CARNet
HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

TLS protokol

CCERT-PUBDOC-2009-03-257

+CERT.hr u suradnji s LSS

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

Rezultat toga rada je i ovaj 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 4

2. TLS PROTOKOL 5

2.1. RAZVOJ KROZ POVIJEST 5
2.2. SVRHA 5
2.3. ALGORITMI 5
2.4. OSI MODEL 8
2.4.1. TLS u OSI modelu 10

3. FUNKCIONALNOST I OBILJEŽJA 11

3.1. NAČIN RADA 11
3.1.1. Protokol rukovanja 11
3.1.2. Protokol zapisa 13
3.2. PRIMJENE TLS-A 14
3.2.1. HTTPS 15
3.2.2. FTPS 16
3.2.3. STRATTLS 17
3.3. PREDNOSTI UPORABE 17
3.4. NEDOSTACI UPORABE 18

4. SSL PROTOKOL 19

4.1. KRATKI OPIS 19
4.2. SLIČNOSTI I RAZLIKE 20

5. POZNATE TLS/SSL IMPLEMENTACIJE 21

5.1. OPENSSL 21
5.2. GNUTLS 21
5.3. NSS 22
5.4. JSSE 23
5.5. USPOREDBA IMPLEMENTACIJA 23

6. SIGURNOSNE RANJIVOSTI SSL/TLS PROGRAMSKIH RJEŠENJA 24

6.1. USKRAĆIVANJE USLUGA 24
6.2. POKRETANJE PROIZVOLJNOG PROGRAMSKOG KODA 25
6.3. OTKRIVANJE OSJETLJIVIH PODATAKA 26
6.4. ZAOBILAŽENJE SIGURNOSNIH MEHANIZAMA 26
6.5. LAŽIRANJE CERTIFIKATA 27

7. BUDUĆI RAZVOJ 28

8. ZAKLJUČAK 28

9. REFERENCE 29

1. Uvod

Internet mreža se svakodnevno koristi za prijenos raznovrsnih podataka, od kojih neki mogu sadržavati posebno osjetljive informacije. Korisnicima je u cilju održati integritet i osigurati zaštitu takvih podataka. Jedno od glavnih svojstava koje je potrebno pri tome zadovoljiti je ostvarivanje autentifikacije, tj. primanje potvrde da je druga strana upravo ta za koju se izdaje. Potreba za posjedovanjem takvih karakteristika dovela je do razvoja nekih standardnih rješenja.

Jedan od prvih protokola koji su pružali siguran prijenos podataka mrežom bio je SSL (eng. Secure Sockets Layer) protokol. Ubrzo nakon njega dolazi do razvoja sličnog protokola nazvanog TLS (eng. Transport Layer Security) protokol. Navedeni protokol našao je razne primjene u autentifikaciji poslužitelja, udaljenom pristupu resursima, osiguravanju poruka elektroničke pošte i sl. Kako bi se korisnicima olakšala uporaba, dostupne su implementacije TLS protokola koje, kao i sam protokol, podržavaju mnogobrojne kriptografske algoritme.

Ovaj dokument daje uvod u TLS protokol, korištene kriptografske algoritme te njegovo značenje u OSI modelu. Također, prikazan je način rada i primjena protokola, kao i prednosti i nedostaci uporabe. Kako bi se dobio bolji pregled obilježja, napravljena je kratka usporedba SSL i TLS protokola. Također, predstavljene su neke osnovne implementacije TLS protokola i navedene njihove najčešće sigurnosne ranjivosti.

2. TLS protokol

2.1. Razvoj kroz povijest

Rana istraživanja o sigurnosti transportnog sloja OSI (eng. Open Systems Interconnection) modela uključivala su SNP (eng. Secure Network Programming) API (eng. Application Programming Interface) komponente. Godine 1993. istražuje se posjedovanje sigurnih API-ja na transportnom sloju koji su vrlo slični spojnicama (eng. sockets). SNP projekt je 2004. godine dobio ACM (eng. Association for Computing Machinery) nagradu za razvoj programskih sustava koji imaju trajan utjecaj (ACM Software System Award). Ovi napori vodili su prema razvoju posebnog standarda koji će osigurati sigurnost podataka prenošenih preko Internet infrastrukture.

SSL (eng. Secure Sockets Layer) protokol originalno je razvila Netscape (Netscape Communications Corporation) organizacija koja pruža razne računalne usluge, a najbolje je poznata po svom web pregledniku. Inačica 1.0 protokola SSL nije nikada javno objavljena dok je inačica 2.0 dostupna od veljače 1995. godine. Budući da je ta inačica sadržavala brojne ranjivosti, dolazi do razvoja inačice 3.0, koja je objavljena 1996. godine. Navedena inačica je kasnije poslužila kao osnova za razvoj TLS (eng. Transport Layer Security) protokola inačice 1.0 kao IETF (eng. Internet Engineering Task Force) standardnog protokola, definiranog u RFC (eng. Request for Comments) 2246 preporuci u siječnju 1999. godine. Mnoge vodeće financijske institucije (poput Visa, MasterCard, American Express i drugih) potvrdile su važnost primjene su SSL protokola za financijsko poslovanje preko Internet mreže.

2.2. Svrha

TLS protokol omogućuje aplikacijama komunikaciju preko mreže na način da se spriječi prisluškivanje, izmjenu i lažiranje poruka. Pruža autentifikaciju krajnjih točaka i povjerljivost komunikacije preko Internet mreže koristeći kriptografiju. Autentifikacija poslužitelja ima različito značenje za preglednik i za korisnika. Na razini preglednika označava posjedovanje valjanog certifikata, tj. provjeren digitalni potpis poslužiteljevog certifikata. Kada je jednom provjeren, preglednik je ovlašten, a samo provjera certifikata ne identificira poslužitelj krajnjem korisniku. Da bi korisnik znao identitet poslužitelja, potrebno je obaviti identifikaciju, tj. preslikavanje poznate oznake na nepoznati entitet kako bi se njega jednoznačno odredilo. Obično se koriste identifikatori (ID brojevi) koji moraju biti jedinstveni.

TLS protokol također pruža podršku za sigurniju dvosmjernu vezu, u kojoj oba krajnja čvora moraju znati s kim komuniciraju (obostrana autentifikacija). To zahtjeva da svi krajnji korisnici posjeduju certifikate (dakle i klijenti, a ne samo poslužitelji).

2.3. Algoritmi

Tipični algoritmi koji se koriste kod TLS/SSL protokola (tablica 1) su:

- Razmjena ključa: RSA, Diffie-Hellman, ECDH, SRP, PSK;
- Autentifikacija: RSA, DSA, ECDSA;
- Simetrično kriptiranje: RC4, DES, AES, IDEA, Triple DES ili Camellia, a u starijim inačicama SSL protokola koristi se i RC2;
- Kriptografske *hash* funkcije: HMAC-MD5 ili HMAC-SHA za TLS, MD5 i SHA za SSL te MD2 i MD4 za starije inačice SSL protokola.

Razmjena ključa	Autentifikacija	Simetrično kriptiranje	Kriptografske hash funkcije (TLS)	Kriptografske hash funkcije (SSL)
RSA	RSA	RC4	HMAC-MD5	MD5
Diffie-Hellman	DSA	DES	HMAC-SHA	SHA
ECDH	DSA	AES		MD2
SRP		IDEA		MD4
PSK		Triple DES		
		Camellia		
		RC2		

Tablica 1. Popis algoritama

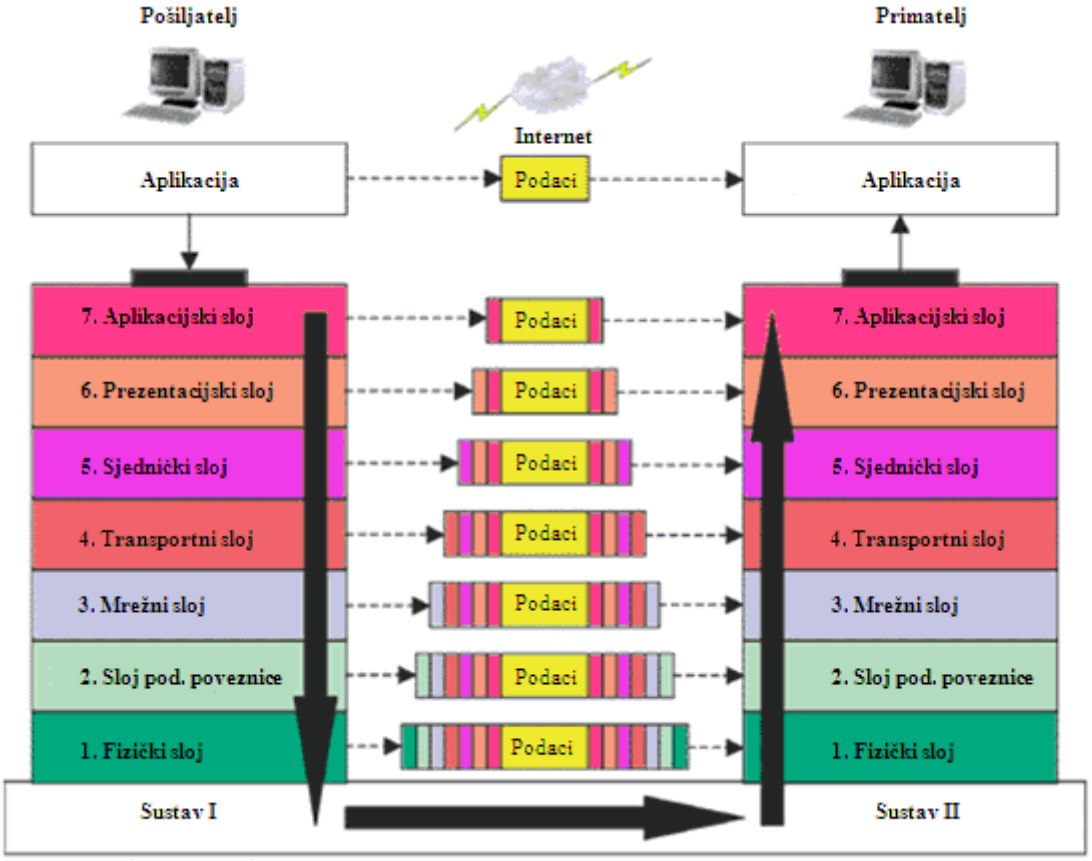
Kratki opis pojedinog algoritma:

- **RSA** algoritam je algoritam za kriptiranje uporabom javnog ključa kojeg su razvili 1977.g. Ron Rivest, Adi Shamir i Leonard Adleman na MIT (eng. Massachusetts Institute of Technology) sveučilištu. Obuhvaća tri koraka: generiranje ključa, kriptiranje i dekriptiranje. Sigurnost algoritma se temelji na dva matematička problema: faktorizacija velikih brojeva i RSA problem, tj. zadatak pronalaženja *n*-tog korijena modula kompozitnog broja N, čiji faktori nisu poznati.
- **Diffie-Hellman** algoritam je kriptografski protokol koji omogućava sudionicima komunikacije uspostavljanje dijeljenog (tajnog) ključa preko nesigurnih mrežnih kanala. Shemu su predložili Whitfield Diffie i Martin Hellman 1976. godine, a uključuje: dogovor oko ključa, uspostavljanje ključa, pregovaranje oko ključa, eksponencijalnu razmjenu ključa i sam protokol kriptiranja podataka. Eksponencijalna razmjena ključa je metoda kriptiranja koja koristi određene brojeve (npr. primarne brojeve) i odnose među njima kako bi kreirala ključ dekriptiranja. Sigurnost algoritma se zasniva na rješavanju Diffie-Hellman problema, tj. pronalaženju vrijednosti g^{xy}, ako je poznat g (generator grupe) te x i y (proizvoljni cjelobrojni brojevi).
- **ECDH** (eng. Elliptic Curve Diffie-Hellman) algoritam je inačica Diffie-Hellman algoritma koja koristi kriptografiju eliptične krivulje, tj. kriptografije javnog ključa koja se temelji na algebarskog strukturi eliptične krivulje preko konačnih polja. Metodu su predložili Neal Koblitz i Victor S. Miller 1985. godine.
- **SRP** (eng. Secure Remote Password Protocol) je protokol koji omogućava dogovor ključa za autentifikaciju i/ili prijavu. Protokol ima niz poželjnih svojstava kao što su:
  - autentifikacija korisnika poslužitelju,
  - otpornost na *„dictionary"* napad i
  - ostvarivanje komunikacije bez treće povjerljive strane (eng. trusted third party).
- **PSK** (eng. pre-shared key) je „dijeljena tajna" koja je prvotno razmijenjena između dva sugovornika pomoću sigurnih kanala. Takvi sustavi gotovo uvijek koriste simetrične algoritme kriptiranja.
- **DSA** (eng. Digital Signature Algorithm) algoritam je standard koji se koristi za digitalne potpise, a predstavila ga je NIST (eng. National Institute of Standards and Technology) organizacija 1991. godine za potrebe DSS (eng. Digital Signature Standard) standarda. Algoritam uključuje generiranje ključa, potpisivanje i provjeru.
- **ECDSA** (eng. Elliptic Curve DSA) algoritam je inačica DSA algoritma uz dodatnu uporabu grupe eliptičnih krivulja. Sadrži algoritme za generiranje i provjeru potpisa.
- **RC4** (drugim imenom ARC4 ili ARCFOUR) je široko korišteni program koji se koristi u raznim protokolima poput SSL (osiguravanje prometa) i WEP (osiguravanje bežičnih mreža). Dizajnirao ga je Ron Rivest u RSA Security odjelu 1987. godine. (RC je akronim za "Ron's Code"). Algoritam generira pseudo-slučajni niz bita koji se kombinira sa tekstom korištenjem XOR funkcije (dekriptiranje se obavlja na isti način). Za generiranje niza koriste se: permutacije 256 bitova i dva pokazivača duljine 8 bita.

