Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Sustavi za praćenje i vođenje procesa

Seminarski rad:

SSL

(Secure Sockets Layer)

Student:
Ivica Pavlić

Nastavnik:
Dipl.ing Predrag Pale

U Zagrebu, svibanj 2005.

Sadržaj:

1. Što je SSL i čemu služi

SSL odnosno *Secure Sockets Layer* predstavlja protokol koji omogućava siguran kanal (sigurnu komunikaciju) između dva uređaja uz mogućnost identifikacije uređaja s kojim komunicirate. SSL je također metoda enkripcije podataka putem transportnog protokola poput TCP-a. Razvijen je od Netscape-a, 1994.god. SSL inačica 3.0 objavljena je 1996. godine, a njegov nasljednik TLS predstavljen je 1999.godine od IETF-a.

Siguran kanal o kojemu je ovdje riječ je *transparentan* što znači da podaci koji su poslani s jedne strane nepromijenjeni stižu do druge strane. Ova karakteristika omogućuje jednostavnu implementaciju SSL-a u postojeće standarde. To praktički znači da bi uz malu modifikaciju svaki protokol koji radi preko TCP-a *(Transmission Control Protocol)* mogao raditi i preko SSL-a.

Kao što smo rekli SSL je razvijen u Netscape-u, vodećoj tvrtki web preglednik. Netscape je trebao način sigurnog komuniciranja koji bi mogli iskoristiti u svojim aplikacijama. Prvenstveno se to odnosilo na Web, no tu su bili i mail te news servisi. Web kao najpopularniji među spomenutim servisima je prvi zahtijevao pažnju. Tipična situacija koja je zahtijevala primjenu sigurnog kanala bila je kupovina preko Interneta u kojoj broj kreditne kartice predstavlja informaciju koju treba štiti. Dakle, prvi zadatak koji je SSL trebao ispuniti je *povjerljivost*. Informacije koje se šalju između klijenta i poslužitelja trebaju ostati poznate samo njima.

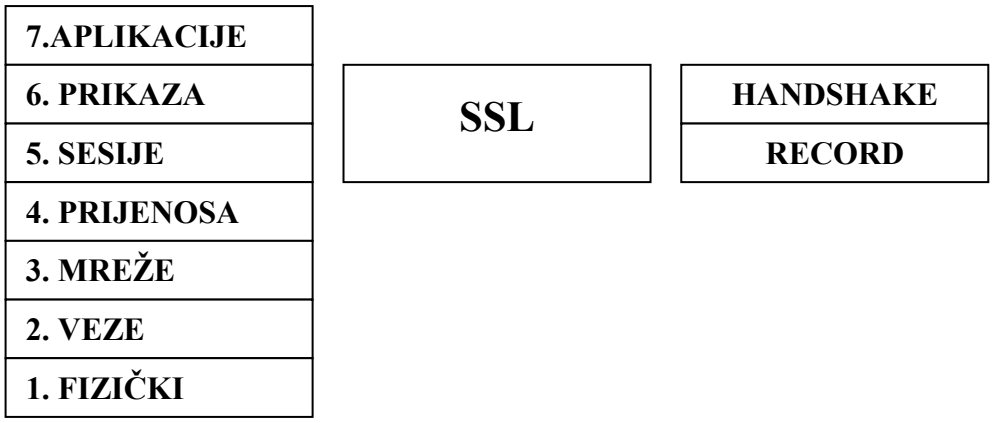
Slučaj kupovine preko Interneta otkriva i druge zadatke koje je SSL morao ispuniti. *Autentifikacija* – korisnik je morao biti siguran da komunicira s onim poslužiteljem s kojim želi ostvariti komunikaciju, a ne nekim trećim koji bi mogao zloupotrijebiti danu informaciju. Sljedeća bitna karakteristika je *spontanost*. To znači da je korisnik mogao spontano izabrati web poslužitelj s kojim želi izvršiti određenu transakciju tj. poslovati s onim s kim dosad nije imao nikakvog poslovnog iskustva. Treći cilj i zadatak je bio povezati SSL sa HTTP-om *(Hyper Text Transfer Protocol)* koji je najviše korišten na Internetu. Odatle proizlazi zahtjev za *transparentnošću* i *pouzdanošću* tog protokola. Kasnije je SSL bio proširen na druge protokole.

Novija verzija SSL-a sadržavala je ranije navedene zahtjeve, no protokol je unatoč tome bio poprilično složen. U sljedećoj inačici ispravljeni su neki sigurnosni propusti. Poboljšana je sigurnosna razmjene kriptografskih algoritama, te je proširen i broj dostupnih algoritama.

Zašto izabrati baš SSL? Spomenimo samo nekoliko podataka. Prema Forrester Research-u u 2001. godini vrijednost on-line prodaje iznosila je 41 milijardu dolara. SSL kao što smo rekli može štiti i druge protokole poput Usenet newsa, POP, SMTP, LDAP itd.

