HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Upravljanje digitalnim certifikatima korištenjem OpenSSL programskog paketa

CCERT-PUBDOC-2005-09-135

CARNet CERT u suradnji s LS&S

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.

Rezultat toga rada ovaj je dokument, koji je nastao suradnjom CARNet CERT-a i LS&S-a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr - nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.lss.hr - laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

Sigurne transakcije i izmjena povjerljivih elektroničkih informacija postavljaju važne sigurnosne zahtjeve kao što su očuvanje integriteta, autentičnost i povjerljivost. Navedeni sigurnosni ciljevi često se ostvaruju korištenjem asimetričnih kriptografskih algoritama, odnosno kriptografskih algoritama javnog ključa (eng. *public key cryptography*). Ovakav pristup omogućuje samo onim entitetima koji posjeduju valjani ključ izvršenje određenih akcija, kao što su digitalno potpisivanje dokumenata ili dekriptiranje kriptiranih podataka.

Koncept koji povezuje vlasnike i njihove javne ključeve na pouzdan način naziva se infrastruktura javnih ključeva, odnosno PKI (eng. *Public Key Infrastructure*) infrastruktura. Digitalni certifikat je jedan od ključnih elemenata PKI infrastrukture, te je za razumijevanje načina rada digitalnih certifikata potrebno poznavanje osnova PKI infrastrukture.

Digitalni certifikat je jedna od tehnologija koja tvori bazu Internet sigurnosti. To su računalne datoteke koje djeluju kao *online* propusnice, te omogućuju autentikaciju svojih vlasnika, kao i zaštitu podataka izmijenjenih preko javnih kanala. Uloga certifikata u PKI infrastrukturi je automatizacija procesa distribuiranja javnih ključeva i sigurne izmjene podataka. Temelj ove tehnologije predstavlja X.509 standard, kojim su definirane informacije koje se pohranjuju u certifikat, te format zapisa tih informacija. Iako PKI nije zaživio u onoj mjeri kako je bilo predviđano u prošlosti, certifikati su ipak pronašli svoju primjenu, od kojih je jedna od najočitijih ona u Web preglednicima koji podržavaju komunikaciju putem SSL kanala.

U dokumentu će ukratko biti opisana PKI infrastruktura, kao i osnovni elementi koji je sačinjavaju, s posebnim naglaskom na digitalne certifikate. Također, ukratko će biti opisane osnovne funkcionalnosti OpenSSL programskog paketa, te način komunikacije SSL protokolom. Konačno, dan je detaljan primjer generiranja certifikata za potrebe Web poslužitelja.

2. PKI infrastruktura

PKI infrastruktura je jedna od robusnijih metoda za sigurnu komunikaciju putem nesigurnih kanala, kao što je Internet. Digitalni certifikat je osnovni element ove strukture koja služi za podizanje razine sigurnosti informacijskog sustava.

Digitalni certifikat je elektronički dokument koji identificira osobe i resurse na javno dostupnim mrežama, a temelji se na tehnologiji javnih i privatnih ključeva. Ovi ključevi se uvijek generiraju u paru. Podaci koji se kriptiraju pomoću javnog ključa, dekriptiraju se privatnim ključem, i obratno. Svaki ključ je jedinstven, te se stoga pomoću njega može identificirati vlasnik. Javni ključevi su dostupni svima koji žele sigurno komunicirati s vlasnikom, dok se privatni ključ nikad ne prenosi nesigurnim kanalima, te je poznat samo vlasniku. S ciljem očuvanja autentičnosti poruke PKI tehnologija omogućuje i digitalno potpisivanje poruka. Potpis se kriptira privatnim ključem, te je stoga primatelj u mogućnosti provjeriti autentičnost poruke dekriptiranjem potpisa javnim ključem entiteta koji je poslao poruku. Uobičajeno je da se potpisom smatra sažetak (*hash*) same poruke.

Digitalni certifikat potvrđuje povezanost javnog i tajnog ključa koje posjeduje neki entitet. Identitet entiteta, koji može biti osoba ili aplikacija, je sadržan u certifikatu zajedno s javnim ključem. Kako bi ovakva struktura bila valjana, ona mora biti potpisana od treće strane, certifikacijskog autoriteta CA (eng. *Certificate Authority*), koji predstavlja autoritet od povjerenja. Razina povjerenja ovisi o samom CA-u. Proces izdavanja certifikata može uključivati i registracijski autoritet, RA (eng. *Registration Authority, Request Authority*), koji je zadužen za registraciju i eventualnu verifikaciju identiteta korisnika. Digitalni certifikati se nakon objave mogu slobodno razmjenjivati ili objavljivati u odgovarajućim repozitorijima, kao što su LDAP ili X.500 direktoriji. X.500 je ISO/ITU standard koji definira način strukturiranja globalnih direktorija.

Vrlo bitna stavka digitalnih certifikata je i trajanje njihove valjanosti. Ova vrijednost predstavlja vremenski period unutar kojeg certifikat vrijedi, te je unaprijed definirana. U slučaju kompromitacije korisnika ili digitalnog certifikata valjanost može biti ukinuta i prije isteka vremenskog roka na koji je digitalni certifikat inicijalno izdan. Povlačenje digitalnih certifikata moguće je obaviti objavom informacije o povlačenju na CRL (eng. *Certificate Revocation List*) revokacijskoj listi, ili dobivanjem informacija o povučenim digitalnim certifikatima korištenjem OCSP (eng. *Online Certificate Status Protokol*) protokola. Više o načinu povlačenja digitalnih certifikata moguće je saznati iz dokumenta dostupnom na adresi http://www.cert.hr/filehandler.php?did=202. Osim upravo navedenih elemenata PKI infrastrukture, postoje i dodatni elementi, kao što su točke povjerenja koje omogućuju provjeru valjanosti, upravljanje privatnim ključevima koje pruža obnovu digitalnih certifikata prije isteka valjanosti, te *back-up* privatnih ključeva. Ovi elementi PKI infrastrukture neće biti detaljno razmatrani u sklopu ovog dokumenta.