- **DES** (eng. Data Encryption Standard) algoritam je blokovska šifra, koja se temelji na simetričnoj kriptografiji uporabom ključa veličine 56 bita. Pojavio se još 70-ih godina 20. stoljeća, a danas se smatra nesigurnim za mnoge aplikacije upravo zbog nedovoljne duljine ključa.
- **AES** (eng. Advanced Encryption Standard) algoritam je kriptografski standard koji obuhvaća AES-128, AES-192 i AES-256 inačice algoritma. Svaka od navedenih inačica ima blokove veličine 128 bita te ključeve veličina 128, 192 ili 256 bita. Standard je objavila NIST organizacija u studenom 2001. godine, a danas je najpopularniji standard za simetrično kriptiranje.
- **IDEA** (eng. International Data Encryption Algorithm) algoritam je blokovska šifra, koju su dizajnirali Xuejia Lai i James Massey 1991. godine kao zamjenu za DES algoritam. Nastao je izmjenom PES (eng. Proposed Encryption Standard) algoritma, a koristio se u PGP (eng. Pretty Good Privacy) programu inačice 2.0.
- **Triple DES** je blokovska šifra nastala iz DES algoritma, na način da se on uporabio tri puta. Nastao je kao zamjena za DES algoritam, zbog jednostavnog načina povećanja duljine ključa bez potrebe za prelaskom na novi algoritam. Polako izlazi iz uporabe jer je zamijenjen AES standardom.
- **Camellia** je blokovska šifra koju su razvile Mitsubishi i NTT kompanije 2000. godine, a ima blokove duljine 128 bita te ključeve duljina 128, 192 ili 256 bita (isto kao AES).
- **RC2** je blokovska šifra koju je dizajnirao Ron Rivest 1987. godine, a sadrži blokove veličine 64 bita i ključeve varijabilne duljine. Ovaj algoritam je ranjiv na napad povezanim/sličnim ključevima (eng. related-key attack).
- **HMAC** (eng. Hash Message Authentication Code) ili KHMAC (eng. Keyed-Hash Message Authentication Code) je jedna inačica MAC (eng. Message Authentication Code) vrijednosti, koja se računa pomoću posebnih algoritama uporabom kriptografskih *hash* funkcija u kombinaciji s tajnim ključem. Omogućava provjeru integriteta i autentifikacije poruke.
- **MD5** (eng. Message-Digest algorithm 5) algoritam je široko korištena kriptografska *hash* funkcija, koju je razvio profesor Ronald Rivest na MIT sveučilištu. Kao Internet standard koristio se u mnogim aplikacijama, često kako bi se provjerio integritet datoteka. Nakon otkrića neotpornosti algoritma na koliziju, povučen je iz primjene.
- **SHA** (eng. Secure Hash Algorithm) algoritam je kriptografska *hash* funkcija koju je dizajnirala NSA (eng. National Security Agency) organizacija. Postoje tri inačice spomenutog algoritma: SHA-0, SHA-1 i SHA-2.
- **MD2** (eng. Message Digest Algorithm 2) algoritam je kriptografska *hash* funkcija koju je razvio Ronald Rivest 1989. godine za 8-bitne sustave.
- **MD4** (eng. Message Digest Algorithm 4) algoritam je kreirao Ronald Rivest na MIT sveučilištu 1990. godine, a koristi se za provjeru integriteta poruke. Nedostaci navedenog algoritma objavljeni su 1991. godine u radu Den Boer i Bosselaers.

2.4. OSI model

OSI (eng. Open Systems Interconnection Reference Model) model je apstraktni opis slojevite komunikacije i dizajna mrežnih protokola. Razvijen je kao dio OSI (eng. Open Systems Interconnection) inicijative, a dijeli arhitekturu mreže u 7 slojeva (kako je prikazano na slici 1).



Slika 1. OSI model

Svaki sloj čini skupina sličnih funkcija koje pružaju usluga višem sloju a pri tome koriste usluge nižeg sloja u modelu. Na primjer, sloj koji pruža uslugu komunikacije bez pogrešaka preko mreže pruža put koji treba aplikacija iznad, ali također treba usluge nižih slojeva za slanje i primanje paketa.

Opis slojeva OSI modela:

- **Aplikacijski sloj** je 7. sloj OSI modela najbliži krajnjem korisniku. Pruža mrežne usluge korisničkim aplikacijama, a ne drugim slojevima. Funkcije aplikacijskog sloja obično uključuju identificiranje sudionika komunikacije, određivanje dostupnosti sredstava i sinkronizacija komunikacije. Prilikom identifikacije sudionika komunikacije, aplikacijski sloj određuje identitet i dostupnost sudionika komunikacije za aplikaciju. Kada se određuje dostupnost sredstava, ovaj sloj mora odrediti da li zahtijevani mrežni resursi postoje. U sinkroniziranju komunikacije, svaka komunikacija između aplikacija zahtijeva suradnju kojom upravlja aplikacijski sloj. Neki primjeri implementacija aplikacijskog sloja su: Telnet, HTTP (eng. Hypertext Transfer Protocol), FTP (eng. File Transfer Protocol) i SMTP (eng. Simple Mail Transfer Protocol) protokoli.
- **Prezentacijski sloj**, 6. sloj modela, omogućuje suradnju entiteta aplikacijskog sloja, tj. entiteti višeg sloja mogu koristiti različitu sintaksu i semantiku. Podatkovne jedinice se enkapsuliraju u SPDU blokove (eng. Session Protocol Data Units) te šalju nižim slojevima. Ovaj sloj pruža neovisnost prilikom predstavljanja podataka zahvaljujući prevođenju iz aplikacijskog u mrežni oblik, i obrnuto. Prezentacijski sloj transformira podatke u oblik koji prihvaća aplikacijski sloj. Također, aplikacije koje rade na ovom sloju, oblikuju i kriptiraju podatke kako bi mogli biti slani

preko nesigurne mreže, pružajući neovisnost od problema usklađenosti. Izvorna struktura koristi osnovna pravila šifriranja ANS.1 (eng. Abstract Syntax Notation One) skupa kriptografskih pravila. Prema ANS.1 pravilima podatke prvo treba definirati preko ANS.1 notacije, a zatim kriptirati kao niz bitova. Postoje razna pravila kriptiranja (eng. Encoding rules), a najpoznatija je DER (eng. Distinguished Encoding Rules) metoda. Slika 2 prikazuje primjer uporabe ANS.1 notacije i DER pravila kriptiranja.

ANS.1 notacija

```
FooProtocol DEFINITIONS ::= BEGIN

    FooQuestion ::= SEQUENCE {
        trackingNumber INTEGER,
        question       IA5String
    }

    FooAnswer ::= SEQUENCE {
        questionNumber INTEGER,
        answer         BOOLEAN
    }

END
```

Poruka koju pošiljatelj odašilje

```
myQuestion FooQuestion ::= {
    trackingNumber     5,
    question           "Anybody there?"
}
```

DER pravilo kriptiranja

```
30 -- tag indicating SEQUENCE
13 -- length in octets

02 -- tag indicating INTEGER
01 -- length in octets
05 -- value

16 -- tag indicating IA5String
0e -- length in octets
41 6e 79 62 6f 64 79 20 74 68 65 72 65 3f -- value ("Anybody there?" in ASCII)
```

Konačni niz bitova (21 oktet)

```
30 13 02 01 05 16 0e 41  6e 79 62 6f 64 79 20 74  68 65 72 65 3f
```

Slika 2. ANS.1 pravila

- **Sjednički sloj**, kao 5. Sloj OSI modela, upravlja vezom među računalima. Uspostavlja, upravlja i određuje vezu među lokalnim i udaljenim aplikacijama. Pruža dvosmjerne (eng. full-duplex) i jednosmjerne (eng. half-duplex) operacije, uspostavljanje točaka provjere (eng. checkpoint) te odgađanje, prekidanje i ponovno pokretanje procedura. Ovaj sloj odgovaran je za zatvaranje i obnovu sjednice. Obično je implementiran eksplicitno u okruženje aplikacije koristeći RPC (eng. Remote Procedure Call) pozive.

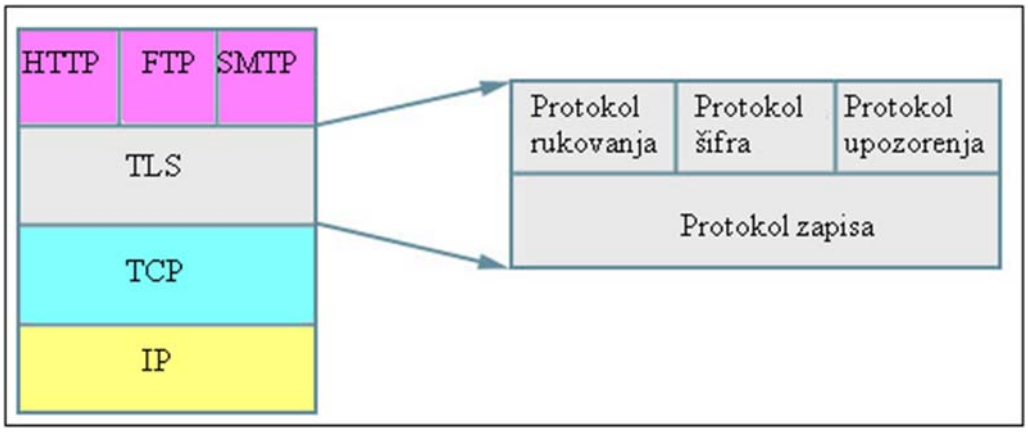
- **Transportni sloj**, ili 4. sloj modela, omogućuje transparentni prijenos podataka između krajnjih korisnika, pružajući usluge stvarnog prijenosa podataka višim slojevima. Upravlja pouzdanošću dane veze kroz kontrolu toka, segmentaciju i upravljanje pogreškama. Neki protokoli ovog sloja imaju mogućnost retransmisije, tj. ponovnog slanja izgubljenih jedinica podataka. Iako nisu razvijeni unutar OSI modela, najbolji primjeri protokola ovog sloja su TCP (eng. Transmission Control Protocol) i UDP (eng. User Datagram Protocol) protokoli. Postoji pet klasa OSI protokola transportnog sloja od klase TP0 (nema funkcionalnosti ispravljanja pogrešaka, dizajniran za mreže bez pogrešaka) do TP4 (dizajniran za stvarne mreže).
- **Mrežni sloj** je 3. sloj u OSI modelu, a pruža funkcionalnost prijenosa niza podataka varijabilne duljine od izvora do odredišta preko jedne ili više mreža uz upravljanje kvalitetom usluga. Omogućuje funkcije usmjeravanja (na ovom sloju djeluju usmjerivači) te može provoditi fragmentiranje paketa i dojavu pogrešaka u prijenosu. Protokol IP (eng. Internet Protocol) je predstavnik protokola mrežnog sloja.
- **Sloj podatkovne poveznice**, 2. sloj modela, omogućuje funkcije i procedure potrebne za prijenos podataka između mrežnih entiteta te detekciju i po mogućnosti ispravljanje pogrešaka u nastalih fizičkom sloju. I WAN (eng. Wide Area Network) i LAN (eng. Local Area Network) mreže spajaju bitove s fizičkog sloja u logičke nizove zvane okviri.
- **Fizički sloj** ili 1. sloj modela je najniži sloj koji definira elektroničke i fizičke specifikacije uređaja tj. određuje vezu između uređaja i fizičkog medija. Definiraju se naponski nivoi, broj pinova na konektorima (odnosno parica u kabelima) ili debljina koaksijalnog kabela. Primjeri uređaja na fizičkom sloju su mrežne kartice (integrirane na matičnoj ploči ili samo utaknute u sabirnicu na matičnoj ploči), mrežni koncentratori (eng. hub) i ponavljači (eng. repeater). Osnovne funkcije fizičkog sloja su: uspostavljanje i prekid veze s komunikacijskim medijem, modulacija (pretvorba digitalnih podataka u odgovarajuće signale) i sl.