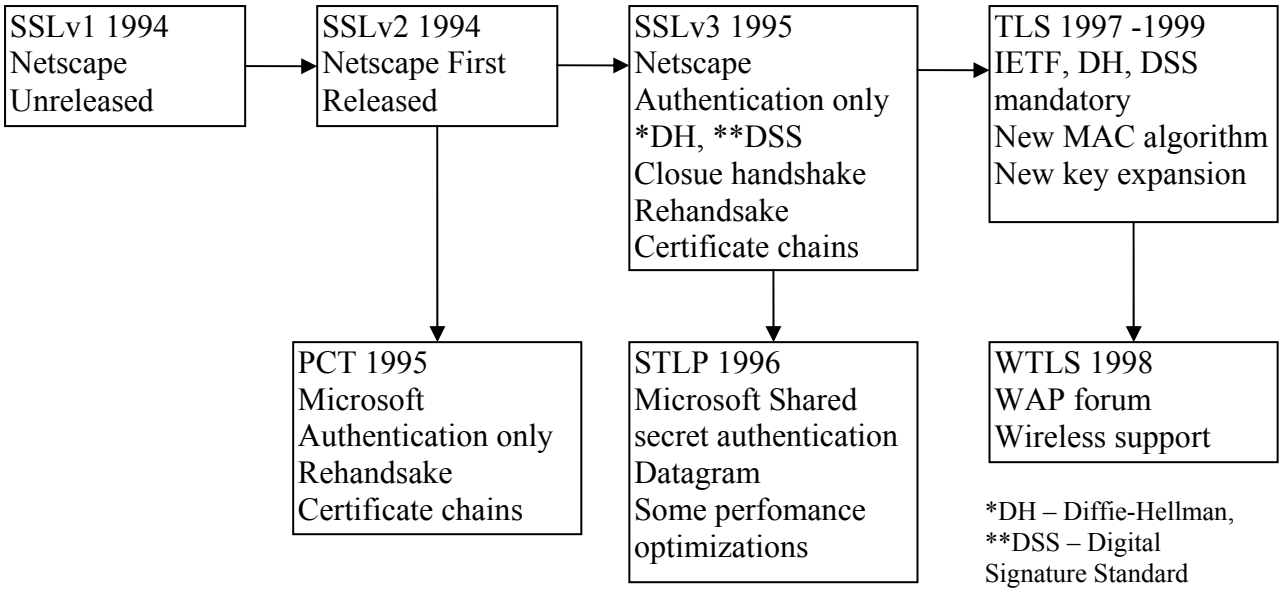
2. SSL i TCP/IP

Sve verzije SSL-a i TLS-a *(Transport Layer Security)* kao njegovog nasljednika dijele zajednički cilj uspostave sigurnog kanala komunikacije preko kojega će aplikacijski programi slati sigurno svoje podatke. Stvaratelji SSL-a uzor su imali u TCP-u s kojim SSL dijeli sličnu semantiku.



Slika 1: Položaj SSL sloja

SSL u svome radu pretpostavlja da je niži sloj pouzdan tj. da će podaci koji su poslani preko mreže sigurno stići do svog odredišta u onom obliku u kojemu su i poslani. Taj zadatak prepušta drugim protokolima poput TCP-a. U praksi se ne koristi UDP protokol koji predstavlja *connectionless (nespojni)* protokol i ne jamči istu sigurnost kao TCP. Ipak u novije vrijeme Microsoft–ov STLP *(Secure Transport Layer Protocol)* i Wireless Application Forum–ov WAP1999 su inačice SSL-a koje bi trebale korektno raditi preko UDP-a. U praksi SSL radi na dobro poznatom portu 443 *(HTTPS – SSL secured HTTP)*.



Slika 2: Razvoj SSL – a

3. TLS

Godine 1996. od strane IETF-a *(Internet Engineering Task Force)* osnovana je grupa za izradu TLS-a, nasljednika SSL-a. Zadatak te grupe bio je uskladiti Microsoft–ov i Netscape–ov SSL. Microsoft je u međuvremenu proizveo STLP *(Secure Transport Layer Protocol)* koji je bio modifikacija SSL-a i donio je neke promjene u njegovom dizajnu. STLP je imao podršku za UDP, povećao je brzinu transakcija i koristio je veći ključ pri autentifikaciji strana.

Na mnogim od sastanaka dogovoreno je da će TLS imati podršku za *Diffie Hellman* algoritam za ključeve te DSS *(Digital Signature Standard)* za autentifikaciju i *Triple DES* za enkripciju. Problem je bio u tome što je Netscape implementirao samo RSA algoritam za autentifikaciju i razmjenu ključeva. Triple DES je predstavljao i problem što se tiče zakona u SAD–u koji su zabranjivali izvoz jake kriptografije. No odlučeno je da će IETF napraviti protokol po principima dobrog i sigurnog protokola, što je uključivalo Triple DES.

U okvirima ovog seminara, kada govorim o SSL–u riječ je o verziji 3.0. Iako je razvoj SSL–a stao na toj verziji, za sam TLS se često govori da je on zapravo verzija 3.1 SSL pa kad govorim o SSL–u isto se odnosi i za TLS.

4. Struktura SSL-a

Glavna uloga SSL-a bila je zaštiti HTTP promet. SSL radi tako da se ostvari veza preko TCP–a, a potom se uspostavlja sigurna veza na višem sloju (SSL). Preko njega se odvija daljnja komunikacija.

SSL protokol sastoji se od 2 dijela:

- sloja poruka i
- sloja zapisa.

Najvažnije komponente *sloja poruka* su *faza rukovanja* te *faza slanja podatka.* U *fazi rukovanja* dolazi do autentifikacije strana pa onda i uspostave kriptirane komunikacije dok u *fazi slanja podataka* korisnički podaci putuju između aplikacija. SSL Record protocol ima zadatak da cijepka i paketira te podatke.

Prilikom početka komunikacije potrebno je web poslužitelju reći da se pripremi za SSL vezu. Postupak uspostave sigurne veze podijeljen je u dva dijela:

- *rukovanje (handshake)* i
- *prijenos podataka (data transfer).*

U fazi rukovanja dolazi do autentifikacije poslužitelja te odabira kriptografskih algoritama i ključeva kojima će se štiti kanal. Faza rukovanja mora završiti prije nego započne razmjena podataka među aplikacijama na višim slojevima. Kada je faza rukovanja gotova, podaci se dijele u sigurne blokove i takvi enkriptirani šalju kroz kanal.

