

SSL protokol

Za zaštitu komunikacije između klijenta i servera npr. *web* klijenta (*web* pretraživača) i *web* servera, najčešće se koristi SSL (*Secure Sockets Layer*) protokol, odnosno njegova standardizovana verzija TLS (*Transport Layer Security*). SSL protokol nastao je u Netscapu, ali je standardizovan u IETF-u pod nazivom Transport Layer Security (TLS). Naziv SSL se zadržao, pa ćemo ga i mi češće koristiti. SSL može biti ugrađen u softverski paket (npr. Microsoft Explorer pretraživač dolazi sa ugrađenim SSL protokolom i većina *web* servera ima implementiran protokol). Alternativno SSL/TLS može biti instaliran kao deo TCP/IP protokol steka i tako transparentan za aplikativni nivo.

SSL (*Secure Sockets Layer*) protokol funkcioniše po klijent-server modelu. Klijent je strana koja inicira sigurnu komunikaciju, dok server odgovara na zahtev klijenta. Najčešći primer korišćenja SSL protokola je *https* koji predstavlja sigurnu verziju *http*-a (*secure http*) i koristi se za uspostavu zaštićene veze sa nekim *web* serverom, najčešće za potrebe elektronskog plaćanja. U tom slučaju, *web* pretraživač predstavlja SSL klijenta, a *web* server, odnosno sajt kome se pristupa, je SSL server.

Sa stanovišta SSL protokola, ono što pravi razliku između klijenta i servera su akcije koje oni preduzimaju prilikom pregovora oko sigurnosnih parametara. Klijent inicira komunikaciju, porukom u kojoj se nalazi i njegov predloženi skup SSL opcija koje će se koristiti za uspostavu zaštićenog kanala. Server, na osnovu onoga što je klijent ponudio, bira skup opcija koje će se koristiti. Iako je konačna odluka na serveru, server može da bira samo iz skupa parametara koje je klijent inicijalno predložio.

B.1 SSL poruke

Najkompleksniji deo SSL protokola je deo oko započinjanja i dogovora o parametrima komunikacije između klijenta i servera, *SSL Handshake*. On omogućava serveru i klijentu da međusobno autentifikuju jedan drugog, da izaberu algoritam za šifrovanje, MAC algoritam i da razmene kriptografske ključeve koji će se koristiti za zaštitu podataka u SSL zapisu.

SSL Handshake se sastoji iz tri pod-protokola: *SSL Handshake*, *SSL Change Cipher Spec* i *SSL Alert*.

Poruke koje se razmenjuju prilikom *SSL Handshake* su sledeće:

Alert - Obaveštava drugu stranu o mogućoj sigurnosnoj pretnji ili o prekidu komunikacije.

ApplicationData - Sami korisni podaci koje dve strane razmenjuju, šifrovani, autentifikovani i/ili verifikovani od strane SSL-a.

Certificate - Poruka koja nosi digitalni sertifikat pošiljaoca (koji sadrži i njegov javni ključ).

CertificateRequest - Zahtev koji server šalje klijentu tražeći mu da pošalje svoj sertifikat.

CertificateVerify - Poruka koju šalje klijent kako bi dokazao da poseduje privatni ključ koji odgovara javnom ključu koji se nalazi u njegovom sertifikatu.

ChangeCipherSpec - Poruka kojom se označava početak korišćenja sigurne komunikacije sa prethodno dogovorenim parametrima.

ClientHello - Poruka koju šalje klijent, u kojoj navodi listu sigurnosnih parametara koje podržava i želi da koristi za uspostavu sigurne komunikacije.

ClientKeyExchange - Poruka od klijenta koja sadrži kriptografske ključeve za uspostavu sigurne komunikacije.

Finished - Potvrda da je inicijalno pregovaranje završeno i da je uspostavljena sigurna komunikacija.

HelloRequest - Zahtev od servera da klijent ponovo (ili restartuje) SSL proces za dogovor oko parametara koji će se koristiti.