2.4.1. TLS u OSI modelu

TLS protokol radi na slojevima ispod aplikacijskih protokola poput HTTP, FTP, SMTP, NNTP i XMPP, ali i iznad pouzdanih protokola transportnog sloja poput TCP protokola. Smještaj protokola TLS u OSI modelu, na 5 i 6 sloju, prikazuje slika 3. Budući da može dodati sigurnost bilo kojem protokolu koji koristi pouzdane veze, često se koristi u kombinaciji s HTTP protokolom kako bi se dobio HTTPS protokol. Dobiveni protokol koristi se za osiguravanje Web stranica na kojima se nalaze aplikacije namijenjene elektroničkom poslovanju i upravljanju imovinom.

Protokol također omogućava stvaranje tunela kroz Internet mrežu kako bi se kreirala VPN (eng. Virtual Private Network) mreža. To donosi neke prednosti u vatrozidu i NAT (eng. network address translation) komponenti (poput mogućnosti kriptiranja svih podataka koji se prenose tunelom).

Osim toga, TLS protokol se sve više koristi kao standardna metoda za zaštitu SIP (eng. Session Initiation Protocol) aplikacijske signalizacije. Može se iskoristiti za autentifikaciju i kriptiranje SIP signalizacije povezane sa VoIP (eng. Voice over Internet Protocol) i drugim aplikacijama temeljenim na SIP protokolu.



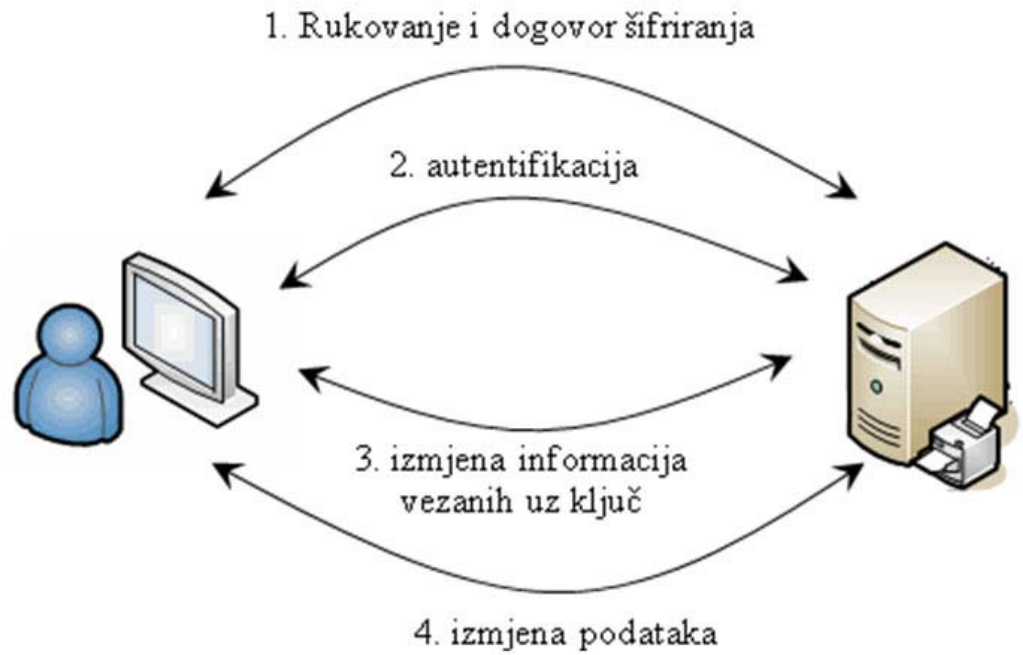
Slika 3. Smještaj TLS protokola u OSI modelu

3. Funkcionalnost i obilježja

3.1. Način rada

Kod TLS protokola komunikacija poslužitelja i klijenta uključuje (slika 4):

1. rukovanje i dogovor okruženja šifriranja (eng. cipher suite) – korisnik i poslužitelj dogovaraju šifriranje koje će se koristiti tijekom komunikacije,
2. autentifikaciju sugovornika – poslužitelj dojavljuje identitet korisniku, a u nekim slučajevima i korisnik poslužitelju (npr. potreba izmjene podataka na poslužitelju). Temelj ove autentifikacije je razmjena javnih i privatnih ključeva,
3. izmjenu informacija vezanih uz ključ – izmjena slučajno odabrane vrijednosti i posebne PMS (eng. Pre-Master Secret) vrijednosti. Razmijenjene vrijednosti koriste se za kreiranje *Master Secret* vrijednosti, koja služi za kreiranje ključa sjednice.
4. izmjena podataka.



Slika 4. Komunikacija poslužitelja i korisnika unutar TLS protokola

3.1.1. Protokol rukovanja

TLS klijent i poslužitelj dogovaraju vezu korištenjem procedure rukovanja (eng. handshaking procedure). Tijekom te procedure oni dogovaraju parametre koji se koriste za uspostavljanje sigurnosti veze:

- Rukovanje počinje kada se klijent spoji na poslužitelj (koji ima implementiran TLS protokol), zahtijevajući sigurnu vezu i predstavljajući popis podržanih šifri i *hash* funkcija.
- Poslužitelj odabire najjaču šifru s popisa i *hash* funkciju te obavještava klijenta. Zatim, poslužitelj šalje informacije o svom identitetu u obliku digitalnog potpisa.
- Certifikat obično sadrži ime poslužitelja, povjerljivi CA (eng. certificate authority) entitet i poslužiteljev javni ključ kriptiranja.

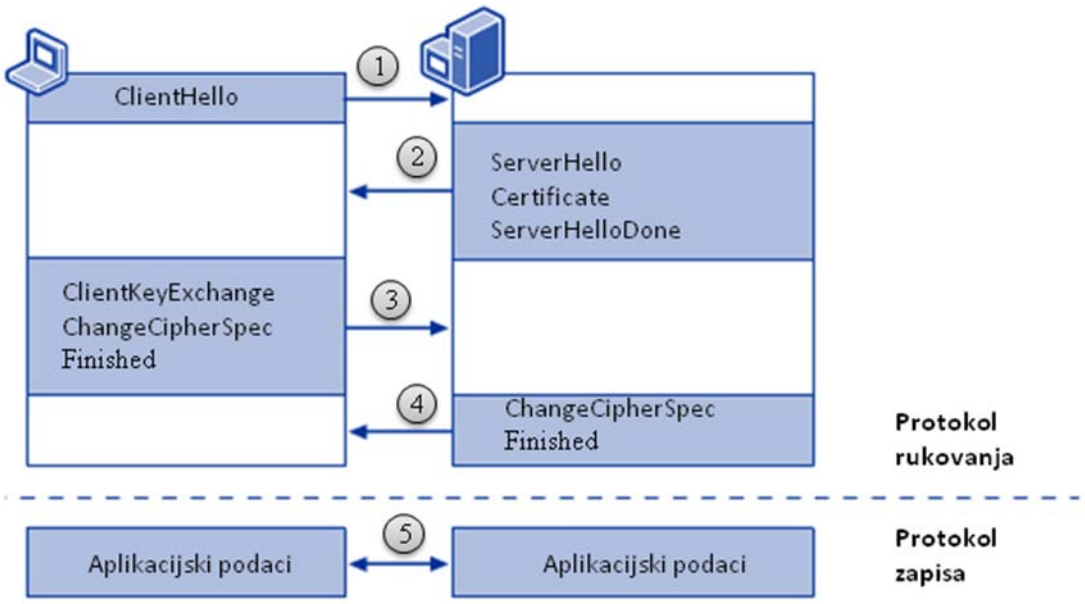
Korisnik može kontaktirati poslužitelj koji izdaje certifikat i provjeriti da li je taj certifikat autentičan (prije obrade):

- Kako bi se generirao ključ sjednice za sigurne veze, korisnik kriptira proizvoljni broj (RN-random number) uporabom poslužiteljevog javnog ključa (PbK) te šalje rezultat poslužitelju. Samo on može dekriptirati poruku (korištenjem privatnog ključa – PvK) pa je ključ skriven ostalim korisnicima. Korisnik zna PbK i RN, a poslužitelj PvK te nakon dekriptiranja i RN.
- Iz proizvoljno odabranog broja, obje strane generiraju potrebe podatke za kriptiranje i dekriptiranje.

Ovo zaključuje fazu rukovanja i inicira osiguranu vezu, koje je kriptirana do njenog prekida. Ako je bilo koji korak neuspješno obavljen, veza nije uspostavljena.

Osnovni oblik rukovanja naziva se jednostavno rukovanje (slika 5). Ovaj oblik rukovanja, ponekad zvan i potpuno rukovanje, uključuje sljedeće korake:

1. Korisnik šalje poruku „ClientHello“ specificirajući najviši TLS protokol koji podržava, proizvoljni broj, popis predloženih šifra i metoda kompresije.
2. Poslužitelj odgovara porukom „ServerHello“, koja sadrži odabrani protokol, proizvoljni broj, šifru i metodu kompresije. Poslužitelj može poslati i identifikacijski broj sjednice.
3. Poslužitelj šalje poruku „Certificate“.
4. Poslužitelj šalje poruku „ServerHelloDone“ kojom potiče kraj dogovaranja rukovanja.
5. Korisnik odgovara porukom „ClientKeyExchange“, koja sadrži javni ključ, PMS ili nikakve podatke.
6. Korisnik i poslužitelj zatim koriste slučajne brojeve i PMS kako bi dobili vrijednost „master secret“ (iz koje se dobiju ostale vrijednosti vezane uz ključ sjednice).
7. Korisnik šalje zapis „ChangeCipherSpec“ kojim javlja da započinje slanje kriptiranih podataka.
8. Konačno, korisnik šalje kriptiranu „Finished“ poruku, koja sadrži sažetak i MAC vrijednosti prethodne poruke.
9. Poslužitelj dekriptira poruku i provjerava sažetak i MAC vrijednosti. Ako provjera nije uspješna veza se odbacuje.
10. Poslužitelj šalje zapis „ChangeCipherSpec“ i svoju kriptiranu poruku „Finished“, a korisnik ponavlja postupak dekriptiranja.
11. Rukovanje je završeno i razmjena podataka je kriptirana.



