Odsjek: Matematika i informatika

Smjer: Informatika

UNIVERZITET U BANJOJ LUCI

PRIRODNO-MATEMATIČKI FAKULTET BANJA LUKA

SECURE SOCKET LAYER (SSL/TLS)

SEMINARSKI RAD IZ PREDMETA INFORMACIONE TEHNOLOGIJE I DRUŠTVO

Profesor:

Prof. dr Dragan Matić

Student:

Dejan Tamamović, 25/19

Asistent:

Milana Grbić, MA

Banja Luka, April 2020.

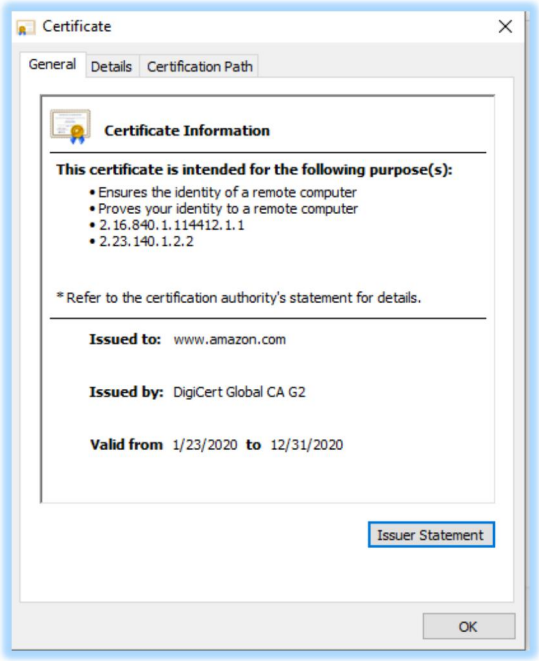
Sadržaj

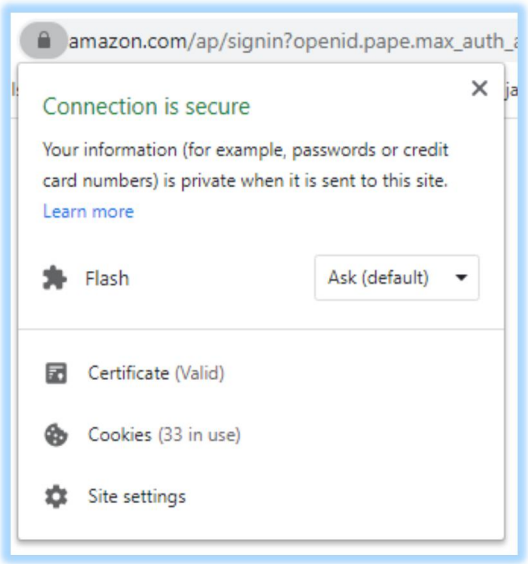
1. Šta je SSL/TLS protokol?

Protokol razvijen od strane kompanije Netscape-a 1994. godine.

Od 1999. se koristi TLS, međutim termin SSL se koristi i dalje.
Danas se koristi napredna verzija SSL-a, Transport Socket Layer dok je termin SSL ostao u upotrebi uglavnom zbog popularnosti.

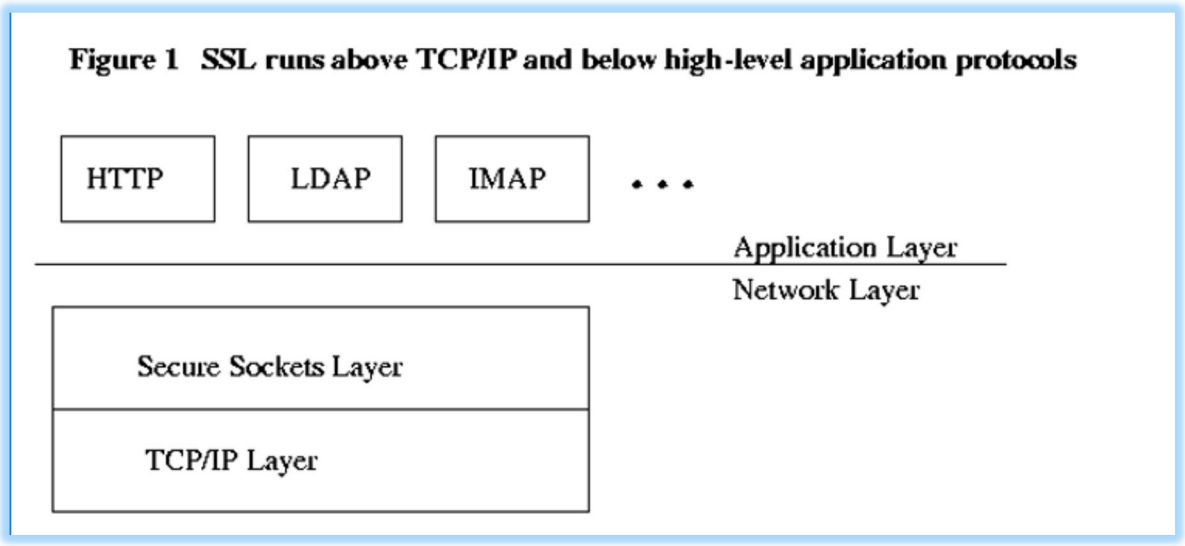
SSL predstavlja standard za osiguravanje internet konekcije i zaštitu osjetljivih podataka koji se razmjenjuju između dva sistema (najčešće veb browsera i veb servera).