ServerHello - Poruka od servera koja specificira sigurnosne servise koji će se koristiti u komunikaciji.

ServerHelloDone - Potvrda koju šalje server kako bi potvrdio klijentu da je završio sa slanjem svih zahteva klijentu za uspostavu sigurne komunikacije.

ServerKeyExchange - Poruka od servera koja sadrži kriptografske ključeve za uspostavu sigurne komunikacije.

B.2 Uspostavljanje zaštićene komunikacije

Osnovna funkcija koju SSL klijent i server mogu da urade je uspostavljanje kanala po kome se odvija sigurna komunikacija.

Na slici B.2.1 je prikazana razmena poruka između SSL klijenta i servera koja prethodi uspostavljanju zaštićene komunikacije, a u tabeli B.2.1 je dato objašnjenje odgovarajućih koraka u razmeni.



Slika B.2.1 SSL koristi 9 poruka prilikom uspostavljanja zaštićene veze

Razmena poruka prilikom uspostavljanja zaštićene veze

Poruka	Akcija
1	Klijent šalje <i>ClientHello</i> poruku sa predlozima za izbor SSL parametara
2	Server odgovara <i>ServerHello</i> porukom sa SSL parametrima koje je izabrao
3	Server šalje informaciju o svom javnom ključu u <i>ServerKeyExchange</i> poruci
4	Server zaključuje svoj deo pregovora <i>ServerHelloDone</i> porukom
5	Klijent šalje ključ koji će se koristiti tokom sesije (kriptovan javnim ključem servera) u <i>ClientKeyExchange</i> poruci
6	Klijent šalje <i>ChangeCipherSpec</i> poruku kojom aktivira korišćenje zaštićene komunikacije sa dogovorenim parametrima za sve buduće poruke
7	Klijent šalje <i>Finished</i> poruku kako bi server proverio novoaktivirane sigurnosne opcije
8	Server šalje <i>ChangeCipherSpec</i> poruku kojom aktivira korišćenje zaštićene komunikacije sa dogovorenim parametrima za sve buduće poruke
9	Server šalje <i>Finished</i> poruku kako bi klijent proverio novoaktivirane sigurnosne opcije

Tabela B.2.1 Razmena SSL poruka tokom uspostavljanja sigurnog kanala komunikacije

B.3 Autentifikacija servera

SSL sadrži mehanizme koji omogućavaju svakoj strani da autentifikuje drugu stranu u komunikaciji.

Autentifikaciju servera, klijent može zahtevati tokom uspostavljanja veze. Na ovaj način, korisnik proverava identitet servera sa kojim komunicira, da nije došlo do krađe identiteta, odnosno, lažnog predstavljanja od strane nekog napadača.

Kada klijent zahteva autentifikaciju, server mu odgovara *Certificate* porukom umesto porukom *ServerKeyExchange*. *Certificate* poruka sadrži lanac sertifikata koji počinje sertifikatom samog servera, a završava se sertifikatom korenog sertifikacionog tela (root CA). Klijent je dužan da proveri da li može da veruje sertifikatu koji dobije od servera. To uključuje proveru lanca poverenja i provoru validnosti sertifikata.

Nakon što je utvrdio identitet servera, klijent nastavlja proceduru opisanu u predhodnom poglavlju. Naravno ključ K, koji klijent šalje serveru je šifrovan javnim ključem servera koji se nalazi u sertifikatu dobijenom od servera. Klijent je siguran da samo server koji poseduje odgovarajući privatni ključ, može da dešifruje poruku klijenta i uspešno nastavi komunikaciju.

Da bi opisani scenario funkcionisao, server mora da ima svoj digitalni sertifikat (koji nazivamo serverski SSL sertifikat) i on mora biti instaliran na serveru.

Na slici B.3.1 je prikazana razmena poruke tokom autentifikacije servera na zahtev klijenta, dok je u tabeli B.3.1 je dato objašnjenje za odgovarajuće korake u razmeni.



