseminat/e normal să fie cunoscut de către toată lumea.

Cheia privata poate fi memorată și necriptată, cu observația că se recomanda setarea drepturilor de acces corespunzătoare (citire/scriere doar pentru utilizator) pe fișierul care va memora cheia privata decriptată:

```
$ # decriptarea cheii private
$ openssl rsa -in CA.private.encrypted.key -out CA.private.decrypted.key
```

Cheia publică a autorității de certificare poate fi obținută prin două procedee: poate fi calculată atât pe baza cheii private, dar poate fi extrasă și din cadrul certificatului digital al autorității de certificare:

```
$ # generarea chei publice pe baza cheii private
$ openssl rsa -in CA.private.decrypted.key -out CA.public.decrypted.key -
pubout

$ # extragerea cheii publice din cadrul certificatului
$ openssl x509 -in CA.crt -pubkey -noout > CA.public.decrypted.key2
```

(cele două fișiere `CA.public.decrypted.key` și `CA.public.decrypted.key2` trebuie să fie identice).

Informațiile despre certificat in Linux pot fi previzualizate cu comanda:

```
$ openssl x509 -text -in CA.crt
```

Informațiile afișate de comanda de mai sus trebuie sa fie identice cu cele din Figura 1.

Verificarea unui certificat se poate face cu comanda:

```
$ openssl verify CA.crt
CA.crt: C = RO, ST = Cluj, L = Cluj-Napoca, O = UBB, OU = Facultatea de
Matematica si Informatica, CN = Decanat, emailAddress = root@cs.ubbcluj.ro
error 18 at 0 depth lookup:self signed certificate
OK
```

## Generarea unui certificat pentru un utilizator al infrastructurii de Securitate bazată pe chei publice/chei private

În cele ce urmează vom genera un certificat digital și o cheie privată unui cadru didactic din facultate care dorește să utilizeze infrastructura de Securitate bazată pe chei publice/chei private pe care o prezentăm. Cheia privată a cadrului didactic va fi folosită la semnarea unei note spre exemplu, iar certificatul său digital va putea fi folosit de oricine dorește să verifice semnătura digitala a cadrului didactic relativ la o notă.

Mai întâi de toate, cadrul didactic are nevoie și el de o cheie privată. Aceasta cheie privată trebuie să fie cunoscută doar de el (și numai de el!), autoritatea de certificare - Decanatul în cazul nostru, sau orice altă autoritate de certificare emitentă de certificare digitale nu trebuie să cunoască această cheie:

```
$ # generare cheie privata a unui cadru didactic (utilizator)
$ openssl genrsa -des3 -out teacher.private.encrypted.key 4096
$ openssl rsa -in teacher.private.encrypted.key -out
teacher.private.decrypted.key
```

În urma acestor comenzii, fișierele `teacher.private.encrypted.key` și `teacher.private.decrypted.key` vor conține cheia cadrului didactic criptată și respectiv decriptată. Aceste fișiere sunt „sensibile" din puncte de vedere al securității, conținutul acestora neavând voie să fie cunoscut de către nimeni altcineva în afara de proprietarul acestei chei private.

După generarea cheii private, calculam cheia publică a cadrului didactic pe baza cheii sale private:

```
$ # generarea cheii publice pe baza cheii private
$ openssl rsa -in teacher.private.decrypted.key -out
teacher.public.decrypted.key –pubout
```

În urma acestei comenzi, fișierul `teacher.public.decrypted.key` conține cheia publică a cadrului didactic (informație publica, neconfidențială).

Odată generata perechea cheie privată / cheie publică a unui utilizator, acesta poate cere autorității de certificare (Decanatului) eliberarea unui certificat digital. Autoritatea de certificare (Decanatul) are nevoie de cheia publica (nu si de cheia privata) a celui căruia îi eliberează certificatul (cadrul didactic) și de informații legate de identitatea sa (certificatul digital semnat de autoritatea de certificare conținând printre altele informația: cheie publică utilizator + identitatea utilizatorului semnată de autoritatea de certificare).