Slika 5. Jednostavno rukovanje

Također, moguće je prilikom rukovanja provesti i autentifikaciju korisnika, a ne samo poslužitelja. Prilikom rukovanja uz korištenje autentifikacije korisnika razmjenjuju se poruke:

1. Korisnik šalje poruku „ClientHello“ specificirajući najviši TLS protokol koji podržava, proizvoljni broj, popis predloženih šifra i metoda kompresije.
2. Poslužitelj odgovara porukom „ServerHello“, koja sadrži odabrani protokol, proizvoljni broj, šifru i metodu kompresije. Poslužitelj može poslati i identifikacijski broj sjednice.
3. Poslužitelj šalje poruku „ServerCertificate“.
4. Poslužitelj zahtjeva certifikat korisnika preko poruke „CertificateRequest“, kako bi se veza mogla međusobno autentificirati.
5. Poslužitelj šalje „ServerHelloDone“ poruku, kojom označava kraj dogovaranja.
6. Korisnik odgovara s porukom „Certificate“, koja sadrži certifikat.
7. Korisnik šalje poruku „ClientKeyExchange“ sa PMS vrijednošću, javnim ključem ili bez podataka.
8. Korisnik šalje „CertificateVerify“ poruku, potpis preko prethodne poruke napravljan pomoću vlastitog privatnog ključa (moguće ga provjeriti preko javnog ključa korisnika).
9. Korisnik i poslužitelj koriste slučajne bojeve i PSM za računanje MS vrijednosti, iz koje dobiju sve ostale podatke za ključ.
10. Korisnik šalje „ChangeCipherSpec“ zapis, koji označava početak kriptiranja.
11. Zatim, korisnik šalje kriptiranu „Finished“ poruku, koja sadrži sažetak i MAC vrijednosti prethodne poruke.
12. Poslužitelj dekriptira poruku i provjerava sažetak i MAC vrijednosti. Ako provjera nije uspješna veza se odbacuje.
13. Poslužitelj šalje zapis „ChangeCipherSpec“ i svoju kriptiranu poruku „Finished“, a korisnik ponavlja postupak dekriptiranja.
14. Završeno je rukovanje i razmjena podataka je kriptirana.

Funkcije vezane uz javni ključ su vrlo skupe u smislu snage računala. TLS pruža sigurnu vezu u mehanizmu rukovanja kako bi se izbjegle takve funkcije. U potpunom rukovanju, poslužitelj šalje identifikacijski broj sjednice kao dio „ServerHello“ poruke. Korisnik povezuje tu vrijednost s IP adresom i TCP priključkom, kako bi, prilikom ponovnog spajanja na poslužitelj, korisnik mogao koristiti isti identifikacijski broj. Ta vrijednost označava kriptografske parametre koji su prethodno dogovoreni (pogotovo MS vrijednost). Sugovornici moraju imati istu MS vrijednost, jer inače rukovanje nije uspješno. Slučajni podaci u porukama „ClientHello“ i „ServerHello“ jamče da će generirani ključevi veze biti različiti od prijašnjih. Ovakav postupak rukovanja još se naziva skraćeno rukovanje ili rukovanje ponovnim pokretanjem.

3.1.2. Protokol zapisa

TLS protokol zapisa osigurava aplikacijske podatke koristeći ključ kreiran tijekom rukovanja. Ovaj protokol odgovoran je, osim za osiguravanje podataka aplikacije, i za provjeru integriteta podataka. Upravlja sa sljedećim:

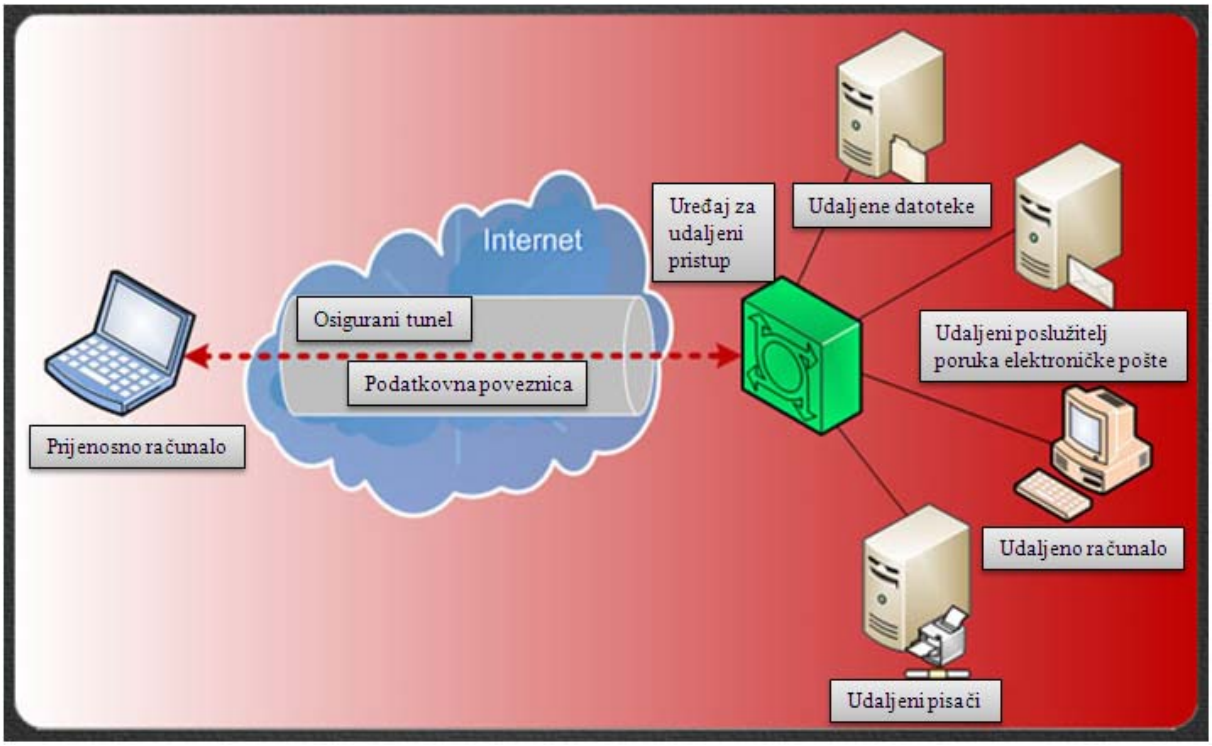
- podjelom odlaznih poruka u blokove pogodne za upravljanje te ponovno sastavljanje ulaznih podataka,
- kompresijom odlaznih blokova i dekompresijom dolaznih blokova (opcionalno),
- primjenom MAC vrijednosti na odlazne podatke i provjerom MAC vrijednosti ulaznih podataka i
- kriptiranjem odlaznih podataka i dekriptiranjem dolaznih poruka.

Kada aplikacija završi svoj rad, odlazni kriptirani podaci predaju se donjem TCP sloju na transport.

3.2. Primjene TLS-a

Osnovna primjena TLS protokola je osiguravanje sustava prilikom pregleda web stranica i informacija u HTTPS komunikaciji. Osim toga, ako je potrebno, protokol se može koristiti u svrhu autentifikacije i zaštite. Neki primjeri uporabe TLS protokola su:

- **Siguran prijenos podataka** za potrebe trgovanja putem Internet mreže (eng. e-commerce) – protokol se primjenjuje između preglednika i poslužitelja. Najbolji primjer je uporaba kreditnih kartica za plaćanje proizvoda/usluga putem Internet mreže. Protokol prvo potvrđuje valjanost certifikata web stranice te šalje podatke o kreditnoj kartici u obliku šifriranog teksta. U ovakvom obliku komunikacije, kada certifikat dolazi od povjerljivog izvora, autentifikacija se provodi samo za poslužitelj. TLS mora biti omogućen na web stranici gdje se događa promet podataka.
- **Autentificiran pristup** web stranicama – kako bi se ostvarila autentifikacija, i korisnik i poslužitelj trebaju certifikate od CA entiteta. Certifikati se mogu preslikati na korisničke račune po dvije osnove:
  - jedan na jedan – koristi se kada poslužitelj ima kopiju korisnikovog certifikata. Prilikom svake prijave poslužitelj provjerava identitet korisnika. Obično se primjenjuje za rukovanje privatnim podacima, poput obavljanja bankarskih usluga preko Internet mreže (jer samo jedan korisnik ima prava pregleda podataka).
  - više na jedan – upotrebljava se kada se želi nekoj grupi korisnika dati pristup povjerljivim materijalima. Tada se kreira grupa, pridrži joj se određeni certifikat te dodaju potrebne ovlasti.
- **Udaljeni pristup** – omogućuje korištenje sredstava i usluga na udaljenim računalima (slika 6). Pri tome, TLS protokol se može koristiti za provedbu autentifikacije i zaštite podataka (kada se korisnik prijavljuje na udaljeni sustav). Na taj način korisnici mogu pristupati porukama elektroničke pošte ili aplikacijama uz smanjen rizik razotkrivanja informacija drugim korisnicima Internet usluga.



Slika 6. Udaljeni pristup

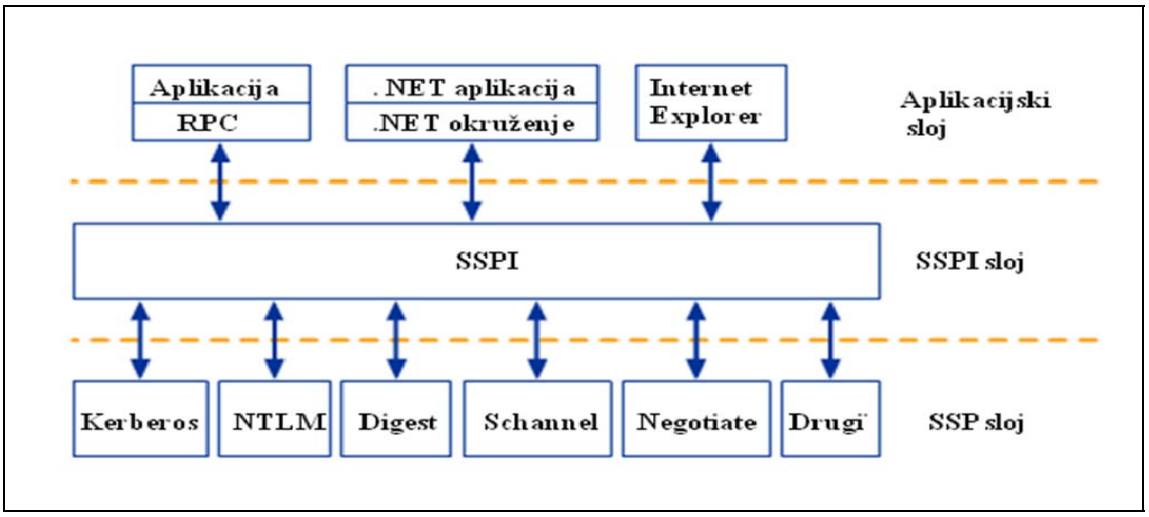
- **SQL pristup** – uporabom Microsoft SQL Server poslužitelja (ili nekog drugog) korisnik može zahtijevati autentifikaciju klijenata prilikom spajanja na poslužitelj na kojem je pokrenut SQL Server. Također, moguće je definirati zahtjeve za kriptiranjem podataka koji se razmjenjuju. Ograničavanjem pristupa podacima omogućuje se zaštita i razotkrivanje vrlo osjetljivih podataka (npr. informacija o financijama).

- **Poruke elektroničke pošte** – prilikom uporabe poslužitelja za razmjenu (eng. Exchange servers), moguće je primijeniti TLS protokol kako bi se osigurala zaštita podataka koji se prenose među poslužiteljima ili među mrežama. Kako bi se osigurala potpuna sigurnost prijenosa poruka elektroničke pošte potrebno je koristiti S/MIME (eng. Secure/Multipurpose Internet Mail Extensions) protokol. Ipak, osiguravanje zaštite podataka u razmjeni između poslužitelja omogućava firmama prijenos poruka dijeljenjem unutar iste firme, jedinice i sl. Opisanu funkcionalnost je moguće ostvariti i bez uporabe S/MIME protokola.

  Kada se TLS protokol omogući na poslužitelju za razmjenu poruka elektroničke pošte pošiljatelja i primatelja, informacije koje se razmjenjuju među njima su kriptirane. Ti poslužitelji koriste SMTP protokol za slanje i primanje poruka. Prilikom slanja kriptirane poruke, razmjena poruka elektroničke pošte funkcionira na slijedeći način:

  1. Svaki poveznik (eng. gateway) se konfigurira kako bi omogućio TLS komunikaciju za SMTP promet.
  2. Pošiljatelj provjerava da li su ponuđene TLS usluge.
  3. Ako primatelj nudi TLS usluge, pošiljatelj inicira početak TLS rukovanja. Primatelj šalje svoj certifikat pošiljatelju.
  4. Ako pošiljatelj vjeruje certifikatu primatelja, dogovara se enkripcijski ključ TLS sjednice, započinje sjednica i SMTP pouka se prenosi.