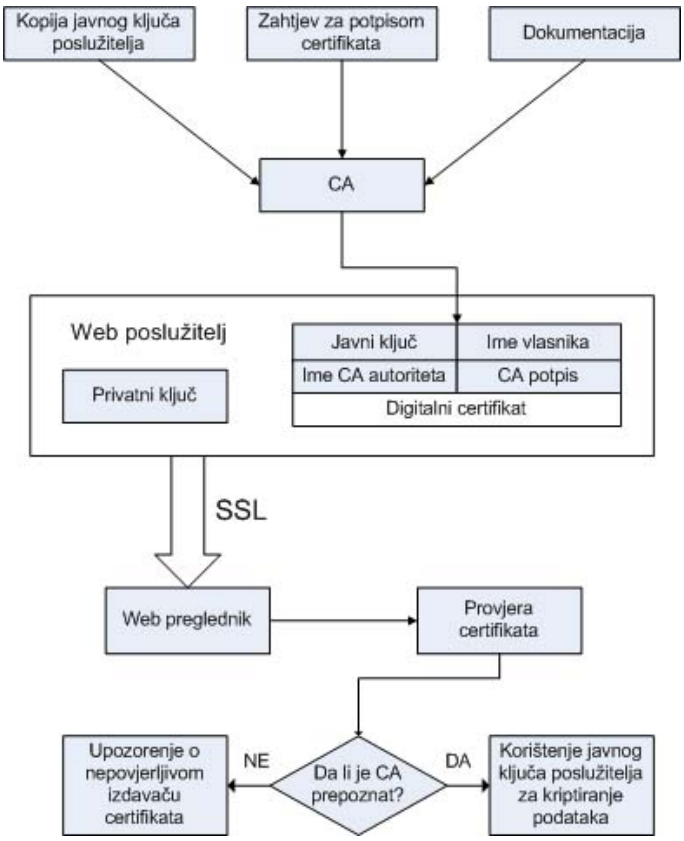
2.1. Format digitalnog certifikata

Format digitalnog certifikata definiran je X.509 standardom, koji ga opisuje kao strukturu od 3 polja:

- `tbsCertificate` – niz, odnosno struktura koja se sastoji od informacija koje identificiraju entitet kojem se izdaje certifikat i izdavača digitalnog certifikata. U strukturi su sadržani ime izdavača certifikata, ime entiteta i njegov javni ključ, period valjanosti, inačica, te eventualne dodatne ekstenzije.
- `signatureAlgorithm` – polje koje sadrži identifikator algoritma kojim je digitalni certifikat digitalno potpisan od strane izdavača (npr. RSA-MD5, RSA-SHA1, DSA-SHA1).
- `signatureValue` – polje koje sadrži digitalni potpis, koji predstavlja `tbsCertificate` strukturu enkodiranu ASN.1 (eng. *abstract syntax notation*) DER (eng. *distinguished encoding rules*) pravilima nad kojom je proveden algoritam naveden u `signatureAlgorithm` polju.

3. Komunikacija putem SSL protokola

Za sigurnu komunikaciju između Web poslužitelja i korisnika koji ga posjećuju potrebno je generiranje para javnog i privatnog ključa poslužitelja. Svim korisnicima koji moraju poslužitelju proslijediti povjerljive informacije dodjeljuje se javni ključ poslužitelja. Ovaj ključ služi za dogovaranje detalja sjednice, kao što su algoritam kriptiranja te sjednički ključ (eng. *session key*) koji će biti korišten u komunikaciji. Nakon što su dogovoreni, ovi parametri se koriste za sigurnu komunikaciju između Web poslužitelja i njegovih klijenata. Problem koji se javlja u ovakvoj koncepciji je pitanje vjerodostojnosti poslužitelja kojem se povjerljive informacije predaju, gdje do izražaja dolazi potreba za uporabom digitalnih certifikata. Web poslužitelj bi trebao posjedovati digitalni certifikat kojim dokazuje da je uistinu onaj za kojeg se izdaje, pri čemu je certifikat potpisan od strane CA kojem korisnici mogu vjerovati. Na Internetu je dostupan velik broj tvrtki koje se bave ovom djelatnošću, od kojih su poznatije Thawte (http://www.thawte.com/) i VeriSign (http://www.verisign.com/).
Prilikom podnošenja zahtjeva za izdavanjem digitalnog certifikata uz javni ključ se predaje i dokumentacija kojom se dokazuje identitet entiteta. CA generira digitalni certifikat koji sadrži javni ključ, ime vlasnika poslužitelja, vlastito ime kao CA autoritet koji je izdao certifikat, te vlastiti digitalni potpis. Na Slici 1 je prikazan put digitalnog certifikata na sigurnom Web poslužitelju.



Slika 1. Put digitalnog certifikata na Web poslužitelju

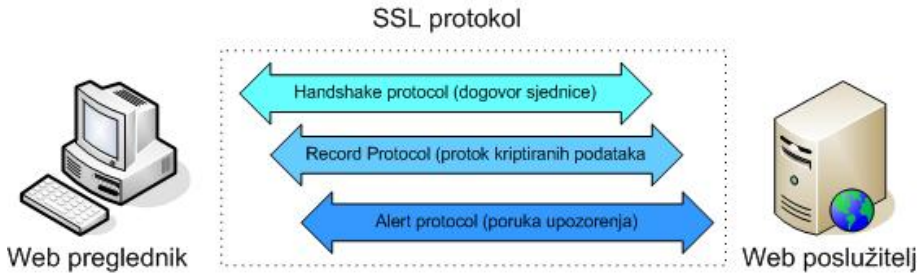
Provjera digitalnog certifikata Web poslužitelja na kojem postoji SSL podrška je slijedeća:

- Korisnik se spaja na Web stranicu na kojoj je omogućena SSL komunikacija korištenjem https://url umjesto http://url adrese.
- Poslužitelj prezentira svoj digitalni certifikat Web pregledniku korisnika, koji provjerava certifikat, te pokušava potvrditi da je CA koji ga je potpisao jedan od onih koje preglednik prepoznaje i kojem vjeruje, te da li je u mogućnosti provjeriti digitalni potpis.