4.1. Faza rukovanja

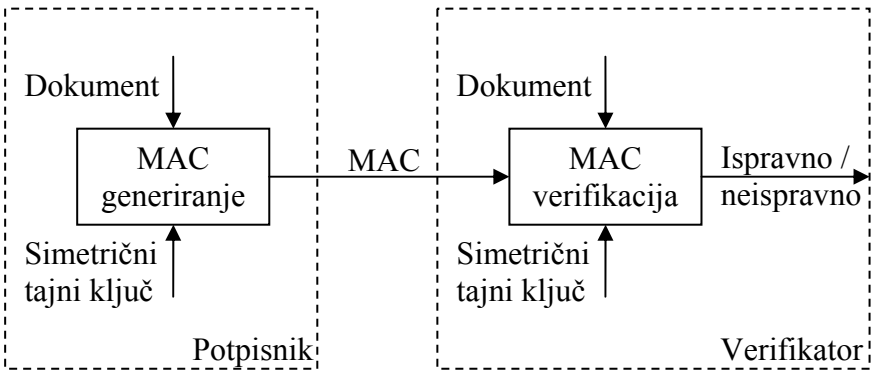
Svrha faze rukovanja je trostruka. Prvo što klijent i poslužitelj moraju učiniti je dogovor oko algoritama koji će se koristiti u komunikaciji. SSL podržava više algoritama. Potrebno je naravno da klijent i poslužitelj posjeduju iste. Drugi zadatak je odabir ključeva koji će se

koristiti u komunikaciji. Treći zadatak je autentifikacija klijenta (poslužitelja). Cijeli proces izgleda po koracima ovako:

1. Klijent šalje poslužitelju listu algoritama koje podržava, zajedno sa slučajnim (random) brojem koji će se koristi u procesu generiranja ključeva.
2. Poslužitelj izabire šifru s liste i šalje je nazad s certifikatom koji služi za njegovu autentifikaciju, a sadrži njegov javni ključ. Također šalje i slučajni broj koji služi za generiranje ključeva
3. Klijent provjerava poslužiteljev certifikat i izvlači javni ključ. Zatim, generira slučajni string tj. ključ zvan *pre_master_secret* i enkriptira ga sa poslužiteljevim javnim ključem.
4. Klijent i poslužitelj nezavisno rade enkripciju i MAC *(Message Authentication Code)* ključeve iz *pre_master_secret* i slučajno generiranih brojeva
5. Klijent šalje MAC iz svih poruka poslanih poslužitelju
6. Poslužitelj šalje MAC iz svih poruka poslanih klijentu

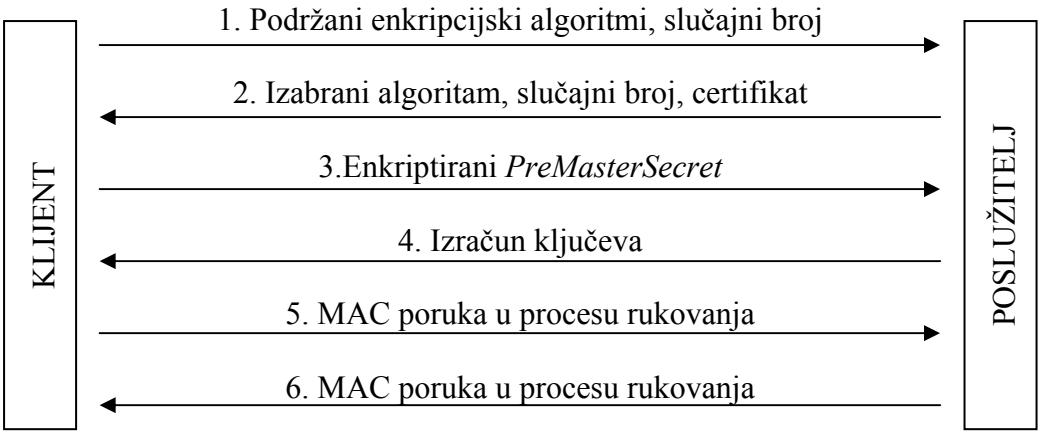
Dakle, prva dva koraka omogućuju razmjenu mogućih algoritama i razmjenu ključeva. Klijent šalje listu algoritama koje podržava, a poslužitelj izabire jednog od njih. U koracima dva i tri dolazi do razmjene ključeva, *pre_master_secret* je generiran od klijenta, enkriptiran javnim ključem od poslužitelja i poslan njemu. O trećem koraku ovisi sigurnost cijele transakcije. U trećem koraku klijent javnim ključem poslužitelja enkriptira ključ koji će dijeliti za komunikaciju. Poslužitelj svojim tajnim ključem dekriptira zajednički ključ. Ostatak procesa rukovanja je samo osiguranje da se razmjena odvija sigurno. U koracima pet i šest dolazi do razmjene MAC poruka koje služe da bi se provjerila vjerodostojnost svih poruka u procesu rukovanja, jer se MAC računa tako da se hash funkcijom djeluje na tijelo *(body)* poruke uz neke dodatne podatke kao broj sekvence, dužina poruke, itd. uz *MAC_Write_Secret*. Za hash funkciju bitno je naglasiti da je **jednosmjerna** tj. da ukoliko poznajemo ***X*** ne možemo dobiti ***v*** za funkciju ***H(v) = X***, te da NE POSTOJE ***v1*** i ***v2*** za koje vrijedi ***H(v1) = H(v2).***

MAC_Write_Secret je izveden iz *MasterSecret* poruke i za napadače je gotovo nemoguće kopirati MAC za neku poruku. MAC, dakle pruža sigurnost da poruka u komunikacijskom kanalu nije promijenjena. Funkcije koje se koriste za izračun MAC-a su MD5 i SHA1. Slika 3 pokazuje to grafički:



Slika 3: Izračun MAC-a

Gornji postupak započinjanja SSL sesije, grafički prikazuje sljedeća slika.