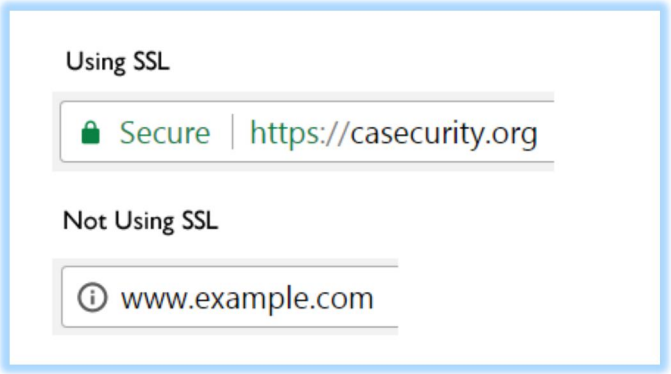
Slika 1.1 i 1.2 - Klikom na 'katanac' prije URL-a možemo pregledati detalje sertifikata HTTPS sajta.

HTTPS (Hyper Text Transfer Protocol Secure) se pojavljuje na početku URL-a kada je web stranica zaštićena SSL sertifikatom.



Slika 1.3 - SSL zauzima poseban sloj (layer), iznad TCP/IP a ispod aplikativnih protokola.
Enkripcija TCP paketa.

HTTP smatramo nesigurnim protokolom jer podaci od web browsera do web servera putuju u tekstualnom (plain-text) obliku.
Ovo znači da se osjetljivi podaci mogu presresti te pregledati (npr. podaci sa kreditnih kartice, login podaci i drugo).
Kod HTTPS-a, SSL osigurava da su pomenuti podaci enkriptovani pomoću algoritama za enkripciju te zaštićeni od 'presretača'.



Slika 1.4 - Primjer URL adrese web stranice osigurane SSL protokolom
i primjer 'nesigurne' stranice ispod.

2. Zašto nam je SSL potreban?

Broj transakcija i uopšte cjelokupna komunikaciju koja se dešava online, a kroz koju svakog časa prođe nebrojeno mnogo povjerljivih i osjetljivih korisničkih podataka, nameće se potreba za sigurnosnim kanalom komunikacije.

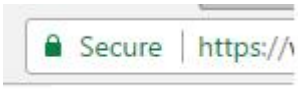
SSL promoviše sljedeće sigurnosne principe:

Enkripcija: Štiti prenos podataka (npr. browser - server, server - server, aplikacija - server).

Autentikacija: Osigurava da je server na koji se kačimo, pravi server.
(SSL, TLS handshake), digitalni sertifikati, CA.

Integritet podataka: Osigurava da su podaci koji su poslati odnosno zatraženi oni koji se i dostave.

Pored već pomenute sigurne konekcije koju omogućava, korištenjem SSL-a dajemo korisnicima naše stranice osjećaj sigurnosti, ovo je posebno važno ukoliko je naša stranica iz domena eCommerce čija su osnova upravo finansijske transakcije.
Ovo je u konačnici slučaj sa stranicama iz raznih domena, ne samo eCommerce.
(Prevencija Man-In-The-Middle napada nad plain-text podacima poslati nesigurnim HTTP protokolom).