- Ukoliko provjera prođe uspješno, Web preglednik korisnika prihvaća javni ključ zapisan u digitalnom certifikatu.
- U slučaju neuspješne provjere Web preglednik objavljuje upozorenje korisniku da ne prepoznaje CA autoritet.

Za obostrano sigurno komuniciranje pomoću asimetričnog algoritma, svaki korisnik bi također trebao posjedovati svoj par javnog i privatnog ključa. Tek na ovaj način je moguća i obostrana autentikacija. Korisnik mora posjedovati digitalni certifikat jedino u slučaju potrebe dokazivanja svog identiteta poslužitelju.

PKI tehnologija se u SSL komunikaciji koristi jedino u fazi rukovanja. Faza rukovanja uključuje pregovor poslužitelja i Web preglednika o simetričnom algoritmu kojim će se kriptirati podaci tijekom komunikacije, te za dogovor ključa koji će vrijediti samo za vrijeme trajanja sjednice. U ovoj fazi javni ključ poslužitelja služi za autentikaciju poslužitelja i zaštitu dogovora o sjedničkom ključu. Nakon što SSL rukovanje završi, slijedi drugi dio protokola (eng. *Record Protocol*) u kojem se svi podaci koji se izmjenjuju između poslužitelja i Web preglednika kriptiraju dogovorenim simetričnim algoritmom. Ukoliko tijekom sjednice dođe do potrebe za bilo kakvom vrstom upozorenja, SSL protokol ima i za takav tip poruke definirani protokol, nazvan *Alert* protokol. Iz upravo navedenog može se zaključiti da se SSL protokol sastoji od tri različita protokola, što je prikazano slikom 2.



Slika 2. Dijelovi SSL protokola

Sva komunikacija koja se odvija putem SSL kanala je osigurana kriptiranjem, ali vrlo često se krivo pretpostavlja da je snaga kriptografskog algoritma određena digitalnim certifikatom. Snaga SSL sjednice zapravo je funkcija Web preglednika i mogućnosti Web poslužitelja. Ako poslužitelj podržava 128-bitne ključeve, a Web preglednik samo 40-bitne ključeve, sjednica će se ostvariti korištenjem 40-bitnog ključa.

4. OpenSSL programski paket

OpenSSL je besplatan kriptografski alat koji implementira SSL v2/v3 (eng. *Secure Socket Layer*) i TLS v1 (eng. *Transport Layer Security*) mrežne protokole, te ostale kriptografske standarde koji se odnose na ove protokole. OpenSSL pruža mogućnost pozivanja raznih kriptografskih funkcija koje su ugrađene u zbirke datoteka OpenSSL programskog paketa iz komandne linije. Lista funkcionalnosti koju pruža OpenSSL je slijedeća:

- kreiranje parametara za RSA, DS i DSA ključeve,
- kreiranje X.509 digitalnih certifikata, CRL lista, te CSR (eng. *Certificate Signing Request*) zahtjeva,
- izračunavanje sažetaka poruka,
- kriptiranje i dekriptiranje,
- podrška SSL/TLS komunikaciji,
- rukovanje e-mail porukama potpisanim ili kriptiranim S/MIME standardom.

OpenSSL programski paket nudi veliki broj naredbi, a svaka od njih koristi dodatne opcije i argumente. Neke od važnijih i češće korištenih naredbi uključuju:

- `ca` – upravljanje certifikacijskim autoritetom,
- `crl` – upravljanje revokacijskim listama digitalnih certifikata,
- `dgst` – izračunavanje sažetka,
- `rand` – generiranje pseudoslučajnih vrijednosti,
- `x509` – upravljanje X.509 certifikatima.

U paketu dolaze i pseudo naredbe `list-standard-commands`, `list-message-digest-commands` i `list-cipher-commands` koje redom služe za ispis svih dostupnih standardnih naredbi, naredbi za izračunavanje sažetka, te kriptografskih naredbi.

4.1. Instalacija

OpenSSL programski paket moguće je besplatno dohvatiti sa službenih stranica OpenSSL projekta koje su dostupne na adresi http://www.openssl.org/. Nakon dohvaćanja izvornog kôda OpenSSL programskog paketa instalacija na Linux operacijskim sustavima je moguća kroz uobičajenu proceduru:

- Pokretanje konfiguracijske skripte, koja bez dodatnih parametara priprema instalaciju u direktoriju `/usr/local/openssl`. Konfiguriranje je moguće prilagoditi dodatnim parametrima čiju je kompletnu listu moguće pronaći u dokumentaciji OpenSSL paketa.
- Pokretanje `make` skripte za izgradnju zbirki datoteka.
- Provjera rezultata prethodnog koraka pokretanjem `make test` naredbe.
- Instalacija naredbom `make install`, ukoliko je testiranje prošlo bez pogrešaka.

Instalacija OpenSSL programskog paketa na Debian sustavima je moguća i korištenjem `apt-get` naredbe, što instalaciju čini vrlo jednostavnom i brzom.