Slika 4: Postupak započinjanja SSL sesije

U stvarnosti proces razmjene poruka je malo detaljniji i svaki od gore navedenih koraka se sastoji on jedne ili više poruka rukovanja (*handsake messages)*. Korak jedan odgovara *ClientHello* poruci, korak dva je serija poruka koja se sastoji o *ServerHello* poruke, *Certificate* poruke u kojoj se šalje certifikat i *ServerHelloDone* kojom se završava taj korak. Treći korak odgovara poruci *ClientKeyExchange.* Koraci pet i šest odgovaraju *Finished* porukama. *CipherSuite* poruka donosi obavijest poslužitelju koje enkripcijske algoritme klijent podržava, a ista poruka sa poslužiteljeve strane odabire jedan od tih „paketa".

SSL podržava veliki niz raznih algoritama i razine sigurnosti, uključujući i algoritme koji pružaju minimalnu razinu sigurnosti ili čak nikakvu. Spomenimo da se 40 bitni ključ lako probija, pa treba voditi računa o izboru dužine ključa. Preporuka standarda je da se ne koristi anoniman *Diffie Hellman* standard za digitalne potpise, a poznato je da se RSA 512 bitni ključ koji štiti certifikat pokazao nedovoljno sigurnim. Dio podržanih „paketa" naveden je dolje:

CipherSuite	Is Exportable	Key Exchange	Cipher	Hash
SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5	*	RSA_EXPORT	RC2_CBC_40	MD5
SSL_RSA_WITH_IDEA_CBC_SHA		RSA	IDEA_CBC	SHA
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA	*	RSA_EXPORT	DES40_CBC	SHA
SSL_RSA_WITH_DES_CBC_SHA		RSA	DES_CBC	SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA		RSA	3DES_EDE_CBC	SHA
SSL_DH_DSS_EXPORT_WITH_DES40_CBC_SHA	*	DH_DSS_EXPORT	DES40_CBC	SHA
SSL_DH_DSS_WITH_DES_CBC_SHA		DH_DSS	DES_CBC	SHA
SSL_DH_DSS_WITH_3DES_EDE_CBC_SHA		DH_DSS	3DES_EDE_CBC	SHA
SSL_DH_RSA_EXPORT_WITH_DES40_CBC_SHA	*	DH_RSA_EXPORT	DES40_CBC	SHA
SSL_DH_RSA_WITH_DES_CBC_SHA		DH_RSA	DES_CBC	SHA

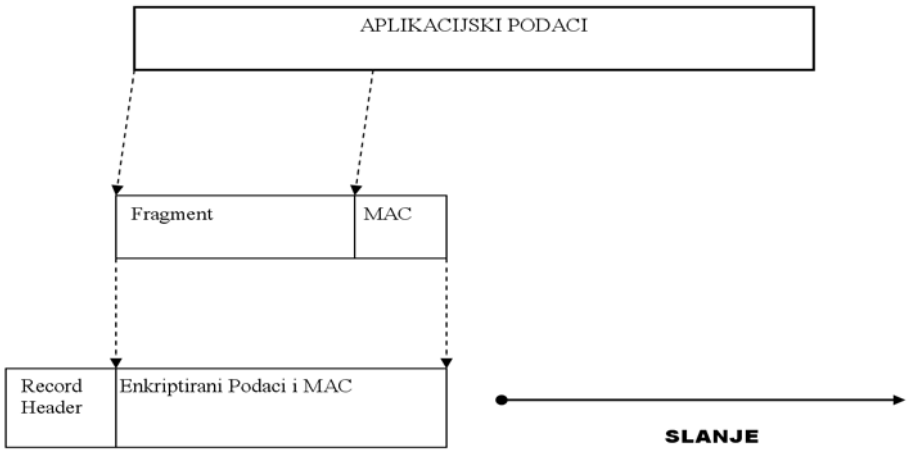
4.2. Faza prijenosa podataka

Tek nakon što je završila faza rukovanja, faza razmjene podataka može započeti. U ovoj fazi poslužitelj i klijent razmjenjuju enkriptirane podatke. Svakoj poruci dodan je MAC s kojim se provjerava ispravnost poruke. Ključevi koji se koriste pri enkripciji poruke i MAC-a dobiveni su u koraku razmjene ključeva. Protokol koji brine o razmjeni je *Record Protocol*.

4.2.1. SSL Record protokol

Kada koristimo SSL želimo imati sigurnu, vjerodostojnu i autentificiranu komunikaciju. Zadatak procesa rukovanja je opisan gore, a sam proces slanja stvarnih aplikacijskih podataka zadatak je *SSL Record* protokola.

SSL Record protokol radi tako da segmentira podatke u *fragmente* koji se svaki zasebno enkriptiraju i šalju. Na strani primatelja svaki takav fragment se posebno prima i dekriptira. Prije nego što se pošalje, fragment se mora zaštiti od napada. Da bi se osigurala autentifikacija poruka, uz fragment se šalje i njegov MAC koji se mora verificirati na strani primatelja da bi bio primljen. MAC je dodan fragmentu i takav paket se šalje enkriptiran. Tom paketu je dodan i *Record header*. Postupak je pojašnjen na slici 7.



Slika 5: Funkcija SSL Record protokola

Ovakav postupak slanja podataka često se naziva „enkapsulacija“ paketa (paketiranje). On omogućuje višim protokolima da nesmetano rade svoj posao. Na ovom mjestu možemo ponoviti zahtjeve koje takav protokol treba ostvariti. Uz opće zahtjeve koji se postavljaju pred jedan siguran sustav za SSL Record Protocol moramo navesti i

- *kriptografsku sigurnost* – temeljni zadatak koji se postavlja na sustav,
- *interoperabilnost* – nezavisni programeri bi trebali moći razvijati SSL aplikacije koje će međusobno dobro raditi,
- *proširivost* – mogućnost nadogradnje SSL-a (sprečava razvoj novih protokola – „open source“)
- *efikasnost* – racionalna upotreba dostupnih računalnih resursa

Record Header sadrži informacije o tipu podatka koji se nalazi u paketu, njegovu dužinu i SSL verziju. To je potrebno kako bi se komunikacija mogla ne smetano odvijati.