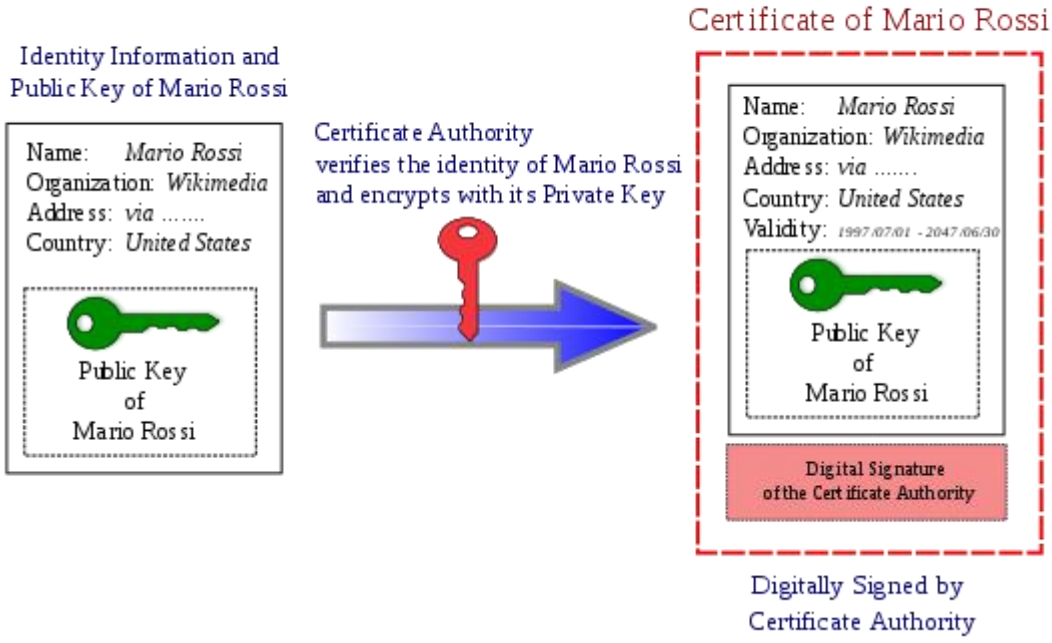


Takođe, potpisan sertifikat garantuje autentičnost internet stranice, kada korisnik posjećuje našu 'digitalno potpisanu' stranicu, third-party organizacija zvana Certificate Authority, provjerava validnost našeg sajta.

Ukratko integritet podataka se obezbjeđuje digitalni potpis poruke na izlazu pošiljaoca, Svrha *digitalnog potpisa* je da potvrdi autentičnost sadržaja poruke tj. dokaže da poruka nije promjenjena na putu od pošiljaoca do primaoca i da obezbjedi garantovanje identiteta pošiljaoca. Digitalni potpis čini sadržaj same poruke. Pošiljalac primenom kriptografskih algoritama prvo od svoje poruke koja je proizvoljne dužine stvara zapis fiksne dužine (512 ili 1024 bita) koji predstavlja sadržaj poruke. Svaka promjena u sadržaju poruke dovodi do promjene potpisa. Dakle, posiljalac kreira digitalni potpis na osnovu poruke koju zeli da posalje. Sifruje ga svojim tajnim ključem i šalje zajedno sa porukom.
Primalac dešifruje potpis posiljaoca njegovim javnim ključem, kreira potpis na osnovu poruke koju je primio i upoređuje ga sa primljenim potpisom. Ako su potpisi jednaki tj. identični, poruku je poslao pravi pošiljalac (jer je njegovim javnim ključem uspesno dešifrovao potpis) i da je ona stigla nepromjenjena (jer je utvđjeno da su potpisi identični).

Pored velike sigurnosti koju pruza ovaj metod zastite, i dalje postoji mogućnost za prevaru. Neko je mogao poslati svoj javni ključ tvrdeći da je od pravog pošiljaoca, a zatim slati poruke za koje bi primaoc mislio da ih šalje pravi pošiljaoc.
Upravo rješenje za ovaj problem nalazimo u digitalnim sertifikatima.

CA ili Certificate Authority je kompanija (organizacija) koja je zadužena za izdavanje i provjeru naše digitalne lične karte (sertifikata)
Prvo proslijedimo naš javni ključ u CA, CA kreira digitalni potpis i izdaju sertifikat koji potvrđuje da je taj ključ zaista naš te ako dalje želimo da komuniciramo sa nekim, pri prvom kontaktu mu šaljemo digitalni sertifikat i svoj javni ključ. Primalac onda lako utvrdjuje validnost naseg sertifikata.



Slika 1.5 - CA Kreira digitalni potpis i izdaje sertifikat koji potvđuje da je taj ključ zaista naš.