Slika B.3.1 Dve dodatne SSL poruke autentifikuju server

Razmena poruka pri autentifikaciji servera	
Poruka	Akcija
1	Klijent šalje <i>ClientHello</i> poruku sa predlozima za izbor SSL parametara
2	Server odgovara <i>ServerHello</i> porukom sa SSL parametrima koje je izabrao
3	Server šalje <i>Certificate</i> poruku koja sadrži sertifikat servera
4	Server zaključuje svoj deo pregovora <i>ServerHelloDone</i> porukom
5	Klijent šalje ključ koji će se koristiti tokom sesije (kriptovan javnim ključem servera koji se nalazi u sertifikatu dobijenom od servera) u <i>ClientKeyExchange</i> poruci
6	Klijent šalje <i>ChangeCipherSpec</i> poruku kojom aktivira korišćenje zaštićene komunikacije sa dogovorenim parametrima za sve buduće poruke
7	Klijent šalje <i>Finished</i> poruku kako bi server proverio novoaktivirane sigurnosne opcije
8	Server šalje <i>ChangeCipherSpec</i> poruku kojom aktivira korišćenje zaštićene komunikacije sa dogovorenim parametrima za sve buduće poruke
9	Server šalje <i>Finished</i> poruku kako bi klijent proverio novoaktivirane sigurnosne opcije

Tabela B.3.1 Razmena SSL poruka pri autentifikaciji servera

B.4 Razdvajanje enkripcije od autentifikacije

Nedostatak postupka autentifikacije iz predhodnom poglavlja, je u tome što se isti javni ključ servera koristi za potvrdu identiteta servera i za enkripciju ključa K koji će se koristiti za šifrovanje sadržaja komunikacije tokom trajanja sesije. U nekim slučajevima ni ne postoji odgovarajuća podrška za sprovođenje opisanog postupka, budući da se neki sigurnosni algoritmi (npr. DSA - *Digital Signature Algorithm*) mogu koristiti samo za digitalno potpisivanje poruke, ali ne i za kriptovanje. U takvoj situaciji, nije izvodljivo da poruka sa ključem K bude šifrovana javnim ključem servera koji se nalazi u sertifikatu kojim je dokazan njegov identitet.

Dakle, potrebno je razdvojiti enkripciju od autentifikacije, pa je server dužan da odgovori i *Certificate* porukom i *ServerKeyExchange* porukom.

Certificate poruka sadrži sertifikat servera. Javni ključ servera koji se nalazi u njegovom sertifikatu koristi se samo kako bi se potvrdio identitet servera, odnosno, izvršila njegova autentifikacija.

ServerKeyExchange poruku sadrži drugi javni ključ servera, koji klijent treba da koristi za enkripciju informacije o ključevima K koji će se koristiti tokom sesije. Razlika je u tome što je informacija o drugom javnom ključu servera sada može biti potpisana privatnim ključem servera, čiji je odgovor javni ključ predhodno poslat sa njegovim sertifikatom. Tako klijent može da potvrdi da server zaista poseduje privatni ključ koji odgovara javnom ključu sadržanom u sertifikatu servera.

Na slici B.4.1 je prikazana razmena poruka koja odgovara objašnjenoj situaciji.



Slika B.4.1 Tri dodatne SSL poruke odvajaju autentifikaciju od enkripcije

U tabeli B.4.1 su objašnjeni svi koraci u razmeni poruka.

