SEMINARSKI RAD

Zaštita podataka primenom kriptografskih metoda

Projekat je radio: Milutin Milosavljević

Sadržaj

1 Uvod

Uporedno sa razvojem informacionih tehnologija i telekomunikacionih sistema raste i mogućnost zloupotreba podataka koji se tim putem prenose. Brzi razvoj hardvera i softvera, koji se koriste za prenos podataka, uslovljava česte promene i usavršavanja istih. U toj trci proizvođači često nemaju vremena za detaljno testiranje opreme. Usled nedovoljnog testiranja, često se javljaju propusti koji čine osnovu za rad potencijalnih napadača. Napadači razvijaju sopstvene alate i tehnike pomoću kojih zaobilaze sigurnosne mere i na taj način dolaze do „važnih" podataka. U cilju eliminisanja ovih nedostataka nameće se novi pravac istrazivanja u sferi sigurnosti informacionih sistema, koji se odnosi na zaštitu podataka od neovlašćenog pristupa, modifikacija ili različitih zloupotreba. Na važnost zaštite podataka ukazuje činjenica da je ona postala jedan od najvažnijih komponenata nacionalnih bezbednosti zemalja, državnih administracija, banaka i sl. Isto tako, veliki broj kompanija sve više poklanja pažnju merama sigurnosti sopstvenih informacionih sistema i podataka.

Kao najzastupljeniji medijum za prenos podataka se koriste računarske mreže. Kao takve logično predstavljaju usko grlo po pitanju sigurnosti informacija koje se tim putem prenose. Iz tog razloga, posebna pažnja posvećena je zaštiti podataka koji se prenose putem računarske mreže.

U ovom radu je dat pregled osnovnih principa zaštite podataka u računarskim mrežama sa naglaskom na metode zaštite bazirane na Kriptografiji. Razlog za to se nalazi u činjenici da kriptografski protokoli čine osnovicu svih protokola za zaštitu podataka koji se implementiraju na višim nivoima. Analizirani su simetrični i asimetrični kriptosistemi, a kasnije je dat princip rada tehnike digitalnog potpisa, i izloženi su problemi upravljanja ključevima kao i upotreba sigurnosnih servisa u okviru TCP/IP modela.

Na početku, u prvom poglavlju, je dat pregled vrsta napada na podatke koji se prenose putem računarskih mreža. Pored toga, navedeni su i mehanizmi koji se mogu koristiti za zaštitu navedenih podataka.

U okviru drugog poglavlja se preko osnovnih termina koji se javljaju u kriptografiji i kratkog istorijskog pregleda, spoznaju osnove koje su neophodne radi lakšeg razumevanja kriptografskih protokola i algoritama koji su dati u nastavku istog poglavlja. Algoritmi su podeljeni na simetrične i asimetrične. Nakon opisa algoritama, iskorišćena je prilika za razmatranje međusobnih prednosti i nedostataka. Na kraju poglavlja na primeru PGP algoritma, ukratko je pojašnjen hibridni način kriptovanja.

Treće poglavlje sadrži prikaz metoda za verifikaciju dokumenata ili poruka koje se šalju putem mreže, odnosno tehnikama digitalnog potpisa. Razmatrane su metode potpisivanja cele poruke i metode potpisivanja sažetka poruke i date su dve najčešće korišćene tehnike za formiranje sažetka poruke.

Verifikacija dokumenata u savremenim komunikacionim sistemima često ne predstavlja siguran vid razmene poruka. Daleko bolje rešenje predstavljaju sistemi koji podržavaju međusobnu autentifikaciju obe strane u komunikaciji. Upravo ove metode za autentifikaciju entiteta su prikazane u okviru četvrtog poglavlja.

Uokviru petog poglavlja se razmatra jedan od najvećih problema sistema za prenos podataka putem računarske mreže, a to je problem upravljanja ključevima. Upravljanje ključevima se ne odnosi samo na distribuciju ključeva, već i na njegovo kreiranje, pamćenje, zamenu i na kraju uništenju.

Poglavlje pod naslovom „Sigurnosni servisi u TCP/IP modelu" sadrži kratak pregled protokola koji se implementiraju u TCP/IP model u cilju unapređenja modela po pitanju bezbednosti paketa podataka koji se prenose putem Interneta. U nastavku su dati opisi dva protokola koji se koriste na najnižim slojevima pomenutog modela.

1.1 Vrste napada i zaštita podataka na mreži

Kako računarske mreže predstavljaju deo informacionog sistema pomoću koga se prenosi najveći broj podataka, posebnu pažnju treba obratiti upravo na njihovu zaštitu. Podaci u okviru informacionih sistema se mogu javiti u sledećim oblicima:

- Javni podaci – podaci koji nisu poverljivi i čiji integritet nije važan, može ih koristiti bilo ko bez ikakvih posledica. Primeri ovakvih podataka su javni servisi za pružanje informacija.
- Interni podaci – pristup ovim podacima je dozvoljen samo za određene grupe korisnika. Javno objavljivanje internih podataka nije dozvoljeno, ali objavljivanje ove vrste podataka nije od kritične važnosti. Primer su podaci u razvojnim grupama, firmama, radni dokumenti i projekti, interni telefonski imenici.
- Poverljivi podaci – poverljivi podaci unutar određene grupe (kompanije) koji su zaštićeni od neovlašćenog pristupa. Neivlašćeni pristup ovim podacima može prouzrokovati značajane posledice kao što su: finansijski gubitak kompanije ili dobitak konkurentskoj kompaniji, smanjenje poverenja korisnika usluga ili potrošača proizvoda. Primeri ovih informacija su: podaci o platama, podaci o zaposlenima, projektna dokumentacija, računovodstveni podaci, poverljivi ugovori.
- Tajni podaci – podaci kod kojih je neautorizovan pristup strogo zabranjen. Integritet podataka je na najvišem nivou. Broj ljudi koji može da pristupi ovim podacima trebalo bi da bude ograničen. Prilikom pristupa ovim informacijama moraju se poštovati veoma striktna pravila. Primer ovih podataka su: vojni podaci, podaci o većim finansijskim transakcijama, podaci od državnog značaja i slično. Ova vrsta podataka se treba čuvati u kriptovanom obliku ili u uređajima sa hardverskom zaštitom [1].

Kada se govori o zaštiti podataka na mreži, uglavnom se misli na zaštitu poverljivih i tajnih podataka koji se prenose putem računarske mreže. Sve masovnija upotreba računarskih mreža iziskuje korišćenje mehanizama i mera za zaštitu podataka koji se tim putem prenose. Mere za zaštitu podataka uopšte, se zasnivaju na tri principa [5]:

- Prevencija – odnosi se na preduzimanje preventivnih aktivnosti za zaštitu podataka i računarskih sistema od mogućih napada
- Detekcija – otkrivanje kako je narušena zaštita, kada je narušena i ko je narušio
- Reakcija – preduzimanje aktivnosti koje dovode do restauracije podataka ili do restauracije računarskog sistema

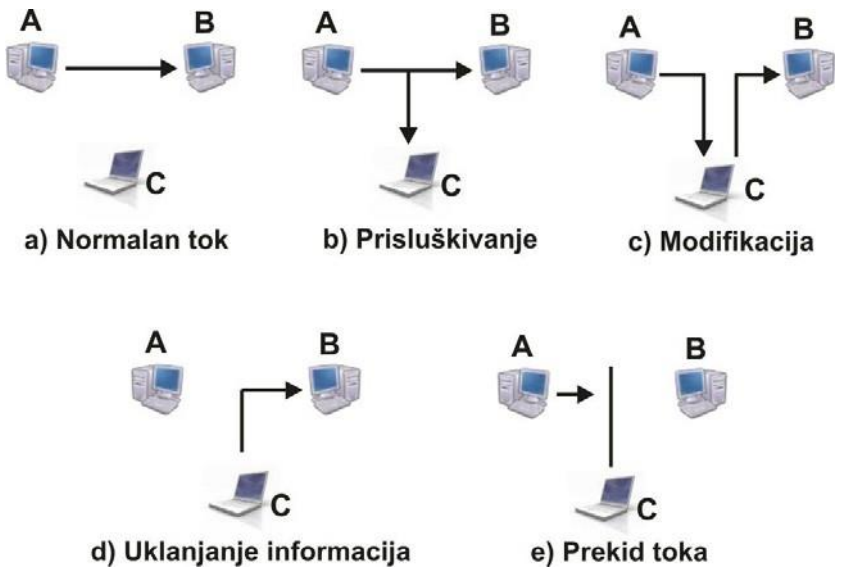
Zaštita podataka koji se prenose putem računarske mreže zasniva se na sva tri principa, ali je ovaj rad fokusiran na zaštitu u vidu prevencije. Najveća pretnja podacima koji se prenose putem računarske mreže javlja se usled slabosti komunikacione opreme pomoću koje se vrši prenos podataka. Ugrožavanje podataka u računarskim mrežama se odnosi na prisluškivanje, analizu, menjanje, uklanjanje informacija kao i lažno predstavljanje [2]. Na Slici 1.1 prikazane su neke od mogućnosti napada na računarske mreže. Treba napomenuti da do ovih napada može doći na bilo kom mestu prenosa informacija od izvora do odredišta.

Svi napadi na podatke koji se prenose mrežom se mogu podeliti u dve grupe:

- Pasivni napadi i
- Aktivni napadi

Pasivni napadi se odnose na sva prisluškivanja i nadgledanja informacija tokom prenosa, bez ikakvih izmena. Ovom vrstom napada napadač na relativno jednostavan način dolazi do informacija. Pasivni napadi se teško otkrivaju. Kao najčešće korišćeni mehanizam zaštite od pasivnih napada primenjuje se kriptovanje podataka koji se prenose putem komunikacionih linija. Kriptovanje podataka se odnosi na modifikaciju istih na način da postanu nerazumljivi ili besmisleni za sve one korisnike

kojima nisu namenjeni. Kao takvo, kriptovanje predstavlja najvažniji element zaštite podataka u računarskim mrežama.



Slika 1.1
Vrste napada na mreži

Aktivni napadi su svi napadi koji vrše promenu sadržaja ili toka informacija. Ova vrsta napada je daleko komplikovanija i teža za otkrivanje nego pasivni napadi. U aktivne napade se ubrajaju modifikacije paketa informacija koji se kreću putem mreže, slanje lažnih paketa, prekidi toka informacija kao i razne vrste preusmeravanja paketa na mreži. Zbog raznovrsnosti ove vrste napada, mehanizmi zaštite moraju biti daleko komplikovaniji i napredniji nego kod pasivnih napada [2].

1.2 Sigurnosni servisi

Generalno govoreći sigurnosni servisi predstavljaju skup pravila koja se odnose na aktivnosti korisnika koje doprinose bezbednosti podataka na mreži. Postoji šest vrsta osnovnih sigurnosnih servisa:

- Autentifikacija (engl. *authentication*),
- Poverljivost ili tajnost podataka (engl. *data confidentiality*),
- Neporicljivost poruka (engl. *non-repudation*),
- Integritet podataka (engl. *data integrity*),
- Autorizacija i kontrola pristupa (engl. *authorization and access control*) i
- Raspoloživost resursa (engl. *resource availability*).

Autentifikacija se odnosi na potvrdu originalnosti poruka, odnosno na identifikaciju izvora poruke ili dokazivanje identiteta korisnika. Pojam autentifikacije se odnosi na ličnosti, terminale, kreditne kartice, smart kartice, biometrijske čitače i slično. Umesto termina autentifikacija u literaturi se često koristi i termin identifikacija.

Poverljivost ili tajnost podataka obezbeđuje zaštitu podataka od neovlašćenih lica. Podatke kroz mrežu treba slati u kriptovanom obliku, osim toga, podaci se, takođe, trebaju i čuvati u kriptovanom obliku. Poverljivost se najčešće ostvaruje kriptovanjem podataka ili fizičkom zaštitom komunikacione linije.

Servis neporicljivosti pruža prevenciju od lažnog poricanja slanja date poruke/dokumenta. Isto tako, sprečava pojave da primalac poruke sam izmeni sadržaj iste i da tvrdi da je takvu primio. Najčešće korišćeni mehanizam koji obezbeđuje zloupotrebe ove vrste je digitalni potpis.

Integritet poruka obezbeđuje konzistentnost i tačnost poruke, odnosno onemogućava bilo kakvu izmenu, uništenje ili lažno generisanje poruke od strane neovlašćene osobe. Ovaj servis obezbeđuju heš (eng. *hash*) funkcije, kontrolne sume, logovanja i razni kodovi.

Autorizacija je servis koji vrši proveru da li je identifikovanom korisniku dozvoljen pristup određenim podacima, a servis kontrole pristupa utvrđuje prava pristupa korisnika. Za realizaciju ovog servisa potrebno je omogućiti postavljanje privilegija objektima koji im pripadaju i sprečiti korisnike sistema da pristupaju korišćenjem prava pristupa drugih korisnika.

Raspoloživost resursa se odnosi na reakciju u cilju održavanja funkcionalnosti resursa u slučaju detekcije otkaza ili napada. Ovaj servis se obezbeđuje uvođenjem redundansa i primenom tehnika za toleranciju otkaza ili primenom redundanse [2], [3].

2 Kriptografija

2.1 Osnovni pojmovi i terminologija u kriptografiji

Termini *Plaintext* ili *Cleartext* odnose se na izvornu poruku ili izvorni tekst koji je moguće pročitati i razumeti bez primene bilo kakvih posebnih metoda. Ukoliko se takav tekst treba preneti sa mesta A do mesta B, on se naziva poruka. Poruka se može slati putem računarske mreže kao izvorni tekst (svima razumljiva) ili kao nerazumljiv sadržaj koji se naziva šifrovan ili kriptovan tekst (eng. *ciphertext* ili *chipher*).

Postupak pomoću koga se izvorni tekst transformiše u šifrovan tekst se naziva kriptovanje (eng. *encryption*). Kriptovanje se koristi da bi se obezbedilo da nijedan korisnik, osim korisnika kome je poruka namenjena, ne može da sazna sadržaj poruke [4]. Ako neovlašćeni korisnici dođu u posed kriptovanog teksta i vide njegov sadržaj ne mogu pročitati izvorni tekst. Kriptovanje izvornog teksta se obavlja pomoću određenog pravila za kriptovanje odnosno kriptografskog algoritma. Svaki kriptografski algoritam kao ulazne podatke ima izvorni tekst i ključ a kao izlaz daje kriptovani tekst.

Postupak koji omogućava da se od kriptovanog teksta dobije originalni izvorni tekst naziva se dekriptovanje (eng. *decryption*). Dekriptovanje odnosno dešifrovanje predstavlja inverzni postupak od kriptovanja. Kriptovani tekst za koji nije poznat ključ zove se kriptogram.

Nauka koja proučava kriptovanja i dekriptovanja podataka se naziva Kriptografija. Kriptografija se oslanja na matematiku i omogućava čuvanje važnih podataka kao i njihov prenos preko računarske ili telekomunikacione mreže a da pri tome niko ne može da ih pročita osim korisnika kome su namenjeni. Dok se Kriptografija bavi zaštitom podataka, Kriptoanaliza je nauka o otkrivanju odnosno "razbijanju" kriptovanih poruka. Objedinjene Kriptografija i Kriptoanaliza se nazivaju Kriptologija.

Protokol predstavlja skup pravila i konvencija koji definiše komunikacioni okvir između dva ili više učesnika u komunikaciji. Tu spadaju: uspostavljanje veze, održavanje veze, raskid veze i obnavljanje veze u slučaju prekida. Kriptografski protokoli se upotrebljavaju za uspostavljanje sigurne komunikacije preko nepouzdanih globalnih mreža i distribuiranih sistema. Oslanjaju se na kriptografske metode zaštite kako bi korisnicima obezbedili osnovne sigurnosne usluge poverljivosti, integriteta i neporicljivosti.

Treba napomenuti razlike između termina kodiranje i šifrovanje. Pojam kodiranje se odnosi na transformaciju izvornog teksta koje se vrši na osnovu obimne „knjige“ kodova, u kojoj se reči i fraze zamenjuju slučajnim nizom znakova. Na primer, "UWRT" može biti kod za "Ja se zovem Petar". Nasuprot tome, šifra radi na nižem nivou: na nivou pojedinačnih slova, malih grupa slova, ili u modernim šemama nad pojedinačnim bitovima. Uz to se umesto “knjige” kodova koriste algoritmi koji su utemeljeni nekom matematičkom formulom.

2.2 Kratak istorijski pregled, opšti pojmovi i podela u kriptografiji

Kao što je već napomenuto, kriptografija je nauka koja se bavi očuvanjem tajnosti informacija i podataka. Kriptografija kao sredstvo za zaštitu informacija datira još od pojave prvih pisama, kada je bilo neophodno preneti poruku na daljinu i to sačuvanu od tuđih neželjenih pogleda.

Prve metode kriptovanja je koristio Julije Cezar kada je slao poruke svojim vojskovođama. On je te poruke šifrovao tako što je pojedina slova u tekstu pomerao za tri, četri ili više mesta u abecedi. Takvu poruku mogli su da dešifruju samo oni koji su poznavali pravilo pomeranja. Prva poznata studija o kriptografiji pojavila se 1467 godine a napisao je italijanski arhitekta *Leone Batista*. On je takođe tvorac

takozvanog šifarskog kruga i nekih drugih rešenja dvostrukog prikrivanja teksta koja su u XIX veku prihvatili i kasnije usavršavali nemci, englezi i francuzi. Pola veka nakon toga pojavljuje se prva knjiga iz oblasti kriptografije. U toku drugog svetskog rata na Nemačkoj strani pojavila se mašina koja je šifrovala poruke na do tada još neviđen način. Nemci su mašinu nazvali Enigma. Međutim ma koliko je ona u to vreme bila revoluciona saveznici su uspeli da razbiju poruke šifrovane Enigmom. Pojavom prvih računara kriptografija se naglo razvija. Kako je vreme odmicalo računari su bili sve brži i brži, radeći i po nekoliko stotina, a kasnije i miliona operacija u sekundi. Novom brzinom rada je omogućeno "probijanje" šifara za sve manje vremena. Uporedo s tim, radilo se i na razvoju novih, sigurnijih i komplikovanijih šema za šifrovanje [6].

Pojavom računarskih mreža kriptografija naglo dobija na značaju. Naročito je bitno obezbediti zaštitu važnih podataka (na primer finansijskih) koji se prenose mrežom. Naime, podaci se razmenjuju računarskom mrežom u formi paketa podataka i oni dospevaju do većeg broja računara na putu od polaznog do odredišnog računara. Na svakom usputnom računaru moguće je te pakete podataka "uhvatiti" i pročitati njihov sadržaj, korišćenjem analizatora protokola ili nekog programa (*sniffera*). Kriptovanje podataka podrazumeva korišćenje raznih kriptografskih algoritama tj. skupova pravila po kojima se vrši kriptovanje. Algoritmi za kriptovanje se mogu podeliti u dve grupe:

- Tajni algoritmi: bezbednost se zasniva na tajnosti algoritma (istorijski interesantni).
- Algoritmi zasnovani na ključu: bezbednost se zasniva na ključevima, a ne na detaljima algoritma koji se može publikovati i analizirati. Ovde je algoritam javno poznat, a ključ se čuva u tajnosti, da nije tako korisnici bi morali da razumeju i ključ i algoritam. Ključ je niz podataka koji se koristi za kriptovanje drugih podataka i koji, prema tome, mora da se koristi i za dekriptovanje podataka.

Danas se najviše koriste algoritami za kriptovanje zasnovani na ključu, a mogu se klasifikovati u tri grupe:

- Simetrični, kod kojih se koristi jedan ključ,
- Asimetrični, kod kojih postoje dva ključa i
- Hibridni, kombinacija predhodna dva.

U cilju postizanja što bolje zaštite podataka algoritam za kriptovanje mora zadovoljiti sledeće zahteve:

- Cena "probijanja" algoritma mora da bude veća od cene šifrovanih podataka;
- Vreme potrebno za "probijanje" algoritma mora da bude duže od vremena u kome podaci moraju da ostanu tajni;
- Broj podataka kriptovanih pomoću jednog ključa mora da bude manji od broja potrebnih podataka da se dati algoritam "probije" [1].

Isto tako, prilikom formiranja algoritama za kriptovanje/dekriptovanje, teži se da postupci kriptovanja odnosno dekriptovanja budu identični tj. inverzni. Na taj način se postiže neophodna kompatibilnost između postupaka kriptovanja i dekriptovanja u smislu korišćenja istih operacija ali obrnutim redosledom i korišćenje istog ključa u oba postupka.

Veoma je važno i da proces kriptovanja/dekriptovanja ima što je moguće kraće vreme izvršavanja. Postizanje što boljih performansi se ostvaruje ukoliko se ovi procesi realizuju hardverski. Kako bi algoritmi za kriptovanje bili što jednostavniji za hardversku realizaciju potrebno je da se izračunavanja u okviru njih baziraju na skupu jednostavnih operacija, kao što su aritmetičko sabiranje, XOR, operacije rotiranja i druge.

2.3 Bit stringovi, stream šifrovanje i blok šifrovanje

Najjednostavnije metode za šifrovanje teksta koriste klasičnu zamenu slova, odnosno kodiraju poruku u sekvencu binarnih cifara (bitova). Najpoznatija metoda za kodiranje teksta je ASCII (eng.

American Standard Code for Information Interchange). Ova metoda kodira izvorni tekst u odgovarajuće bit stringove.

Algoritam za kodiranje na osnovu izvornog teksta i određenog pravila formira bit stringove. Otuda se nameće podela na šifarske redove, gde se izvorni podaci kodiraju metodom bit po bit, i šifarske blokove, gde se izvorni podatak deli na blokove određene veličine. Ilustracije radi ASCII metoda koristi osam bitova radi reprenzetacije jednog karaktera, odnosno za šifarski blok koji se sastoji od 8 karaktera, primenom ASCII algoritma se dobija blok veličine 64 bita [8].