5. Primjer generiranja certifikata OpenSSL programskim paketom

Za potrebe primjera pretpostavlja se da na poslužitelju na kojem će se koristiti digitalni certifikat već postoji funkcionalni Web poslužitelj sa instaliranom SSL podrškom. Kako se usluge potpisivanja digitalnog certifikata naplaćuju, u primjeru će se kreirati vlastito potpisani certifikat. Digitalni certifikat će biti potpisan privatnim ključem samog Web poslužitelja. Alternativno rješenje je i kreiranje vlastitog CA autoriteta na poslužitelju koji će potom potpisati digitalni certifikat Web poslužitelja. Ovakva procedura je korisna ako će CA autoritet biti iskorišten za potpisivanje više digitalnih certifikata. U primjeru se generira certifikat samo za Web poslužitelj, te nema potrebe za kreiranjem CA, već će se u nastavku dati samo smjernice kreiranja vlastitog CA.
U praksi je vrlo čest slučaj da su izdavač certifikata i subjekt certifikata jedan te isti entitet. Iako ovakav pristup nije valjan za komercijalne Web stranice, dovoljno je dobar za testiranje ispravnosti instalacije i konfiguracije certifikata.

5.1. Vlastiti CA

U sklopu OpenSSL programskog paketa dolazi i podrška za upravljanje vlastitim CA autoritetom. Cijela procedura kreiranja vlastitog CA se sastoji od kreiranja direktorija i datoteka potrebnih za pohranu podataka, kreiranja privatnog ključa CA, te konfiguriranja određenih postavki OpenSSL konfiguracijske datoteke. CA također mora posjedovati svoj certifikat koji bi trebao biti potpisan od višeg autoriteta.
Kreiranje vlastitog CA na poslužitelju započinje generiranjem odgovarajuće strukture direktorija. U direktoriju gdje je instaliran OpenSSL, `/etc/ssl/`, potrebno je kreirati direktorij za CA autoritet, te u njemu sve potrebne podstrukture:

```
/etc/ssl# mkdir CA
/etc/ssl# cd CA
/etc/ssl/CA# mkdir certs crl newcerts private
/etc/ssl/CA# echo "01" > serial
/etc/ssl/CA# cp /dev/null index.txt
```

Nakon kreiranja CA direktorija, u njemu su kreirani još direktoriji za izdane certifikate (`/certs`), za izdanu crl listu povučenih certifikata (`/crl`), za zahtjeve certifikata (`/newcerts`), te direktorij za pohranu privatnog ključa CA autoriteta (`/private`). Inicijaliziran je i brojač serijskih brojeva izdanih certifikata (`serial`), te datoteka za bazu podataka digitalnih certifikata (`index.txt`).
Nakon kreiranja potrebnih struktura potrebno je konfigurirati postavke OpenSSL konfiguracijske datoteke koji se odnose na CA (opisane u poglavlju 5.2.1),
Slijedi generiranje privatnog i javnog ključa CA autoriteta, te digitalnog certifikata. Ova procedura je analogna onoj koja će biti predstavljena u primjeru, te ju je moguće pronaći u poglavlju 5.3.

5.2. Konfiguracija

U direktoriju gdje je instaliran OpenSSL programski paket nalazi se i `openssl.cnf` konfiguracijska koja se sastoji od nekoliko dijelova. Svaki dio ima svoju namjenu:

- `ca`, `CA_default` – definiranje postavki CA autoriteta,
- `policy_match`, `policy_anything` – definiranje različitih politika zahtjeva,
- `req`, `req_distinquished_name`, `req_attributes` – definiranje standardnih vrijednosti zahtjeva.

5.2.1. Definiranje konfiguracije CA autoriteta

Prije korištenja certifikacijskog autoriteta potrebno je podesiti konfiguracijsku datoteku, pri čemu treba posebnu pozornost obratiti na direktivu `dir` koja definira direktorij u kojem se nalaze svi potrebi poddirektoriji. U ovom konkretnom primjeru to je jedina direktiva koju je obavezno promijeniti, te ju je potrebno postaviti na vrijednost prethodno generiranog direktorija `/CA`. U dijelu konfiguracije certifikacijskog autoriteta se između ostalog nalaze putanje do prethodno definiranih direktorija:

```
#####################################################################
[ ca ]
```

```
default_ca      = CA_default            # The default ca section

######################################################################
[ CA_default ]

dir             = /CA                   # Glavni direktorij za pohranu
certs           = $dir/certs            # Pohrana izdanih certifikata
crl_dir         = $dir/crl              # Pohrana izdanih crl listi
database        = $dir/index.txt        # indeks datoteka baze podataka.
#unique_subject = no                    # 'no' dozvoljava kreiranje vise
                                        # certifikata s istim subjektom.
new_certs_dir   = $dir/newcerts         # Novi certifikati.

certificate     = $dir/cacert.pem       # CA certifikat
serial          = $dir/serial           # Trenutni serijski broj
#crlnumber      = $dir/crlnumber        # the current crl number must be
                                        # commented out to leave a V1 CRL
crl             = $dir/crl.pem          # Trenutna CRL lista
private_key     = $dir/private/cakey.pem# Privatni kljuc
RANDFILE        = $dir/private/.rand    # private random number file

x509_extensions = usr_cert              # The extentions to add to the cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt        = ca_default            # Subject Name options
cert_opt        = ca_default            # Certificate field options

# Extension copying option: use with caution.
# copy_extensions = copy

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crlnumber must also be commented out to leave a V1 CRL.
# crl_extensions        = crl_ext

default_days    = 365                   # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md      = md5                   # which md to use.
preserve        = no                    # keep passed DN ordering
```