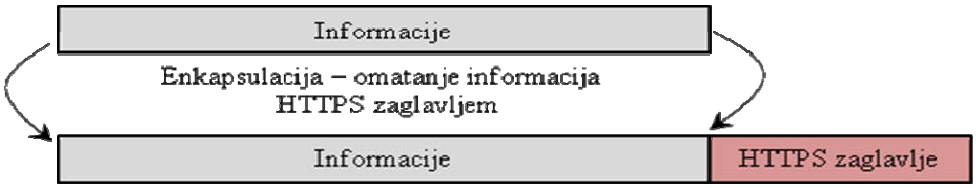
Postoje još mnoge druge razne primjene TLS protokola. Mogućnost pristupa protokolu kroz SSPI (eng. Security Service Provider Interface) sučelja znači da se može primijeniti u gotovo svakoj aplikaciji. Arhitekturu spomenutog sučelja prikazuje slika 7. Sve više informacija među različitima aplikacijama se izmjenjuje na način kako bi iskoristile prednosti TLS protokola.



Slika 7. Arhitektura SPPI sučelja

3.2.1. HTTPS

HTTPS (eng. Hypertext Transfer Protocol Secure) protokol je kombinacija HTTP protokola s protokolima za mrežnu sigurnost (SSL ili TLS). HTTPS nije poseban protokol, već se odnosi na kombinaciju „obične" HTTP komunikacije preko SSL ili TLS veze. Ova kombinacija osigurava zaštitu od prisluškivanja i MITM (eng. man-in-the-middle) napada, pružajući mehanizme za provjeru valjanosti certifikata. Slika 8 prikazuje enkapsuliranje podataka u novi paket s HTTPS zaglavljem.



Slika 8. Izgled HTTPS paketa

Kako bi se poslužitelj konfigurirao za uporabu HTTPS veza, administrator mora kreirati certifikat uporabom javnog ključa za poslužitelj. Takav certifikat mora potpisati CA entitet, koji time potvrđuje identitet vlasnika certifikata. Web preglednici su obično distribuirani s certifikatima potpisnim od glavnih, korijenskih CA entiteta. Neke web stranice koriste certifikate koje samostalno potpisuju (uporaba HTTPS protokola bez CA entiteta), obično u uvjetima kada klijent posjeduje poslužitelj. Opisana situacija donosi određene probleme kod prihvaćanja certifikata u web preglednicima. Ipak takav način korištenja certifikata sprečava napad prisluškivanjem, ali ne i MITM napad.

Ovaj sustav može se koristiti i za potrebe autentifikacije korisnika kako bi se ograničio pristup poslužitelju. Postupak zahtjeva kreiranje certifikata za svakog korisnika koji ima pravo pristupa te ugrađivanje istog u preglednik tog korisnika. Obično sadrži ime i adresu elektroničke pošte (u posebnim slučajevima i lozinku) te se provjerava pri svakom ponovnom spajanju na poslužitelj.

Neka ograničenja HTTPS protokola su:

- Razina zaštite ovisi o ispravnoj implementaciji web preglednika, programa na poslužitelju i podrške za kriptografske algoritme. HTTPS je nesiguran kada se primjenjuje na javno dostupan statički sadržaj. Cijela stranica može biti indeksirana pomoću „Web crawler" programa, a URI niz kriptiranog izvora može biti otkriven poznavanjem samo veličine zahtjeva/odgovora. Opisani postupak napadaču omogućuje pristup zaštićenom tekstu.
- Budući da SSL/TLS protokol djeluje ispod HTTP protokola i nema znanja o protokolu višeg sloja, poslužitelj može samo predstaviti jedan certifikat za određenu kombinaciju IP adrese i priključka. Ovo znači da, u većini slučajeva, nije izvedivo koristiti usluge pohrane sadržaja na poslužitelje drugih korisnika (tzv. hosting) temeljene na imenima s HTTPS protokolom. Preporuka RFC-3546 TLS Extensions definira rješenje zvano SNI (eng. Server Name Indication) značajku, za koju podršku imaju: Firefox 2.0, Opera 8, Mozilla 1.8 i Internet Explorer 7 na operacijskom sustavu Windows Vista. SNI značajka omogućava korisniku zahtjev imena domene prije predaje certifikata poslužitelju. Funkcionira na način da SNI šalje ime virtualnog poslužitelja kao dio rukovanja kod TLS protokola. To poslužitelju omogućava spajanje na taj virtualni poslužitelj i slanje pregledniku certifikata s ispravnim imenom domene.

3.2.2. FTPS

FTPS (eng. File Transfer Protocol Secure) protokol je proširenje FTP (eng. File Transfer Protocol) protokola koje dodaje podršku za TLS i SSL kriptografske protokole. Uključuje potpunu podršku za TLS i SSL kriptografske protokole, kao i šifre AES, RC4, RC2, Triple DES i DES te *hash* funkcije SHA, MD5, MD4 i MD2.

Razvijene su dvije metode uvođenja sigurnosti FTP klijenta:

1. **Eksplicitno** – FTPS klijent mora eksplicitno zahtijevati sigurnosni oblik od FTPS poslužitelja te se zatim prilagoditi međusobno dogovorenoj kriptografskoj metodi. Ako klijent ne zahtjeva sigurnost, FTPS poslužitelj može dopustiti klijentu da nastavi nesigurnu vezu ili odbiti/ograničiti vezu. Mehanizmi za dogovaranje autentifikacije i sigurnosti definirani su preporukom RFC-2228. U ovoj metodi korisnik određuje koji će dijelovi veze biti kriptirani, a promjena kriptiranja nekog dijela veze može se donijeti u bilo kojem trenutku (jedino ograničenje dolazi od poslužitelja – npr. odbacivanje naredbi).
2. **Implicitno** – nije dopušteno pregovaranje s FTPS poslužiteljem. Klijent šalje poruku koja označava iniciranje veze poslužitelju. U slučaju da ne primi takvu poruku, poslužitelj odbacuje vezu. Implicitni FTPS poslužitelji trebaju osluškivati priključnice za FTPS kontrole (IANA Well Known Port 990/TCP) i podatkovne (989/TCP) kanale. Prilikom uporabe ovog postupka cijela sjednica je kriptirana.

U nekim slučajevima nije potrebno koristiti kriptiranje, kao npr.:

- prijenos podataka koji nisu osjetljivi,
- prijenos podataka koji su već kriptirani na razini datoteke i
- dostupne kriptografske funkcije ne dostižu zahtijevanu razinu.

Ova praksa se preporuča zbog manjih zahtjeva za računalnim resursima (nije potrebno procesorsko vrijeme za kriptiranje i dektirptiranje) i boljeg iskorištavanja mrežnih resursa (nema enkapsulacije paketa).

3.2.3. STRATTLS

STARTTLS je proširenje komunikacijskih protokola koji prenose nekriptirane poruke. Pruža način nadogradnje nekriptirane veze na kriptiranu vezu, umjesto uporabe posebnih priključaka za kriptografsku komunikaciju.

Budući da omogućava uspostavu kriptirane veze među poslužiteljima, nakon uspostave veze sva komunikacija među poslužiteljima je kriptirana.

Preporuke koje definiraju STARTTLS su:

1. RFC 2595 za IMAP i POP3 (http://tools.ietf.org/html/rfc2595),
2. RFC 2487 za SMTP (http://tools.ietf.org/html/rfc2487),
3. RFC 4642 za NNTP (http://tools.ietf.org/html/rfc4642).

3.3. Prednosti uporabe

Neke od prednosti koje donosi primjena TLS protokola su:

- **Jaka autentifikacija te privatnost i integritet poruka** – TLS protokol korištenjem kriptiranja podataka može očuvati tajnost i integritet. Također, omogućuje autentifikaciju poslužitelja te (opcionalno) autentifikaciju klijenta kako bi se pružio identitet sugovornika u sigurnoj komunikaciji. Integritet podataka osigurava se pomoću vrijednosti za provjeru. Kako bi se spriječilo otkrivanje podataka, TLS sigurnosni protokol može se iskoristiti za zaštitu od:
  - napada u kojem jedan sustav pogađa/prisvaja identitet drugog (eng. masquerade attack),
  - jednog od oblika MITM (eng. man-in-the-middle) napada gdje napadač presreće poruke u razmjeni javnog ključa, te ih ponovno šalje potpisujući vlastiti javni ključ (eng. bucket brigade attack),
  - napada u kojem napadač na sustavu koristi nesigurne značajke nekih starijih inačica (eng. rollback attack),
  - jednog od oblika mrežnog napada pri kojem je valjani mrežni promet zlonamjerno ili lažno izmijenjen i/ili izbrisan (eng. replay attack).
- **Mogućnost suradnje** (eng. Interoperability) – TLS protokol radi na većini web preglednika (uključujući Microsoft Internet Explorer, Mozilla Firefox i Netscape Navigator), kao i na većini operacijskih sustava i web poslužitelja (uključujući Microsoft Windows, UNIX, Novell, Apache inačica 1.3 i kasnije, Netscape Enterprise Server i Sun Solaris). Obično je ugrađen u novije čitače, LAPD poslužitelje i razne druge aplikacije.
- **Fleksibilnost algoritma** – TLS protokol pruža razne opcije za mehanizme autentifikacije, kriptografske algoritme i *hash* algoritme koji se koriste tijekom osiguravanja sjednice.
- **Lakoća razvoja** – mnoge aplikacije koriste TLS transparentno na operacijskim sustavima kao npr. Windows Server 2003, Unix i dr. Korisnik može uporabiti TLS protokol za osiguravanje pretraživanja web stranica kada koristi preglednik Internet Explorer te IIS (eng. Internet Information Services) usluge. Tada na primjer ako poslužitelj ima ugrađen certifikat, potrebno je samo odabrati provjeru. Slična praksa primijenjena je i kod Mozilla Firefox te Netscape Navigator web preglednika.
- **Lakoća uporabe** – budući da se TLS protokol implementira ispod aplikacijskog sloja, većina njegovih operacija su potpuno neovisne o korisniku. Ovo korisniku omogućuje zaštitu od napada uz posjedovanje minimuma znanja o sigurnosti i komunikaciji. Također, budući da nije

potrebno instalirati neki program na računalu, kriptiranje je „uvijek uključeno“, a rad je skriven od krajnjeg korisnika.

- **Zaštita** – svi podaci koje razmjenjuju poslužitelji ili poslužitelj i korisnik su kriptirani. Slanje nekriptiranih poruka povećava rizik od presretanja ili izmjene poruka. Osim toga, moguće je obaviti skeniranje i analizirati poruke elektroničke pošte kako bi se otkrili razni zlonamjerni programi.
- **Globalna raširenost/prihvaćenost** - Budući da su specifikacije protokola definirane preporukom, on je globalno prihvaćen te je njegova uporaba široko raširena. Većina financijskih institucija implementirala je TLS protokol u svoje poslovanje, a sve više ih planira to učiniti u određenom vremenskom roku.