Jedan bit string može biti zapisan na više načina, što zavisi od veličine blokova na koje se string deli. Na primer, sekvencu 100111010110 možemo podeliti u blokove veličine tri bita: 100 111 010 110. Takođe, budući da bit string dužine tri bita predstavlja jednu BCD cifru iz opsega od 0 do 7, naša sekvenca se može zapisati i u obliku 4 7 2 6.

Ukoliko istu sekvencu podelimo u blokove veličine 4 bita: 1001 1101 0110, i četiri bita predstavljaju brojeve od 0 do 15 imamo sekvencu 9 13 6.

Ako je veličina bloka unapred zadata, za male brojeve, neophodno je dodavati nule počev od MSB bita. Na primer, binarna reprezentacija broja 5 je 101, a koristi se blok veličine 6 bita, potrebno je 5 predstaviti kao 000101, ili ako je veličina bloka 8 bitova broj 5 bi bio 00000101. Dosta korišćena metoda je i predstavljanje bit stringova korišćenjem HEX notacije (*Hexadecimal*) od po 4 bita.

Tokom izračunavanja šifarski algoritam često koristi operacije nad bit stringovima kao što je XOR ili Exclusive OR. Ova jednostavna operacija se obavlja nad dva operanda iste dužine.

Mnogi autori često spominju termin *stream* šifrovanja, misleći pri tom na metode koje se zasnivaju na šifrovanju pojedinačnih reči ili karaktera iz teksta proizvoljne dužine. Znači izvorni tekst se šifruje reč po reč ili karakter po karakter, gde je pravilo kodiranja svake reči ili karaktera određeno pozicijom u okviru izvornog teksta. Međutim moderniji način korišćenja engleskog termina *stream* šifrovanja se odnosi na šifrovanje izvornog teksta bit po bit. Sve što se može desiti svakom pojedinačnom bitu je da promeni vrednost ili da ostane nepromenjen. Zadatak onoga ko želi dešifrovati ovako kodirani tekst, bio bi da se identifikuju sve pozicije na kojima su bitovi promenjeni, i promenjene vrednosti treba vratiti na originalne vrednosti. Da bi izvorni tekst mogao da se dešifruje na jednostavan način, neophodno je poznavati šemu koja identifikuje ili na neki način predviđa promene bitova. Jedna ovakva šema se može predstaviti u obliku ključa 1000110 i izvornog teksta 1100101. Ustanovimo pravilo da ukoliko je jedinica u okviru ključa to znači da treba promeniti bit u okviru izvornog teksta koji je na istoj poziciji. Prateći ovo pravilo dobija se šifrovani niz 0100011. Ukoliko se isti ključ primeni dva puta ponovo se dobija izvorni tekst [8]. Ovo znači da je postupak dešifrovanja identičan procesu šifrovanja. Pošto operacija zamene bitova odgovara XOR funkciji može se koristiti formula koja prati postupak kodiranja:

$$C_i = P_i \; XOR \; K_i,$$

gde su C_i kodiran tekst, P_i izvorni tekst i K_i ključ koji se formira kao slučajno definisana vrednost, koja je poznata samo pošiljaocu i primaocu poruke.

Stream šifrovanje zahteva dugačke nizove ključa, za čije formiranje se koriste generatori koji proizvode dugačke alfabetske nizove nadovezivanjem kratkih ključeva. Ovaj način nije najbolje rešenje zato što postoji velika verovatnoća da će se jedan isti kratak ključ ponoviti veći broj puta u dugačkom nizu ključeva, što olakšava otkrivanje celog ključa. Zato generatori ključeva koji se praktično koriste prilikom *stream* šifrovanja koriste sofisticiranije metode. Projektovanje kvalitetnog generatora ključa je često veoma težak prblem, koji zahteva korišćenje veliki broj matematičkih formula.

Dobra strana *stream* ovakvog načina šifrovanja je ta da ukoliko do odredišta stigne pogrešna informacija, nakon dešifrovanja doći će do gubitka samo onih bitova izvornog teksta koji su izmenjeni u prenosu. To znači da ovaj način šifrovanja nije podložan greškama u prenosu podataka usled povećanog nivoa šuma. Ova osobina čini stream šifrovanje jako povoljno za korišćenje u okviru GSM mobilnih mreža. Ostale prednosti se ogledaju u veoma velikoj brzini šifrovanja/dešifrovanja i lakoći implementacije.

2.4 Blok šifrovanje

Blok šifrovanja se koriste za šifrovanje kratkih poruka kao što su ključevi, lozinke, potpisi ili autentifikacije korisnika. Kod ovog načina šifrovanja, izvorni podatak se deli u blokove simbola određene veličine koji se individualno i nezavisno kodiraju u cilju formiranja šifrovanog bloka podataka. Blok šifrovanje odlikuju sledeće osobine:

- Svaki blok simbola se šifruje uvek na isti način, nezavisno od mesta koje zauzima u poruci.
- Jednake poruke, šifrovane sa istim ključem, uvek daju jednake šifrovane poruke.
- Da bi se dešifrovao deo poruke, nije neophodno dešifrovati je od početka, dovoljno je dešifrovati blok koji nas interesuje.

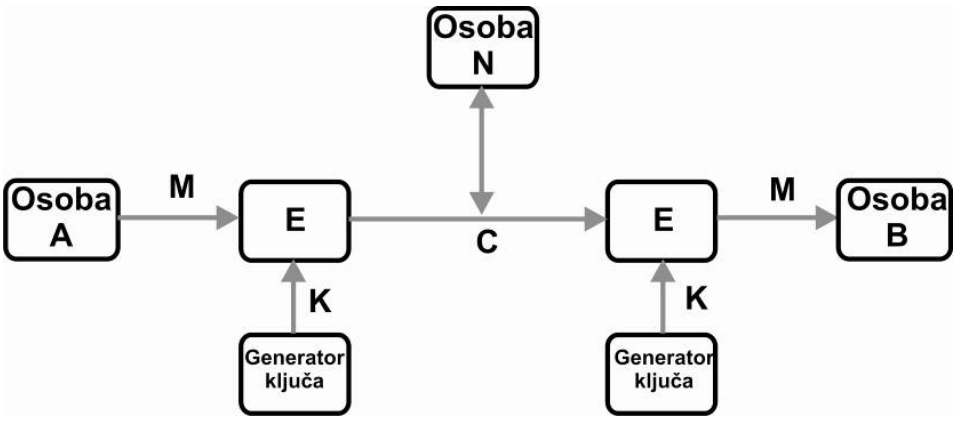
Kada se ima u vidu način rada, blok šifre se mogu javiti u sledećim oblicima [8]:

- Direktna upotreba blok-šifre ECB (eng. *Electronic Codebook*)
- Ulančavanje šifrovanih blokova CBC (eng. *Cipher Block Chaining*).
- Šifrovanje pomoću povratne sprege CFB (eng. *Cipher Feedback*).
- Formiranje izlaza u povratnoj sprezi OFB (eng. *Output Feedback*).

Blok šifrovanje je primenjeno u sledećim algoritmima: LUCIFER, DES, FEAL, IDEA, RC5, SKIPJACK, BLOWFISH, TWOFISH, AES (RIJNDAEL), itd.

2.5 Simetrična kriptografija

Osnovna osobina simetričnih kriptosistema ili kriptosistema sa tajnim ključem (*Symetric-Key* ili *Secret-key Criptosystems*) je da se za kriptovanje/dekriptovanje poruka koristi isti ključ. Na Slici 2.1 je, u vidu blokova, prikazan princip rada simetričnog kriptosistema. Osoba *A* ima za cilj slanje poruke *M* osobi *B* preko nezaštićenog komunikacionog kanala. Osoba *A* najpre generiše poruku *M* (izvorni tekst) koja se upućuje u blok za šifrovanje *E*. U ovom bloku se vrši kriptovanje poruke *M* uz korišćenje ključa *K* dobijenog uz pomoć generatora ključa. Na taj način se kreira kriptovana poruka *C*. Potom se tako dobijena poruka komunikacionim kanalom šalje do osobe *B* [9], [10].



Slika 2.1
Blokovski prikaz simetričnog kriptosistema

Postupak dekriptovanja se obavlja inverznim postupkom od kriptovanja u bloku za dekriptovanje *D*. Dekriptovanje poruke *C* se vrši pomoću istog ključa *K* koji je korišćen prilikom kriptovanja. Nakon dekriptovanja se dobija izvorna poruka *M*.

Ukoliko na kanalu za prenos postoji osoba *N* (napadač) može da presretne kriptovanu poruku i ukoliko dođe u posed ključa može pročitati ili zloupotrebiti izvornu poruku. Da bi se izbegle manipulacije, obe strane moraju držati ključ u tajnosti, odnosno ključ se ne sme prenositi nezaštićenim komunikacionim kanalom. Za razliku od ključa, kriptovana poruka može da se šalje i po nezaštićenom kanalu s obzirom na to da sadržaj izvorne poruke može da protumači samo onaj korisnik koji ima ključ. Primer slanja ključa različitim komunikacionim kanalom je kada se PIN (eng. *Personal Identification Number*) kod za pristup zaštićenim Internet sajtovima (npr. Banke za uvid stanja na računu) dostavljaju korisnicima poštom, a ne Internetom.

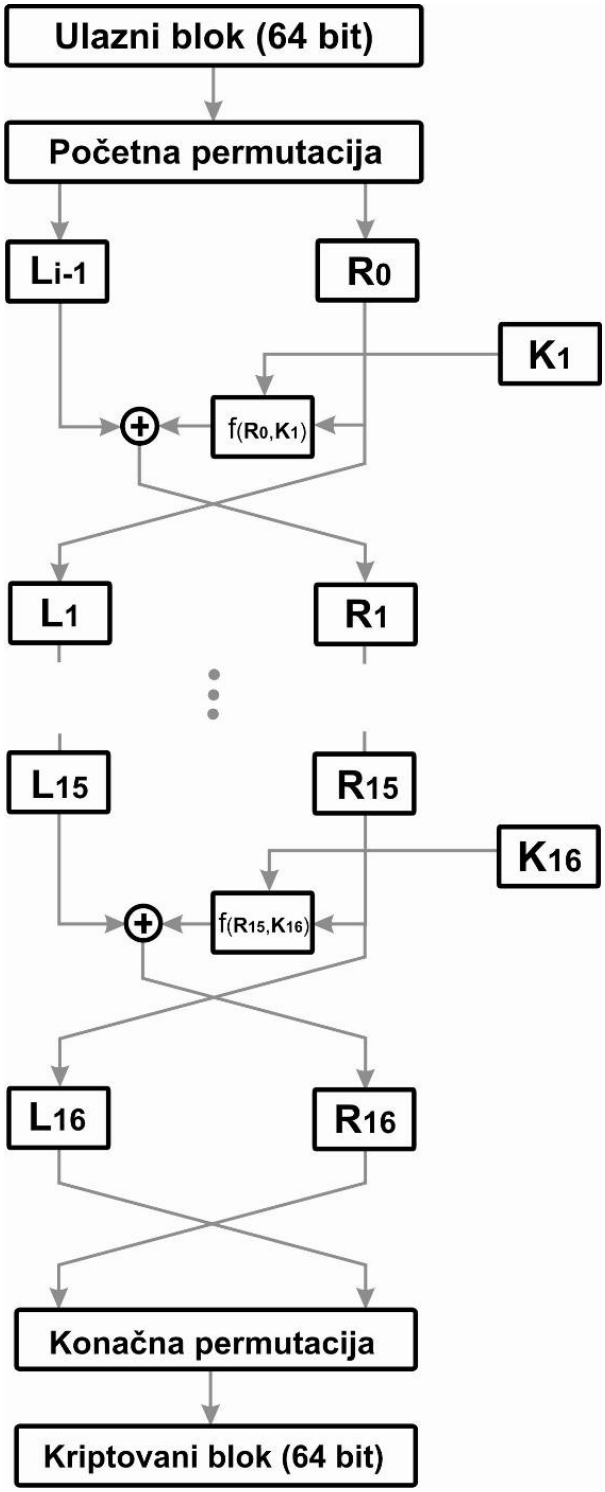
Najpoznatiji algoritmi simetričnih kriptosistema koji se danas koriste su: DES, 3DES, DES-CBC, IDEA, RC5, RC6, AES i drugi.

2.5.1 DES algoritam

Početkom šezdesitih godina prošlog veka, kompanija IBM je pokrenula istraživački projekat u cilju zaštite podataka pod nazivom *Lucifer*. Ovaj projekat okončan je 1971 godine i *Lucifer* je bio prvi šifrat sa blokovima veličine 64 bita koji je koristio ključ od 128 bita. Kompanija je kasnije komercijalizovala ovaj način kodiranja i nazvala ga DES (eng. *Data Encryption Standard*). 1976 godine DES je prihvaćen kao federalni standard za enkripciju podataka i korišćen je u komunikacijama Američke vlade. DES je narednih dvadesetak godina bio najviše korišćeni standard na svetu. Tokom eksploatacije, DES standard je bio modifikovan i unapređivan svakih pet godina. Svoju slavu završio je nakon mnogo debata kasnih devedesetih. Nasledio ga je 2001. god. AES (eng. *Advanced Encryption Standard*), takođe poznat pod nazivom *Rijndael* algoritam. U poređenju sa DES novi algoritam je bio dosta napredniji po pitanju sigurnosti podataka. Danas, DES algoritam i dalje koristi veliki broj organizacija u svetu čime je nastavio život pružajući zaštitu u mrežnim komunikacijama, skladištenjima podataka, lozinkama i sistemima za kontrolu pristupa [11], [12].

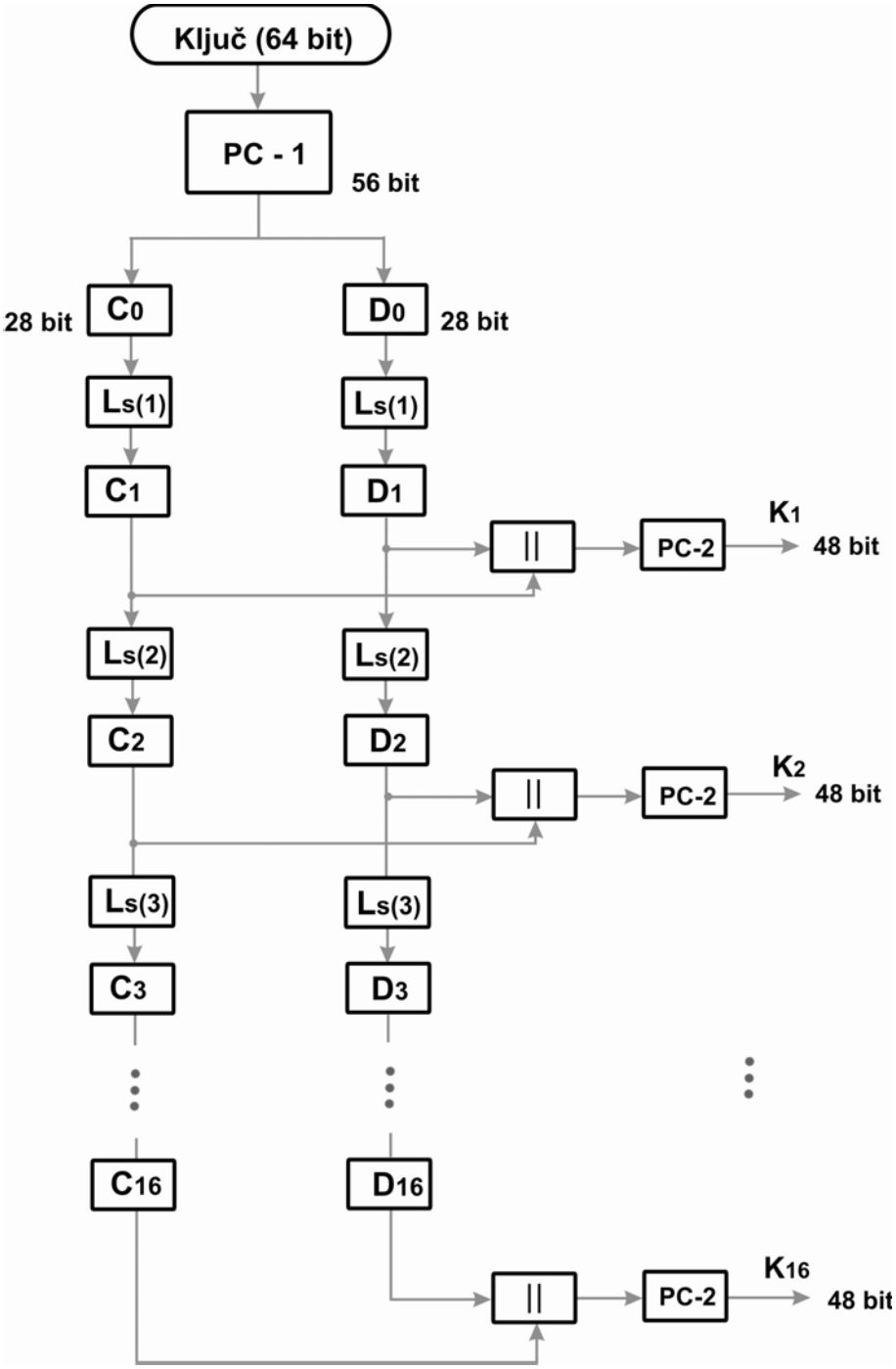
DES predstavlja simetrični algoritam za kriptovanje blokovskog tipa, odnosno predstavlja direktnu upotrebu blok-šifre (ECB mod). Kao ulaz u algoritam se koristi blok od 64-bita izvornog teksta i 56-bitni ključ. Izlaz iz algoritma je 64-bitni kriptovan tekst koji se dobija nakon 16 iteracija koje se sastoje od identičnih operacija. Ključ od 56 bita se formira od inicijalnog 64-bitnog ključa informacije ignorisanjem svakog 8 bita, tj. odsecanjem ukupno 8 bitova. Na Slici 2.2 je prikazan izgled DES algoritam za kriptovanje.

Kriptovanje pomoću DES algoritma se sprovodi u nekoliko koraka. Prvo se bitovi ulaznog bloka dužine 64 bita permutuju početnom permutacijom. Radi se o permutaciji koja jednostavno vrši zamenu pesta bitova Permutovan ulazni blok deli na dva dela od po 32 bita, levi L_0 i desni R_0 deo. Nad desnim delom bloka se obavlja funkcija $f(R_0,K_1)$, gde je R_0 desnih 32 bita, a K_1 je 48-bitni ključ. Ova funkcija generiše 32-bitni rezultat. Nad dobijenim rezultatom funkcije f i L_0 vrši se operacija XOR. Rezultat XOR operacije predstavlja novu 32-bitnu vrednost R_1 koja se koristi za dalje operacije. Kao levi deo L_1 se koristi vrednost R_0 iz prethodne iteracije. Nakon ponavljanja 16 istovetnih koraka, blokovi međusobno menjaju mesta te se spajaju. Na kraju se obavlja konačna permutacija koja je inverzna početnoj. Dobijena 64-bitna vrednost čini kriptovani blok podataka [11].



Slika 2.2
Izgled DES algoritma za kriptovanje

Prilikom kriptovanja/dekriptovanja u svakoj iteraciji se koriste različiti ključevi K_1, ..., K_{16} veličine 48-bita. Za generisanje ovih ključeva se koristi poseban algoritam. Na Slici 2.3 ilustrovan je postupak dobijanja ključeva koji se koriste prilikom DES kriptovanja/dekriptovanja.



Slika 2.3
Formiranje ključeva u okviru DES algoritma

Postupak generisanja šesnaest 48-bitnih delova ključeva (K_1, K_2, … ,K_{16}) od zadatog tajnog ključa sprovodi se u nekoliko koraka. Prvo se iz 64-bitnog ključa odstrane bitovi parnosti tj. *8-*, *16-*, *24-*, 32-, *40-*, *48-*, *56-* i 64-bit, zatim se pomoću zadate tabele permutacije PC-1 (Tabela 1) iz ključa generišu dva bloka po 28 bita (C_i i D_i). Blokovi se formiraju tako što svaki bit sa pozicije *i* ide na poziciju koja se čita iz *i*-te stavke ove tabele. Nakon toga, sledi 16 koraka u okviru kojih se svaki blok rotira ulevo za određeni broj bita (*Ls(i)* zavisi o kom koraku se radi). Nastali blokovi se spajaju (operacija ||), a onda se pomoću permutacije koja je definisana tabelom PC-2 (Tabela 2) generše odgovarajući deo ključa K_i, gde je *i* redni broj koraka [11].