Razmena poruka pri razdvajanju autentifikacije od enkripcije	
Poruka	Akcija
1	Klijent šalje <i>ClientHello</i> poruku sa predlozima za izbor SSL parametara
2	Server odgovara <i>ServerHello</i> porukom sa SSL parametrima koje je izabrao
3	Server šalje <i>Certificate</i> poruku koja sadrži sertifikat servera
4	Server šalje javi ključ koji će klijent da koristi za enkripciju u <i>ServerKeyExchange</i> poruci; ova poruka je potpisana privatnim ključem servera
5	Server zaključuje svoj deo pregovora <i>ServerHelloDone</i> porukom
6	Klijent šalje ključ koji će se koristiti tokom sesije (kriptovan javnim ključem servera dobijenim u <i>ServerKeyExchange</i> poruci) u <i>ClientKeyExchange</i> poruci
7	Klijent šalje <i>ChangeCipherSpec</i> poruku kojom aktivira korišćenje zaštićene komunikacije sa dogovorenim parametrima za sve buduće poruke
8	Klijent šalje <i>Finished</i> poruku kako bi server proverio novoaktivirane sigurnosne opcije
9	Server šalje <i>ChangeCipherSpec</i> poruku kojom aktivira korišćenje zaštićene komunikacije sa dogovorenim parametrima za sve buduće poruke
10	Server šalje <i>Finished</i> poruku kako bi klijent proverio novoaktivirane sigurnosne opcije

Tabela B.4.1 Razmena SSL poruka pri razdvajanju autentifikacije od enkripcije

B.5 Autentifikacija klijenta

Server takođe može da zahteva autentifikaciju klijenta, tokom razmene inicijalnih *Hello* poruka između klijenta i servera. SSL specificira da server ne može da traži autentifikaciju klijenta ukoliko se on prethodno nije autentifikovao klijentu.

Napomena: SSL koristi javni ključ klijenta samo za digitalno potpisivanje, odnosno autentifikaciju klijenta. Za razliku od slučaja sa serverom, ne postoji potreba za enkripcijom korišćenjem javnog ključa klijenta.

Na slici B.5.1 je prikazana odgovarajuća razmena poruka, a u tabeli B.5.1 su objašnjeni odgovarajući koraci.



Slika B.5.1 Tri dodatne SSL poruke autentifikuju klijenta

Razmena poruka pri autentifikaciji klijenta	
Poruka	Akcija
1	Klijent šalje <i>ClientHello</i> poruku sa predlozima za izbor SSL parametara
2	Server odgovara <i>ServerHello</i> porukom sa SSL parametrima koje je izabrao
3	Server šalje <i>Certificate</i> poruku koja sadrži sertifikat servera
4	Server šalje <i>CertificateRequest</i> poruku kojom traži da autentifikuje klijenta
5	Server zaključuje svoj deo pregovora <i>ServerHelloDone</i> porukom
6	Klijent šalje <i>Certificate</i> poruku koja sadrži sertifikat klijenta
7	Klijent šalje ključ koji će se koristiti tokom sesije (kriptovan javnim ključem servera) u <i>ClientKeyExchange</i> poruci
8	Klijent šalje <i>CertificateVerify</i> poruku koja sadrži važne informacije o sesiji potpisane privatnim ključem klijenta; server koristi javni ključ klijenta iz sertifikata klijenta, kako bi mogao da potvrdi identitet klijenta
9	Klijent šalje <i>ChangeCipherSpec</i> poruku kojom aktivira korišćenje zaštićene komunikacije sa dogovorenim parametrima za sve buduće poruke
10	Klijent šalje <i>Finished</i> poruku kako bi server proverio novoaktivirane sigurnosne opcije
11	Server šalje <i>ChangeCipherSpec</i> poruku kojom aktivira korišćenje zaštićene komunikacije sa dogovorenim parametrima za sve buduće poruke
12	Server šalje <i>Finished</i> poruku kako bi klijent proverio novoaktivirane sigurnosne opcije

Tabela B.5.1 Razmena SSL poruka pri autentifikaciji klijenta

Samo slanje *Certificate* poruke od strane klijenta ne obezbeđuje potpunu autentifikaciju klijenta. Klijent mora da dokaže da poseduje i odgovarajući privatni ključ. Zato, klijent šalje *CertificateVerify* poruku koja sadrži digitalno potpisan heš dobijen od informacije koja je poznata i klijentu i serveru. Na taj način server može da proveriti potpis i utvrdi da li klijent poseduje odgovarajući privatni ključ.