3. Način funkcionisanja

Objasnivši nekoliko glavnih pojmova koje susrećemo u priči o SSL protokolu ukratko ćemo vidjeti i osnove njegovog funkcionisanja.

HTTPS Enkriptuje podatke koji se prikupljaju pomoću HTTP, pomoću sigurnosnih alogoritama za enkripciju.

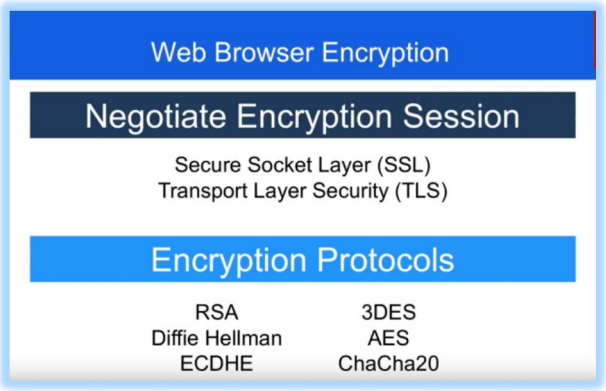
Pomoću sertifikata web sajt na koji se povezujemo se identifikuje slanjem kopije SSL sertifikata. (Autentikacija sajta koji posjećujemo).

Handshake će biti objašnjen na idućim stranicama.

CA ili Certificate Authority obezbjeđuju digitalne sertifikate organizacijama koje svojim korisnicima trebaju omogućiti sigurnu konekciju (npr. Digicert).
CA nije ništa drugo do kompanija (organizacija) koja se bavi izdavanjem digitalnih sertifikata.

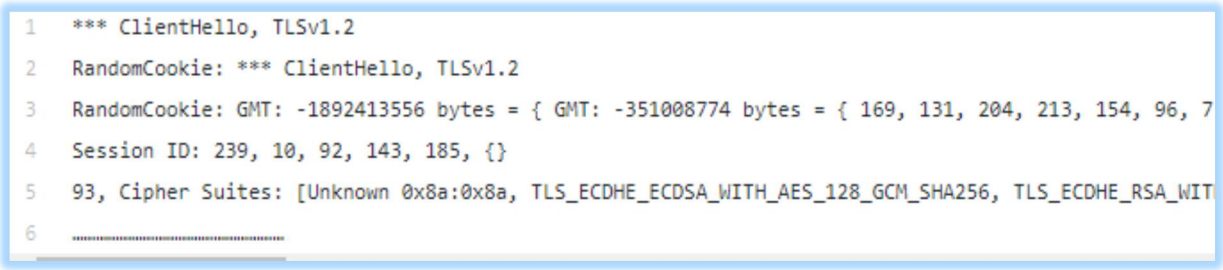
3.1 Handshake

U najkraćim crtama SSL/TLS predstavlja 'dogovor' između dva sistema, o vrsti enkripcijskog algoritma koji će se koristiti za enkripciju podataka.



Slika 2.2 - SSL/TLS kao sesija dogovora, o vrsti algoritma za enkripciju.

Tokom handshake-a, klijent i server razmjenjuju važne informacije poput verzije TLS-a, liste podržanih algoritama za enkripciju (cipher suite) iz koje će server izabrati najpoželjniji odnosno onaj koji podržavaju i klijent i server.



Slika 2.3 - Primjer informacija koje klijent šalje serveru pri uspostavljanju SSL handshake-a, korak poznat i kao 'Client Hello'.

U kratkim crtama, zamislimo da server šaljemo neki zahtjev za određenim informacijama, dešava se sljedeće:

a) Klijentsku poruku enkriptujemo odabranim kriptoalgoritmom i šaljemo serveru zajedno sa tajnim ključem ('shared key').

b) Dobijenu enkriptovanu poruku šaljemo preko javne mreže, te kada dođe do servera, on će koristeći isti algoritam i isti ključ (iz tog se razloga i zove 'shared' ili djeljeni ključ, raspakovati našu enkriptovanu poruku.
Način na koji 'dobijamo' pomenuti shared key opsan je u tekstu koji slijedi i ovaj vid enkripcije se naziva simetrična enkripcija.

Generisanje shared ključa odvija se na idući način:

Kao što već znamo podaci koji se šalju između klijenta i servera u HTTPS konekciji su enkriptovani.
Obično se zbog performansi koristi simetrična enkripcija, za koju je neophodan isti zajednički ključ na obe strane.