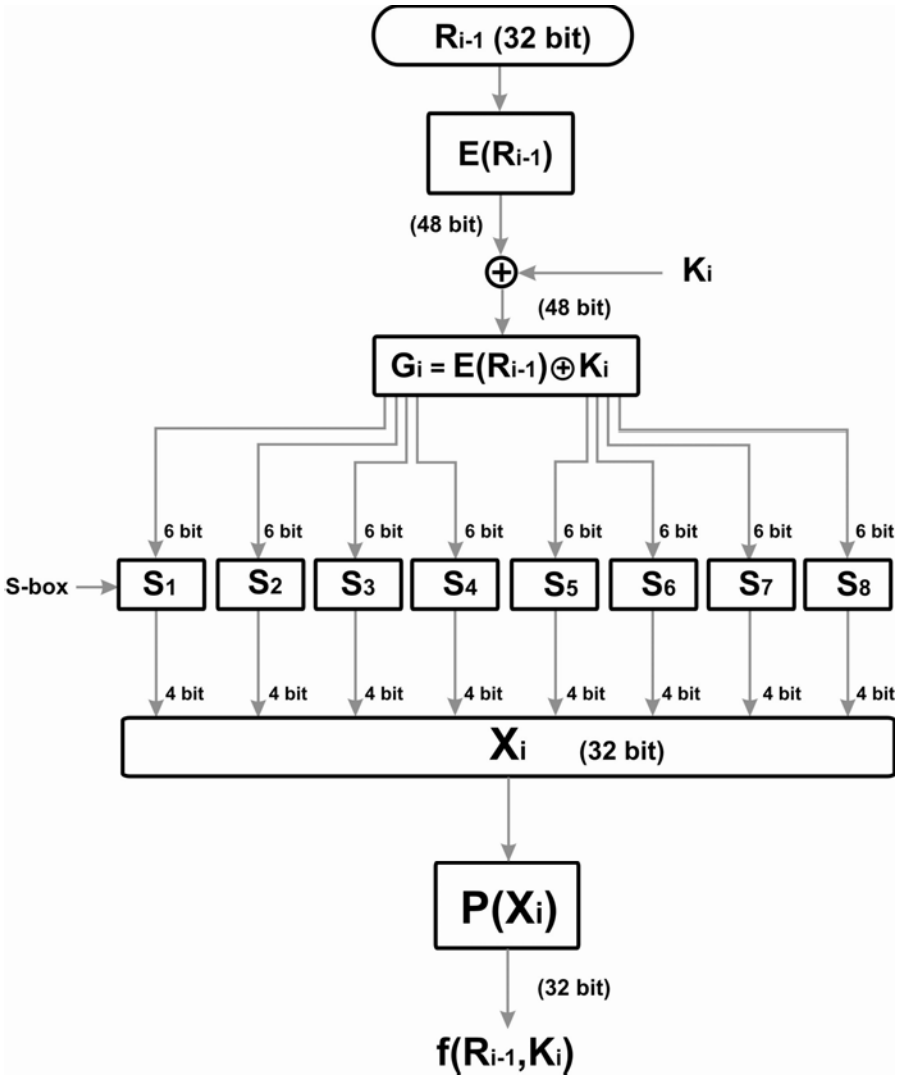
Tabela 1
Tabela permutacije PC-1

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

Tabela 2
Tabela permutacije PC-2

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

Suštinski, a ujedno i najkritičniji deo algoritma predstavlja funkcija kriptovanja *f* tj. *Feistel* funkcija. Zbog njene kompleksnosti pretpostavlja se da ne postoji način razotkrivanja DES-a. Funkcija *f* se formira u nekoliko koraka, što je prikazano na Slici 2.4.



Slika 2.4
Formiranje pomoćne DES funkcije f

Od ulaznih 32-bitnih R_i se proširenjem, na osnovu tabele *E*, dobija podatak dužine 48 bita. Dobijena vrednost se bit po bit "sabira" sa ključem K_i (operator " $\oplus$ " predstavlja XOR logičku operaciju). Rezultat je 48-bitna vrednost koja se deli na osam delova od po šest bita. Tako dobijene vrednosti se vode u osam blokova za supstituciju tkz. S-box (S_1,S_2, …,S_8). Svaki S-box ulaznih 6 bita zamenjuje sa 4 bita na izlazu. Kao primer, u okviru Tabele 3, je prikazana tabela substitucije S_5. Prvi i zadnji bit svakog 6-bitnog dela informacije na ulazu u S-box predstavljaju adresu reda, a srednja četiri adresu kolone u tabeli substitucije. Slično se formiraju i ostali S-box-ovi [11], [12].

Tabela 3

Primer tabele za substituciju S_5

S_5	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x10100x	x1101x	x1110x	x1111x
0yyyy0	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
0yyyy1	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
1yyyy0	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
1yyyy1	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

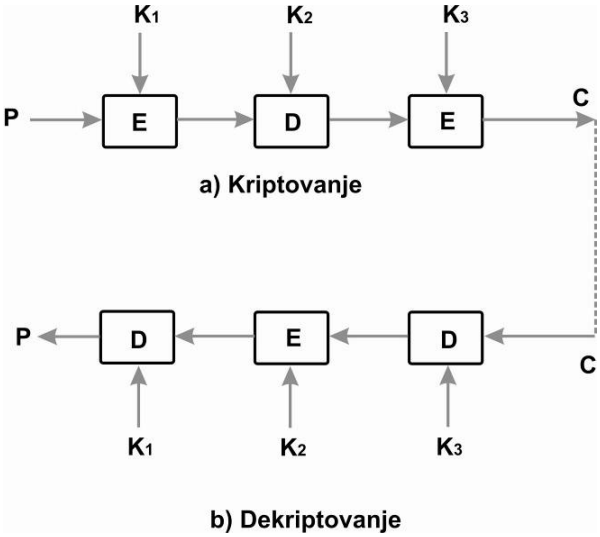
Istim postupkom nad svakom šestorkom od ulaznih 48 bita substitucijom se dobija ukupno 32 bita. Tako formiranih 32 bita dodatno se permutuje zadatom tabelom $P(X_i)$ te se dobija konačna vrednost funkcije *f*. Budući da je algoritam za kriptovanje identičan algoritmu za dekriptovanje, postupak dekriptovanja se može sprovesti tako što se operacije korišćene u postupku kriptovanja obavljaju obrnutim redosedom. Zbog simetričnosti algoritma to se postiže tako što se kriptovani blok pusti kroz isti algoritam s tom razlikom da se umesto ključa K_i u i-tom koraku upotrijebi ključ K_{16-i+1}.

Zanimljivo je da se napomene jedna slabost vezana za DES algoritam. Naime, zbog samog načina kreiranja delova ključeva, postoje četiri ključa za koje je dekripcija jednaka enkripciji. To znači, ako se kriptuje neka poruka dva puta i to sa baš jednim od ta 4 ključa, kao rezultat se dobija originalna poruka. Kako je verovatnoća da se od svih mogućih ključeva odabere baš neki od tih mala, to ne utiče na sigurnost algoritma.

DES algoritam koristi 56-bitni ključ za kriptovanje što znači ukupno 72.057.594.037.927.936 mogućnosti za osobu koja hoće da probije tkz. brutalnim napadom tekst kriptovan sa DES-om. 1999. god. objedinjene grupe *EFF* i *Distributed.net* kao DES *Cracker* uz pomoć 100.000 PC računara na Internetu, probili su DES za vreme 22 sata i 15 minuta. Pritom je brzina pretraživanja bila 245 biliona ključeva/sekundi [13].

2.5.2 Trostruki DES (3-DES) algoritam

Trostruki DES algoritam predstavlja unapređeni DES standard za kriptovanje podataka i najviše se koristi u okviru aplikacija koje se baziraju na internetu. To je DES algoritam, koji koristi dva ili tri različita DES ključa. Na Slici 2.5 je prikazan način kriptovanja/dekriptovanja podataka P pomoću tri ključa (K_1,K_2 i K_3).



Slika 2.5
Postupci 3-DES: a) Kriptovanja; b) Dekriptovanja

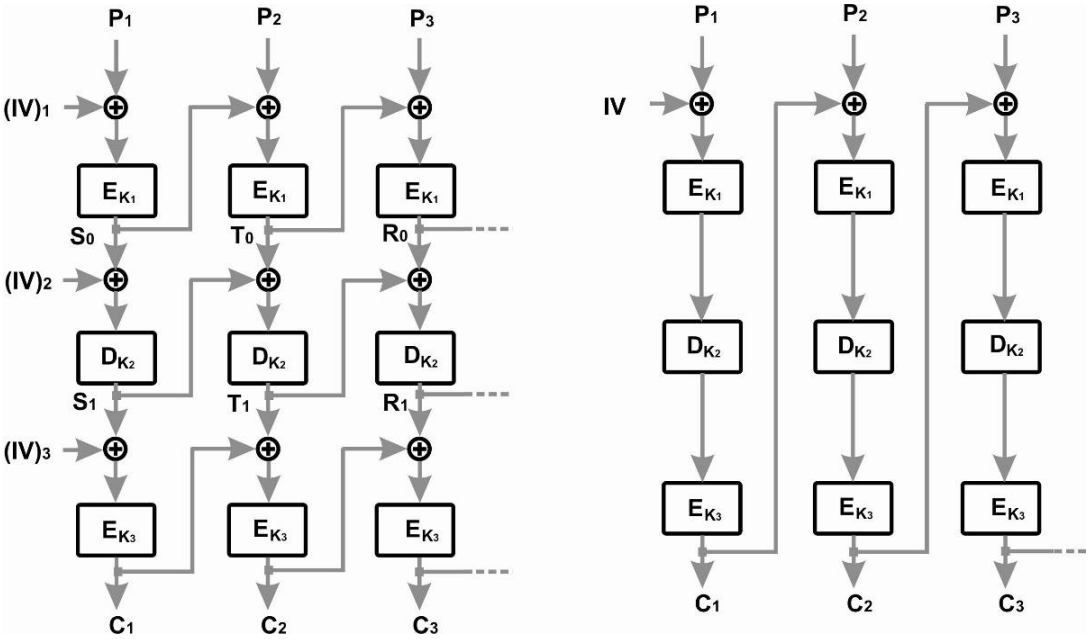
Prvi ključ K_1 se koristi za kriptovanje bloka podataka izvorne poruke P pomoću standardnog DES algoritma. Tako kriptovana poruka se dekriptuje drugim ključem K_2. Dekriptovanjem poruke sa ovim ključem se dobija nova šifrovana poruka. Na kraju se razultat dekriptovanja opet kriptuje, ovaj put ili trećim ključem K_3 ili opet prvim K_1. Tako je konačno formirana kriptovana poruka C. Naizmeničnim korišćenjem različitih ključeva povećava se efektivna dužina ključa na ukupno 168 bita, a tako i broj kombinacija koje bi eventualni napadač morao probati da bi došao do izvorne poruke. Broj kombinacija za dva različita ključa je 2^{112}, dok za 3 različita ključa ima čak 2^{168} kombinacija. 3-DES algoritam, kako ga još nazivaju, rešava problem dužine ključa običnog DES-a, a sa druge strane nedostatak mu je to što je mnogo sporiji od standardnog DES-a [11].

Dekriptovanje poruke koja je kriptovana pomoću 3-DES algoritma se, kao i kod klasičnog DES-a, obavlja inverznim funkcijama u odnosu na kodiranje. Prvo se obavlja dekriptovanje pomoću trećeg ključa, sledi kriptovanje pomoću ključa K_2 i na kraju se izvorna poruka P dobija postupkom dekriptovanja pomoću ključa K_1.

2.5.3 DES-CBC algoritam sa inicijalizacionim vektorom

DES-CBC predstavlja DES algoritam koji prilikom kriptovanja koristi ulančavanje šifrovanih blokova CBC. Ovaj algoritam se koristi kao pouzdan mehanizam u zaštiti IP paketa. Ova varijacija DES algoritma koristi inicijalizacioni vektor, IV, dužine 64 bita, tj. iste dužine kao i blok koji se želi kriptovati. IV mora biti slučajna vrednost kako ne bi došlo do formiranja dva identična kriptovana podatka.

Postoji više načina kriptovanja pomoću DES-CBC algoritma kao što su: unutrašnji CBC i spoljašnji CBC, koji su prikazani na Slici 2.6. Razlika između ova dva algoritma je ta što unutrašnji CBC algoritam zahteva tri različita inicijalizaciona vektora (IV_1, IV_2 i IV_3) a spoljašnji CBC samo jedan IV.



Slika 2.6
Unutrašnji i spoljašnji CBC algoritmi

Na početku se blok izvornog teksta P podeli u delove $P_1,P_2,..,P_n$ i ti delovi se saberu sa IV pomoću XOR logičke operacije, bez obzira o kom se načinu kriptovanja radi [11]. Slede tri koraka: kriptovanje pomoću DES algoritma i ključa K_1, zatim dekriptovanje pomoću DES algoritma i ključa K_2 i ponovo kriptovanje pomoću DES algoritma i ključa K_3. Matematički modeli koji opisuju spoljašni CBC algoritam za kriptovanje su:

$$C_1 = E_{K3}(D_{K2}(E_{K1}(P_1 \oplus IV)))$$
$$C_2 = E_{K3}(D_{K2}(E_{K1}(P_2 \oplus C_1)))$$
$$C_3 = E_{K3}(D_{K2}(E_{K1}(P_3 \oplus C_2)))$$

Međurezultati koji se koriste prilikom kriptovanja metodom unutrašnjeg DES-CBC-a su opisani sledećim formulama:

$$S_0 = E_{K1}(P_1 \oplus (IV)_1) \qquad T_0 = E_{K1}(P_2 \oplus S_0) \qquad R_0 = E_{K1}(P_3 \oplus T_0)$$
$$S_1 = D_{K2}(S_0 \oplus (IV)_2) \qquad T_1 = D_{K2}(T_0 \oplus S_1) \qquad R_1 = D_{K2}(R_0 \oplus T_1)$$

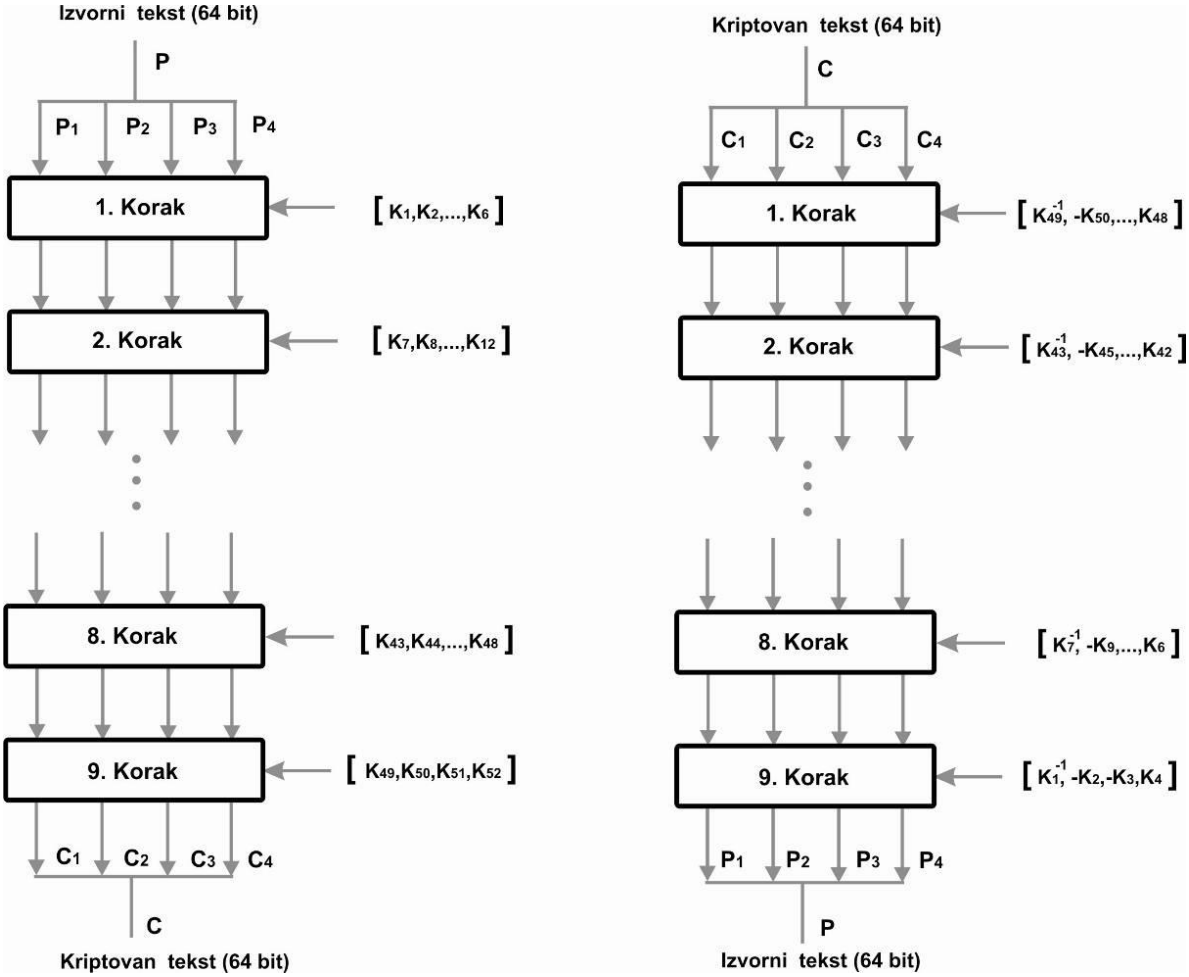
Formule koje opisuju kriptovanje metodom unutrašnjeg CBC-a su:

$$C_1 = E_{K3}(S_1 \oplus (IV)_3) \qquad C_2 = E_{K3}(T_1 \oplus C_1) \qquad C_3 = E_{K3}(R_1 \oplus C_2)$$

2.5.4 IDEA algoritam

1990. godine *Xuejia Lai* i *James Massey* sa Švajcarskog državnog instituta za tehnologiju su ustanovili novi sistem za šifrovanje blokova podataka. Originalna verzija ovog algoritma za enkripciju je nazvana *Proposed Encryption Algorithm* (PES). Kasnije je PES preimenovan u IDEA. Ovaj algoritam koristi 128-bitni ključ za kriptovanje blokova od po 64 bita podataka [14].

Ilustracija IDEA standarda za kriptovanje/dekriptovanje podataka je prikazana na Slici 2.7. Kao i kod svih ostalih blokovskih šifrovanja, i ovde imamo dve ulazne informacije i to izvorni tekst P i ključ za enkripciju K.



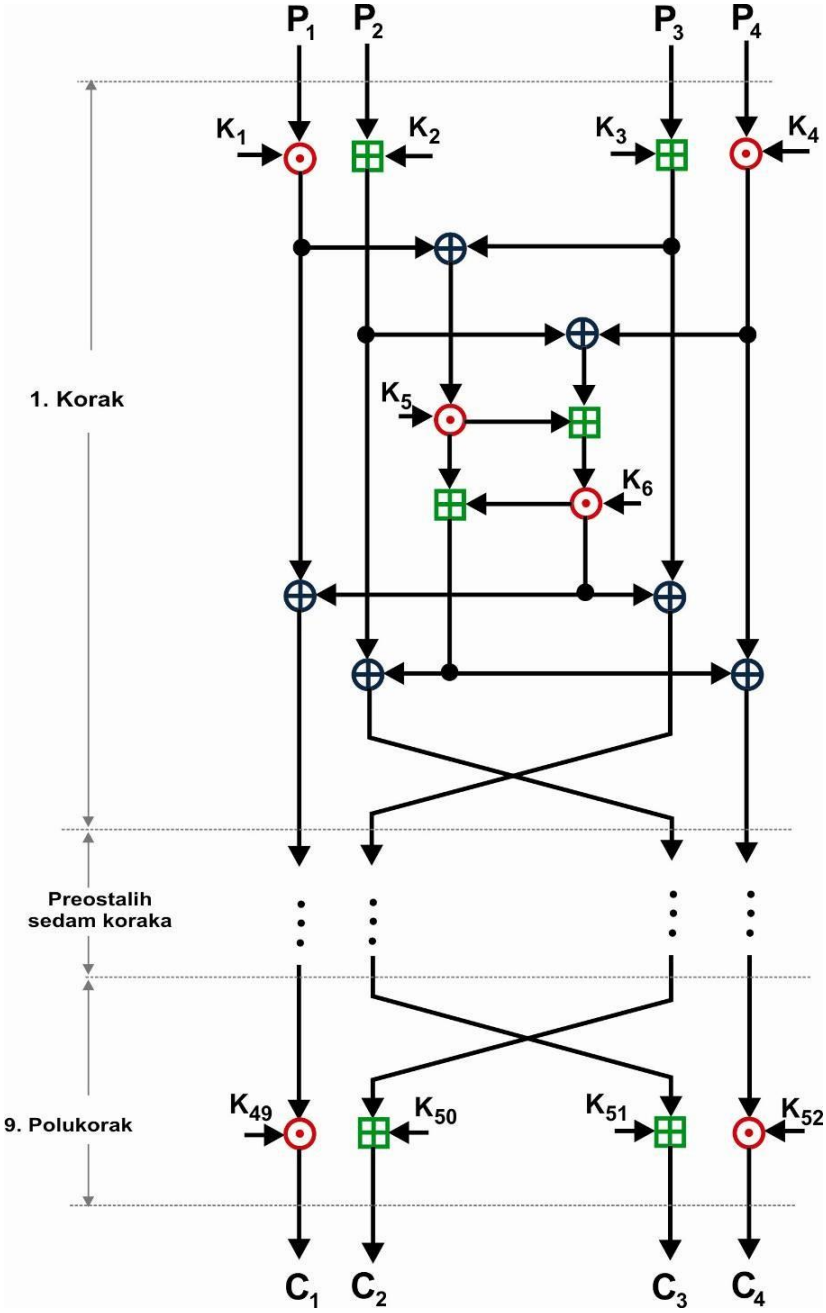
Slika 2.7
IDEA algoritam za: a) Kriptovanje; b) Dekriptovanje

Kriptovani blok podataka C se dobija primenom devet koraka nad izvornim tekstom P i delovima ključa K. 64-bitni ulazni blok podataka P se deli na četiri dela od po 16 bit-a (P_1, P_2, P_3 i P_4). Od originalnog 128-bitnog ključa formira se 52 dela ključa (od K_1 do K_{52}) od po 16 bita. Prvih osam delova ključa (K_1,K_2,...,K_8) se izdvajaju direktno od 128-bitnog ključa, i to počevši od K_1 kao najnižih 16 bitova, K_2 kao sledećih 16 bitova i tako dalje. Svaka sledeća grupa od po osam ključeva se formira na isti način s tim što se prethodno 128-bitni ključ rotira ulevo za 25 bita [11].

IDEA algoritam za kriptovanje se bazira na ukupno osam koraka u okviru kojih se obavljaju kombinacije tri različite operacije i to:

- ⊕ Bit po bit XOR nad 16 bitnim sub-blokovima
- ⊞ Sabiranje 16 bitnih celobrojnih vrednosti osnove 2^{16}
- ⊙ Množenje 16 bitnih celobrojnih vrednosti osnove $2^{16} + 1$.

Na Slici 2.8 je prikazan redosled operacija u okviru jednog koraka, a nakon obavljenih osam identičnih koraka sledi deveti polu-korak koji je takođe prikazan na istoj Slici 2.8 [14].