3.4. Nedostaci uporabe

Osnovni nedostaci/ograničenja TLS protokola su:

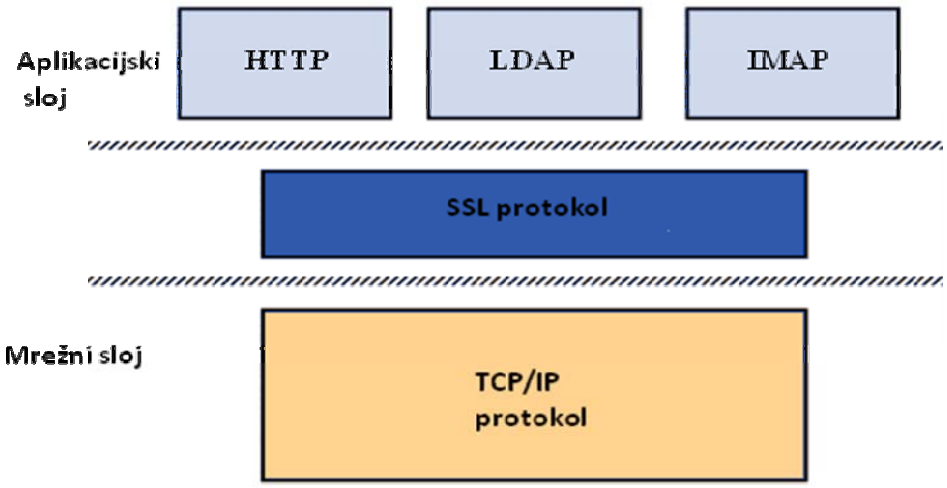
- **Povećan rad procesora** – osnovno ograničenje prilikom implementacije TLS protokola. Funkcije poput kriptiranja, a posebno operacije oko distribucije javnog ključa, zahtijevaju dodatan rad procesora. Također, nije moguće točno odrediti koliko će se smanjiti performanse sustava tj. koliki je trošak CPU (eng. Central Processing Unit) jedinica. Performanse variraju u ovisnosti o učestalosti uspostavljanja veze i njenom trajanju. Najviše resursa se troši prilikom uspostavljanja veze.
- **Dodatan rad administratora** – TLS okruženje je poprilično složeno i zahtjeva održavanje pa administratori moraju konfigurirati sustav i upravljati certifikatima.
- **Veličina paketa** – budući da TLS protokol dodaje određene informacije u pakete koji se razmjenjuju preko mreže, veličina paketa se povećava. Posljedica povećanja paketa je povećanje potrebnog vremena za obradu, kao i za prijenos podataka. To uvodi kašnjenje u prijenos velike količine podataka.

4. SSL protokol

SSL (eng. Secure Sockets Layer) protokol je standardizirana sigurnosna tehnologija za kreiranje kriptirane veze između poslužitelja i preglednika. Njome se osigurava zadržavanje privatnosti i sigurnosti svih podataka koji se razmjenjuju između sudionika komunikacije. Jedna od najraširenijih primjena SSL standarda jest zaštita podataka u tzv. *online* transakcijama (npr. plaćanje putem Interneta, bankarske transakcije i sl.).

4.1. Kratki opis

SSL protokol radi iznad TCP/IP sloja i ispod protokola aplikacijskog sloja, kako je prikazano na slici 9. Omogućava autentifikaciju korisnika i poslužitelja pružajući pouzdanu kriptiranu vezu.



Slika 9. Smještaj SSL protokola u OSI model

Protokol sadrži slijedeće funkcionalnosti:

- **Autentifikaciju poslužitelja** – omogućuje korisniku otkrivanje/potvrdu identiteta poslužitelja. Korisnik može koristiti tehnike poput kriptografije javnog ključa kako bi provjerio valjanost certifikata. Važnost ovog postupka dolazi do isticanja u sustavima gdje se razmjenjuju važni (npr. financijski) podaci.
- **Autentifikacija korisnika** – omogućuje poslužitelju potvrdu korisničkog identiteta. Uporabom istih tehnika kao kod autentifikacije poslužitelja, poslužitelj može provjeriti certifikat korisnika. Ovaj postupak ima veliko značenje kod slanja povjerljivih informacija korisniku.
- **Kriptirana SSL veza** – zahtjeva kriptiranje svih informacija koje se razmjenjuju između korisnika i poslužitelja, kako bi se osigurao visoki stupanj povjerljivosti. Pri tome je moguće detektirati izmjenu podataka u prijenosu.

SSL protokol sadrži dvije podgrupe:

- **protokol zapisivanja** – definira oblik korišten u prijenosu podataka i
- **protokol rukovanja** – omogućuje razmjenu podataka između poslužitelja i klijenta kako bi se uspostavila veza. Ovaj protokol obuhvaća:
  - autentificiranje poslužitelja,
  - odabir kriptografskog algoritma ili šifre koje podržavaju i klijent i poslužitelj,
  - (opcionalno) autentificiranje klijenta i
  - generiranje dijeljenih tajnih podataka pomoću javnog ključa.

Navedeni protokol podržava uporabu raznih kriptografskih algoritama ili šifra za funkcije autentifikacije, prijenosa certifikata i uspostave sjedničkog ključa. Slijedi popis algoritama koje podržava SSL protokol:

- DES (eng. Data Encryption Standard),

- DSA (eng. Digital Signature Algorithm),
- KEA (eng. Key Exchange Algorithm) – algoritam za razmjenu ključa,
- MD5 (eng. Message Digest algorithm 5),
- RC2 and RC4,
- RSA (Rivest, Shamir, Adleman),
- RSA za razmjenu ključa – algoritam za razmjenu ključa temeljen na RSA algoritmu,
- SHA-1 (eng. Secure Hash Algorithm),
- SKIPJACK – složeni simtrični algoritam implementiran u FORTEZZA sklopovlju,
- Triple-DES.

4.2. Sličnosti i razlike

TLS je standard usko povezan sa SSL protokolom inačice 3.0 (ponekad se i odnosi na SSL protokol inačice 3.1). On zamjenjuje protokol SSL inačice 2.0 i trebao bi se koristiti u novim programskim implementacijama. Aplikacije koje zahtijevaju veću razinu međusobne suradnje trebaju podržavati SSL protokol inačice 3.0, kao i TLS protokol.

Proširenja TLS protokola:

- Kod TLS protokola KHMAC (eng. keyed-Hashing for Message Authentication Code) algoritam zamjenjuje MAC (eng. Message Authentication Code) algoritam koji se koristi kod SSL protokola. KHMAC postupak pruža više sigurnosti od MAC algoritma. Osim toga, stvara vrijednost za provjeru integriteta (eng. integrity check value), kao i MAC algoritam, ali uporabom *hash* funkcija što ju čini složenijom za otkivanje napadaču.
- U specifikaciji protokola TLS dodane su mnoge nove poruke za upozoravanje,
- Kod TLS protokola nije uvijek potrebno uključiti certifikate od korijenskih CA entiteta, nego je dovoljno koristiti središnje CA entitete. Više informacija o PKI infrastrukturi moguće je pronaći u dokumentu „Nedostaci PKI infrastrukture“ dostupnom na web stranicama CERTa.
- TLS protokol definira vrijednosti za povećanje blokova (eng. padding block values) koje se koriste u blokovskim algoritmima šifriranja. Kod RC4 algoritma, koji koriste operacijski sustavi Microsoft Windows, ova funkcionalnost nije bitna.
- Algoritmi koji pružaju sigurnost zasnovanu na PC karticama (eng. Fortezza algorithm) nisu uključeni u TLS RFC preporuku, jer nisu javno dostupni.
- Određena polja poruka kod TLS protokola su izmijenjena.

5. Poznate TLS/SSL implementacije

5.1. OpenSSL

OpenSSL je alat otvorenog koda koji implementira protokole SSL (inačice 2.0 i 3.0) i TLS (inačice 1.0) te pruža kriptografsku biblioteku koja se može koristiti u drugim alatima. Temelji se na „SSLeay" biblioteci, koju su razvili A. Young i Tim J. Hudson. Licenciran je Apache-style licencom, što znači da je besplatan za dohvat i uporabu za komercijalne i nekomercijalne svrhe. Trenutna inačica (0.9.8j), dostupna je od 7. siječnja 2009. godine, a moguće ju je preuzeti na web stranici: http://www.openssl.org/source/. Program je dostupan za većinu UNIX-a (Solaris, Linux) i Mac OS X te četiri BSD operacijska sustava otvorenog koda operacijskih sustava, kao i za OpenVMS te Microsoft Windows operacijske sustave.

OpenSSL program koristi različite kriptografske funkcije za:

- kreiranje i upravljanje privatnim i javnim ključevima te njihovim parametrima,
- operacije vezane uz kriptiranje uporabom javnog ključa,
- kreiranje X.509 certifikata te lista valjanih i povučenih certifikata,
- računanje MD (eng. Message Digests) vrijednosti poruke,
- kriptiranje i dekriptiranje uporabom šifra,
- testiranje klijenta i poslužitelja,
- rukovanje S/MIME potpisanim ili kriptiranim porukama elektroničke pošte i
- rukovanje, generiranje i provjera određenih zahtjeva.

Obuhvaća razne algoritme:

- Šifre: Blowfish, Camellia, DES, RC2, RC4, RC5, IDEA, AES;
- Kriptogrfske *hash* funkcije: MD5, MD2, SHA, MDC-2;
- Kriptogrfija uporabom javnog ključa: RSA, DSA, Diffie-Hellman algoritam, algoritme zasnovane na eliptičnim krivuljama (eng. Elliptic curve).

Program sadrži velik skup naredbi s raznim opcijama i argumentima, čiji je popis moguće dobiti uporabom sljedećih pseudo-naredbi:

- popis standardnih naredbi – „list-standard-commands",
- popis MD naredbi – „list-message-digest-commands" i
- popis naredbi za šifre – „list-cipher-commands".

Osim toga, moguće je dohvatiti popis svih algoritama:

- popis šifra – „list-cipher-algorithms",
- popis MD algoritama – „list-message-digest-algorithms" i
- popis algoritama koji podržavaju javne ključeve – „list-public-key-algorithms".

Opisani proizvod razvija tim volontera, a svaki korisnik koji želi pripomoći razvoju, može se prijaviti na web stranici: http://www.openssl.org/support/.

5.2. GnuTLS

Program GnuTLS (eng. GNU Transport Layer Security Library) je besplatno dostupna implementacija SSL i TLS protokola, čija je svrha pružiti API (eng. application programming interface) sučelja aplikacijama kako bi se omogućila sigurna komunikacija. Izdan je pod GNU LGPL (eng. Lesser General Public License) licencom, dok su neki dijelovi izdani pod licencom GNU GPL (eng. GNU General Public License). Navedene licence omogućuju besplatno kopiranje i distribuciju alata. Osnovna razlika je u tome što LGPL licenca omogućuje povezivanje s besplatnim programima koji nisu pod istom licencom te uvodi neka dodatna prava izmjene programa. Prvotno je razvijen za GNU projekte, a koristi se u programima poput GNOME, CenterIM, Exim, Mutt, Slrn, Lynx i CUPS.

GnuTL je dostupan za većinu UNIX operacijskih sustava te za operacijski sustav Microsoft Windows, a moguće ga je preuzeti preko poveznice:

http://www.gnu.org/software/gnutls/download.html.

GnuTLS obuhvaća sljedeća obilježja:

- podrška za SSL protokol inačice 3.0 i TLS protokol inačica 1.0 i 1.1,
- podrška za PSK (eng. pre-shared key) algoritam pri autentifikaciji,
- mehanizam proširivanja TLS protokola,
- podrška za jake algoritme kriptiranja (SHA-256/384/512 i Camellia),
- kompresija i
- rukovanje s X.509 i OpenPGP certifikatima.

Podržava mnoge algoritme:

- Razmjena ključeva: Anon-RSA, RSA, RSA EXPORT, DHE-RSA, DHE-DSS, SRP-DSS, SRP-RSA, SRP, PSK i DHE-PSK;
- Kriptografski algoritmi: AES-256, AES-128, 3DES, DES, RC4-128, RC4-40, RC2-40 i Camellia.

Program razvija tim stručnjaka, ali svaki korisnik može prijaviti pogreške ili prijedloge te pridonijeti razvoju proizvoda (http://www.gnu.org/software/gnutls/help.html).

5.3. NSS

Program NSS (eng. Network Security Services) je skup biblioteka, koje služe kao podrška SSL i S/MIME protokolima. Razvila ga je Netscape organizacija, a koriste ga AOL, Red Hat, Sun Microsystems operacijski sustavi u raznim aplikacijama (Mozilla Firefox, Thunderbird i SeaMonkey, AOL Instant Messenger, Evolution, Pidgin, OpenOffice.org 2.0., Red Hat Directory Server i dr.).

Licenciran je s tri licence: „Mozilla Public License“, „GNU General Public License“ i „GNU Lesser General Public License“. Trenutna inačica izdana je 28. ožujka 2008. godine, a riječ je o inačici 3.12.