5.2.2. Definiranje politika zahtjeva

Dio konfiguracijske datoteke koji se odnosi na politike zahtjeva definira različite politike potpisivanja digitalnih certifikata. U standardnom formatu definirane su samo dvije politike, najblaža politika (`policy_anything`), i nešto striktnija politika (`policy_match`). Ove politike rade na taj način da se prilikom potpisivanja certifikata provjeravaju ona polja u zahtjevu digitalnog certifikata za koje politika zahtjeva "`match`", odnosno poklapanje s vrijednošću CA certifikata. Ukoliko se vrijednosti ne slažu, digitalni certifikat neće biti potpisan. Osim ključne riječ "`match`" koja zahtjeva poklapanje, postoje još i „`optional`“ koja definira polja kao opcionalno, te „`supplied`“ koja traži da je polje prisutno, ali se ne mora poklapati s vrijednošću iz digitalnog certifikata CA autoriteta. U dijelu definiranja politika, osim postavljanja ključnih riječi na polja koja se pojavljuju u zahtjevu certifikata, postoji i direktiva `policy`, koja određuje koja će se politika koristiti od postojećih.

```
policy          = policy_anything

# For the CA policy
[ policy_match ]
countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
```

```
emailAddress            = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional
```

5.2.3. Definiranje standardnih vrijednosti zahtjeva

Dio konfiguracijske datoteke `req` se koristi prilikom kreiranja zahtjeva za digitalnim certifikatom, te definira standardne vrijednosti i minimalne i maksimalne duljine polja koja se pojavljuju u zahtjevima. Neka od ovih polja, kao što je npr. `commonName`, koje identificira entitet koji zahtjeva certifikat, će biti različita za svaki novi zahtjeva digitalnog certifikata, dok će neka druga polja iskoristiti standardne vrijednosti (npr. `countryName`). Za potrebe primjera ovaj dio konfiguracijske datoteke je popunjen na slijedeći način:

```
####################################################################
[ req ]
default_bits            = 1024
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
x509_extensions = v3_ca # The extentions to add to the self signed
cert

# Passwords for private keys if not present they will be prompted for
# input_password = secret
# output_password = secret

# This sets a mask for permitted string types. There are several
options.
# default: PrintableString, T61String, BMPString.
# pkix   : PrintableString, BMPString.
# utf8only: only UTF8Strings.
# nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).
# MASK:XXXX a literal mask value.
# WARNING: current versions of Netscape crash on BMPStrings or
UTF8Strings
# so use this option with caution!
string_mask = nombstr

# req_extensions = v3_req # The extensions to add to a certificate
request

[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = HR
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Hrvatska

localityName                    = Locality Name (eg, city)

0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = LSS
```

```
organizationalUnitName              = Organizational Unit Name (eg,
section)
#organizationalUnitName_default =

commonName                      = Common Name (eg, YOUR name)
commonName_max                  = 64

emailAddress                    = Email Address
emailAddress_max                = 64

# SET-ex3                       = SET extension number 3

[ req_attributes ]
challengePassword               = A challenge password
challengePassword_min           = 4
challengePassword_max           = 20

unstructuredName                = An optional company name
```

Konfiguracijska datoteka ne zahtjeva puno preinaka u odnosu na standardno distribuirani oblik, ali je nužno ispravno podesiti putanje direktorija. Za brže kreiranje zahtjeva pogodno je unaprijed definirati standardne, odnosno očekivane vrijednosti polja, dok je kroz kreiranje politika moguće postaviti ograničenja na pojedine vrijednosti polja digitalnog certifikata.

5.3. Generiranje certifikata

Apache 2.x distribucija standardno uključuje SSL podršku, dok se za Apache 1 seriju mogu koristiti dvije različite implementacije SSL podrške. Jedna je mod_ssl modul, koji se koristi u kombinaciji s Apache 1 programskim paketom, dok je druga zasebna instanca poslužitelja Apache-SSL koji se također bazira na Apache seriji 1. Ne postoje konkretni razlozi zašto je jedan odabir bolji od drugoga, stoga se odluka prepušta administratoru sustava. Bez obzira koji se Web poslužitelj koristi, i uz koju podršku, prisutnost OpenSSL programskog paketa na sustavu je neophodna, jer se sve ovdje navedene implementacije oslanjaju na metode OpenSSL paketa.
Nakon što su na sustavu prisutni OpenSSL programski paket i Apache Web poslužitelj s osiguranom SSL podrškom, moguće je generirati certifikate te konfigurirati Web poslužitelj da se služi njima.

5.3.1. Generiranje ključeva

Web poslužitelj ne može biti pokrenut nakon što je SSL omogućen sve dok se privatni ključ i certifikat pravilno ne konfiguriraju. Prvi korak je kreiranje privatnog ključa Web poslužitelja, iz kojeg će potom biti generiran i javni ključ. Privatne ključeve je moguće zaštiti zaporkom (eng. *Passphrase*) za povećanje razine sigurnosti. Ipak, generalna preporuka kod kreiranja privatnog ključa Web poslužitelja je da se ključ ne kriptira zaporkom. Razlog tome leži u činjenici da kriptiranje privatnog ključa zahtjeva unošenje zaporke pri svakom ponovnom pokretanju Web poslužitelja, te također pruža lažni osjećaj sigurnosti. Iako je zaštita privatnog ključa zaporkom sigurnija, razlika u sigurnosti uistinu nije velika. Ukoliko je napadač uspio doći do privatnog ključa, poslužitelj je već kompromitiran, te napadač ima pristup sustavu. Napadač do zaporke može doći korištenjem *keylogger* programa, ili čak pronaći datoteku u kojoj se zaporka nalazi u jasnom obliku zbog potreba dekriptiranja podataka.
Slijedeći primjer generira nezaštićeni, 1024 bita dug ključ pomoću RSA algoritma:

```
/etc/apache2# mkdir ssl
/etc/apache2# cd ssl
/etc/apache2/ssl# openssl genrsa -out server.key 1024
Generating RSA private key, 1024 bit long modulus
.........++++++
...................++++++
e is 65537 (0x10001)
```