Postoje dva metoda za kreiranje i razmjenu zajedničkog ključa, a to su RSA i Diffie-Hellman.

Kada se konektujemo preko HTTPS konekcije, server šalje sertifikat ka CA na provjeru validnosti, ali još jedan veoma bitan dio
sertifikata su i dva broja, ova dva broja klijent i server koriste za enkripciju svojih privatnih ključeva (niko nema pristup ovom broju!).

$$g^a \text{ MOD } p = \text{encrypted key}$$

a - privatni ključ klijenta
p - jedan od brojeva iz digitalnog sertifikata
g - drugi broj iz digitalnog sertifikata

U slučaju Diffie-Hellman metoda, klijent generiše 48-bajtni broj (poznat samo klijentu/serveru), njega enkriptuje pomoću ključa iz sertifikata
o kojem je prethodno bilo riječi (poznat i klijentu i serveru), te šalje ovaj ključ ka serveru, server ovaj broj čuva, potom bira svoj 48-bajtni privatni ključ te ga istom formulom enkriptuje te šalje klijentu.

Zadnji korak je enkripcija ta dva broja na klijentu i serveru te dobijanje našeg 'shared ključa', Magija je u tome da u ovom zadnjem koraku dobijamo isti broj na obe strane, bez da iko zna za njega te ovaj ključ koristimo za dalju enkripciju komunikacije.

$$(\text{enc key})^a \text{ MOD } p = \text{key}$$

ALICE	BOB
Public Keys available = P, G	Public Keys available = P, G
Private Key Selected = a	Private Key Selected = b
Key generated = $x = G^a mod P$	Key generated = $y = G^b mod P$
Exchange of generated keys takes place	
Key received = y	key received = x
Generated Secret Key = $k_a = y^a mod P$	Generated Secret Key = $k_b = x^b mod P$
Algebraically it can be shown that $k_a = k_b$	
Users now have a symmetric secret key to encrypt	

Slika 2.4 - Handshake, Diffie-Hellman key-exchange algoritam

4. Podešavanje SSL-a

U slučaju da naš web sajt prikuplja bilo kakve osjetljive podatke, kao što su login podaci, brojevi platnih kartica i sl. tada naš sajt mora biti osiguran.
U narednim koracima biće ukratko objašnjen princip podešavanja HTTPS-a.

SSL je prilično jednostavan za podesiti, te jednom kada je podešen sve što trebamo je da 'rutiramo' korisnike da koriste HTTPS umjesto HTTP-a.
Ako prije podešavanja SSL-a pokušamo da pristupimo sajtu koristeći https:// ispred URL-a, dobićemo grešku jer nismo instalirali SSL sertifikat.

Korak 1: Hostujemo sajt sa jedinstvenom IP adresom
Sa jedinstvenom IP adresom sigurni smo da traffic usmjeren ka toj adresi dolazi isključivo do našeg sajta.
Danas postoji mnogo hosting planova koji nude takozvane dijeljene IP adrese, gdje nekolicina web sajtova koristi zajedničku lokaciju.

Korak 2: Kupovina sertifikata
Da bi osigurali da je naša web stranica zaista naša, nešto poput lične karte za naš sajt. Ovo se izvodi kupovinom SSL sertifikata, SSL sertifikati su detaljnije spomenuti u prethodnim poglavljima.
Zgodno je spomenuti da sertifikate možemo kreirati i sami, tzv. 'self-signed cert' ali većina pregledača komunicira sa Certificate Authority.

Korak 3: Aktivacija sertifikata
Ovo obično prepuštamo web host-u da uradi za nas. Ako se ipak odlučimo da ovo sami radimo, to se izvodi u web hosting kontrolnom panelu, kao što su WHM ili cPanel.

Korak 4: Instalacija sertifikata
Takođe web host može ovo da uradi za nas. Ako radimo sami, opet koristimo web hosting kontrolni panel.

Korak 5: 'Update-amo' našu stranicu da koristi HTTPS
Obično treba da zaštitimo stranice kao što su login ili cart checkout, ukoliko se radi o nekoj eCommerce stranici, nema potrebe omogućavati HTTPS na stranicama koje ne barataju sa osjetljivim podacima, čak i nije preporučljivo, jer usporava cjelukupni sajt.
Ovo obično rješavaju serverske redirekcije, kada je rječ o PHP stranicama ovo možemo podesiti u .htaccess fajlu, pomoću mod-rewrite redirekcija.

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule ^(cart/|checkout/) https://%{HTTP_HOST}%{REQUEST_URI}
```