Program podržava razne sigurnosne standarde:

- SSL protokol inačice 2.0 i 3.0,
- TLS protokol inačice 1.0
- PKCS standarde:
    - PKCS #1. RSA standard koji definira implementaciju kriptografije uporabom javnog ključa temeljenu na RSA algoritmu,
    - PKCS #3. RSA standard koji definira implementaciju Diffie-Hellman algoritma,
    - PKCS #5. RSA standard koji definira kriptografiju zasnovanu na lozinkama,
    - PKCS #7. RSA standard koji definira primjenu kriptografije na podatke (digitalni potpisi),
    - PKCS #8. RSA standard koji definira pohranu i kriptiranje privatnih ključeva,
    - PKCS #9. RSA standard koji definira uporabu potrebnih tipova atributa za kriptiranje,
    - PKCS #10. RSA standard koji definira sintaksu zahtjeva za certifikatom,
    - PKCS #11. RSA standard koji definira komunikaciju s kriptografskim uređajima (npr. pametne kartice) i omogućava neovisnost aplikacije o posebnim algoritmima i implementacijama,
    - PKCS #12. RSA standard koji definira oblik za pohranu i prijenos privatnih ključeva, certifikata i ostalih tajnih informacija.
- CMS (eng. Cryptographic Message Syntax) korišten u S/MIME protokolu,
- X.509 certifikate,
- OCSP (eng. Online Certificate Status Protocol) certifikate,
- PKIX certifikate,
- algoritme: RSA, DSA, ECDSA, Diffie-Hellman, EC Diffie-Hellman, AES, Triple DES, DES, RC2, RC4, SHA-1, SHA-256, SHA-384, SHA-512, MD2, MD5, HMAC i
- FIPS generator pseudo slučajnih brojeva.

5.4. JSSE

JSSE (eng. Java Secure Socket Extension) programski paket sadrži skupinu programa (API sučelja, alata, implementacija algoritama i sl.) koji omogućavaju sigurnost komunikacije preko Internet mreže. Implementira Java inačicu SSL i TLS protokola.

Uključuje funkcionalnost:

- kriptiranja podataka,
- autentifikacije poslužitelja i (opcionalno) autentifikacije klijenta,
- osiguravanja integriteta poruka,
- kriptografije i
- PKI (eng. Public Key Infrastructure) infrastrukture.

Paket je bio opcionalni dodatak Java programu inačica 1.2 i 1.3, dok je u inačicu 1.4 ugrađen.

5.5. Usporedba implementacija

U ovom dijelu dokumenta dana je usporedba tri prethodno opisana alata: OpenSSL, GnuTLS i NSS, budući da su to i najpoznatije implementacije.

Tablica 2 prikazuje usporedbu podrške za protokole SSL i TLS raznih inačica. Alat GnuTLS sadrži podršku za sve navedene inačice protokola, dok ostali alati podržavaju samo neke inačice.

	SSLv2.0 [1]	SSLv3.0	TLSv1.0	TLSv1.1	TLSv1.2
GnuTLS	Da	Da	Da	Da	Da
OpenSSL	Ne	Da	Da	Ne	Ne
NSS	Da	Da	Da	Ne	Ne

Tablica 2. Podrška protokola

Prikaz podrške algoritama za razmjenu ključa nalazi se u tablici 3. Također, naviše algoritama podržava alat GnuTLS.

	Anon-RSA	RSA	RSA EXPORT	DHE-RSA	DHE-DSS	SRP-DSS	SRP-RSA	SRP	PSK	DHE-PSK	ECC
GnuTLS	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Ne
OpenSSL	Da	Da	Da	Da	Da	Ne	Ne	Ne	Ne	Ne	Da
NSS	Da	Da	Da	Da	Da	Ne	Ne	Ne	Ne	Ne	Da

Tablica 3. Algoritmi za razmjenu ključa

U tablici 4 nalazi se pregled kriptografskih algoritama koje se koriste u implementacijama. Naviše algoritama podržava alat GnuTLS.

	AES-256 CBC	AES-128 CBC	3DES CBC	DES CBC	RC4-128 CBC	RC4-40 [1]	RC2-40 [1]	Camellia
GnuTLS	Da	Da	Da	Da	Da	Da	Da	Da
OpenSSL	Da	Da	Da	Da	Da	Ne	Ne	Da
NSS	Da	Da	Da	Da	Da	Ne	Ne	Da

Tablica 4. Kriptografski algoritmi

Podrška postupaka kompresije nalazi se u tablici 5, gdje je pokazano da GnuTLS podržava oba navedena postupka, OpenSSL samo ZLIB postupak, a NSS nijedan.

	ZLIB	LZO [1]
GnuTLS	Da	Da
OpenSSL	Da	Ne
NSS	Ne	Ne

Tablica 5. Kompresija

6. Sigurnosne ranjivosti SSL/TLS programskih rješenja

U svim SSL/TLS implementacijama uočene su razne sigurnosne ranjivosti uzrokovane nepravilnom implementacijom ili pogreškama u određenim programskim komponentama. Posljedice takvih propusta mogu biti vrlo različite, a jedna od najozbiljnijih je mogućnost izvođenja napada uskraćivanja usluga (eng. Denial of Service) na ugroženom računalu. Osim toga, napadač može iskoristiti određeni propust kako bi lažirao certifikate ili na neki drugi način ugrozio sigurnost sustava.

U nastavku dokumenta opisane su osnovne ranjivosti uočene kod SSL/TLS implementacija, te dani stvarni primjeri nedostataka implementacija koji uzrokuju opisane ranjivosti. Za sve opisane ranjivosti proizvođači su već izdali odgovarajuće programske ispravke.

6.1. Uskraćivanje usluga

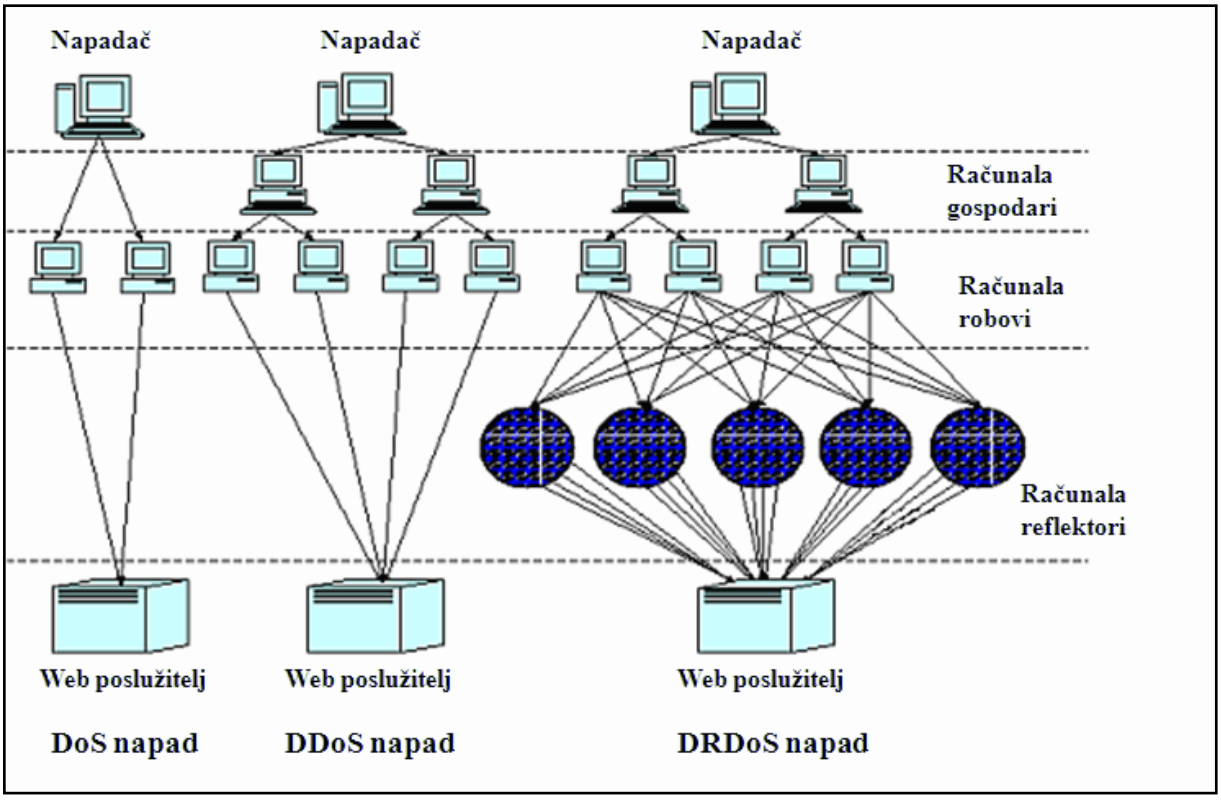
DoS (eng. Denial of Service) napad ili napad uskraćivanja usluga je pokušaj onemogućavanja dostupnosti ili korisnosti resursa računala legalnim korisnicima. Općenito se dijeli na dva oblika:

- navođenje računala na korištenje resursa kako ne bi mogao obavljati ostale funkcije i
- narušavanje veze između korisnika i računala kako oni više ne bi mogli komunicirati.

Napadi se mogu izvoditi na razne načine, a najučestaliji oblik pokretanja napada je poplavljivanje (eng. flooding) resursa. To označava da napadač šalje velike količine zahtjeva računalu koje ih ne stiže obraditi. Također, napad može doći iz jednog ili više izvora, kao i *zombie* računala (računala zaražena zlonamjernim virusom koji omogućuje napadaču udaljeno upravljanje). Napadač može postaviti zlonamjerni kod na više legalnih računala koji se nazivaju gospodari. Svaki gospodar može zaraziti određena računala (robovi) te upravljati pokretanjem DoS napada s njih. Razne inačice DoS napada prikazuje slika 10.

Zaštitu od DoS napada omogućava:

- primjena programa za filtriranje prometa (vatrozid i sl.),
- korištenje novih, ispravljenih inačica svih programskih proizvoda instaliranih na računalo,
- uporaba antivirusnih programa i sl.



Slika 10. Vrste DoS napada

U programskom paketu OpenSSL otkriveni su razni nedostaci koji zlonamjernim korisnicima mogu omogućiti izvođenje DoS (eng. Denial of Service) napada. Neki od sigurnosnih problema koji mogu dovesti do DoS uvjeta su:

- Nepravilnosti prilikom SSL/TLS rukovanja kada se koristi "do_change_cipher_spec()" funkcija. Ranjivost se javlja zbog pogrešne dodjele NULL pokazivača, a zahvaća sve inačice paketa od 0.9.6c do 0.9.6k i 0.9.7a do 0.9.7c.
- Pogreška čitanja izvan granica (eng. out-of-bounds read error) u rutini korištenoj tijekom SSL/TLS rukovanja u slučaju kada se koristi Kerberos sustav autentifikacije. Nedostatak je pronađen kod sljedećih inačica: 0.9.7a, 0.9.7b i 0.9.7c.
- Nespecificirana nepravilnost u inačici 0.9.6d, koja dovodi do beskonačnih petlji eng. (infinite loop).
- Pojava cjelobrojnog prepisivanja prilikom rukovanja ANS.1 oznakama.
- Nepravilna obrada nevaljanih javnih ključeva korisničkih certifikata u slučaju kada je aplikacija konfigurirana da ignorira pogreške dekriptiranja javnog ključa.
- Pogrešno rukovanje određenih podacima kada je „server_name“ varijabla postavljana na vrijednost 0x00, a zlouporaba zahtjeva slanje posebno oblikovanih TLS 1.0 Client Hello paketa.
- Greška prilikom izostavljanja „Server Key exchange message" poruke pri rukovanju u slučaju kada se koristi Diffie-Hellman razmjena ključa.