Što se tiče tipa podataka, SSL ih podržava četiri:

- podaci aplikacija *(application_data),*
- protokol za izvanredne događaje *(alert),*
- protokol za rukovanje *(handshake)* i
- protokol za promjenu načina šifriranja *(change_cipher_spec).*

Application_data su podaci aplikacija koji koriste SSL, a ostala tri protokola služe za upravljanje vezom. Tako se na primjer *alert* koristi kod signalizacije grešaka, *handshake* kod tzv. rukovanja, a *change_cipher_spec* za promjenu enkripcijski ključeva kojima se štiti veza.

Dakle, uz *Record header* jedan „SSL paket“ (ili zapis) sadrži enkriptirane podatke i MAC. MAC se izračunava primjenom sljedeće formule:

```
hash(MAC_write_secret + pad2 +
      hash(MAC_write_secret  +  pad1  +  seq_number  +  length  +
content))
```

MAC_write_secret predstavlja zajednički glavni tajni ključ, *pad1* niz ASCII znakova kojima se povećava sigurnost MAC-a, sadržaj je proizvoljan. *Pad2* predstavlja drugi niz znakova, a *seq_number* predstavlja sekvencijski broj poruke.

Protokol za izvanredne događaje sastoji se od dva dijela:

- *razine* i
- *opisa*.

Zapisani su kao 8 bitni brojevi. SSLv3.0 definira 13 različitih izvanrednih događaja. Prikazani su u tabeli 2. Postoje dvije razine *alert* protokola. Jedna je *warning* koja upozorava na grešku, ali ne prekida vezu, dok *fatal* trenutno prekida vezu..

Razina	**Ime**	**Opis**
1	Upozorenje *(Warning)*	SSL upozorenje da problem nije kritičan
2	Kritično *(Fatal)*	SSL izvanredni kritični događaji trenutno prekidaju trenutnu sjednicu

Tabela 1: Razine *Alert* protokola

Kada je otkrivena greška, strana koja je otkrila grešku šalje poruku drugoj strani. U trenutku slanja i primitka takve poruke, obje strane zatvaraju sesiju (vezu). Poslužitelj i klijent su u tom slučaju obavezni zaboraviti bilo kakve podatke o sesiji, identifikatore sesije, ključeve i ostale tajne koje su vezane uz sesiju.

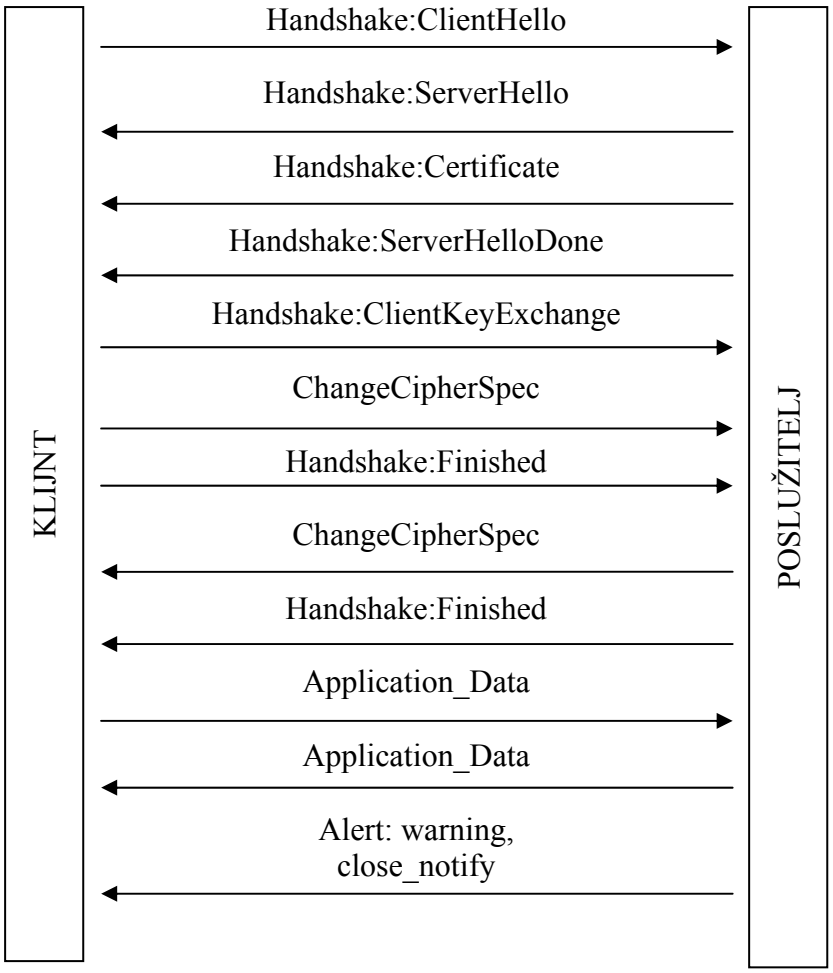
Detaljniji prikaz poruka koji je definirao SSL standard za *alert* protokol možemo pronaći u tablici 2 (13 različitih opisa izvanrednih događaja). Takvi događaji, tj. poruke se šifriraju i komprimiraju.