## Cererea de eliberare a unui certificat digital

Utilizatorul trimite autorității de certificare cererea de eliberare a unui certificat digital sub forma unui fișier .csr (Certificate Signing Request). Acest fișier se generează pe baza cheii utilizatorului (va conține doar cheia publica a utilizatorului)și a informațiilor legate de identitatea sa:

```
$ # cererea de eliberare a certificatului digital emisa de utilizator catre
autoritatea de certificare
$ openssl req -new -key teacher.private.decrypted.key -out teacher.csr
```

Cererea de eliberare a certificatului digital (fișierul .csr) se trimite autorității de certificare pe un canal securizat (HTTPS spre exemplu). La nivelul acesteia, are loc emiterea certificatului digital către utilizator (a fișierului .crt ce va fi pus la dispoziția utilizatorului):

```
# eliberarea pe baza cererii de catre decanat a certificatului digital
cadrului didactic
openssl x509 -req -days 60 -in teacher.csr -out teacher.crt -CA CA.crt -CAkey
CA.private.decrypted.key
```

Remarcați că la emiterea unui certificat digital de către autoritatea de certificare este nevoie de cheia privată a acesteia (fișierul CA.private.decrypted.key) pentru a se semna certificatul emis profesorului.

Observații:

1. Pentru a putea emite certificatul, autoritatea de certificare trebuie sa atribuie certificatului un număr serial. Acest număr serial este util spre exemplu în cazul în care pe viitor se dorește/este necesara revocarea certificatului. Acest număr serial trebuie specificat în fișierul `CA.srl` (fișier inițializat cu "01") și este incrementat de fiecare dată când are loc eliberarea unui certificat.
2. Eliberarea unui certificat digital unei entități, presupune faptul că entitatea respectivă deține o cheie privată, iar autorității de certificare i se trimite cheia publică corespunzătoare acestei chei private (împreună cu informații despre identitatea celui ce cere / urmează sa i se elibereze certificatul). Pentru că generarea unei chei private nu este un proces facil pentru mulți utilizatori, autoritățile de certificare pot genera ele o cheie privată utilizatorilor care solicită certificate digitale, aceste chei private sunt eliberate (trimise) utilizatorilor odată cu certificatul digital corespunzător pe canale securizate (HTTPS spre exemplu), urmând ca ulterior autoritatea de certificare să șteargă cheile private generate utilizatorilor (acestea nu au voie sub nicio formă să memoreze aceste chei).

Fișierul `teacher.crt` generat în urma comenzii anterioare reprezintă certificatul digital al profesorului:

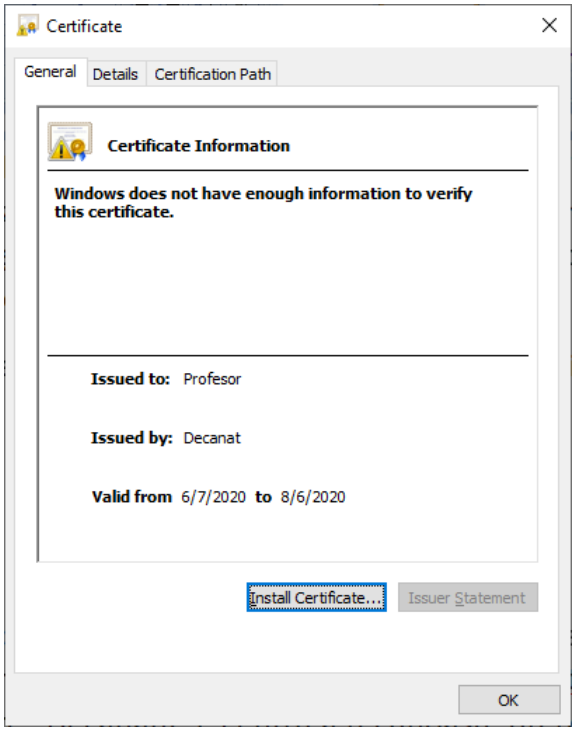


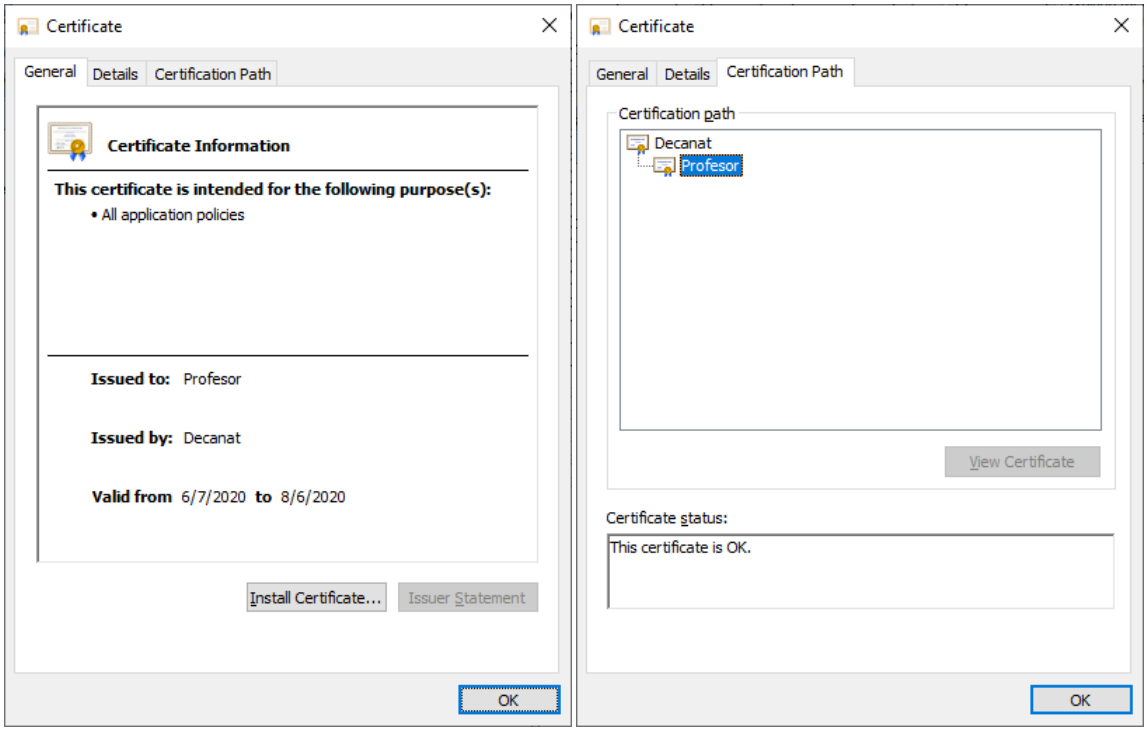

**Figura 2: Certificatul digital emis unui utilizator**

În imaginea de mai sus, se poate vedea că certificatul digital eliberat profesorului nu este recunoscut drept valid pentru că sistemul de operare (Windows) nu recunoaște certificatul autorității de certificare (nu are încrederea în autoritatea de certificare „Decanat”). Pentru a recunoaște drept valid certificatul emis profesorului, certificatul Decanatului trebuie importat ca Trusted Root Certificate Authority.

DISCLAIMAR: Dacă vă importați certificatul digital self-signed al autorității de certificare Decanat ca Trusted Root Certificate Authority să îl ștergeți imediat după parcurgerea acestui material. Teoretic, folosind cheia privată corespunzătoare acestui certificat pot fi semnate certificate „false” în care browserul / sistemul vostru de operare să aibă încredere ca fiind valide (lucru care poate duce la atacuri de tip phishing spre exemplu dacă cheia privată generată pentru autoritatea de certificare „Decanat” este compromisă). Pentru a înlătura un certificat al unei autorități de certificare din lista de Trusted Root Certificate Authorities, în Internet Explorer/Edge căutați în zona de Settings/Options opțiunea Manage Certificates.

Pentru a importa certificatul Decanatului în lista de Trusted Root Certificate Authority trebuie doar să urmați pașii următori: dublu click pe CA.crt (certificatul Decanatului), Install Certificate, Current User, Place all the certificates in the following store, Trusted Root Certificate Authorities.