Slika 2.8
Redosled operacija u procesu kriptovanja

Postupak dekriptovanja pomoću IDEA algoritma je sličan postupku kriptovanja, s tom razlikom što se koriste različite kombinacije delova ključa u okviru devet koraka izvršavanja algoritma. Algoritam koji se koristi za dekriptovanje takođe je prikazana na Slici 2.7.

2.5.5 RC5 algoritam

Algoritam opisan u ovom delu formirao je *Ronald Rivest* sa *Massachusetts* instituta zatehnologiju 1994. godine i nazvao ga RC5. Kompanija *RSA Data Security* je procenila da će RC5 i njegov naslednik RC6 biti dostojni sledbenici DES algoritma za kriptovanje podataka [16].

RC5, takođe pripada simetričnim šifarskim blokovima pogodan kako za softversku tako i za hardversku realizaciju. RC5 predstavlja algoritam sa promenjivom veličinom bloka ulaznih podataka,

promenjivim brojem koraka izvršavanja i promenjivim dužinama ključa. Na ovaj način RC5 obezbeđuje veliku fleksibilnost u performansama i nivoima sigurnosti podataka.

RC5 algoritam karakterišu sledeći parametri RC5-*w/r/b*. Gde je *w* broj bitova u reči, *r* predstavlja broj koraka RC5 algoritma, a *b* je dužina ključa izražena u bajtovima. Različitim izborom parametara dobijaju se različiti RC5 algoritmi. Standardnu reč čini 32 bita, dok su moguće i vrednosti 16 bita i 64 bita. RC5 kriptovanje koristi blokove od dve reči (2*w*) izvornog ili kriptovanog teksta. Moguće vrednosti paramera *r* su 0,1,…,255. Broj bajtova koji čine ključ, *b*, kreće se od 0 do 255. Tajni ključ K se često pre upotrebe proširi i na taj način se formira takozvana proširena tabela ključeva S [11], [15].

RC5 algoritam čine tri komponente: algoritam za proširenje ključa, algoritam za kriptovanje i algoritam za dekriptovanje. Sva tri algoritma koriste sledeće jednostavne operacije:

⊞ Aritmetičko sabiranje po modulu 2^w ,
⊕ Bit po bit XOR,
<<< Rotacija ulevo za određeni broj bitova.

Tokom procesa kriptovanja, algoritam koristi podatke iz proširene tabele ključeva *S*[*0,1,...,t-1*], od $t = 2(r + 1)$ reči. Proširena tabela ključeva se formira od vrednosti tajnog ključa *K*. Algoritam za proširenje ključa koristi dve konstante P_w i Q_w koje se definišu kao:

$$P_w = Odd\ ((e - 2)2^w)$$
$$Q_w = Odd\ ((\phi - 1)2^w)\ ,$$

gde su e = 2.71828… (osnova prirodnog logaritma) i $\phi = (1+ \sqrt{5}) / 2 = 1.61803\ldots$, a *Odd(x)* je celodrojni ostatak od *x*. Procedura formiranja proširene tabele ključeva se obavlja u okviru tri koraka. U okviru prvog koraka pri formiranju proširene tabele ključeva, tajni ključ *K*[0,1,…*b*-1] se preslikava u niz *L*[0,1,…,*c*-1] koji se sastoji od $c = b/u$ reči, gde je $u = w/8$. Proces preslikavanja se obavlja po sledećem pravilu:

for $i = b - 1$ down to 0 do
$L[i/u] = (L[i/u] <<< 8) + K[i]$

Drugi korak se odnosi na inicijalizaciju niza *S* pomoću dve konstante P_w i Q_w pomoću sledeće procedure:

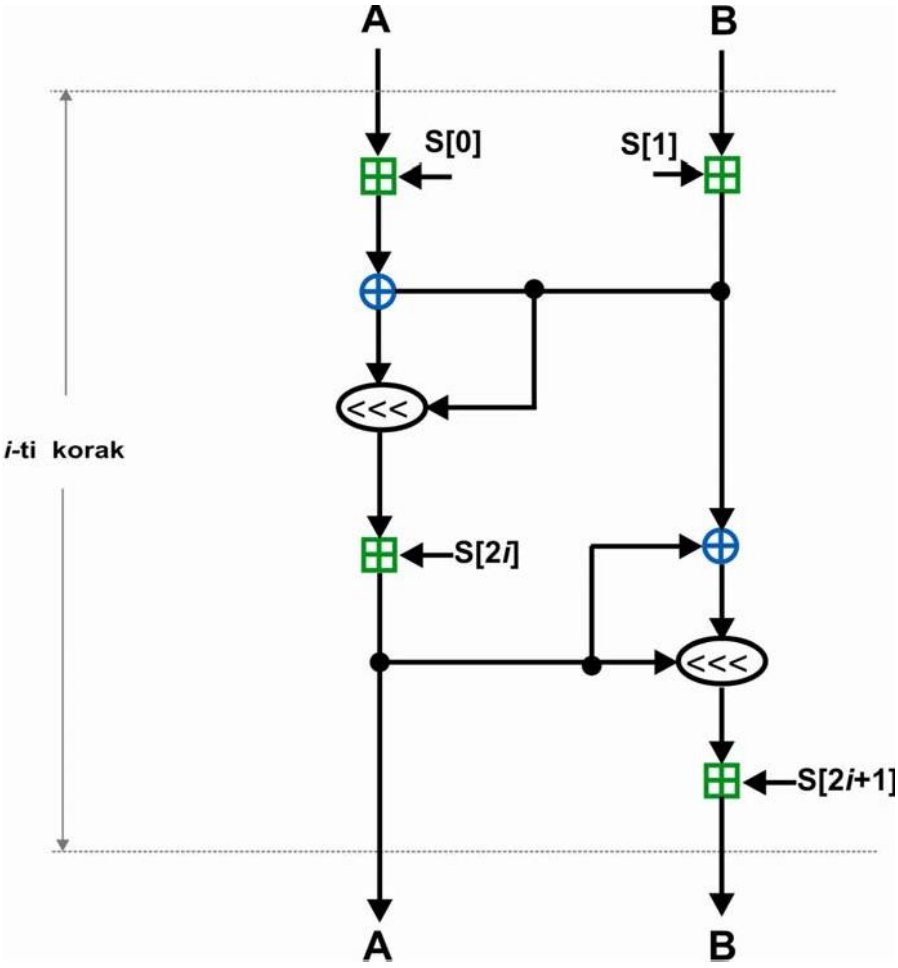
$S[0] = P_w$
for $i = 1$ to $t - 1$ do
$S[i] = S[i - 1] + Q_w$

U okviru trećeg koraka se na osnovu tajnog ključa *K* i nizova *L* i *S* formira proširena tabela ključeva koja se sastoji od elementata *S*[*i*] i *L*[*j*]. U okviru ovog koraka se koristi sledeća procedura:

$i = j = 0$
$A = B = 0A = S[i] = (S[i] + A + B) <<< 3$
$B = L[j] = (L[j] + A + B) <<< (A + B)$
$i = (i + 1) \pmod t$
$j = (j + 1) \pmod c$

Opisana procedura se primenjuje 3 x max(*t, c*) puta.

Na Slici 2.9 je prikazan algoritam koji se koristi za kriptovanje podataka. Ulazni blok RC5 algoritma sadrži dve *w*-bitne reči koje se čuvaju u dva registra *A* i *B*. U ovim registrima se takođe smeštaju i informacije koje su dobijene na izlazu iz algoritma [11].

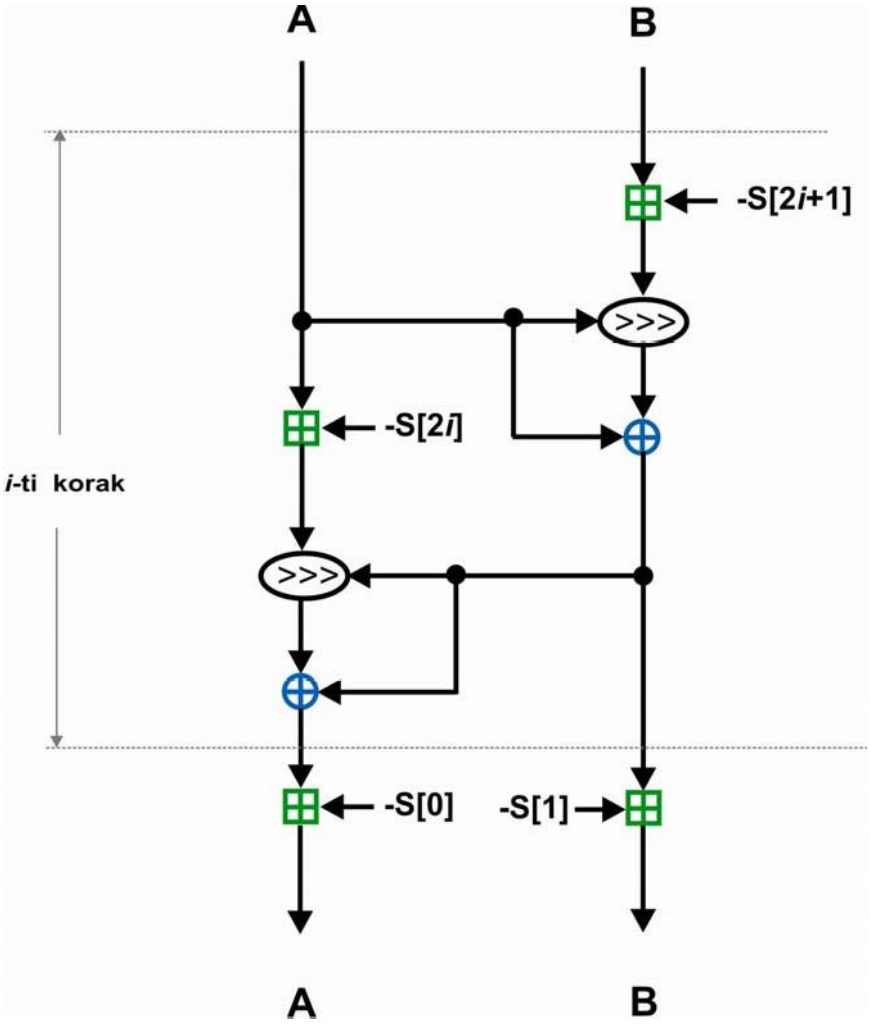


Slika 2.9
Operacije u okviru RC5 algoritma

Sledeće procedure opisuju operacije prilikom kriptovanja:

$A = A + S[0];$
$B = B + S[1];$
for $i = 1$ to r do
$A = ((A \oplus B) <<< B) + S[2i];$
$B = ((B \oplus A) <<< A) + S[2i+1];$

Algoritam za dekriptovanje je prikazan na Slici 2.10. Primećuje se da algoritmi za kriptovanje i dekriptovanje imaju određenu sličnost koja se ogleda u obrnutom redosledu operacija.



Slika 2.10
Operacije u postupku dekriptovanja pomoću RC5 algoritma

Formule koje opisuju postupak dekriptovanja podataka su:

for $i = r$ down to 1 do
$B = ((B - S[2i+1]) >>> A) \oplus A$
$A = ((a - s[2i]) >>> B) \oplus B$
$B = B - S[1]$
$A = A - S[0]$

Zbog jednostavnosti operacija RC5 je lak za implementaciju. Uz to iznos pomaka u operacijama rotiranja nije fiksan, već zavisi od ulaznog podatka.

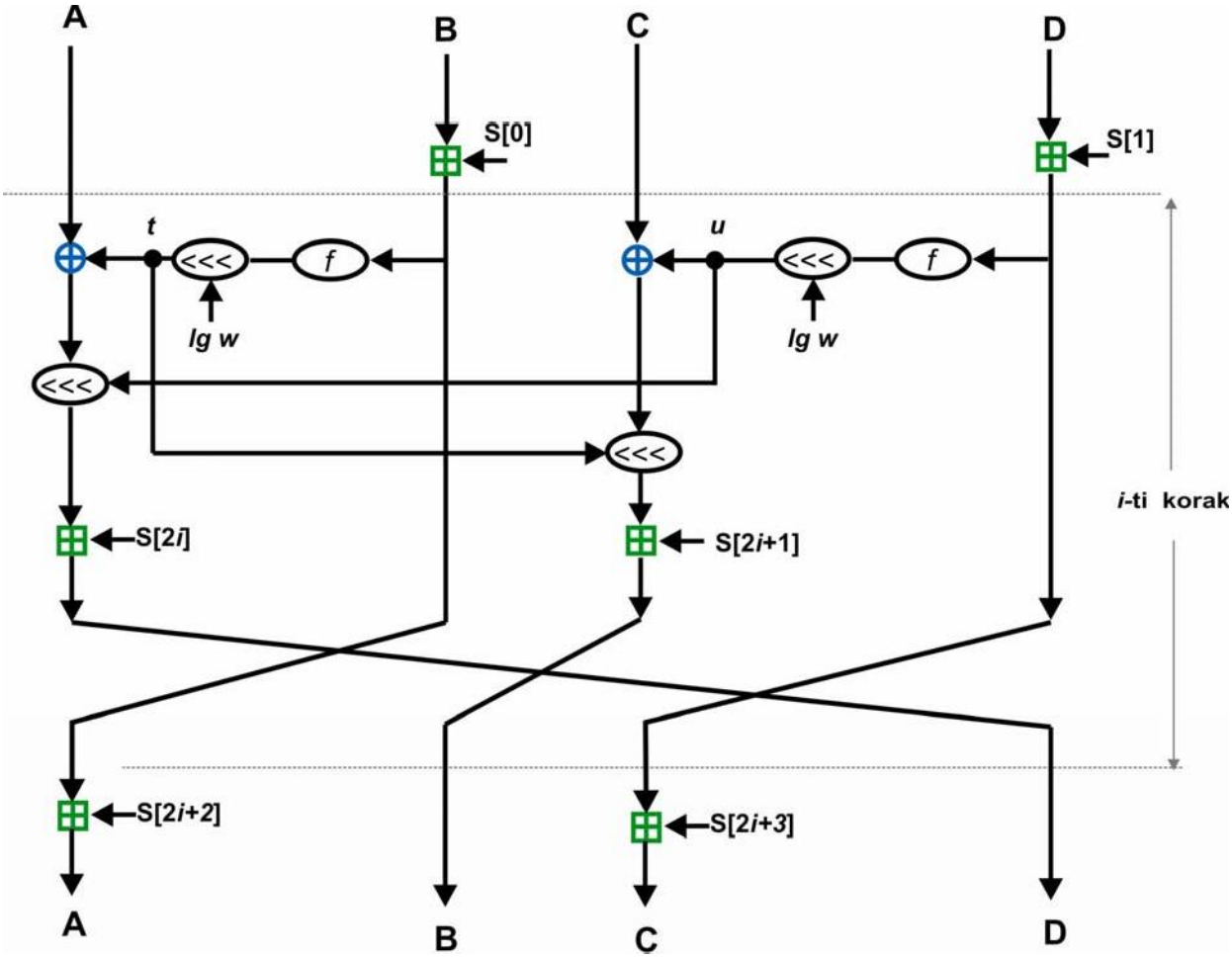
2.5.6 RC6 algoritam

RC6 algoritam se pojavio kao unapređenje RC5 algoritma, naravno sa strožijim zahtevima po pitanju sigurnosti i boljim performansama. Kao i kod RC5 i ovaj algoritam koristi rotacije sa promenjivim pomakom. Novo je jedino to što RC6 koristi četiri umesto dva bloka reči (radna registra). RC6 podržava blokove podataka od po 128 bitova i koristi ključeve veličine 128, 192 i 256 bita [17].

Varijante RC6 se specifikuju kao RC6-*w/r/b* gde je *w* veličina reči u bitovima, *r* predstavlja broj koraka algoritma i *b* je dužina ključa u bajtovima. Operacije koje se koriste prilikom kriptovanja su:

- $a + b$ Celobrojno sabiranje po modulu 2^w
- $a - b$ Celobrojno oduzimanje po modulu 2^w
- $a \oplus b$ Bit po bit XOR nad w-bit rečima
- $a \times b$ Celobrojno množenje po modulu 2^w
- $a <<< b$ Rotacija w-bitne reči a ulevo za b bita
- $a >>> b$ Rotacija w-bitne reči a udesno za b bita

Algoritam koji opisuje RC6 šemu za kriptovanje je prikazan na Slici 2.11.



Slika 2.11
Izgled koraka za kriptovanje pomoću RC6 algoritma

Četiri w-bitna radna registra koja sadrže izvornu poruku su A, B, C i D. $S[0,1,\ldots, 2r + 3]$ predstavljaju w-bitne delove ključa iz proširene tabele ključeva. Dobijeni kriptovani tekst se skladišti u registrima A, B, C i D. Formiranje proširene tabele ključeva se vrši isto kao i kod RC5, pomoću konstanti P_w i Q_w. Procedura koja prati proces kriptovanja je [11]:

$B = B + S[0]$
$D = D + S[1]$
for $i = 1$ to r do
{
$t = (B \times (2B + 1)) <<< \lg w$
$u = (D \times (2D + 1)) <<< \lg w$
$A = ((A \oplus t) <<< u) + S[2i]$

$C = ((C \oplus u) <<< t) + S[2i+1]$

$(A, B, C, D) = (B, C, D, A)$

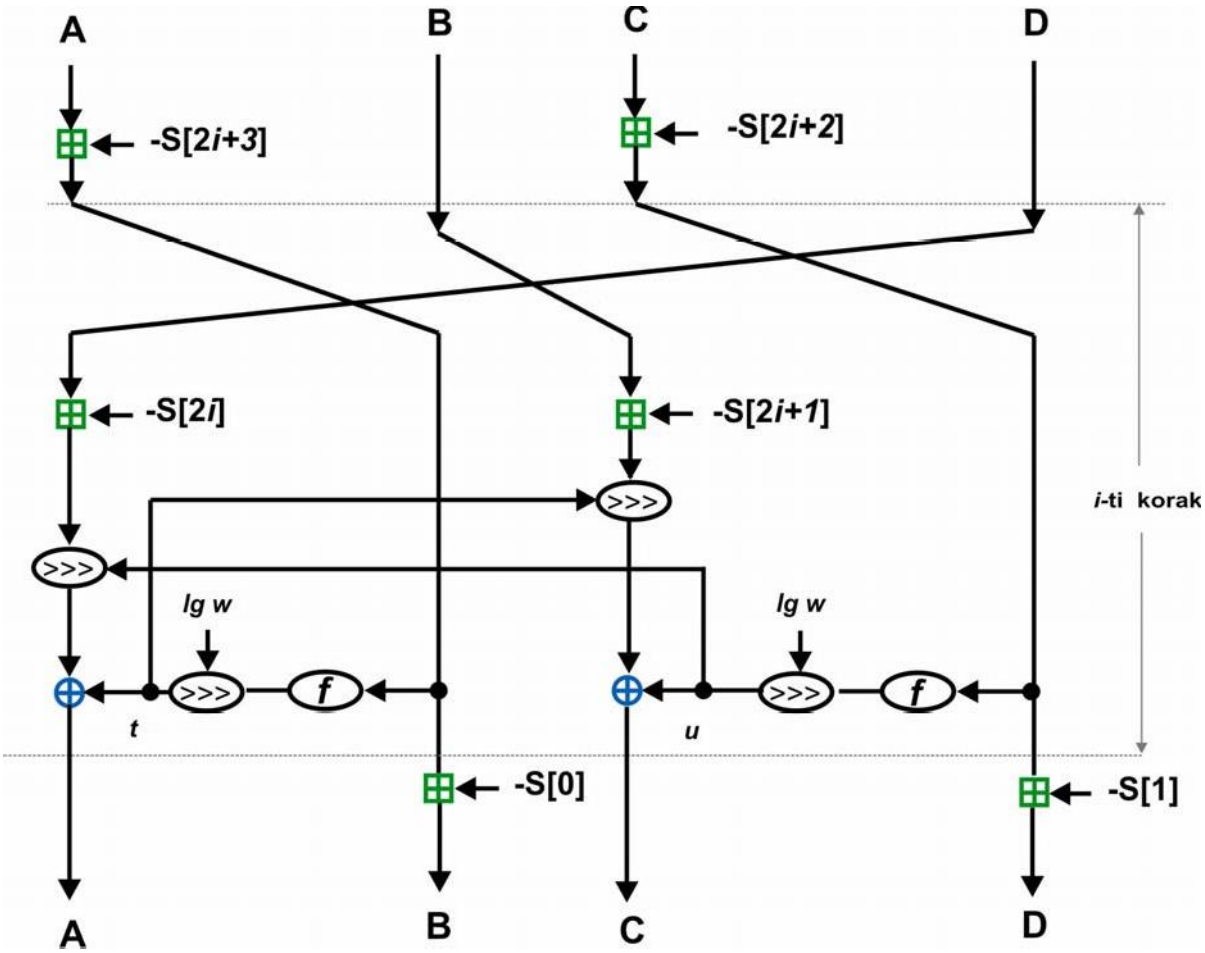
}

$A = A + S[2R+2]$

$C = C + S[2R+3]$

Gde je lg *w* logaritam od reči w za osnovu dva.

Proces dekriptovanja je sličan kriptovanju s tim što se menja redosled operacija. Na Slici 2.12 je prikazan algioritam koji se koristi za dekriptovanje poruka kriptovanih pomoću RC6 algoritma.



Slika 2.12

Operacije za dekriptovanje pomoću RC6 algoritma

Proces dekriptovanja se može opisati u vidu sledeće procedure:

$C = C - S[2R+3]$

$A = A - S[2R+2]$

for $i = r$ down to 1 do

{

$(A, B, C, D) = (D, A, B, C)$

$u = (D \times (2D+1)) <<< \lg w$

$t = (B \times (2B+1)) <<< \lg w$

$C = ((C - S[2i+1]) >>> t) \oplus u$

$A = ((A - S[2i]) >>> u) \oplus t$

}

$$D = D - S[1]$$

$$B = B - S[0]$$

2.5.7 AES (*Rijndael*) algoritam

Napredni standard za kriptovanje AES (eng. *Advanced Encryption Standard*) je zamenio dugogodišnji standardni DES način kriptovanja podataka. Naglim razvojem računarskih tehnologija i sirove snage procesora algoritmi stari desetak i više godina postaju neupotrebljivi u smislu da više ne mogu pružiti zadovoljavajući nivo sigurnosti podataka.