Privatni ključ Web poslužitelja je generiran i pohranjen u server.key datoteku:

```
/etc/apache2/ssl# ls
```

```
server.key
/etc/apache2/ssl# cat server.key
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQDLyJpqEaynvg0aW88tRvZ2cFlrFqgaUoDGEpfxxISl31CyyZo4
ogDrLsB2VjsqSRK708FtRrQy4hLXkK4Ej3hy64HCzYlnfTMpQxmQ1cPBuIAO8yPJ
Ew2maA8lj5mD2UORl4J13OV8dAQ+A4mkege0b+w2UbXWnkSQYITmtgiENQIDAQAB
AoGAExRdIME24R2yLF2cMCkMuloahSRB6PUAiYsEHua/ZKhb7eFiRxsj6uEds/au
426kCmHxIkIUbQ8tpHEvWc31AOKAHjs1BXmSTVEs6DbhL2262WU6JNm2z3BRiu1S
4k8uIMTf5Z2DiZkp9GvIwba9w3gwbIjiExPfDDk1aNgB4qECQQD8ZX+7G1bS2LxV
vEXsVK98T54Zu5CuFd8CWPinXv9e2uo+RcfLIt9QQ7cERTcwwEcJBFOBwtP/sQIU
z42Sk4LPAkEAzrFs/O7e/iNmcMa0juyjeCwjaXJ6/K331HtL/q6vHUTxUd3JNm00
ztOnIhjOxHYQCEl7n5xmKwEHG+lUktJZuwJBAOEHobVxr5x3TptmLDWEX52G4t/w
ghMkPtu59aBMK4Uny1dQIcz0P4Qxi3hMOePGWs8oQ0fbK+a5q7BgSMoZqLMCQH1e
QtQb8yfEwxLug4AZaoGujAAejxiKUMMR0iN1QwQG2DSmyK/5h0lYOYzF7UxerBTK
tXA3hgTZh71XRpQTgJcCQEwwyFMVQhNPVx5li/TlQ+B0pXS1KeCego1huJ8zPOB8
V3PybW55+iepV3Y/gRvDiWRx34Gp1jir69TVgxvwaHQ=
-----END RSA PRIVATE KEY-----
```

Javni ključ je moguće izdvojiti iz privatnog na slijedeći način:

```
/etc/apache2/ssl# openssl rsa -in server.key -pubout
writing RSA key
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDLyJpqEaynvg0aW88tRvZ2cFlr
FqgaUoDGEpfxxISl31CyyZo4ogDrLsB2VjsqSRK708FtRrQy4hLXkK4Ej3hy64HC
zYlnfTMpQxmQ1cPBuIAO8yPJEw2maA8lj5mD2UORl4J13OV8dAQ+A4mkege0b+w2
UbXWnkSQYITmtgiENQIDAQAB
-----END PUBLIC KEY-----
```

5.3.2. Generiranje zahtjeva za potpisivanje certifikata

Slijedeći korak je generiranje zahtjeva za potpisivanje certifikata, CSR (eng. *Certificate Signing Request*). Ovo je dokument kojim se od CA autoriteta zahtjeva potpisivanje, a sadrži javni ključ entiteta, u ovom slučaju Web poslužitelja, te sve relevantne podatke o podnositelju zahtjeva. Sve informacije navedene u zahtjevu postaju dio certifikata nakon potpisivanja.

Proces kreiranja zahtjeva je interaktivan, te se od korisnika očekuje da unese informacije o podnositelju zahtjeva. Ovaj proces može ubrzati dio konfiguracijske datoteke koji sadrži standardne vrijednosti pojedinih polja zahtjeva. U slučaju da se ništa ne unese u polje, biti će postavljena standardna vrijednost. Potrebno je obratiti pozornost na upute pri kreiranju zahtjeva, jer ako se polje želi ostaviti prazno potrebno je unijeti točku. Kao ulazni parametar koristi se privatni ključ, dok će zahtjev biti pohranjen u datoteku `server.csr`.

```
/etc/apache2/ssl# openssl req -new -key server.key -out server.csr
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or
a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [HR]:
State or Province Name (full name) [Hrvatska]:
Locality Name (eg, city) []:.
Organization Name (eg, company) [LSS]:
Organizational Unit Name (eg, section) []:.
Common Name (eg, YOUR name) []:test.cert.hr
Email Address []:webmaster@test.cert.hr

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Zahtjev je kreiran i nalazi se u `server.csr` datoteci, te je spreman za potpisivanje. Moguće ga je potpisati samostalno, ili zahtjev poslati u CA autoritet. Za potrebe ovog primjera zahtjev će biti vlastito potpisan.

5.3.3. Potpisivanje zahtjeva

Sve što je potrebno za potpisivanje zahtjeva su sam zahtjev i privatni ključ. Za vlastito potpisani certifikat koristi se `x509` naredba u kombinaciji s `–req` parametrom. Ulazni parametri su zahtjev i ključ kojim će biti obavljeno potpisivanje. Dodatno se još definira vrijeme trajanja certifikata, te ime datoteke u koju će certifikat biti pohranjen.