BROJ	IME	OPIS
0	close_notify	Indicira da pošiljatelj više ne kani slati podatke. Ukoliko je ovaj opis poslan sa upozorenjem, sjednica može biti obnovljena; ukoliko je pak poslan uz kritičnu razinu događaja, sjednica ne može biti obnovljena
10	unexpected_message	Primljena je neočekivana poruka. Ovaj opis se ne bi smio dogoditi; indicira pogrešku u jednoj os SSL implementacija (klijentskoj ili poslužiteljskoj). Kritičan
20	bad_record_mac	Pošiljatelj je primio zapis s neispravnim autentikacijskim kodom poruke (MAC). Kritičan
30	decompression_failure	Informacija u zapisu se ne može ispravno dekomprimirati. Kritičan
40	handshake_failure	Indicira da pošiljatelj nije u mogućnosti prihvatiti skup sigurnosnih parametara (npr. pošiljatelj nije zadovoljan s primateljevim algoritmima za šifriranje i njihovom snagom). Kritičan
41	no_certificate	Događa se kao odgovor na zahtjev za uvjerenjem ukoliko ne postoji odgovarajuće uvjerenje
42	bad_certificate	Događa se ukoliko je zahtjev za uvjerenjem neuspješan (uvjerenje je neispravno ili je neispravan digitalni potpis)
43	unsupported_certificate	Događa se ukoliko pošiljatelj ne podržava određeni tip uvjerenja
44	certificate_revoked	Događa se ukoliko pošiljatelj primi uvjerenje koje je već prije povučeno
45	certificate_expired	Događa se ukoliko pošiljatelj primi uvjerenje koje je isteklo
46	certificate_unknown	Događa se ukoliko se prilikom obrade uvjerenja dogodi pogreška
47	illegal_parameter	Događa se ukoliko pošiljatelj ustanovi da je neka vrijednost u protokolu za rukovanje nedozvoljene vrijednosti. Kritičan

Tabela 2: Izvanredni događaji SSLv3.0

4.2.2. SSL poruke

Detaljniji prikaz jedne SSL konekcije izgledao bi otprilike ovako:



Slika 6: Prikaz jedne SSL konekcije

ClientHello poruka sastoji se od:

- *ProtocolVersion client_version* poruke u kojoj je sadržana najnovija verzija SSL-a koju klijent podržava,
- *Random random* poruka koje su slučajna struktura za povećanje sigurnosti, sastoje se od 32-bitne vremenske značajke i 28 okteta generiranih od strane sigurnog generatora slučajnih brojeva i *SessionID session_id* koji predstavlja identifikaciju ID. Polje je prazno za novu sjednicu, a ukoliko nije, to znači da klijent pokušava obnoviti neku prethodnu sjednicu.
- *CipherSuite cipher_suites* je popis načina šifriranje koje klijent podržava

i na kraju

- *CommpressionMethod* popis načina kompresije koje klijent podržava. Kod kompresije je bitno naglasiti da zbog autorskih prava u sve verzije SSL-a nije uključena kompresija. Uz nju dolazi i 32 byte-a slučajnih podataka *(Client.random)*

ServerHello poruka sadrži sljedeće informacije:

- *ProtocolVersion client_version* - SSL verzija koju koristi klijent,
- *Random random* - slučajna struktura,
- *SessionID session_id* - sjednica identifikacija (ovo polje nikad nije prazno). Ukoliko se podudara sa *session_id* poljem iz *ClientHello* poruke to znači da će prethodna SSL sjednica biti obnovljena. Inače *session_id* sadrži ID nove sjednice.
- *CipherSuite* - način šifriranja odabran od poslužitelja za tekuću sjednicu.
- *CompressionMethod* - način kompresije odbran od poslužitelja.

Ceritificate: ukoliko poslužitelj posjeduje javni certifikat on ga prosljeđuje klijentu. Klijent također može koristiti ovu poruku da pošalje svoj certifikat poslužitelju ukoliko je to poslužitelj zatražio. Kada se to dogodi klijent šalje *CertificateVerify* poruku u kojoj enkripitra neku zadanu poruku i šalje ju poslužitelju da bi ovaj mogao provjeriti dali klijent posjeduje svoj privatni ključ. Ova se poruka sastoji od:

- imena izdatnika certifikata,
- imena entiteta za koji je taj certifikat izdan,
- javnog ključa tog entita,
- vremenske komponente koja služi kao neki „rok trajanja“ certifikata. Iako taj rok trajanja može biti beskonačan u praksi se obično iz sigurnosnih razloga on ograničava.

Ovakav postupak istovjetan je ukoliko klijent posjeduje certifikat. Važnost certifikata je u tome da je nama kao kupcu nekog proizvoda ili usluge na Internetu jako važno da smo sigurni da je poslužitelj upravo onaj za kojeg se predstavlja. Kao posrednik u ovom poslu potrebne su usluge treće strane, *Certificate Authority* ustanove. Njena uloga je u tome da „potpiše“ svojim javnim ključem certifikat koji koristi naš poslužitelj. Potpis Certificate Authority-a garantira da je certifikat našeg poslužitelja vjerodostojan. Često se koristi sljedeća jednadžba da se dočara proces:

Povjerenje = Autenitifikacija + Enkripcija + Certificate Authority.

ServerKeyExchange: poslužitelj šalje ovu poruku sam ukoliko ne posjeduje certifikat ili se ono koristi za digitalne potpise, a moguće je i da je zabranjen izvoz ključeva većih od 512 bita. U tom slučaju poslužitelj generira privremeni par javnih/tajnih ključeva i koristi ovu poruku za slanje tih ključeva.

CertificateRequest: poruka kojom poslužitelj traži od klijenta da mu pošalje svoj javni certifikat, služi za autentifikaciju klijentske strane.

CertificateVerify: služi da poslužitelj provjeri da li klijent zna tajni ključ. Ukoliko ga ne zna neće moći potpisati poruku koju će poslužitelj provjeriti s javnim ključem. U TLS-u svi certifikati su X.509 oblika.