Danas se može govoriti o ne baš nedostižnoj sigurnosti podataka koji su kriptovani pomoću DES algoritama. Kako je DES prestao zadovoljavati potrebama NIST-a (*The National Institute of Standards and Technology*) 1997. godine se objavljuje konkurs za novi standard za enkripciju podataka. U konkurenciji su bila 15 kandidata i to: CAST-256, CRYPTON, DEAL, DFC, E2, FROG, HPC, LOKI97, MAGENTA, MARS, RC6, Rijndael, SAFER+, *Serpent te Twofish*, od kojih je izabrano pet finalista: MARS, RC6, Rijndael i Serpent te Twofish. Za novi AES standard se 2000. godine odabira *Rijndael* [11], [19].

AES se zasniva na *Rijndael* algoritmu koji su razvili *Daemen* i *Rijmen* 1999 godine. Ovaj algoritam pripada grupi simetričnih šifarskih blokova, koji kriptuje blokove podataka od po 128 bitova koristeći ključeve od 128, 192 i 256 bita. Sve vrednosti u okviru AES algoritma predstavljaju se u vektorskom obliku (b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0) a koji odgovaraju sledećem polinomu:

$$b_7 x^7 + b_6 x^6 + b_5 X^5 + b_4 X^4 + b_3 x^3 + b_2 x^2 + b_1 x + b_0 = \sum_{i=0}^{7} b_i x^i$$

Kriptovanje podataka pomoću AES algoritma se obavlja iz više iteracija. Izvorni podaci se dele u 4 grupe od po 4 bajta na osnovu kojih se formira dvodimenzionalna matrica ulaznih bajtova [18]. Na osnovu matrice ulaznih podataka formira se matrica zvana matrica stanja. Sve operacije koje se obavljaju u toku AES algoritma se izvode nad matricom stanja. Matricu stanja čine četiri reda dužine jedan bajt. Broj redova matrice stanja se kreće $0 < r < 4$ a broj kolona $0 < c < Nb$, gde je Nb dužina bloka u bitovima podeljena sa 8 x 4 = 32. Za slučaj kada je ulazni blok 128-bitni važi $Nb = 4$.

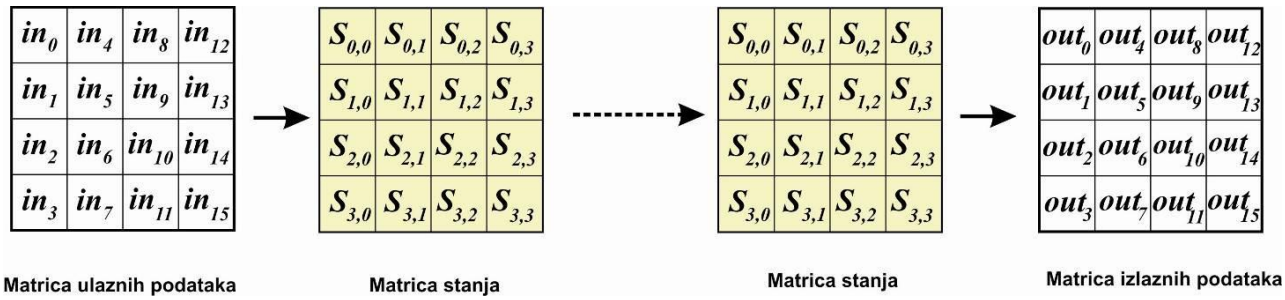
Matrica ulaznih podataka se sastoji od elemenata in_0, in_1, ..., in_{15} dužine od po jedan bajt. Ova matrica se preslikava u matricu stanja pomoću sledeće šeme:

$$S_{r,c} = in_{(r+4c)} \quad \text{za } 0 \le r < 4 \text{ i } 0 \le c < Nb$$

Ukoliko se radi o dekriptovanju proces formiranja matrice izlaznih podataka iz matrice stanja koristi se sledeća šema:

$$out_{(r+4c)} = S_{r,c} \quad \text{za } 0 \le r < 4 \text{ i } 0 \le c < Nb$$

Ilustracija preslikavanja matrica je prikazana na Slici 2.13.

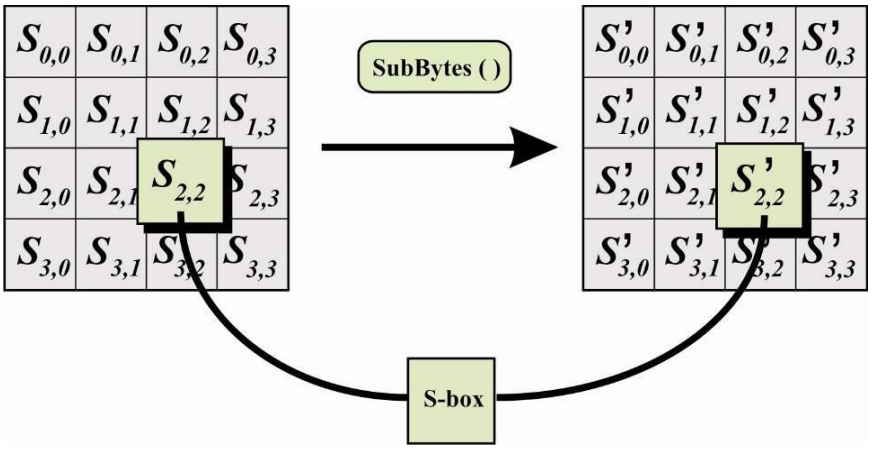


Slika 2.13
Postupci preslikavanja matrica

U svakoj od 9, 11 ili 13 iteracija izvršavaju se operacije nad elementima matrice stanja. Redosled i broj izvršavanja ovih operacija je definisan algoritmom za kriptovanje [18], [19]. Moguće su sledeće operacije:

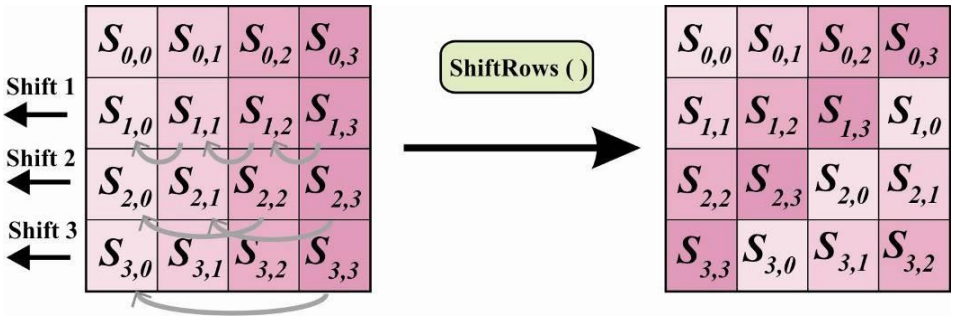
- *SubBytes( )*
- *ShiftRows( )*
- *MixColumns( )*
- AddRoundKey( )

SubBytes( ) je operacija substitucije koja se obavlja nad svakim bajtom iz matrice stanja tako što se svaki element menja sa elementom iz pomoćne lookup tabele odnosno S-box-a. Ilustracija operacije substitucije je prikazana na Slici 2.14.



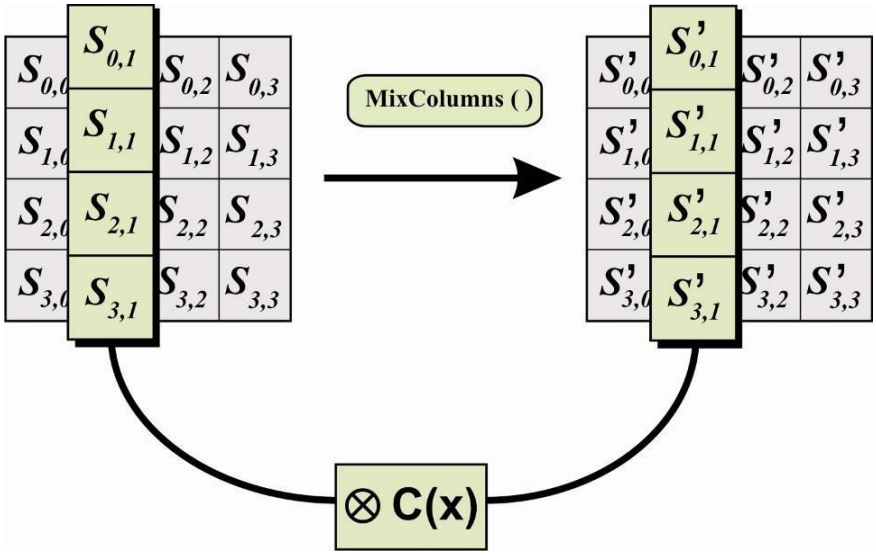
Slika 2.14
Operacija substitucije

ShiftRows( ) operacija se odnosi na pomeranje elemenata (bajtova) u okviru kolona za određeni broj pozicija. Na Slici 2.15 je prikazana ilustracija operacije pomeranja. Prva kolona ostaje nepromenjena, dok se svaka sledeća kružno se pomera i to: druga kolona za po jedno mesto ulevo, treća za po dva mesta ulevo itd [19].



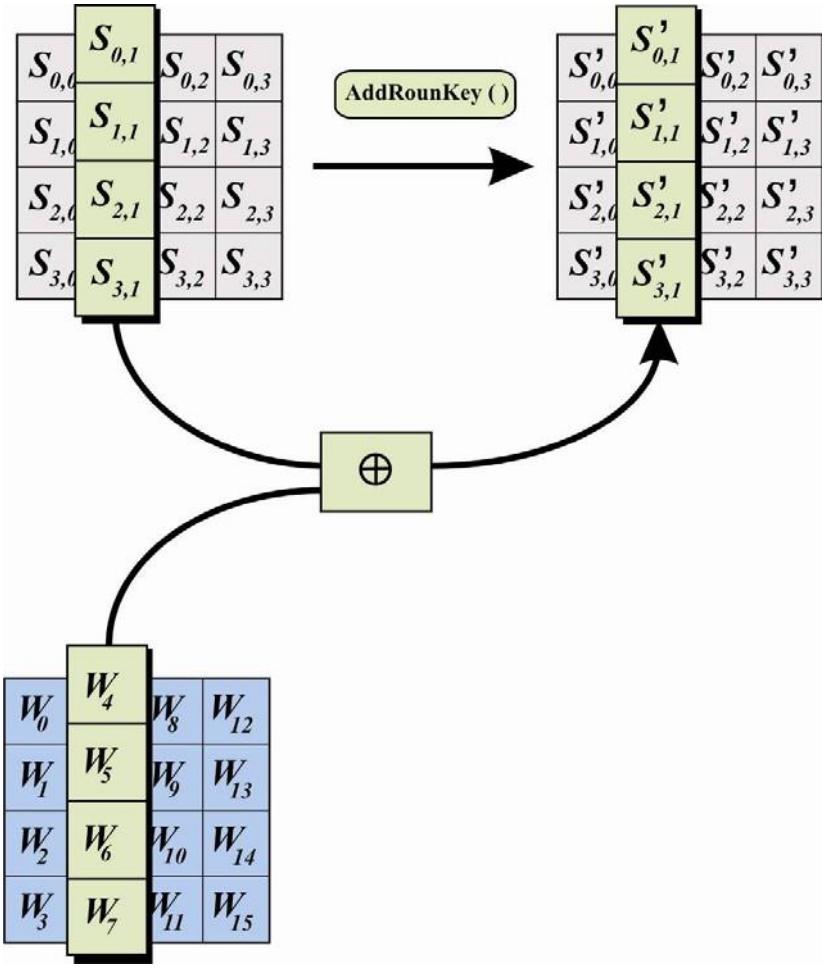
Slika 2.15
Operacija pomeranja

MixColumns( ) operacija koja koristi određenu linearnu transformaciju nad svakom kolonom iz matrice stanja. Primer ove operacije je prikazan na Slici 2.16 gde se nad kolonom obavlja operacija množenja sa konstantom *C (x)*. Svaka kolona se tretira kao polinom u *Galoovom* polju GF(2^8) i množi sa nesvodljivim fiksnim polinomom $C(x)=3x^3+x^2+x+2$.



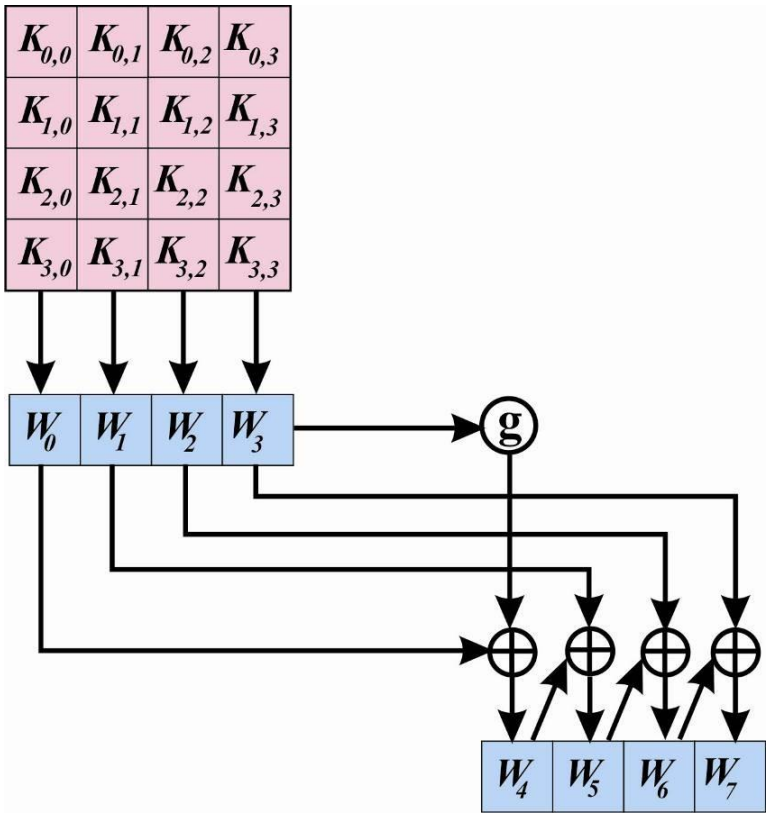
Slika 2.16
MixColumns( )Operacija

AddRoundKey( ) operacija se odnosi na XOR operaciju nad matricom stanja i matricom ključeva. Ilustracija operacije *AddRoundKey( )* je prikazana na Slici 2.17.



Slika 2.17
AddRoundKey( ) operacija

Ključ veličine 128/192/256 bita se predstavlja kao matrica koja ima 4 kolone od po 4/6/8 bajta . Zatim se tako dobijena matrica proširuje u niz reči od po 4 bajta, odnosno ukupno 44/52/60 reči w_i od 128/192/256 bita. Proširenje ključa se započinje kopiranjem prve četiri reči ključa u reč (w_0, w_1, w_2, w_3). Zatim se formira petlja koja generiše sledeće četiri reči w_i (vidi Sliku 2.18) na osnovu predhodne reči w_{i-1} i reči koja je za 4 mesta unazad w_{i-4}. U tri od četiri slučaja se primenjuje samo XOR operacija, a svaki četvrti ima složeniju funkciju *g* [19].



Slika 2.18
Postupak proširavanja ključa

Funkcija g se formira od sledećih operacija:

- *RotWord ( )*
- *SubWord ( )*
- *Rcon (i)*

RotWord ( ) se odnosi na kružno pomeranje reči u levo za jedan bajt. Ukoliko je ulazna reč (b_0, b_1, b_2, b_3) nakon pomeranja se transformiše u (b_1, b_2, b_3, b_0).

SubWord ( ) operacija substitucije koja se obavlja nad svakom od reči veličine četiri bajta. Ova operacija zamenu reči vrši pomoću S-box-a.

Rcon (i) operacija predstavlja matricu konstanti koju čine reči, a računa se po formuli:

$$Rcon(i) = x^{(254+i)}$$

Algoritam se na najvišem nivou sastoji iz sledećih radnji:

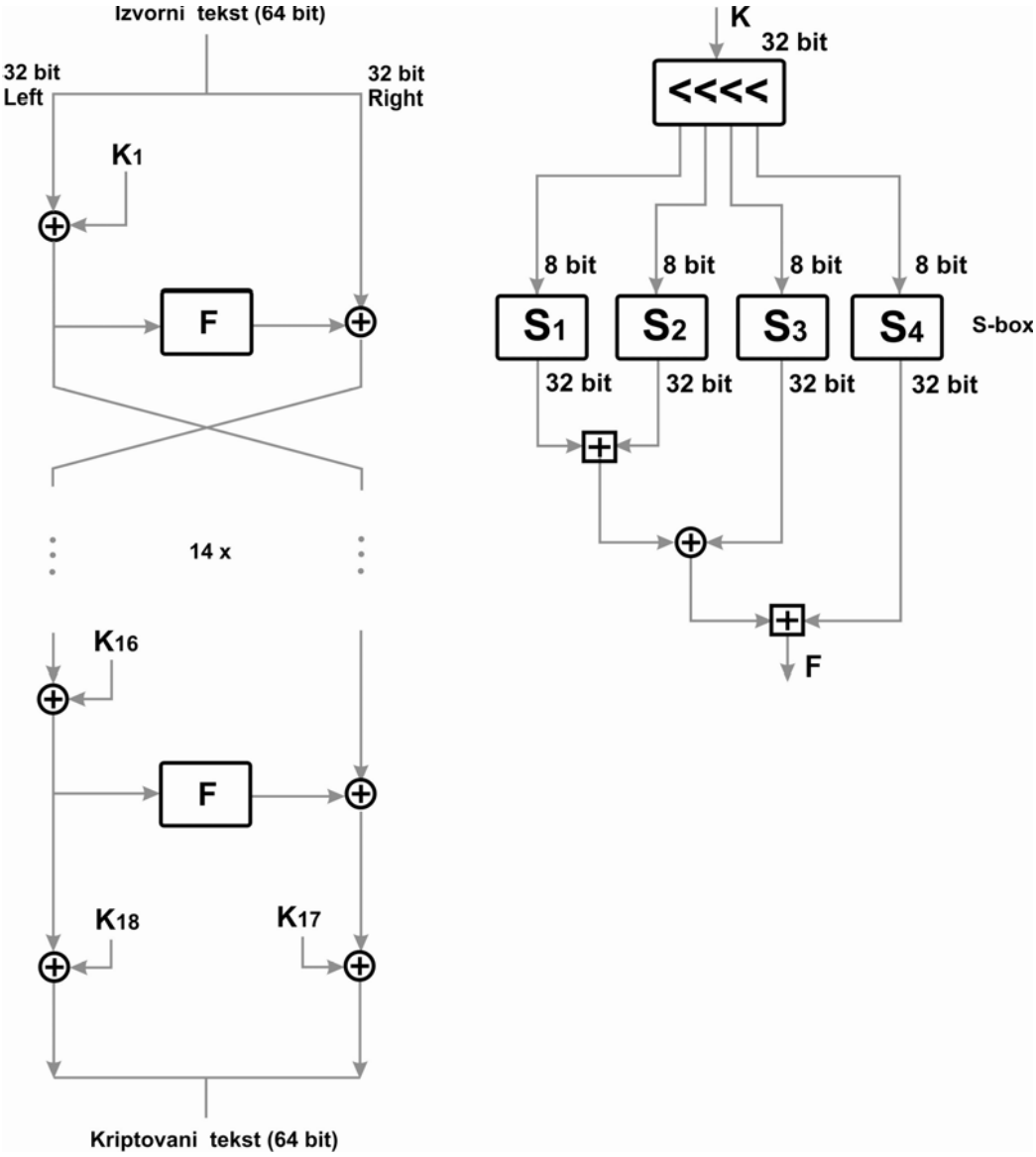
- Proširenje ključa
- Inicijalna iteracija: *AddRoundKey( )*
- Iteracije redom: *SubBytes ( ), ShiftRows ( ), MixColumns ( )* i *AddRoundKey ( )*
- Konačna iteracija: *SubBytes ( ), ShiftRows ( )* i *AddRoundKey ()*

2.5.8 Blowfish

Blowfish pripada grupi simetričnih blok šifarskih algoritama. Ustanovio ga je *Bruce Schneider* 1993. godine. Ovaj algoritam pruža dobre kriptografske karakteristike u softverskim sistemima za kriptovanje i do danas nije pronađen efikasniji algoritam.

Blowfish je zamišljen kao algoritam opšte namene, namenjen da zameni DES algoritam i da pri tom reši probleme i ograničenja koja se javljaju upotrebom ostalih algoritama. Ovaj algoritam je bio oslobođen vlasničkih prava i patenata, odnosno karijeru je započeo kao nepatentiran ali su ga kasnije, kao takvog prihvatile sve zemlje. Tako je algoritam ostao u javnosti kao besplatan za bilo kog korisnika.

Blowfish algoritam koristi 64-bitne blokove izvornih podataka i ključ promenjive veličine od 32 do 448 bita. Algoritam koristi 16 koraka *Feistel* šifrovanja i veliki broj S-box-ova koji se koriste za dobijanje ključa. Na Slici 2.19 je prikazan dijagram koji opisuje strukturu *Blowfish* algoritma [20].



Slika 2.19

Blowfish algoritam: a) Kriptovanje; b) Formiranje pomoćne funkcije F

Izvorni blok podataka se deli u dve linije od po 32 bita. Algoritam prate dve matrice sa pomoćnim ključevima: *P* matrica sa 18 elementata i četiri S-box-a (blokova za substituciju) sa 256 elemenata. Svaki S-box prihvata 8-bitne elemente na ulazu i generiše 32-bitne izlaze. Tokom svakog koraka izvršenja

algoritma (ukupno 16) koristi se po jedan elemenat P matrice (P_1,P_2,...,P_{16}), a preostala dva elementa (P_{17} i P_{18}) se koriste na kraju. Nad elementima P matrice i 32-bitnim podacima se primenjuje XOR logička operacija.