Slični problemi javljaju se i kod paketa GnuTLS te također omogućuju stvaranje DoS stanja. Neki od sigurnosnih propusta su:

- Pogreška prilikom poziva „gnutls_handshake()" za već valjano uspostavljanu sjednicu kod inačica od 2.3.5. do 2.4.0.
- Nepravilnost u DER dekoderu u „libtasn1“ komponenti koja se javlja za inačice prije 1.2.10.
- Problemi kod obrade paketa sa zapisima (eng. record packet).
- Greška prilikom provjere potpisa X.509 certifikata kod inačice 1.0.16. (moguće i ranijih inačica). Zlouporaba je moguća obradom dugog niza certifikata koji sadrže certifikate potpisane uporabom dugih RSA ključeva.
- Nepravilnost u „_gnutls_ciphertext2compressed()" funkciji pri obradi posebno oblikovanih TLS kriptiranih podataka. Ranjivost je uočena kod inačica prije inačice 2.2.4.

Osim navedenih propusta, određeni nedostaci koji omogućuju DoS napad pronađeni su i kod NSS paketa:

- Nedefinirana pogreška u inačicama NSS paketa na Sun Java System Web Server 6.0 (bez Service Pack 10) i Sun ONE Application Server 7 (bez Update 3 paketa) platformama.
- Nepravilnosti u „inftrees.c“ datoteci „zlib“ komponente pri rukovanju određenim nizovima. Navedeni propust javlja se kod inačice 3.10.

Kod JSSE alata ovi propusti omogućuju uzrokovanje DoS uvjeta:

- Nepravilno rukovanje SSL/TLS zahtjevima za dogovaranje algoritma kriptiranja i ostalih specifikacija pri rukovanju.

6.2. Pokretanje proizvoljnog programskog koda

Pokretanje proizvoljnog programskog koda je sposobnost napadača da pokrene bilo koju naredbu na ciljanom računalu ili sustavu. Mogućnost da se aktivira izvršavanje proizvoljnog koda s jednog stroja na drugi se naziva udaljeno pokretanje programskog koda. To je najgori učinak koji ranjivost može prouzročiti, jer omogućuje preuzimanje svih ovlasti nad ugroženim računalom/sustavom.

Obično se postiže kroz kontrolu nad pokazivačem procesora (PC, eng. program counter) koji predstavlja programsko brojilo te pokazuje na sljedeću naredbu koja procesor treba izvesti. Napadač umeće proizvoljni kod te mijenja vrijednost PC pokazivača tako da pokazuje na umetnuti kod koji se tada automatski pokreće.

Kod programskog paketa OpenSSL sljedeći propusti omogućuju napadaču pokretanje proizvoljnog programskog koda:

- Pogreška u rukovanju ANS.1 oznakama, koja se manifestira cjelobrojnim prepisivanjem.
- Nespecificirana pogreška u DTLS (eng. Datagram Transport Layer Security) implementaciji.

Programski paket GnuTLS također sadrži neke ranjivosti koje napadač može iskoristiti za pokretanje proizvoljnog programskog koda:

- Pojava prepisivanja memorije na stogu kada se rukuje s „Client Hello" porukama kod inačica prije 2.2.4. Kako bi iskoristio navedeni nedostatak napadač mora korisniku podmetnuti posebno oblikovan TLS paket.

Unutar paketa NSS sljedeći nedostaci mogu dovesti do pokretanja proizvoljnog programskog koda:

- Višestruko prepisivanje memorije na stogu prilikom obrade SSLv2 poruka poslužitelja kod inačica 3.10 i 3.11.3.

6.3. Otkrivanje osjetljivih podataka

Podaci koji se prenose preko Internet mreže često sadrže osjetljive informacije, tj. povjerljive ili tajne podatke. Takve podatke potrebno je zaštiti, jer obično obuhvaćaju informacije o korisniku koje napadač može zlouporabiti kako bi mu nanio određenu štetu. Osnovna zaštita osjetljivih podataka podrazumijeva njihovo kriptiranje da bi se dobio oblik koji nije poznat napadačima. Zlonamjerni korisnici razvijaju razne metode za prisluškivanje veze, dekriptiranje poruka ili neki drugi način otkrivanja informacija (npr. „phishing" napad).

Unutar programskog paketa NSS otkrivene su određene ranjivosti koje mogu omogućiti otkrivanje osjetljivih podataka:

- Nepravilnosti u implementaciji rukovanja pogreškama u aplikacijama koje koriste CBC šifre u SSL ili TLS protokolu. Mjerenjem razlike vremena za koje aplikacija reagira na novu poruku napadač može otkriti određene informacije (poput lozinki) o prenošenim podacima preko SSL i TLS kanala.

6.4. Zaobilaženje sigurnosnih mehanizama

Budući da na većini poslužitelja postoje određeni pohranjeni podaci, potrebno je osigurati njihovu zaštitu od neautentificiranih korisnika. Kako bi se omogućila zaštita podataka od pregleda, izmjene ili uništenja, potrebno je definirati prava pristupa svim korisnicima. Sigurnosna ograničenja definiraju takva prava pristupa osiguravajući legalnim korisnicima dostupnost potrebnih podataka, te branjenjem pristupa svima ostalima.

U radu implementacije OpenSSL sljedeći problemi omogućuju zaobilaženje sigurnosnih ograničenja:

- Pogreška pri rukovanju sa „SSL_OP_MSIE_SSLV2_RSA_PADDING" opcijom, što omogućuje poticanje dogovora manje sigurnog SSL 2.0 protokola (čak i ako oba sugovornika podržavaju sigurnije SSL 3.0 i TLS 1.0 protokole). Napadač mora izvesti MITM napad, a zlouporaba je moguća samo kada je omogućena „SSL_OP_ALL" opcija. Opisanu ranjivost sadrže inačice prije inačica 0.9.7h i 0.9.8a .
- Nepravilna provjera potpisa certifikata u slučaju uporabe RSA algoritma kod inačica 0.9.7j i 0.9.8b.

Kod programskog paketa GnuTLS zaobilaženje sigurnosni ograničenja omogućuju ove ranjivosti:

- Nepravilnosti kod provjere niza X.509 certifikata čija je zlouporaba moguća lažiranjem proizvoljnih imena prilikom izvođenja MITM napada. Opisani problem otkriven je kod inačica prije 2.6.1.
- Pogreška u provjeri određenih potpisa u slučaju uporabe RSA ključeva kod inačice 1.4.
- Problem u „cURL/libcURL" komponenti koji uzrokuje neispravnu provjeru određenih dijelova SSL/TLS certifikata, a pronađen je kod inačica 7.14.0 - 7.16.3.

U paketu NSS također se javljaju slični problemi:

- Nepravilna provjera potpisa certifikata kada se koriste RSA algoritmi za inačice 3.x.

6.5. Lažiranje certifikata

Certifikati predstavljaju dokumente koji su garancija identiteta određenog poslužitelja, web stranice ili korisnika. Lažiranjem certifikata napadač može navesti korisnika na posjetu zlonamjernih web stranica, preuzimanje zlonamjernog programskog koda, kao i ostvariti krađu određenih podataka, veze s proizvoljnim poslužiteljima i sl. Kako bi napadač mogao lažirati certifikate, mora prikupiti određene podatke poput javnog ključa, potpisa i sl.

Programski paket OpenSSL sadrži neke ranjivosti, koje napadač može iskoristiti kako bi lažirao certifikat:

- Nepravilna provjera povratne vrijednosti funkcije „EVP_VerifyFinal()“ prilikom provjere DSA i ECDSA ključeva. Napadač ga može iskoristiti kako bi zaobišao provjeru certifikata slanjem posebno oblikovanog potpisa. Opisani nedostatak javlja se kod inačica prije inačice 0.9.8j.
- Pogreške prilikom uporabe RSA algoritma, koje omogućuju lažiranje PKCS #1 v1.5 potpisa, a time i lažiranje samog certifikata.

Kod paketa JSSE javljaju se sljedeći propusti koji omogućuju lažiranje certifikata:

- Kriva provjera RSA PKCS #1 v1.5 potpisa ako se koristi RSA algoritam za javne ključeve.

7. Budući razvoj

Budući da je definiran kao standard kroz više preporuka (ovisno o inačici) TLS protokol je široko prihvaćen. Sve više aplikacija ugrađuje njegovu funkcionalnost kako bi iskoristile pogodnosti koje nudi. Razvijene su razne TLS/SSL implementacije koje donose jednostavno korištenje i ne zahtijevaju posjedovanje posebnih, stručnih znanja. Zahvaljujući takvim svojstvima, ali i prednostima koje donosi uporaba protokola (autentifikacija, kriptiranje i sl.), predviđa se sve veća uporaba opisnog standarda.

Izmjene određenih dijelova protokola obavljaju se svakodnevno, u svrhu ispravljanja pogrešnih specifikacija. Velika prednost je sama činjenica da svaki korisnik može prijaviti uočenu ranjivost. Također, preporukama se definirana razna proširenja ovog protokola. Ova obilježja također pokazuju da se može očekivati danji napredak u razvoju i primjeni TLS protokola.

8. Zaključak

Korisnici Internet usluga neprestano razmjenjuju razne sadržaje putem mreže, dohvaćaju podatke smještene na poslužitelje, pristupaju udaljenim resursima i sl. U svim tim situacijama određeni podaci sadrže i osjetljive informacije, koje je potrebno zaštititi od pregleda i uporabe drugih korisnika Internet infrastrukture. Razvoj TLS protokola uvodi standardni način osiguravanja integriteta podataka i autentifikacije sugovornika u komunikaciji. Budući da djeluje ispod aplikacijskog sloja OSI modela, njegov rad je skriven od krajnjeg korisnika, što znači da nisu potrebna posebna znanja za uporabu samog protokola. Također, protokol podržava raznovrsne kriptografske algoritme pa se može primijeniti za razne potrebe gdje se traži kriptiranje podataka. TLS protokol implementiran je u nekoliko programskih rješenja što dodatno olakšava njegovu primjenu na sustav krajnjeg korisnika. Iako te poznate implementacije TLS protokola sadrže određene ranjivosti, njihovi proizvođači, sukladno otkrivanju nedostataka, izdaju potrebne programske zakrpe. Zahvaljujući svim dobrim svojstvima koje pruža njegova primjena protokol je iz dana u dan sve više raširen i prihvaćen kao standard.

9. Reference

[1] TLS, http://en.wikipedia.org/wiki/Transport_Layer_Security, ožujak, 2009.

[2] TLS protocol, http://www.ietf.org/rfc/rfc2246.txt, ožujak, 2009.

[3] SSL protocol, http://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt, ožujak, 2009.

[4] Transport Layer Security Protocol, http://msdn.microsoft.com/en-us/library/aa380516.aspx, ožujak, 2009.

[5] TLS (FAQ), http://www.bnymellon.com/security/tlsencryption.pdf, ožujak, 2009.

[6] What is TLS/SSL?, http://technet.microsoft.com/en-us/library/cc784450.aspx, ožujak, 2009.

[7] SSL/TLS in Detail, http://technet.microsoft.com/en-us/library/cc785811.aspx, ožujak, 2009.

[8] FTPS, http://en.wikipedia.org/wiki/FTPS, ožujak, 2009.

[9] STARTTLS, http://en.wikipedia.org/wiki/STARTTLS, ožujak, 2009.

[10] HTTPS, http://en.wikipedia.org/wiki/Https, ožujak, 2009.

[11] OpenSSL, http://www.openssl.org/, ožujak, 2009.

[12] GnuTLS, http://www.gnu.org/software/gnutls/, ožujak, 2009.

[13] JSSE, http://java.sun.com/javase/technologies/security/, ožujak, 2009.

[14] NSS, http://www.mozilla.org/projects/security/pki/nss/, ožujak, 2009.

[15] Ranjivost SSL/TLS implementacije, http://www.mozilla.org/projects/security/pki/nss/news/vaudenay-cbc.html, ožujak, 2009.

[16] SSL/TLS nedostatak, http://www.cisco.com/warp/public/707/cisco-sa-20070118-certs.shtml, ožujak, 2009.

[17] DoS ranjivost, http://secunia.com/advisories/11139/, ožujak, 2009.

[18] Sigurnosni problemi kod SSL/TLS implementacija, http://www.cert.org/advisories/CA-2003-26.html, ožujak, 2009.

[19] Problemi kod OpenSSL implementacije, http://secunia.com/advisories/product/253/, ožujak, 2009.

[20] Nedostaci implantacije GnuTLS, http://secunia.com/advisories/search/?search=gnutls, ožujak, 2009.