# TIM-22

# Marko Kopanja, Jovan Laković, Miloš Zrnić

**SW34-2015, SW51-2015, SW93-2017**

## Sertifikat x509:

|  |
| --- |
| Verzija |
| ID |
| Algoritam potpisivanja |
| Podaci o Izdavaocu |
| Period Validnosti |
| Podaci o Subjektu |
| Public Key |
| ID Izdavaoca |
| ID Subjekta |
| Ekstenzije |

Struktura

1. Verzija – 3
2. ID – pretpostavka da imamo nase, npr. za CA vojne jedinice u Srbiji ID = EU-SRB-CA-MU01 (zavisi od tipa sertifikata EU-SRB-MU01-CL1, EU-CA01...)
3. Algoritam – po najboljoj praksi
4. Ekstenzije – Client cert (Agent), SSL cert (Centar), CA cert.

## CA Node

Svaka klasa Certificate authority-ja mora da ima:

### Strukture podataka:

* 1. Keystore sa svojim sertifikatom I privatnim kljucem
  2. Keystore sa sertifikatima koje je izdao
  3. Listu ID-jeva trenutno validnih sertifikata koje je on izdao
  4. \*\*Potencijalno\*\* keystore sa simetricnim kljucevima za komunikacije(Centar-Agent) ako PKI posreduje u njihovoj komunikaciji I on im generise simetricni kljuc

### Funkcionalnosti:

\*odnose se na sertifikate/csr-ove koji se nalaze u lancu ispod njega

* 1. Prikaz CSR-ova
  2. Prikaz Sertifikata
  3. Inicijalno izdavanje novog sertifikata
  4. Povlacenje sertifikata
  5. Obnova postojeceg sertifikata
  6. Automatsko posredovanje u komunikaciji – obavezan minimum provera validnosti sertifikata

#### \*Potencijalna implementacija\*

* 1. Sistem notifikacija za admina(npr. kada stigne novi CSR, sutradan istice neciji sertifikat pa da zna da treba da ga zameni itd itd)
  2. Automatsko posredovanje u kominikacijama - generisanje, skladistenje I slanje simetricnih kljuceva

### Inicijalno izdavanje sertifikata:

1. Iz liste CSR-ova ADMIN izabere jedan kako bi izgenerisao novi sertifikat
2. Otvara se forma za pravljenje sertifikata
3. Automatski se popunjava podacima iz CSR-a(SubjectData), svojim podacima (CA)(IssuerData) I generise se ID
4. Admin ima opciju da generise Public I Private key. Javni kljuc se dodaje u formu, private key se negde privremeno cuva.
5. Admin bira ekstenziju od ekstenzija koje su u njegovom ovlascenju
6. Klikom na “Save” sertifikat se cuva I postaje validan - sertifikat se dodaje u keystore, njegov ID se dodaje u listu trenutno validnih sertifikata(\*\*proveriti da li je kod sistema gde CA generise Pub/Priv kljuceve potrebno cuvati privatni kljuc – verovatno da je korisno za “key recovery” ako klijent izgubi nekako kljuc ali bi trebalo da je svakako sigurnije da se pravi ispocetka novi sertifikat u tom slucaju\*\*)
7. Admin ima opciju da download-uje sertifikat I privatni kljuc kako bi oni mogli biti instalirani na klijentskim racunarima – privatni kljuc sertifikata mora minimalno da se cuva do ovog koraka

### Povlacenje sertifikata:

1. Iz liste sertifikata ADMIN izabere onaj koji zeli da povuce
2. Sertifiakt se povuce tako sto se obrise kod CA I njegov ID se brise iz liste validnih sertifikata
3. \*\*Ako je moguce da automatski obrise sertifikat I privatni kljuc I kod klijenta ako nam klijent to bude podrzavao

### Obnova postojeceg sertifikata

1. Iz liste sertifikata admin izabere jedan koji zeli da obnovi
2. Iskoci mu forma za generisanje novog sertifikata koja je automatski popunjena
3. Prati ostale korake kao kod “Inicijalnog izdavanja”
4. Prati korake kao kod “Povlacenja sertifikata”

Pojašnjenje: Ova funkcionalnost moze da bude skroz odradjena preko HTTPS-a, zato sto ovde moze da se uspostavi sigurna komunikacija sa starim sertifikatom (sertifikatom koji se obnavlja) i da se novi sertifikat I privatni kljuc potpisu starim javnim kljucem klijenta. ( ovde bi verovatno moralo da bude podrzano I to automatsko brisanje kod klijenta)

*Druga opcija je da se dostavljanje vrši preko USB-a/Kartice.*

### Automatsko posredovanje u komunikaciji

Komunikacija SIEM Centar – SIEM Agent:

Podrzava servis preko kojeg može da se proveri da li je sertifikat validan (**OCSP**)

1. Centar/Agent posalju sertifikat od druge strane da bi ga proverili
2. PKI Server ga interno proverava I vraca True/False odgovor

\*\*Nećemo raditi sa javnom listom (*CRL*).

#### Potencijalno će biti implementirano:

* Podrzava servis preko kojeg Centar I Agent mogu dobiti simetricni kljuc preko kojeg ce da komuniciraju

1. Nakon provere sertifikata, Centar I Agent ce slati nove zahteve za simetricnim kljucem
2. Simetricni kljuc se generise I salje se potpisan odgovarajucim javnim kljucevima
3. \*\*Mozda se simetricni kljuc cuva negde u PKI-ju kako se ne bi novi generisao svaki put kada centar I agent uspostave komunikaciju

##### \*\*NAPOMENE\*\*

Kod ovakvog sistema skladistenje sertifikata je podeljeno po CA-ovima (svaki sertifikat se cuva samo kod onog CA koji ga je izdao) tako da moramo omoguciti mehanizam da NPR. Root admin ima pristup svim mogucim sertifikatima u sistemu (isto npr. da Europe CA ima pristup svim sertifikatima u Evropi). U “pravom svetu” verovatno da bi svaki CA imao svoj server sa instancom aplikacije pa bi ovo moralo da bude omoguceno HTTPS-om, ali posto kod nas ne mora tako, dovoljno je da mozemo samo da instanciramo objekat subCA I pristupimo direktno njegovom keystore-u.