Na Slici 2.19 (a) se vidi i F funkcija (*Feistel*) koja se u okviru svakog koraka primenjuje nad podacima sa leve strane. Nad tako dobijenom informacijom i podacima sa desne strane se primenjuje XOR operacija. Neposredno pre izvršavanja sledećeg koraka, tako dobijeni podatak i podatak sa leve strane zamene mesta. Blok dijagram koji prikazuje dobijanje F funkcije je prikazan na Slici 2.19 (b). Podatak na ulazu (32-bitni) se deli na četiri 8-bitna podatka koji se koriste kao ulazi u S-box-ove. Izlazi iz S-box-ova S_1 i S_2 se sabiraju po modulu 2^{32} i nad tako dobijenim podatakom i S_3 se primenjuje XOR operacija. Na kraju se u cilju dobijanja konačne 32-bitne funkcije F predhodno dobijeni rezultat sabira po modulu 2^{32} sa S_4.

Formiranje ključeva se započinje inicijalizacijom P matrice i S-box-eva sa vrednostima dobijenim na osnovu heksadecimalne vrednosti broja *pi*. Tajni ključ i elementi P matrice se sabiraju pomoću XOR operacije. Nakon toga se vrši kriptovanje 64-bitnih blokova podataka koji sadrži sve nule i kao rezultat se dobijaju elementi P_1 i P_2. Ovaj postupak se ciklično ponavlja sve dok se ne dobiju svi elementi matrice P i S-box-eva. Da bi se generisali svi elementi potrebno je da algoritam prođe 521 put, što čini ukupno 4KB podataka [20].

Proces dekriptovanja identičan je postupku kriptovanja s tom razlikom što se elementi matrice P upotrebljavaju obrnutim redosledom.

Blowfish je veoma brz kriptografski metod izuzev u sitacijama kada se vrši razmena ključeva. Svaki novi ključ zahteva slanje 4 KB teksta.

2.6 Prednosti i nedostatci simetričnog kriptovanja

Algoritmi koji koriste simetrični ključ za kriptovanje odlikuju se visokom efikasnošću, što se ogleda u kratkom vremenu kriptovanja poruka. Razlog kratkog vremena kriptovanja je upotreba kratkih ključeva. Iz tih razloga se ova vrsta algoritama koristi za kriptovanje/dekriptovanje poruka velike dužine.

Simetrično kriptovanje ima dva osnovna nedostatka. Oba korisnika (osoba A i osoba B) moraju posedovati jedinstveni simetrični ključ, te se javlja problem distribucije ključeva. Naime, korisnici koji žele da razmene poruku prethodno moraju da se dogovore o ključu. Jedini pouzdan način je da se oba korisnika fizički sretnu i izvrše razmenu ključa. Međutim, često su korisnici fizički razdvojeni i ne mogu da dođu u neposredan kontakt, zato moraju da koriste neki zaštićen kanal da bi sigurno razmenili ključeve. Problem je to što zaštićen kanal praktično ne postoji.

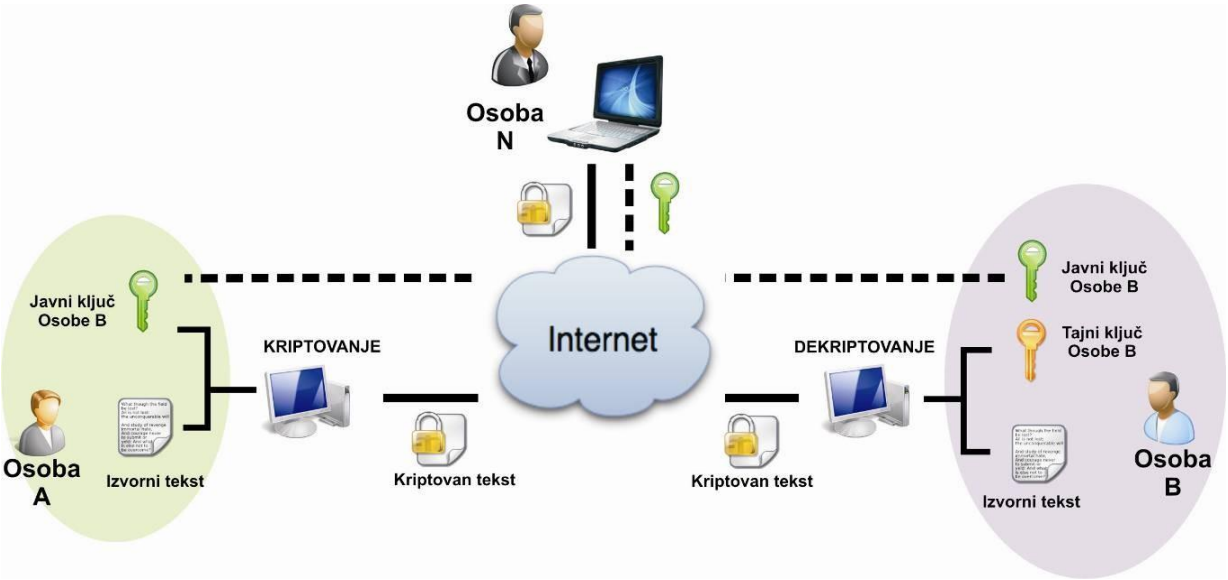
Drugi problem koji se javlja kod simetričnih kriptosistema je veliki broj potrebnih ključeva. Ako imamo N ljudi koji žele da koriste ovu metodu kriptovanja, to zahteva $N(N-1)/2$ simetričnih ključeva. Na primer za 1 milion ljudi potrebno je 500 milijardi simetričnih ključeva. Radi dobijanja toliko velikog broja različitih ključeva moraju se koristiti ključevi veće dužine. Tako na primer, dužina ključa od 56 bita je danas na granici dovoljnog s obzirom na to da savremeni računari mogu relativno brzo da otkriju ključ te dužine.

2.7 Asimetrični kriptosistemi

Tvorci asimetrične kriptografije su *Whitefield Diffie* i *Martin Hellman* koji su 1976. godine opisali ideju kriptografije koja se temelji na dva ključa, privatnom (ili često zvanim tajnim) i javnom ključu. U literaturi pojam asimetričnog kriptovanja se poistovećuje sa terminom *asymmetric-key* ili *public-key* kriptovanjem.

Razlika između simetričnih i asimetričnih algoritama je u tome što simetrični algoritmi koriste isti ključ za kriptovanje i dekriptovanje dok asimetrični algoritmi koriste različite ključeve za kriptovanje odnosno dekriptovanje. Informacije koje su kriptovane javnim ključem mogu se dekriptovati samo tajnim ključem odnosno to može samo osoba koja je vlasnik tajnog asimetričnog ključa. Oba ključa moraju biti povezana pomoću jedinstvene jednosmerne funkcije. Odnosno ne sme se izračunati tajni ključ iz javnog ključa ili se barem ne sme izračunati u razumnom vremenu.

Algoritmi asimetričnih kriptosistema zasnivaju se na određenim svojstvima brojeva. Pri kriptovanju se izvorni tekst tretira kao niz prirodnih brojeva koji se odabranom funkcijom kriptovanja i ključem *Ke* preračunavaju u kriptovani niz teksta. Funkcija kriptovanja mora biti takva da se iz kriptovanog teksta ne može odrediti izvorni tekst, čak ako je poznat i ključ za kriptovanje. Međutim, ukoliko se zna ključ dekriptovanja *Kd* moguće je lako računanje izvornog teksta. Asimetrično kriptovanje predstavlja jako složen vid zaštite podataka. Za njegovu realizaciju svaki od sagovornika mora posedovati dva ključa (javni i tajni). Iako su različiti, ključevi su međusobno povezani određenim transformacijama. Na Slici 2.20 je prikazan primer asimetričnog kriptovanja [9].



Slika 2.20
Postupak asimetričnog kriptovanja

Radi jednostavnije analize rada koristićemo simbole *A, B* i *N*. *A* je osoba koja želi da pošalje izvorni tekst, *B* predstavlja osobu koja bi trebala da primi poslati tekst, a *N* je osoba koja neovlašćeno želi da dođe do sigurnih podataka koje osoba *A* šalje osobi *B*. Scenario asimetričnog kriptovanja bi izgledao ovako: Osoba *A* kodira poruku radi slanja osobi *B* upotrebom javnog ključa osobe *B* koji je svima dostupan (čak i osobi *N*). Osoba *A* je javni ključ je mogla dobiti putem email-a, preuzeti sa *Web* sajta i sl. Međutim bilo ko ili osoba *N* i pored toga što poznaje javni ključ ne može otkriti sadržaj poruke. Poruku može dešifrovati samo osoba *B* korišćenjem svog tajnog ključa. Na ovaj način poruka je zaštićena od trećeg lica. Osnovni nedostatak ovog načina kriptovanja je njegova sporost i neprikladnost za kriptovanje velikih količina podataka. Takođe, ostaje otvoreno pitanje autentičnosti poruke, odnosno kako da osoba *B* bude sigurna da je poruku koju je primila uistinu poslala osoba *A*. Najčešće se koriste sledeći asimetrični algoritmi: RSA (eng. *Rivest-Shamir-Adleman*), *Diffie-Hellman*, *ElGamal*, *Eliptic Curves*, *Rabin* i drugi.

2.7.1 RSA

Najpopularniji asimetrični metod za kriptovanje podataka je RSA algoritam, koji nosi naziv po svojim izumiteljima: *Rivest, Shamir* i *Adleman*.

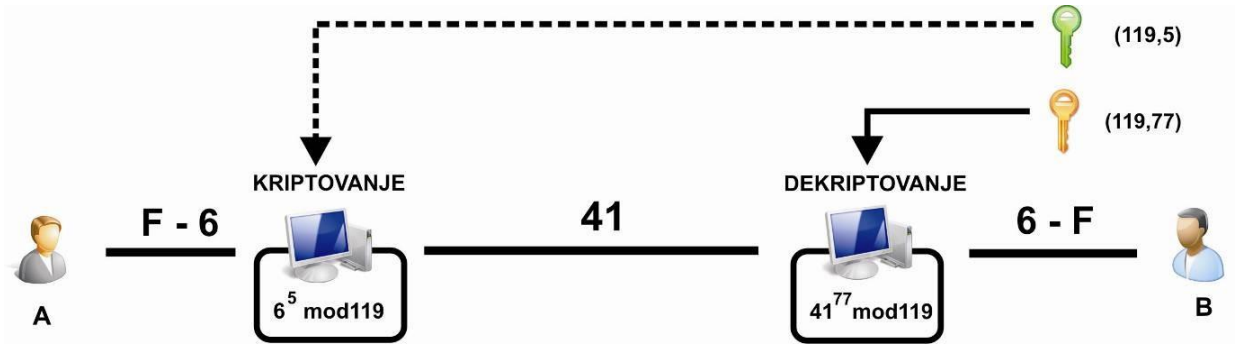
Za generisanje javnog i tajnog ključa se koriste prosti brojevi. Tajni ključ predstavlja uređeni par brojeva (N,d). Javni ključ je takođe uređeni par brojeva (N,e). Treba uočiti da je broj N zajednički za oba ključa. Osoba koja šalje poruku vrši kriptovanje pomoću sledeće jednačine [10]:

$$C = P^e \bmod N$$

Gde je: P, izvorni tekst koji je prikazan u obliku broja; C, broj koji predstavlja kriptovan tekst; brojevi e i N su komponente javnog ključa. Kada se poruka primi potrebno je dekriptovati pomoću sledeće jednačine:

$$P = C^d \bmod N$$

Gde su P i C isto kao i u predhodnoj formuli, a N i d predstavljaju komponente tajnog ključa. Na Slici 2.21 je prikazan primer RSA kriptovanja.



Slika 2.21
Primer RSA kriptovanja

Tajni ključ predstavlja uređeni par (119, 77), a javni ključ je uređeni par (119,5) i osoba A želi da pošalje karakter F osobi B. Ovaj karakter se može predstaviti kao broj 6 (F je šesti karakter u abecedi). Korišćenjem formule algoritma kriptovanja dobija se $C = 6^5 \ mod \ 119 = 41$. Ovaj broj se šalje kao kriptovan tekst osobi B. Osoba B kada primi broj koristi algoritam za dekriptovanje $P = 41^{77} \ mod \ 119 = 6$, a nakon toga inverznom transformacijom se broj 6 vrati u karakter F.

Postavlja se pitanje koliko je ovaj način slanja pouzdan. Ukoliko neka osoba C zna formulu za dekriptovanje i $N = 119$, a nedostaje joj samo $d = 77$, zašto ne bi pokušala da pogodi d. U ovom jednostavnom primeru osoba C bi mogla da pogodi vrednost d, međutim osnovni koncept RSA za d i e predviđa korišćenje veoma velikih brojeva. U praksi korišćenje jako velikih brojeva za d i e znači veliki broj kombinacija koje se moraju isprobati radi otkrivanja šifre tj. veoma dugo vreme razbijanja koda čak i uz pomoć najbržih računara koji su dostupni danas.

Osnovni problem kod RSA algoritma je kako izvršiti izbor brojeva N, d i e (veoma velike vrednosti dužine od 1024 do 2048), a da ujedno zadovoljavaju formule algoritma. Zapravo RSA koristi teoriju prostih brojeva i sledeću proceduru [21]:

- Izabrati dva prosta broja p i q
- Sračunati $N = p \ x \ q$
- Izabrati e (mora biti manje od N) tako da e i proizvod $(p – 1)(q – 1)$ budu uzajamno prosti (da nemaju zajedničkog delitelja osim 1).
- Odrediti broj d tako da zadovoljava jednačinu $(e \ x \ d) \ mod \ [(p – 1)(q – 1)] = 1$

Kao ilustracija formiranja para ključeva data je sledeća procedura: izabrati proste brojeve $p = 2357$, $q = 2551$; izračunati $N = p \ x \ q = 6012707$; izračunati $(p – 1)(q – 1) = 6007800$; bira se slučajni broj $e = 3113390$; računa se $d = 3674911$.

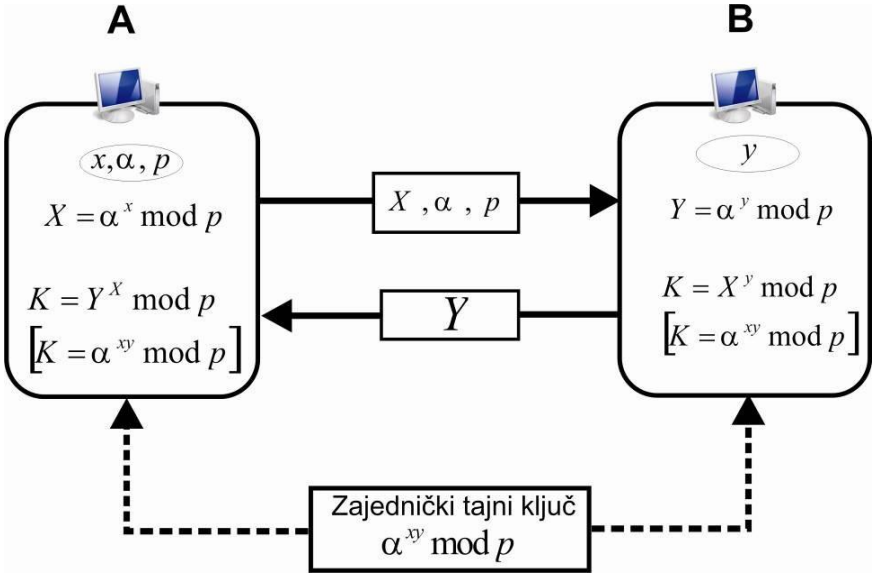
Određivanje originalne poruke na osnovu kriptovane poruke i javnog ključa ekvivalentno je faktorizaciji proizvoda dva velika prosta broja [22]. Ukoliko bi neka osoba C faktorisala broj N i iz toga pronašla proizvod $(p – 1)(q – 1)$ ne može doći u posed broja e, zato što on nema zajedničkog delitelja sa

$(p-1)(q-1)$. Ako se desi da na neki način sazna broj *e*, da bi došla do tajnog ključa *d* mora da faktoriše broj *N*. Pošto je broj *N* veoma veliki i do 309 decimalnih cifara, faktorizacija je skoro nemoguća.

Veliki prosti brojevi koji se koriste u RSA algoritmu nameću više problema praktične prirode. Da bi se množili ovi brojevi, moraju se koristiti posebni algoritmi za množenje, za koje je potrebno više vremena, samim tim se povećava vreme izvršavanja algoritma za kriptovanje. Poređenja radi DES algoritam je u proseku oko 500 puta brži u odnosu na RSA. Sem toga, algoritmi za faktorizaciju su iz dana u dan sve bolji i bolji, te je danas 512-bitni RSA algoritam nedovoljan za bezbedno kriptovanje poruka. Međutim pretpostavlja se da će 1024-bitni RSA biti bezbedan još petnaestak godina.

2.7.2 Diffie-Hellman-ova eksponencijalna razmena ključeva

Diffie i *Hellman* su predložili šemu za kriptovanje podataka koja kao algoritam za razmenu javnog ključa koristi stepen modula *q*. Ova šema se u literaturi često može sresti kao D-H kriptografski protokol. D-H kriptografski protokol se bazira na šemi eksponencijalne razmene tajnog ključa, i nije predviđen za razmene poruka. Slika 2.22 prikazuje pomenutu šemu [11].



Slika 2.22
Postupak D-H kriptovanja

Osoba *A* u cilju razmene tajnog ključa sa osobom B šalje poruku α. Metodom izbora slučajnog broja generiše celobrojnu vrednost *x* iz skupa brojeva $\{1,2,\ldots,p-1\}$. Vrednost *x* koristi za kriptovanje pruke pomoću formule:

$$X = \alpha^x \bmod p$$

Ovako formiran podatak *X* se šalje do odredišta *B*. Kada primi poruku i sazna *X*, *α* i *p* Osoba *B* takođe na osnovu slučajnog izbora generiše broj *y* i pomoću njega računa vrednost *Y* po formuli:

$$Y = \alpha^y \bmod p$$

Ovako dobijenu funkciju vraća osobi *A*. Istovremeno na osnovu podataka koje je dobio od *A* i broja *y* (koga je sam generisao) računa vrednost ključa *K* po formuli:

$$K = X^y \bmod p$$

Kako je i osoba *A* dobila vrednost *Y* može izračunati vrednost ključa *K* po formuli:

$$K = Y^x \bmod p$$

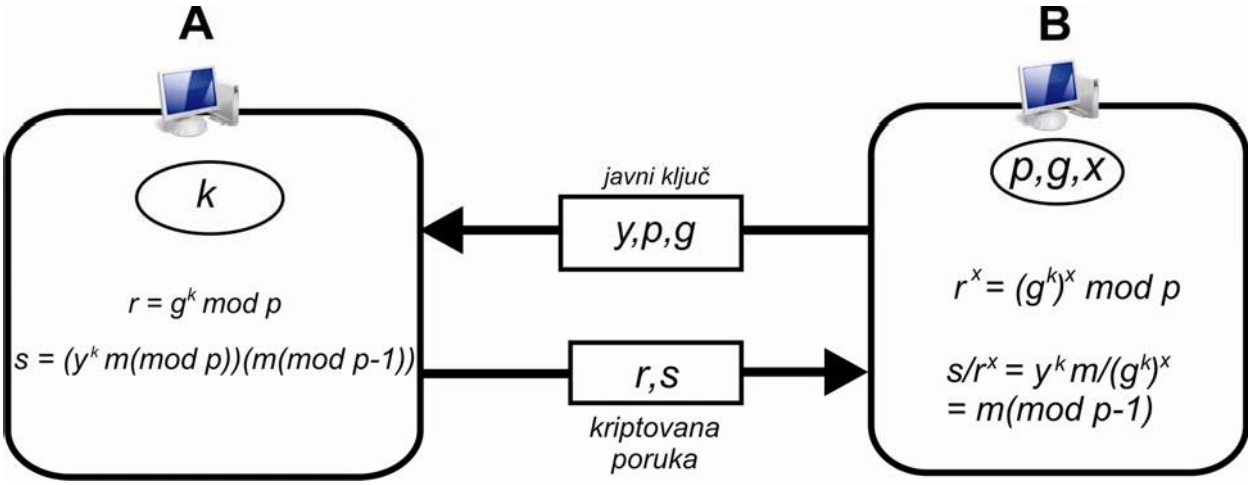
Kako je $X^y = \alpha^{xy} \bmod p = Y^x$ to je zajednički tajni ključ $K = \alpha^{xy} \bmod p$. Na ovaj način osobe *A* i *B* dolaze do zajedničkog tajnog ključa, koga nadalje koriste za kriptovanje odnosno dekriptovanje međusobnih poruka [23].

Ukoliko neka osoba *C* želi da dođe u posed zajedničkog tajnog ključa, znajući *X*, α i *p,* prinuđena je da koristi diskretni logaritam $y = dlog_{\alpha,p}(Y)$. Sigurnost D-H razmene ključeva leži u činjenici da iako je relativno lako naći eksponente po modulu prostog broja, jako je teško naći diskretne logaritme. Ovo je naročito nezamislivo za velike proste brojeve.

2.7.3 ELGamal algoritam

Taher Elgamal je 1985. godine formirao jedan asimetrični kriptografski algoritam koji se bazira na *Diffie-Hellman*-ovoj razmeni ključeva. *ElGamal* algoritam je pogodan za kriptovanje poruka, kao i za digitalno potpisivanje, te se danas koristi u okviru PGP (eng. *Pretty Good Privacy*) softvera za zaštitu podataka. Slično kao i kod D-H algoritma bezbednost *ElGamal*-a se zasniva na teškom ili gotovo nemogućem izračunavanju diskretnog logaritma od velikog prostog broja [11].

Na Slici 2.23 je u vidu blokova prikazan algoritam za slanje izvorne poruke *m* od strane osobe *A* ka osobi *B*.