Podrazumeva se da je server dužan da proveriti da li može da veruje sertifikatu koji dobije od klijenta. To uključuje proveru lanca poverenja i provore validnosti sertifikata.

U navedenom primeru SSL klijent je *web* pretraživač, a digitalni sertifikat koji mu je potreban da bi opisani scenario funkcionisao je lični korisnički sertifikat.

Prednosti korišćenja PKI za autentifikaciju klijenta (krajnjih korisnika) su:

- Osetljive informacije (*password* ili privatni ključ) ne prenose se kroz mrežu.
- Nije potreban repozitorijum za korisničke akreditiv (*credential*) - postoji CA kome svi veruju.
- Autentifikacija nije centralizovan servis.

B.6 Kreiranje i format SSL poruke

SSL protokol se oslanja na TCP protokol preko svog *Record* protokola. *Record* protokol preuzima poruke aplikacije i poruke jednog od tri SSL *Handshake* pod-protokola (slika B.6.1), formatira ih, vrši odgovarajuću enkapsulaciju i prosleđuje ih transportnom sloju



Slika B.6.1 Komponente SSL protokola

B.6.1 SSL i transportni sloj

Pošto zahteva puzdan prenos, bez grešaka, SSL koristi TCP na transportnom sloju. Omogućeno je kombinovanje više SSL poruka u okviru jednog TCP segmenta.



Slika B.6.1.1 Kombinovanje više SSL poruka u jedan TCP segmentu

B.6.2 Record sloj SSL protokola

Record sloj SSL protokola vrši enkapsulaciju u format prepoznatljiv za sve SSL poruke (*Alert*, *ChangeCipherSpec*, *Handshake* i poruke aplikacije). Na slici B.6.2.1 je prikazan format poruke, a u tabeli B.6.2.1 su navedena objašnjenja za odgovarajuća imena polja.



Slika B.6.2.1 Record sloj SSL protokola je zadužen za enkapsulaciju poruke

Polje	Veličina (B)	Namena
<i>Protocol</i>	1	Označava da je poruka deo jednog od tri <i>handshake</i> pod-protokola ili aplikacije. Moguće su sledeće vrednosti: 20 - <i>ChangeCipherSpec</i> ; 21 - <i>Alert</i> ; 22 - <i>Handshake</i> ; 23 - Aplikacija
<i>Version</i>	2	Verzija SSL protokola. Trenutna verzija je 3.0, a TLS koristi verziju 3.1
<i>Length</i>	2	Dužina poruke preuzete od protokola na sloju iznad Record sloja SSLa.
<i>Protocol Messages</i>	n	Poruka

Tabela B.6.2.1 Polja SSL Record Layer poruke

***ChangeCipherSpec* protokol:** je jednostavan protokol koji ima samo jednu poruku, *ChangeCipherSpec*, koja je već pomenuta i objašnjena.

Alert protokol: se koristi da bi signalizirali grešku ili upozorenje drugoj strani uključenoj u komunikaciju.

Handshake protokol: je zadužen za razmenu poruka tokom uspostavljanja i dogovaranja o parametrima SSL sesije. Pripadaju mu poruke *HelloRequest*, *ClientHello*, *ServerHello*, *Certificate*, *ServerKeyExchange*, *CertificateRequest*, *ServerHelloDone*, *CertificateVerify*, *ClientKeyExchange*, *Finished*, koje je značenje opisano na početku u poglavlju B.1 SSL poruke.

From:

<http://www.bpd.amres.ac.rs/> - AMRES wiki

Permanent link:

http://www.bpd.amres.ac.rs/doku.php?id=amres_cbp_wiki:bpd_106_dodatakb_ssl

Last update: 2011/05/15 20:49