ClientKeyExchange: služi za razmjenu ključeva između poslužitelja i klijenta, moguća su 3 oblika poruka.
Prvi je za *Diffie-Hellman*-ovu razmjenu ključeva *opaque dh_Yc<1..216-1>, signature.*
Drugi je za RSA *ServerRSAparams params, Structure signed_params*.
Treći je za Foretezza/DMS *ServerForetzzaParams params.*

ChangeCipherSpec: poruka kojom se izabire način šifriranja.

Alert:Warning close_notify: poruka kojom se zatvara sjednica.

5. Efikasnost

Faza rukovanja je skupa. Njezina skupoća proizlazi iz vremena koje je potrebno da bi se jedna SSL sesija u potpunosti obradila. Takav proces zahtjeva određeno procesorsko vrijeme. Dekripcija poruka *ClientKeyExchange* obavlja se na poslužitelju što usporava fazu rukovanja. Tvorci SSL su prilikom stvaranja postavili i zahtjev za efikasnošću protokola. Zbog toga server vezi pridjeljuje netransparentan ključ i sprema podatke u vezi. Kasnije kada se klijent ponovo spoji, može sa serverom koristiti prijašnje ključeve. Na taj način ubrzava se cijeli proces. U ovom procesu ostvaren je zahtjev koji su stvaratelji SSL imali pri definiranju zahtjeva kod faze stvaranja SSL – *efikasnost.* Informacije koje se čuvaju o svakoj sesiji su *pre_master_secret* koji se čuva u nekom vremenu.

5.1.SSL brzina

Koliko je zapravo brz ili možda spor SSL? Njegovu brzinu usporediti ćemo s simetričnim kriptografskim sustavom. Sljedeća tablica prikazuje brzinu u Mb/s

Stroj	**RC4**[1] (Mbps)	**3DES**[2] (Mbps)
Athlon 600 MHz (Linux)	541	40
Intel P III 450 MHz (Linux)	408	30
IBM RS6000 43P/140 330 MHz	192	17
Sparc Ultra 5 (Solaris 7)	176	14
SGI Indy (IRIX 6)	63	5

Tabela 3: Brzina SSL-a na različitim procesorima

Uspoređujući tu tablicu s tablicom asimetričnih kripto-sustava dolazimo do sljedećih podataka. Na drugoj tablici prikazan je broj prijava po sekundi. Vidimo da na SGI Indy stroju SSL uspijeva ostvariti tek 13 prijava. Nažalost, nedovoljno brzo za jedan ozbiljan on-line busniess (do 100 prijava po sekundi).

[1] RC4 je slijedna šifra koju je dizajnirao Rivest for RSA Data Security (danas RSA Security).Algoritam se temelji na korištenju slučajnih permutacija. Analize pokazuju da period niza bude veći od 10^{100}. Potrebno je 8 do 16 mašinskih operacija po byte-u. Zasebne analize su pažljivo istražile algoritam i on se smatra sigurnim. RC4 s ekoristi za enkripciju u proizvodima kao što su RSA SecurPC. Također se koristi za sigurnu komunikaciju u enkripciji prometa sa web stranica korištenjem SSL protokola

[2] *3DES* je mod DES enkripcijskog algoritma koji enkriptira podatke tri puta. Koriste se tri 64-bitna ključa umjesto jednog (prva enkripcija je enkriptirana drugim ključem i rezultirajući niz je ponovno enkriptiran trećim ključem).

Stroj	Operacijski sustav	Br.prijava/s
Athlon 600 MHz	Linux	100
Intel P III 450 MHz	Linux	73
Sparc Ultra 5	Solaris 7	27
IBM RS6000 43P/140 330 MHz	AIX 4.3	27
SGI Indy	IRIX 6.4	13

Tabela 4: Broj prijava po sekundi

Treba napomenuti da pri obradi ovih zahtjeva dolazi do 100% iskorištenja procesora. To znači da svi ostali servisi pokrenuti na računalu ostaju bez resursa. Načini na koji bi mogli ubrzati proces SSL prijava je da eventualno kupimo brži stroj, koji će koristi brži procesor ili čak više njih. Ubrzanje je moguće ukoliko koristimo ***HTTP1.1/Keepalive*** opciju u web serveru, te ukoliko klijent i server spremaju sesiju u cache.

Dizajneri SSL-a sami su naglasili da SSL **ne može** spriječiti česte sigurnosne propuste. Naveli su i nekoliko tih propusta koje treba imati na umu prilikom uvođenja SSL.

Prvi savjet je da su američki izvozni zakoni zabranjivali izvoz jake enkripcije. Svi RSA ključevi iznad 512 bita bili su zabranjeni. No taj zakon se **nije odnosio** na RSA ključeve koji su se koristili za operacije potpisivanja (gdje treba koristiti veće ključeve jer 512 bita nije dovoljno sigurno). U nemogućnosti izbora većeg ključa preporuka je što češće mijenjati te ključeve.

Drugi savjet je da se obrati pozornost prilikom odabira funkcije za kreiranje slučajnog broja. Treba koristiti provjerenu metodu koja će zaista nuditi nepredvidivi niz bitova.

Treći savjet odnosi se na pitanje certifikata, *Certificate Authority* treba izabrati s pažnjom i mora imati mogućnost revoke-a certifikata.

Fortezza, što na talijanskom jeziku znači *tvrđava* je enkripcijski algoritam koji je razvila američka agencija pod imenom NSA *(National Security Agency)*, a popularan je u državnim krugovima i vojsci. Koristi se u mobilnoj tefoniji. Popularan je i zbog dostupnosti hardvera koji obavlja dekriptiranje *Fortezza* algoritama. Naravno, takav hardver je dobrodošao jer značajno može ubrzati zahtjevan proces SSL komunikacije. Ipak treba imati na umu da ga je razvila NSA, a onda izraziti i sumnju u „sigurnost“ takvog algoritma jer njegova struktura je tajna.