```
/etc/apache2/ssl# openssl x509 -req -days 365 -in server.csr -signkey
server.key -out server.crt
Signature ok
subject=/C=HR/ST=Hrvatska/O=LSS/CN=test.cert.hr/emailAddress=
webmaster@test.cert.hr
Getting Private key
```

Pomoću x509 naredbe moguće je i provjeriti sadržaj certifikata:

```
/etc/apache2/ssl# openssl x509 -text -in server.crt
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number:
            aa:a1:a9:e7:9e:c5:ce:71
        Signature Algorithm: md5WithRSAEncryption
        Issuer:          C=HR,          ST=Hrvatska,          O=LSS,
CN=test.cert.hr/emailAddress=webmaster@test.cert.hr
        Validity
            Not Before: Oct 13 09:51:12 2005 GMT
            Not After : Oct 13 09:51:12 2006 GMT
        Subject:          C=HR,          ST=Hrvatska,          O=LSS,
CN=test.cert.hr/emailAddress=webmaster@test.cert.hr
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:cb:c8:9a:6a:11:ac:a7:be:0d:1a:5b:cf:2d:46:
                    f6:76:70:59:6b:16:a8:1a:52:80:c6:12:97:f1:c4:
                    84:a5:df:50:b2:c9:9a:38:a2:00:eb:2e:c0:76:56:
                    3b:2a:49:12:bb:d3:c1:6d:46:b4:32:e2:12:d7:90:
                    ae:04:8f:78:72:eb:81:c2:cd:89:67:7d:33:29:43:
                    19:90:d5:c3:c1:b8:80:0e:f3:23:c9:13:0d:a6:68:
                    0f:25:8f:99:83:d9:43:91:97:82:75:dc:e5:7c:74:
                    04:3e:03:89:a4:7a:07:b4:6f:ec:36:51:b5:d6:9e:
                    44:90:60:84:e6:b6:08:84:35
                Exponent: 65537 (0x10001)
    Signature Algorithm: md5WithRSAEncryption
        97:8f:11:db:09:9a:7f:58:45:52:df:df:94:ee:99:ca:9c:48:
        fd:e7:d5:bd:49:ea:ed:bc:5b:16:ff:1a:db:44:3a:f5:0d:dc:
        05:e5:d2:b5:1b:2c:55:1d:9c:a3:ea:44:f9:aa:7f:19:f8:9e:
        e0:f7:ac:be:5c:5e:95:33:d5:6d:60:22:48:f6:59:d3:cc:d9:
        d6:1f:92:96:a4:52:31:5a:c1:3c:51:f2:be:f4:be:7a:c4:f0:
        c3:41:16:bd:3a:09:eb:35:14:84:f5:79:c3:dd:80:a1:7b:39:
        b9:a6:4a:41:98:49:8c:3a:1e:e9:1d:b6:12:f7:9d:0e:e7:9b:
        fd:99
-----BEGIN CERTIFICATE-----
MIICTzCCAbgCCQCqoannnsXOcTANBgkqhkiG9w0BAQQFADBsMQswCQYDVQQGEwJI
UjERMA8GA1UECBMISHJ2YXRza2ExDDAKBgNVBAoTA0xTUzEVMBMGA1UEAxMMdGVz
dC5jZXJ0LmhyMSUwIwYJKoZIhvcNAQkBFhZ3ZWJtYXN0ZXJAdGVzdC5jZXJ0Lmhy
MB4XDTA1MTAxMzA5NTExMloXDTA2MTAxMzA5NTExMlowbDELMAkGA1UEBhMCSFIx
ETAPBgNVBAgTCEhydmF0c2thMQwwCgYDVQQKEwNMU1MxFTATBgNVBAMTDHRlc3Qu
Y2VydC5ocjElMCMGCSqGSIb3DQEJARYWd2VibWFzdGVyQHRlc3QuY2VydC5ocjCB
nzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAy8iaahGsp74NGlvPLUb2dnBZaxao
GlKAxhKX8cSEpd9QssmaOKIA6y7AdlY7KkkSu9PBbUa0MuIS15CuBI94cuuBws2J
```

```
Z30zKUMZkNXDwbiADvMjyRMNpmgPJY+Zg9lDkZeCddzlfHQEPgOJpHoHtG/sNlG1
1p5EkGCE5rYIhDUCAwEAATANBgkqhkiG9w0BAQQFAAOBgQCXjxHbCZp/WEVS39+U
7pnKnEj959W9SertvFsW/xrbRDr1DdwF5dK1GyxVHZyj6kT5qn8Z+J7g96y+XF6V
M9VtYCJI9lnTzNnWH5KWpFIxWsE8UfK+9L56xPDDQRa9OgnrNRSE9XnD3YChezm5
pkpBmEmMOh7pHbYS950O55v9mQ==
-----END CERTIFICATE-----
```

Digitalni certifikat je potpisan i spreman za korištenje. Preostaje samo konfiguriranje Apache web poslužitelja. Preporučljivo je također ograničiti mogućnost čitanja privatnog ključa i certifikata samo od strane administratora:

```
/etc/apache2/ssl# chmod 400 server.key server.crt
```

5.3.4. Konfiguriranje SSL podrške Apache Web poslužitelja

Minimalna SSL konfiguracija Apache poslužitelja se sastoji od tri direktive u Apache konfiguracijskoj datoteci:

```
# Omoguci SSL podrsku
SSLEngine On
# Putanja do certifikata posluzitelja
SSLCertificateFile /etc/apache2/ssl/server.crt
# Putanja do privatnog kljuca posluzitelja
SSLCertificateKeyFile /etc/apache2/ssl/server.key
```

U ovisnosti o seriji Apache poslužitelja potrebno je dodatno konfigurirati i neke dodatne parametre koji se u ovom dokumentu neće detaljno obrađivati.

Osim tri navedene direktive koje su nužne, postoji još nekoliko njih koje je moguće podesiti kako bi se povećala razina sigurnosti komunikacije s Web poslužiteljem. Ovdje spadaju direktive koje definiraju podržane protokole i kriptografske algoritme. Kako je poznato da je SSL v2 nesiguran protokol preporučljivo je ukloniti podršku za ovaj protokol, te također omogućiti samo snažne kriptografske algoritme:

```
# Dozvoli SSLv3 i TLSv1, onemoguci SSLv2
SSLProtocol All -SSLv2

# Usklicnik Apche posluzitelja obavjestava da nikad ne koristi navedene
# metode. EXP se odnosi na sve algoritme s 40 i 56 bitnim kljucevima, NULL
# se odnosi na algoritme bez enkripcije, ADH predstavlja anonimnu Diffie-
# Hellman izmejnu kljuceva, dok se LOW svi algoritmi koji pruzaju nisku
# razinu sigurnosti.
SSLCipherSuite ALL:!EXP:!NULL:!ADH:!LOW
```

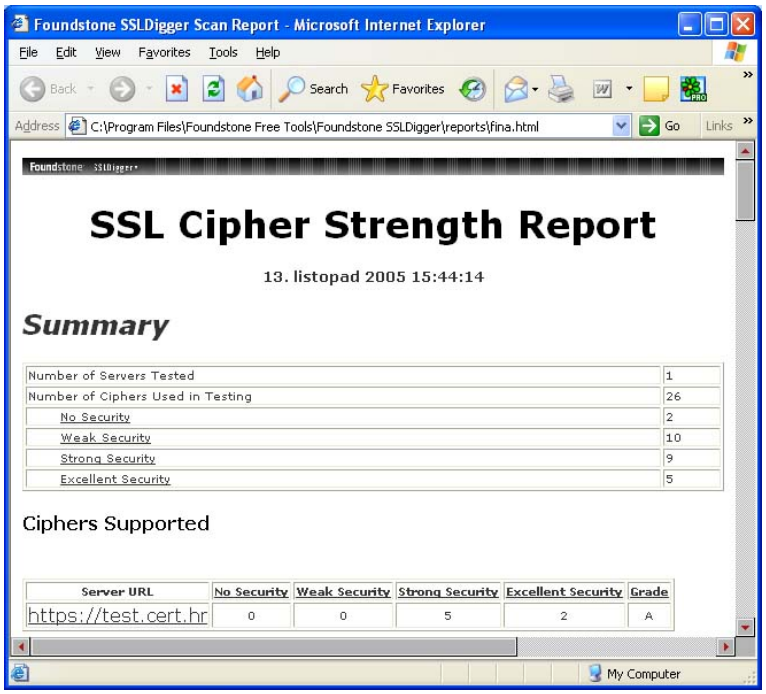
5.3.5. Provjera instalacije

Nakon instalacije digitalnog certifikata na Web poslužitelju moguće je testirati ispravnost instalacije posjetom Web stranicama koje se nalaze na poslužitelju. Prilikom pokušaja spajanja na https://test.cert.hr javlja se obavijest o postojanju certifikata, što je prikazano slikom 3.

Slika 3. Obavijest o nepravilnostima u certifikatu

U ovom slučaju, obavijest se javlja jer Web preglednik ne može prepoznati CA kao autoritet od povjerenja, te o tome obavještava korisnika. Korisnik može pogledati sadržaj certifikata i proslijediti ako smatra da je certifikat valjan. U primjeru se obavijest javlja jer je certifikat vlastito potpisan. Osim nepovjerenja prema CA autoritetu koji je potpisao certifikat, dana je i obavijest da je vremenska valjanost certifikata u redu, te da je vlasnik certifikata zaista Web poslužitelj kojem se pristupa.

Za provjeru razine sigurnosti SSL podrške moguće je iskoristiti besplatni alat za provjeru podržanih kriptografskih algoritama SSLDigger. Nešto više o ovom alatu moguće je doznati na adresi http://www.cert.hr/tools_indetail.php?lang=hr&id=292. Nakon što se izvrši testiranje, SSLDigger generira izvještaj o razini sigurnosti koju pruža Web poslužitelj, što je prikazano na slici 4.



Slika 4. SSLDigger izvještaj o podržanim algoritmima

Iz rezultata provedenog testiranja vidljivo je da poslužitelj podržava samo snažne kriptografske algoritme. Ovo je postignuto podešavanjem `SSLCipherSuite` direktive kojom su isključene sve nedovoljno sigurne metode.

6. Zaključak

Digitalni certifikati, kao osnovni koncept u PKI infrastrukturi, omogućuju automatizaciju procesa autentikacije i sigurne izmjene podataka. Certifikat sadrži podatke koji identificiraju određeni entitet, a da su navedeni podaci točni garantira autoritet koji potpisuje certifikat. Na ovaj način omogućena je autentikacija, dok se sigurni komunikacijski kanal ostvaruje korištenjem SSL protokola. Ostvarenje ovih procesa postiže se asimetričnim i simetričnim kriptografskim algoritmima, te popratnim metodama. Implementacija protokola i vezanih kriptografskih algoritama i servisa objedinjena je u OpenSSL programskom paketu.

Primjerom je demonstrirano vrlo jednostavno generiranje i potpisivanje digitalnog certifikata. Podrška je ostvarena upravo OpenSSL programskim paketom, dok je digitalni certifikat instaliran na lokalnom Web poslužitelju.

7. Reference

[1] SSL: Introduction to Secure Sockets Layer, http://www.cisco.com/en/US/netsol/ns340/ns394/ns50/ns140/networking_solutions_white_paper09186a0080136858.shtml

[2] Apache Documentation, http://httpd.apache.org/docs-project/

[3] OpenSSL homepage, http://www.openssl.org/

[4] OpenSSL certificate cookbook http://www.pseudonym.org/ssl/ssl_cook.html

[5] X.509 Certificates and Certificate Revocation Lists, http://java.sun.com/j2se/1.5.0/docs/guide/security/cert3.html

[6] Ivan Ristic, Apache Security, O'Reilly, March 2005