În urma importării certificatului digital al Decanatului în lista de Trusted Root Certificate Authorities, certificatul emis profesorului este recunoscut drept valid:




Această validare poate fi făcută și în linia de comandă Linux pe baza certificatului digital al autorității de certificare. Acest proces presupune verificarea semnăturii depuse de autoritatea de certificare la semnarea certificatului emis profesorului, verificare realizată folosind cheia publică a autorității de certificare (ce se regăsește in cadrul fișierului `CA.crt`).

```
$ openssl verify -CAfile CA.crt teacher.crt
teacher.crt: OK
```

În continuare, cadrul didactic va completa un catalog sub forma unui fișier denumit `note.txt`:

```
Popescu Ion 9
John Doe 4
```

ulterior, urmând se semneze acest catalog:

```
$ openssl dgst -sha1 -sign teacher.private.decrypted.key -out note.txt.sha1
note.txt
```

Fișierul `note.txt.sha1` conține semnătura pusă de cadrul didactic și va trebui să însoțească catalogul (fișierul `note.txt`). Validarea notelor din catalog se poate face oricând în viitor pe baza semnăturii depuse de profesor și a cheii sale publice (cheie ce se poate obține și extrage din cadrul certificatului său digital `teacher.crt`):

```
$ openssl dgst -sha1 -verify teacher.public.decrypted.key -signature
note.txt.sha1 note.txt
Verified OK
```

Dacă studentul rău intenționat John Doe își modifică nota în fișier:

```
Popescu Ion 9
```

```
John Doe 5
```

urmând sa reclame la Decanat faptul ca el a promovat sau să-și mintă părinți cu un carnet de note „falsificat”, aceștia din urma pot ușor verifica integritatea „catalogului” (faptul că fișierul `note.txt` nu a fost modificat) pe baza semnăturii depuse de profesor și a cheii sale publice:

```
$ openssl dgst -sha1 -verify teacher.public.decrypted.key -signature
note.txt.sha1 note.txt
Verification Failure
```

Materialul de față s-a dorit să vă familiarizeze și să facă mai clare unele aspecte ce țin de procesele interne ce au loc în cadrul unei organizații ce folosește infrastructuri de securitate bazate pe chei publice/chei private. Într-un scenariu real, procesele prezentate manual în prezentul material (eliberare de certificate, cereri de eliberare, semnarea de documente, revocarea certificatelor, etc.) sunt automatizate și integrate în fluxul operațional al organizației respective. În toate cazurile însă, managementul cheilor private din cadrul organizației (și a securității acestor chei private), reprezintă unul din cele mai importante și „delicate” procese din cadrul unei astfel de infrastructuri.

## Concurs :)

Cei care îmi trimit un fișier denumit `note.txt` cu numele lor în fișier și o nota (10 să presupunem) de forma:

```
John Doe 10
```

împreuna cu un fișier semnătură denumit `note.txt.sha1` care să valideze împreună cu cheia publică de mai jos conținutul fișierului `note.txt` vor primi nota 10 la disciplina Protocoale de Securitate în Comunicații pe semestrul în curs.

```
-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAyHg5YGKDQuKSokIJdovX
Fimy+RUAhJJHFA1lEc/4RRP/4dwkiC1lw31DzDsoOkNwxqBlnyR3PrdodW5F0UDq
V4xt8Wk3j2un8lCm9Ji9kFdw/XPpaPedyh21ln1dTH8whRvDmdNJJHN2TbzeVORa
o8u1TsbafdOzQWV2LfBMfsLkLkBK8HI/pJflw4EMsgRdxbotdki1gzeOVJObQrbz
lr1bePW5W5xqUM1bbU/X3EInpOI4Pzj/o2njpv0OlwrPcNKJGSwsPBB6PD4Hxmdd
xTWl8Pdz0jQjbwpkARq+5tyFjTP5a2ypStRUxfu+gFl0cCu9LoCmrvlX5znWpmmv
kWY4hiR1KGxgNedjKM0N8WOw3/SumiTPj3SJcmyXv5Z3BuDPSy+AggVFqDmfBePg
FWN6eBbmeZ5LQMa+M80Pxwojt7ifJXu7N7peNJt7PCEx80hFuC7LeCV2qKMwvkEg
uQmHCvmPU+T39cab4xR0Kt2rljrtWqefIfo6ne8QekdUaohKgiBkseKGwKibFsDg
IJLQfWhqux4l/pXizBZk1ldqIv6ustoLbmPcSyy7ue7/67oQ1hkSjJhkKiL1kupu
BR5i2VSk//wGWF8GeokuTJgoy/2GWFZBuf2tdV2jjudhiZ1A8Vc/9tB/BUXMLYMP
uu652tl86wqvgw8iS6Sf03MCAwEAAQ==
-----END PUBLIC KEY-----
```

Ca de obicei, sunt deschis la orice observații/comentarii legate de calitatea acestui material sau posibile scăpări în cadrul său.