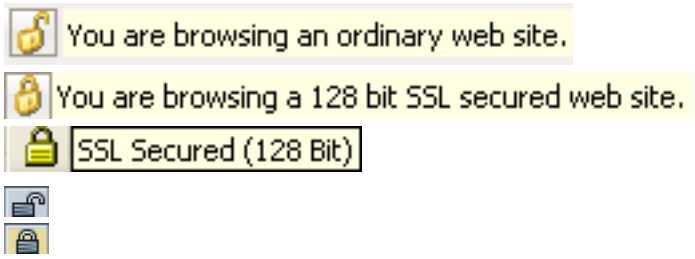
Za izbjegavanje „*man in the middle“* tipa napada, tvorci predlažu korištenje certifikata. Prilikom faze rukovanja server je obavezan poslati svoj certifikat koji je potpisan od *Certificate Authority*-a jer to garantira autentičnost.

6. Primjena SSL-a

SSL je stvoren prvenstveno kao odgovor na sverastuće potrebe za zaštićenim prijenosom podataka putem Interneta. SSL je gotovo neophodan za sigurnost e–trgovine. Unatoč postojanju i drugih rješenja sigurnosti na Internetu kao što su: *Secure-MIME protokol, Secure HTTP, Secure Shell, Private Enhanced Mail, MIME Object Security Services, Private Communication Technology,* SSL se je pokazao najboljim rješenjem za osiguravanje elektroničkih transakcija. Daljnji uspjeh elektroničkog poslovanja uvelike ovisi o potrošačevom pojerenju u funkcioniranje SSL–a. No, kao i svi ostali sustavi, sigurnost SSL–a ovisi o svim karikama u lancu koji ga čine: klijentu, serveru te izdavatelj potvrda. Svi ovi sudionici trebaju brinuti o sigurnosti i redovito održavati svoje sustave kako bi se ostavila što manja mogućnost ometanja komunikacije i krađe, kako informacija tako i novca.

U budućnosti, SSL će moći obavljati mnogo više transakcija u kraćem vremenu. Duljine ključeva će rasti, algoritmi će biti uspješniji, sve u cilju razvoja sigurnosti informacija na Internetu.

Pristupanjem stranicama zaštićenim SSL-om, mijenja se i izgled preglednika (Browser). Iako je ikonografija različita od preglednika do preglednika, svaki od njih daje do znanja korisniku da li se nalazi na običnoj stranici ili na zaštićenoj.



7. Zaključak

Globalna ekspanzija Interneta i elektroničkog poslovanja utjecala je na pitanje same sigurnosti takve trgovine i obratno. Internet kao medij ne pruža mogućnosti zaštite podataka koji se publiciraju na njega. Otvorenost Interneta je velika prednost što se tiče njegovog razvoja, ali takva otvorenost nije poželjna sa sigurnosnog stajališta. Upravo zbog toga, potrebne su brojne tehnologije zaštite podataka i njihovog transfera putem Interneta. No, ni uz sve postojeće metode zaštite, elektroničko poslovanje nije u potpunosti sigurno. Stalno poboljšavanje tih metoda i tehnika zaštite iz dana u dan poboljšavaju tu sliku, ali je i sve je veći broj napada na računalne sustave. Dijeljenje informacija putem mreže, bila ona Internet ili neka druga, postala je neophodna u današnjem svijetu informacija gdje svatko želi što prije doći do konkurentske prednosti nad drugima. Takav način obavještavanja također postavlja imperativ sigurnosti nad informacijama. Vladine organizacije, obavještajne agencije, međunarodne korporacije, interesne skupine, terorističke organizacije, mafija i drugi, svi žele razmjenjivati informacije bez znanja drugih. Kontrola nad dužinama kriptografskih ključeva postala je važna i za nacionalnu sigurnost što su Sjedinjene Američke Države već počele nadzirati. Dakle, osigurati tajnost, ali ograničenu.

Sigurnost postaje sve veći posao. Kompanije koje se bave proizvodnjom sigurnosnog softvera imaju prosječan rast od tridesetak posto godišnje i ukupan promet od 13 milijardi dolara u 2004. godini. Antivirusna zaštita, digitalni certifikati, vatrozidi postaju neophodni za normalno odvijanje poslovnih procesa. Brojke su mjerilo, a moderno informatičko poslovanje ne trpi kompromise.

I na kraju uvijek postoji veliko ali. Unatoč svim raspoloživim sigurnosnim mjerama i uloženim naporima u kvalitetnu sigurnosnu politiku, cijeli sustav će dobro funkcionirati ili pasti na ljudima. Sigurnost je posao s punim radnim vremenom, a ne povremeni posao koji se obavlja tek kad se nešto dogodi jer u najvećem broju slučajeva je tada već kasno. Ako se ljudi zaduženi za održavanje i brigu za sigurnosnu infrastrukturu olako shvaćaju sigurnosnu politiku poduzeća ili neke druge organizacije, ne postoji niti jedan zaštitni sustav, softversko ili hardversko rješenje, ili konzultantska usluga koja može pomoći. No, to je ipak tema izvan dohvata ovog rada te zahtjeva pristup s drugačijeg stajališta.

8. Literatura

1. Rescorla, Eric : **SSL and TLS : Designing and building secure systems**, Addison – Wesley, Boston, 2001.
2. Panian, Ž. : **Izazovi elektroničkog poslovanja**, Narodne novine, Zagreb, 2002.

Internet izvori:

1. http://www.tf.zr.ac.yu Ivković, M : Zaštita računarskih mreža i sigurnost podataka
2. http://www.zemris.fer.hr Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave Fakulteta elektrotehnike i računarstva
3. http://www.entrust.com Entrust inc.
4. http://www.americanexpress.hr PBZ American Express d.o.o.