Slika 2.23
Postupak ElGamal kriptovanja

Prvo se obavlja generisanje i slanje javnog ključa od strane osobe *B*. Iz skupa prostih brojeva se odabere broj *p* kao i dva prosta broja *g* i *x*. Prilikom izbora brojeva potrebno je da važi $g < p$ i $x < p$. Broj *x* predstavlja tajni ključ osobe *B*, a javni ključ se računa po formuli $y = g^x \bmod p$. Vrednosti *y*, *g* i *p* čine javni ključ osobe *B*, koji se šalje osobi *A*.

Kriptovanje podataka *m* obavlja osoba *A*. Nakon što je saznala funkciju *y* (odnosno javni ključ osobe B) osoba *A* generiše slučajni broj *k* iz skupa prostih brojeva koji zadovoljava uslov $NZD(k,p-1) = 1$. Pomoću broja *k* se formira funkcija $r = g^k \bmod p$. Uz pomoć funkcije *y* računa se ključ sesije po formuli $s = y^k \pmod p (m \pmod{p-1})$. Sada se kriptovani tekst može predstaviti kao uređeni par (*r, s*).

Postupak dekriptovanja obavlja osoba *B* nakon primanja kriptovane poruke (*r, s*). Prvo se uz pomoć tajnog ključa x računa vrednost rx po formuli $r^x = (g^k)^x \bmod p$. Nakon toga se računa izvorna poruka pomoću jednačine $s/r^x = y^k m/(g^k)^x = (g^x)^k m/(g^k)^x = m \pmod{p-1}$.

Bezbednost ovog algoritma se može ugroziti ukoliko napadač poseduje par (*r, s*) na osnovu koga može rekonstruisati uređeni par (*r, 2s*) a samim tim i poruku *2m*.

Što se tiče efikasnosti kriptovanja, *ElGamal* algoritam se ne može pohvaliti zato što je odnos izvornog teksta prema kriptovanom tekstu 2:1. Sa druge strane, izvorni tekst se može kriptovati u više oblika kriptovanog teksta što ga čini veoma fleksibilnim [11].

2.8 Prednosti i nedostatci asimetričnih algoritama

Kriptovanje/dekriptovanje pomoću asimetričnog ključa ima dve prednosti: Prvo, rešava nedostatake deljenja ključa kod simetričnih algoritama prilikom komunikacije između dve osobe (*A* i *B*). Kod simetričnog kriptovanja ključ se razmenjuje između dve osobe i ne može se koristiti ukoliko jedna od dve osobe želi komunicirati sa trećom osobom. Kod asimetričnih kriptosisitema svaka osoba kreira po dva ključa, jedan je tajni koji osoba čuva, a drugi je javni koji se razmenjuje sa drugima. Svaki od entiteta je nezavistan i svoj par ključeva može koristiti u komunikaciji sa bilo kime. Druga prednost se ogleda u veoma velikom smanjenju broja ukupno potrebnih ključeva. U sistemu u kome komunicira milion korisnika, potrebno je samo 2 miliona ključeva, dok bi u slučaju korišćenja simetričnog kriptovanja bilo potrebno bar 500 milijardi ključeva.

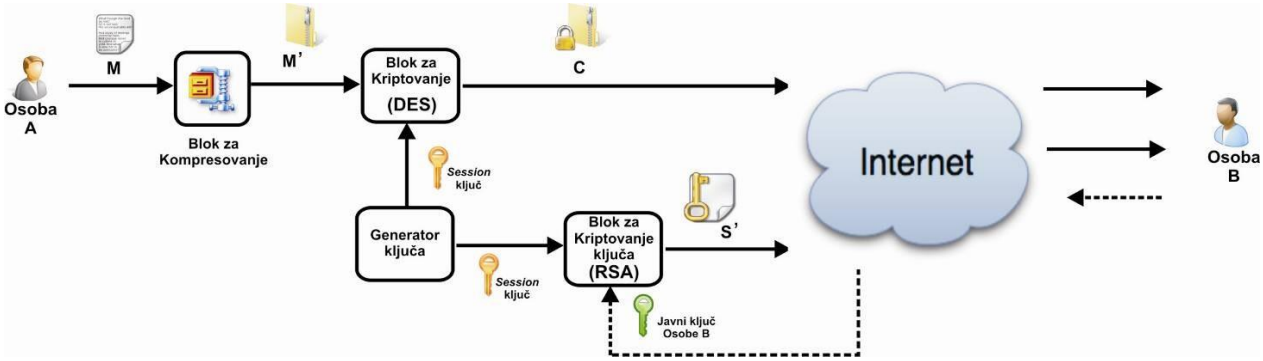
Simetrični algoritmi, takođe, imaju dva nedostatka. Najveći nedostatak je kompleksnost algoritama koji se koriste prilikom kriptovanja. Ako se želi efektno kriptovanje to povlači da algoritam koristi ogromne ključeve prilikom rada. Operisanje sa ogromnim brojevima zahteva mnogo vremena. Zbog toga asimetrični algoritmi nisu preporučljivi za rad sa velikim izvornim podacima. Može se reći da su asimetrični algoritmi mnogo efikasniji u radu sa kratkim porukama. Isto tako, ova vrsta algoritama zbog svoje složenosti nisu pogodni za hardversku implementaciju. Drugi nedostatak je taj što se komunikacija između dve strane i javni ključ moraju verifikovati. Kako osoba *A* šalje svoj javni ključ osobi *B* putem elektronske pošte, osoba *B* na neki način mora biti sigurna da je dobijeni ključ upravo poslat od strane osobe *A*. Ovo je naročito važno ukoliko se radi o korišćenju asimetričnog kriptovanja prilikom identifikacije korisnika na neki sistem.

2.9 Hibridni kriptosistemi

Imajući u vidu da upotreba simetrične ili asimetrične kriptografije pati od izvesnih nedostataka javlja se potreba za sistemima koji kombinuju najbolje pojedinačne karakteristike oba sistema. Tako su nastali Hibridni kriptosistemi. Princip rada ovih sistema se ogleda u sledećem: Izvorni tekst se prvo kriptuje upotrebom ključa (ponekad nazvan i *session* ključ), a zatim se taj ključ zajedno sa kriptovanom porukom pakuje i ponovo kriptuje sa javnim ključem osobe kojoj se šalje poruka. Postupak dekripcije cele poruke se ostvaruje obrnutim redosledom operacija: Osoba koja je primila poruku prvo dekriptuje istu sa svojim tajnim kljucem, pronalazi zapakovani *session* ključ i koristi ga da bi pročitala izvornu poruku [4].

Za generisanje session ključa koristi se generator pseudo-slučajnog broja u kombinaciji sa raznim korisnikovim unosima u toku procesa generisanja. Na ovaj način se postižu zavidne performanse sistema za kriptovanje, jer se asimetrično kriptuje samo kratak simetrični ključ, a ne cela, velika poruka. Znamo da asimetrično kriptovanje nije pogodno za poruke velike dužine.

Primer uspešnog hibridnog kriptosistema je PGP (eng. *Pretty Good Privacy*) programski paket za kriptovanje podataka koji je formirao *Phill Zimmerman* 1991. godine. Princip rada PGP algoritma je prikazan ana Slici 2.24.



Slika 2.24

Postupak kriptovanja pomoću PGP algoritma

Osoba *A* pre slanja poruke *M* Osobi *B* prvo izvrši kompresiju pomoću neke od metoda za kompresiju podataka (npr. ZIP). Komprimovanjem poruke se ubrzava vreme slanja poruke, a istovremeno i dobija na zaštiti poruke. Tako kompresovana poruka *M'* se kriptuje pomoću nekog od simetričnih algoritama (*DES, 3DES, IDEA* ili slično) i *session* ključa. Na taj način se dobija kriptovana poruka *C*. Zatim se vrši kriptovanje *session* ključa pomoću nekog od asimetričnih algoritama (*RSA, ElGamal* ili *Diffie-Hellman*) i javnog ključa osobe *B*. Tako nastaje kriptovana poruka *S'* koja se šalje osobi *B* zajedno sa porukom *C* [24].

Proces dekriptovanja se obavlja obrnutim redosledom radnji od procesa kriptovanja. Kada primi kriptovanu poruku osoba *B* upotrebljava svoj tajni ključ i dekriptuje poruku koja sadrži *session* ključ. Korišćenjem *session* ključa osoba *B* dekriptuje kriptovanu poruku a kasnije je i dekompresuje u cilju dobijanja izvorne poruke.

PGP program nudi nekoliko stepeni zaštite: niski, visoki i vojni. Prvi koristi 512-bitni ključ, drugi 768-bitni a vojni koristi 1024-bitni ključ. Osim generisanja para ključeva, PGP algoritam nudi još dosta sigurnosnih mera tipa: za generisanje ključeva se ne koristi generator ključeva već algoritam traži od korisnika da unese neke podatke pomoću tastature, a za to vreme meri razmake između vremena udaraca i na osnovu njih generiše ključeve. Isto tako PGP nudi potpisivanje poslatih dokumenata u cilju dokazivanja autentičnosti.

3 Tehnika digitalnog potpisa

Digitalni potpisi se koriste za identifikaciju izvora informacije koji može biti neka osoba, organizacija ili računar. Ideja digitalnog potpisa je slična klasičnom potpisivanju dokumenata. Ukoliko se neki dokument želi poslati elektronskim putem, takođe se mora potpisati. Za razliku od klasičnog potpisa, digitalni potpis je gotovo nemoguće falsifikovati. Potpisivanjem dokumenata osoba koja ih šalje garantuje autentičnost, integritet i neporicljivost osobi koja ih prima.

Digitalni potpisi se zasnivaju na rešenjima primenjenim u simetričnim, a češće u asimetričnim kriptografskim sistemima. Potpisivanje dokumenata bazira se na matematičkoj funkciji i dva ključa, tajnom i javnom ili samo tajnom ključu.

Sistemi za digitalno potpisivanje dokumenata bazirani na simetričnoj kriptografiji koriste tajni ključ za potpisivanje i verifikaciju poruke. Ovi sistemi su poznati pod imenom *Message Authentication Code* (MAC). Princip rada MAC-a se zasniva na nekoliko koraka. Prvo se vrši izračunavanje kontrolne sume (eng. *checksum*) izvorne poruke, zatim se koristi tajni ključ za potpisivanje dobijene kontrolne sume. Ovako potpisanu poruku može verifikovati samo osoba koja poseduje tajni ključ. Najpoznatiji tip MAC-a je HMAC ili *Hashed Message Authentication Code*. Za potpisivanje kontrolne sume izvorne poruke HAMAC koristi neku od „heš“ funkcija (eng. *hash functions*) kao što su MD5, SHA-1 ili SHA-256 i tajni ključ.

Sistemi za digitalno potpisivanje dokumenata bazirani na asimetričnoj kriptografiji poseduju dva različita ključa. Prilikom potpisivanja poruka ovom metodom postoje dve tehnike: potpisivanje cele poruke i potpisivanje kratkog sadržaja poruke (sažetka) takozvani *message digest*.

Oblasti u kojima se digitalni potpisi najviše koriste su elektronska trgovina, elektronska pošta i razne finansijske transakcije, mada se ova tehnika može naći i u drugim oblastima u cilju rešavanja mnogih problema koji se odnose na bezbednost informacija

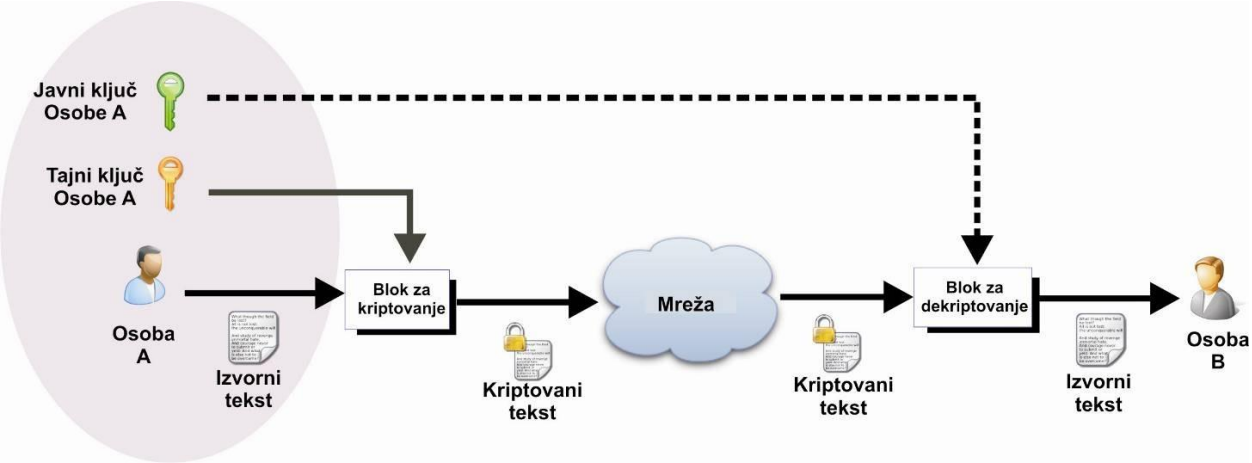
Tehnologija digitalnog potpisa sve više postaje korisnija i prihvatljivija metoda za zaštitu važnih informacija, uzimajući u obzir to da se mnoge zemlje utrkuju u razvoju što boljeg standarda za potpisivanje [27].

3.1 Potpisivanje cele poruke

Osnovu ove metode za potpisivanje čini cela poruka. Osoba koja šalje poruku koristi tajni ključ za kriptovanje cele poruke i na taj način je potpisuje. Onaj ko primi tako potpisanu poruku, koristi javni ključ da bi dekriptovao primljenu poruku i na taj način sprovodi verifikaciju osobe koja je poslala poruku.

Treba zapaziti da se prilikom korišćenja tehnike digitalnog potpisa, za kriptovanje koristi tajni, a za dekriptovanje javni ključ, što je obrnuti postupak u odnosu na postupak koji sprovode asimetrični algoritmi za kriptovanje. Ovo je moguće zato što su algoritmi koji su danas u upotrebi, kao RSA, bazirani na simetričnim matematičkim formulama. Obrnuti postupak korišćenja ključeva ovde je opravdan iz razloga što je cilj verifikacija identiteta osobe koja je poslala poruku, a ne da se zaštiti sadržaj poruke. Na Slici 3.1 je prikazan metod potpisivanja cele poruke.

Osoba A šalje izvorni tekst osobi B ali ga pre toga kriptuje pomoću svog tajnog ključa. Osoba B nakon primanja potpisanog teksta (kriptovani tekst) vrši dekriptovanje pomoću javnog ključa osobe A. Na sledećim primerima se može pokazati kako se postiže autentičnost, integritet i neporicljivost potpisivanja cele poruke, što predstavlja svrhu digitalnog potpisivanja dokumenata.



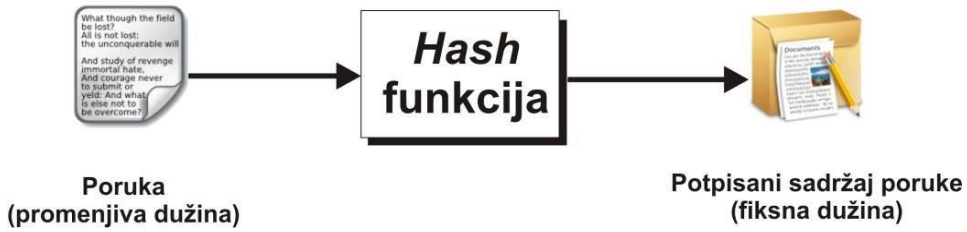
Slika 3.1
Tehnika potpisivanja cele poruke

Ukoliko neka osoba C želi poslati poruku, a da to izgleda kao da ju je poslala osoba A, onda za kriptovanje mora koristiti tajni ključ osobe A. Ako osoba C ne poseduje ovakav ključ i pošalje poruku, takva poruka se ne može korektno dekriptovati korišćenjem javnog ključa osobe A. Ako neka osoba C neovlašćeno dođe do kriptovanog sadržaja poslate poruke (uhvati poruku), pokuša da je promeni bilo celu ili neki njen deo, kasnijim dekriptovanjem takve poruke se dobija nečitljiv sadržaj. Na taj način se održava integritet poruke. Ako se poruka koju je poslala osoba A zabeleži na nekom pomoćnom mestu i osoba A kasnije ospori da je poslala poruku, osoba B može naći poruku na snimljenom mestu i pomoću javnog ključa osobe A može formirati duplikat originalne poruke. Znači tehnika potpisivanja cele poruke zadovoljava karakteristiku neporicljivosti [10].

3.2 Potpisivanje sažetka poruke

Ukoliko su poruke dugačke, korišćenje kriptovanja sa javnim ključem za potpisivanje cele poruke je veoma nepraktično. Nepraktičnost se ogleda u veoma velikim dužinama poruka, što iziskuje dosta resursa i troši mnogo vremena za kriptovanje. Kao logično rešenje ovog problema javlja se mogućnost , potpisivanja samo sažetka umesto potpisivanja cele poruke. Osoba koja šalje poruku kreira skraćenu verziju poruke tj. njen sažetak. Tako formiran sažetak potpisuje i šalje komunikacionim kanalom. Osoba koja primi tako skraćenu poruku proverava njen potpis. Svaka promena izvorne poruke izaziva promenu u sadržaju, što se odražava na promenu potpisa, čime se minimizuje mogućnost zloupotrebe [10].

Za kreiranje sadržaja poruke se koristi pomenuta heš funkcija za sažimanje. Ova funkcija, bez obzira na dužinu poruke, formira sadržaj fiksne dužine, što je ilustrovano na Slici 3.2.

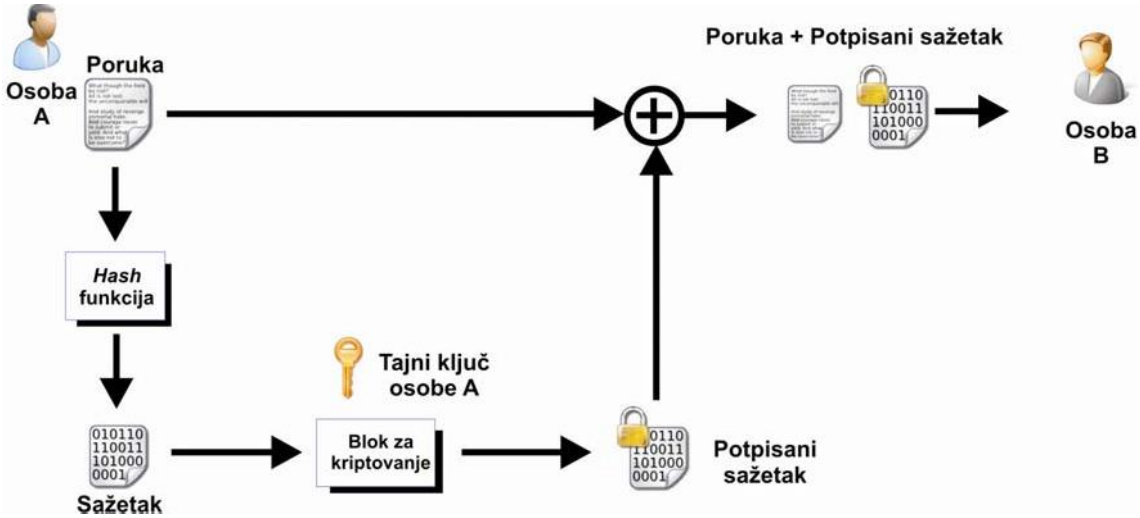


Slika 3.2
Primena hash funkcije

Tehnika potpisivanja sadržaja poruke uglavnom koristi neku od dve heš funkcije: MD5 (*Message Digest 5*) sa 128-bitnim sadržajem i SHA-1 (*Secure Hash Algorithm 1*) sa 160-bitnim sadržajem. Kako

bi garantovala siguran prenos poruke heš funkcija mora da zadovolji dve stvari: Prvo, funkcija sažimanja se obavlja u jednom smeru. Sadržaj se jedino može formirati na osnovu originalne poruke ne i obrnuto. Uz to, formiraje sadržaja treba biti veoma brzo i jednostavno; Drugo, heš funkcija je jednoznačna, tj primena iste heš funkcije na istoj poruci daje isti sažetak.

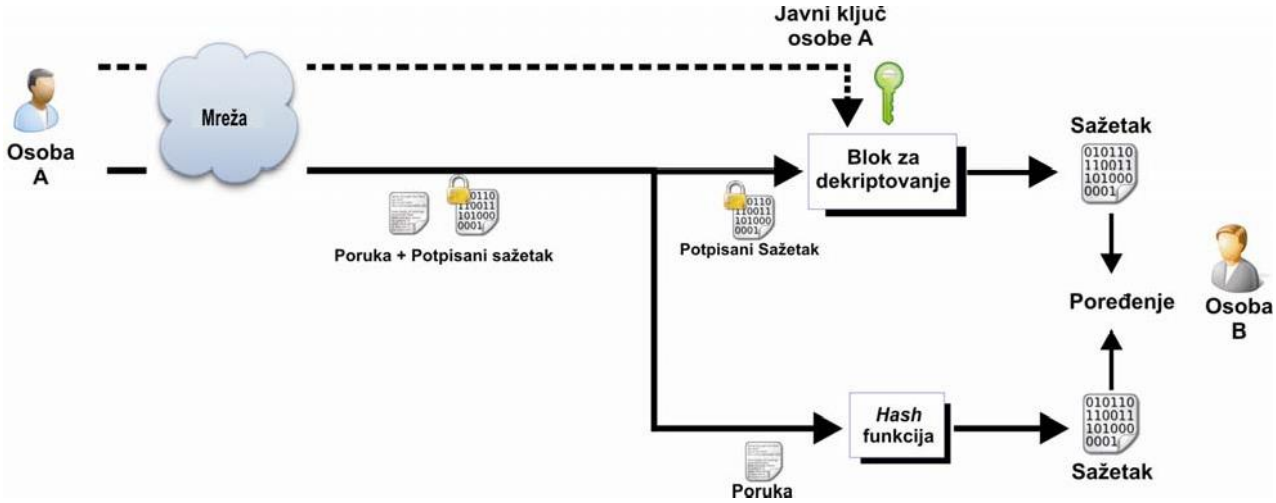
Nakon kreiranja sadržaja poruke, vrši se kriptovanje (potpisivanje) istog korišćenjem tajnog ključa osobe koja šalje poruku (osoba A). Obično se za kriptovanje koristi RSA algoritam. Kriptovani sadržaj se upakovan zajedno sa originalnom porukom šalje osobi B. Na Slici 3.3 je prikazan postupak potpisivanja sažetka. Osoba A primenom heš funkcije formira sažetak koji se potpisuje i upakovan sa porukom šalje osobi B [10].



Slika 3.3
Postupak slanja sadržaja poruke

Osoba B prima originalnu poruku i kriptovani sadržaj zajedno sa potpisom, a nakon prijema vrši razdvajanje.

Na Slici 3.4 je prikazana strana osobe B u procesu primanja poruke. Primenom heš funkcije na originalnu poruku osoba B kreira drugi sadržaj. Takođe, dekriptuje kriptovani sadržaj koji je primila od strane osobe A uz pomoć javnog ključa osobe A. Konačno, vrši poređenje predhodno kreiranih sadržaja i ukoliko su isti uspešno je obavljena verifikacija poruke.



Slika 3.4
Postupak prijema potpisanog sadržaja poruke

Kada se govori o izvornoj poruci sledeća zapažanja ukazuju na njenu sigurnost: Ako sadržaj dolazi od verifikovanog pošiljaoca, isto tako i poruka dolazi u paketu od verifikovanog pošiljaoca, čime je zadovoljena autentičnost poruke. Zbog karakteristike jednoznačnosti sadržaja, ukoliko je sadržaj prenešen nepromenjen, zaključuje se da je i poruka prenešena nepromenjena čime se zadovoljava integritet. Osoba koja šalje poruku ne može se odreći sadržaja, a samim tim i poruke koja se šalje zajedno sa sadržajem, što ukazuje na neporicljivost [10].

3.3 Heš funkcije

3.3.1 MD5

MD5 (eng. *Message Digest Algorithm 5*) je heš funkcija koja se primenjuje u aplikacijama za digitalno potpisivanje dokumenata. Dužina sadržaja koji se formira na osnovu MD5 funkcije je kratka (128 bita) što ga čini pogodnim za brzu proveru identiteta osoba koje šalju obimne dokumente.

Algoritam MD5 funkcije je razvio *Ron Rivest* 1991. godine, kao zamenu za MD4 algoritam. Nakon pet godina otkriveni su mali nedostatci u algoritmu te su kriptografi preporučivali upotrebu drugih heš funkcija. Nekoliko narednih godina su otkriveni dodatni nedostaci te je upotreba ovog algoritma dovedena u pitanje. Tokom 2005. godine grupa istraživača je uspela da formira isti sadržaj primenjujući MD5 na dva različita dokumenta. Zbog pronađenih nedostataka, danas se ovaj algoritam sve ređe koristi za digitalno potpisivanje, ali je našao primenu u proveri integriteta fajlova, gde se koristi za izračunavanje kontrolnih suma, kod kojih sigurnost nije prioritetna [11].

MD5 algoritam kao ulaznu informaciju koristi *w*-bitni broj. Izvorni tekst se može prikazati kao niz brojeva [26]:

$$m_0, m_1, m_2, \ldots, m_{w-2}, m_{w-1}$$

Gde je broj *w,* vrednost iz proširenog skupa prirodnih brojeva.

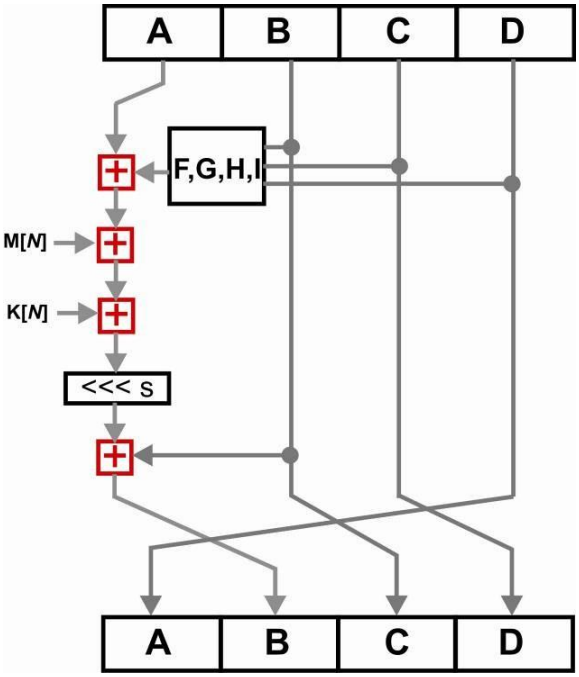
Na početku je potrebno izvršiti dopunu ulazne informacije do vrednosti koja se dobija od broja koji je celobrojni umnožak od 512 bita umanjenog za 64 bita. Na primer, ukoliko se izvorna poruka sastoji od 128 bitova ($w = 127$) potrebno je dopuniti do 448 bitova tj. $512 - 64 = 448$. Dopuna se započinje sa početnim bitom „1“ a svi ostali bitovi za popunjavanje imaju vrednost „0“.

Nakon dopune poruke, izvornoj poruci je potrebno dodati 64-bitnu reprezentaciju broja *w*. Ukoliko je dužina poruke veća i ne može da se predstavi pomoću 64 bita, poruci se dodaje samo nižih 64 bita. Dodavanjem ovih 64 bita dužina cele poruke postaje deljiva sa 512, odnosno deljiva sa 16 reči od 32-bita. Sada se poruka može prikazati kao:

$$M\left[1,2,\ldots,N\right]$$

Gde je *N* broj deljiv sa 16. Ovako pripremljenu poruku algoritam kasnije koristi prilikom formiranja sadržaja.

Na Slici 3.5 je prikazan izgled jednog od moguća 64 koraka izvršenja MD5 algoritma. Nakon predhodne pripreme poruke, potrebno je inicijalizovati 128-bitni bafer koji se sastoji od četiri 32-bitna registra A, B, C i D. Kao inicijalne vrednosti koje se upisuju u ove registre koriste se proizvoljne 32-bitne konstante [27].



Slika 3.5
Korak MD5 algoritma

Posle završene inicijalizacije, pokreće se prvi korak MD5 algoritma. Ukupnih 64 koraka se deli u četiri ciklusa od po 16 koraka. Algoritam je formiran za izvršenje 512 bita poruke, što znači da ukoliko je poruka duža od 512 bita izvršenje algoritma se mora ponoviti.

Algoritam se sastoji od četiri ciklusa koji imaju isti tok s tim što se prilikom izračunavanja u svakom od ciklusa koristi različita logička funkcija F, G, H i I. Funkcije se računaju po formulama:

$$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$
$$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$
$$H(X,Y,Z) = X \oplus Y \oplus Z$$
$$I(X,Y,Z) = Y \oplus (X \vee \neg Z)$$

Gde su $\wedge, \vee, \neg, \oplus$ operacije AND, OR, NOT i XOR respektivno [27].

Tokom ciklusa se koriste operacije aritmetičkog sabiranja po modulu 2^{32} i operacija pomeranja ulevo za *s* pozicija, gde je *s* vrednost različita za svaki ciklus. $M[N]$ predstavlja 32-bitnu ulaznu poruku, a $K[N]$ je konstanta koja je drugačija za svaki ciklus. Ukupno 16 $M[N]$-ova se koristi tokom 16 koraka u okviru svakog od 4 ciklusa.

Rezultat jednog koraka se koristi kao početna vrednost (A, B, C i D) narednog koraka. Na kraju se konačna vrednost formiranog sadržaja upisuje u registre A, B, C i D.

3.3.2 SHA-1 algoritam

SHA (*Secure Hash Algorithm*) se pojavio 1993. godine (u literaturi se često koristi termin SHA-0) od strane američke vladine agencije Nacionalnog instituta za standarde i tehnologiju (NIST), kao zvanični standard za sažimanje poruka. Zbog pronađenih propusta u sigurnosti ubrzo je povučen, a nasledio ga je SHA-1 1995. godine .

Obe varijante SHA-0 i SHA-1 formiraju sadržaje dužine 160 bita, a maksimalna dužina izvorne poruke može biti 2^{64}. Nešto kasnije su se pojavile varijante ovog algoritama koje formiraju i duže

sadržaje. To su SHA-256, SHA-224, SHA-384 i SHA-512 a nazive su dobile po dužini sadržaja. Zbog dužeg sadržaja ove varijante su sigurnije rešenje u odnosu na SHA-0 i SHA-1. Ovi algoritmi čine grupu nazvanu SHA-2. Iako je ova grupa algoritama po pitanju sigurnosti bolje rešenje, danas je zbog jednostavne implementacije i brzine najviše u upotrebi SHA-1.

U osnovi SHA-1 je baziran na idejama MD4 i MD5 algoritama. Ulazna poruka se dopuni po potrebi slično kao i kod MD5 a nakon toga smešta u ulazni bafer koji se sastoji od pet 32-bitnih registra. Izračunavanje se vrši u okviru 80 koraka i pritom se koriste operacije aritmetičkog sabiranja po modulu 2^{32}, pomeranja ulevo za određeni broj pozicija i četiri tipa funkcija. Prvi tip funkcije se koristi u okviru prvih dvadeset koraka a računa se po formuli [11]:

$$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$

U okviru drugih dvadeset koraka se koristi funkcija:

$$G(X,Y,Z) = X \oplus Y \oplus Z$$

Od četrdesetog do šezdesetog koraka se koristi funkcija oblika:

$$H(X,Y,Z) = (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z)$$

Funkcija koja se upotrebljava u okviru poslednjih dvadeset koraka je:

$$I(X,Y,Z) = X \oplus Y \oplus Z$$

Gde su $\wedge, \vee, \neg, \oplus$ operacije AND, OR, NOT i XOR respektivno. Konačan rezultat se smešta u početna pet 32-bitna registra.

SHA-1 predstavlja industrijski standard u kriptografiji i našao je primenu u mnogim aplikacijama kao što su TLS, SSL, PGP, SSH, S/MIME i IPSec (o kojima će više reči biti kasnije).

4 Autentifikacija entiteta

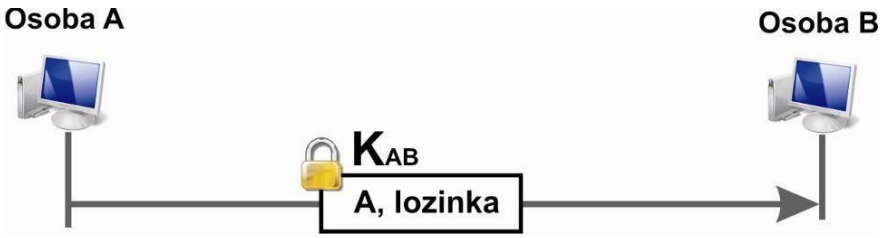
Autentifikacija predstavlja proceduru pomoću koje se vrši uzajamna verifikacija identiteta dva entiteta. Kada se govori o entitetu misli se na neku osobu, proces, klijenta ili server. U nastavku će se, radi jednostavnijeg objašnjenja procedura, pojam entiteta odnositi na neku osobu. Autentifikacija entiteta se može obavljati korišćenjem simetričnog ili asimetričnog kriptovanja.

4.1 Autentifikacija entiteta pomoću simetričnog kriptovanja

Kada se radi o tehnikama digitalnog potpisivanja, diskutovano je o autentifikaciji poruke odnosno njenog sadržaja, što nikako ne treba poistovetiti sa autentifikacijom osoba. Osnovna razlika između ova dva pojma je ta što se kod autentifikacije poruka verifikacija i provera identiteta obavljaju prilikom slanja svake poruke, dok kod autentifikacije osoba, verifikacija obavlja samo jednom, na početku sesije. Autentifikacija entiteta pomoću simetričnog kriptovanja se javlja u tri vida [10].

4.1.1 Prvi vid autentifikacije entiteta

Na Slici 1 je prikazana procedura koja opisuje prvi vid autentifikacije. Osoba A šalje svoj identitet i lozinku u okviru kriptovane poruke. Za kriptovanje ove poruke koristi se tajni ključ K_{AB}. Na Slici 4.1 je ključ prikazan u okviru katanca kako bi prikazao da je poruka kriptovana.

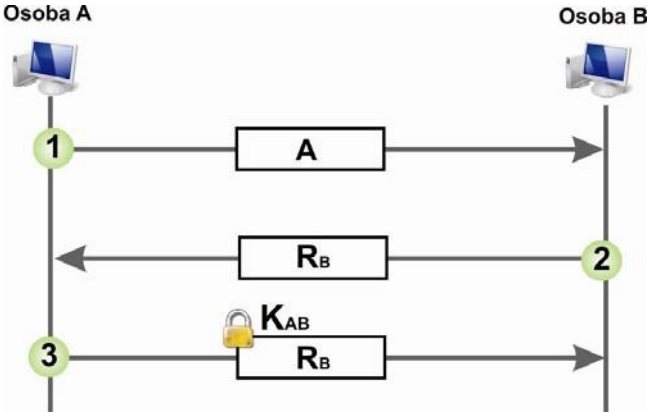


Slika 4.1
Prvi vid autentifikacije entiteta

Postavlja se pitanje da li je ovaj vid autentifikacije dovoljno bezbedan. Činjenica je da napadač, odnosno osoba C ne može videti sadržaj ili lozinku zato što ne poseduje ključ K_{AB}. Međutim, ukoliko osoba C uhvati obe poruke i autentifikacionu i poruku koja nosi podatke, može da ih snimi negde i kasnije ih pošalje ka B. Osoba B ne zna da li je poruka poslata od osobe A ili od osobe C. Znači, tokom ovog vida autentifikacije ne postoji ništa što može garantovati autentičnost izvora poruke. Posledice usled zloupotrebe mogu biti značajne. Na primer, Osoba A pošalje poruku osobi B (na primer bankarski menadžer) da izvrši plaćanje nekoj osobi C. Osoba C može da uhvati poruku i da je pošalje osobi B. Ne znajući da poruka dolazi od osobe C (a ne od A) osoba B može po drugi put izvršiti isplatu. Na taj način je osoba C izvršila zloupotrebu koja se naziva ponovljeni napad [10].

4.1.2 Drugi vid autentifikacije entiteta

Da bi se izbegla opasnost od ponovljenog napada, u proceduru za autentifikaciju se mora dodati nešto što će osobi koja prima poruku omogućiti da identifikuje ponovljeni autentifikacioni zahtev. To se postiže korišćenjem jednog velikog broja (128 bita) dobijenog slučajnim odabirom. Ovaj broj se koristi samo jednom u toku sesije i naziva se jednokratni (*one-time*) broj. Na Slici 4.2 je prikazana procedura drugog vida autentifikacije.



Slika 4.2
Drugi vid autentifikacije

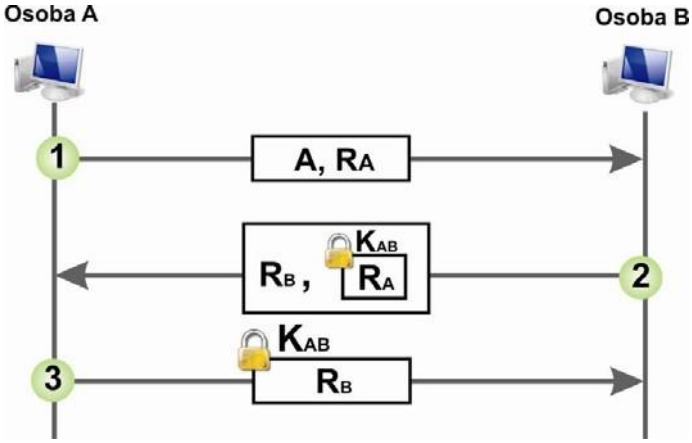
Autentifikacija entiteta se obavlja u tri koraka: Prvo, osoba A šalje osobi B svoju identifikaciju u obliku izvornog teksta. Nakon toga, Osoba B proziva osobu A slanjem *one-time* broja R_B takođe u obliku izvornog teksta. Na ovaj način osoba B saznaje da osoba A želi razmenu poruka, a slanjem broja R_B ona želi da proveri da li je poruka zaista stigla od osobe A. U poslednjem koraku, nakon prepoznavanja prozivke od strane osobe B, osoba A se odaziva vraćanjem kriptovanog broja R_B nazad ka osobi B. Za kriptovanje broja R_B se najčešće koristi metoda simetričnog kriptovanja pomoću tajnog ključa. U okviru ovog koraka osoba A dokazuje da je ona upravo ta koja želi razmenu sa osobom B [10].

Prednost ove metode za autentifikaciju entiteta se ogleda u jednostavnosti implementacije, odnosno sprovođenje uzajamne autentifikacije pomoću relativno malog broja koraka. Osnovni nedostatak je taj što je izvršena samo verifikacija osobe A, dok je identitet osobe B nepoznat.

4.1.3 Bidirekciona autentifikacija

Naprednija verzija drugog vida autentifikacije entiteta predstavlja takozvana bidirekciona autentifikacija. Na Slici 4.3 je prikazana metoda bidirekcione autentifikacije. Prednost ovog vida autentifikacije se ogleda u tome što se za isti broj koraka (tri koraka) ostvaruje uzajamna autentifikacija entiteta [10].

Na početku, osoba A šalje sopstvenu identifikaciju u obliku izvornog teksta i one time broja R_A radi pozivanja osobe B, sa kojom želi izvršiti uzajamnu verifikaciju.



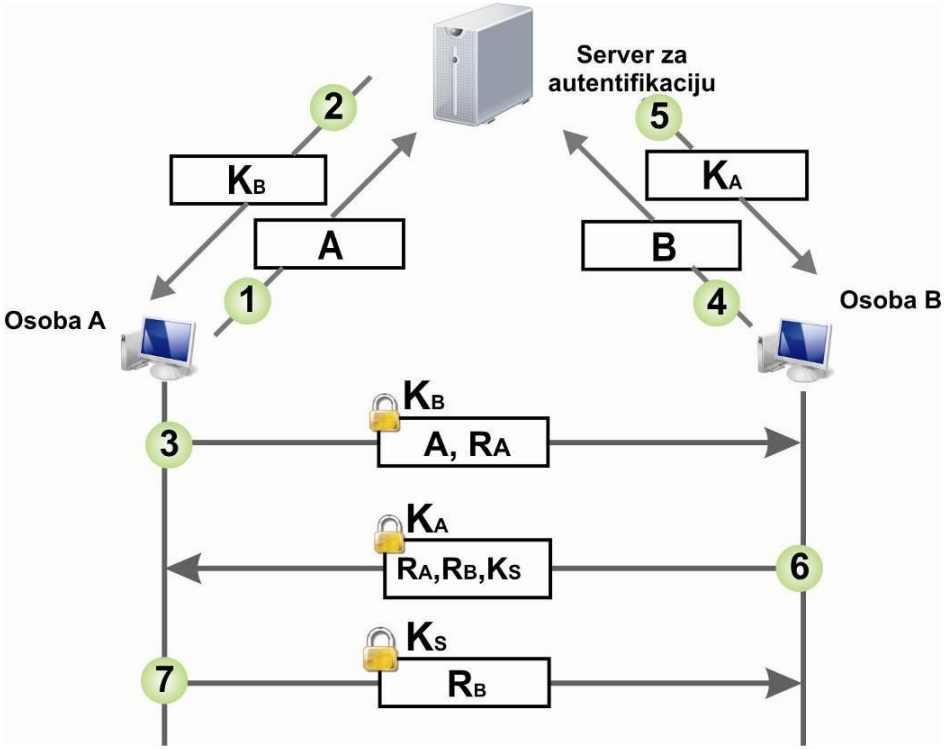
Slika 4.3
Bidirekciona autentifikacija entiteta

U drugom koraku, osoba B se odaziva na poziv osobe A slanjem kriptovanog broja R_A (pomoću tajnog ključa K_{AB}) i slanjem sopstvenog broja R_B. Time osoba B obaveštava osobu A da je primila poziv, a ujedno želi da proveri da li je poruka stvarno poslata od strane osobe A. Nakon poslate poruke osoba B očekuje povratnu poruku koja će joj dokazati da je osoba A pročitala predhodno poslatu poruku i pokazati da poseduje tajni ključ K_{AB}. Kada je primila poruku od strane osobe B, osoba A je ustanovila da je poruka sigurno došla od osobe B (vraćen joj je poslati broj R_A), a ujedno i stekla utisak da osoba B poznaje tajni ključ K_{AB}. Ukoliko to nije tako, već je poruku poslala osoba koja se lažno prestavlja kao osoba B, osoba A bi dekriptovanjem dobijene poruke dobila neispravnu vrednost broja R_A [10].

U trećem koraku, osoba A se odaziva osobi B slanjem kriptovanog (K_{AB}) broja R_B. Imajući u vidu činjenicu da se za svaku sesiju koriste različiti setovi brojeva za R_A i R_B i da prilikom jedne sesije se jednovremeno obavlja samo jedna autentifikacija, može se zaključiti da je ova metoda veoma jednostavna i bezbedna u smislu ponovljenog napada. Kao takva predstavlja odlično rešenje za uzajamnu autentifikaciju entiteta [10].

4.2 Autentifikacija entiteta korišćenjem asimetričnog kriptovanja

Pored korišćenja simetričnog načina kriptovanja za autentifikaciju se mogu koristiti i asimetrične metode. Na Slici 4.4 je prikazana autentifikacija entiteta asimetričnim kriptovanjem. Prilikom autentifikacije entiteta, korišćenjem asimetričnog kriptovanja, osobe u komunikaciji (osoba A i osoba B) koriste usluge servera za autentifikaciju.



Slika 4.4
Autentifikacija entiteta korišćenjem asimetričnog kriptovanja

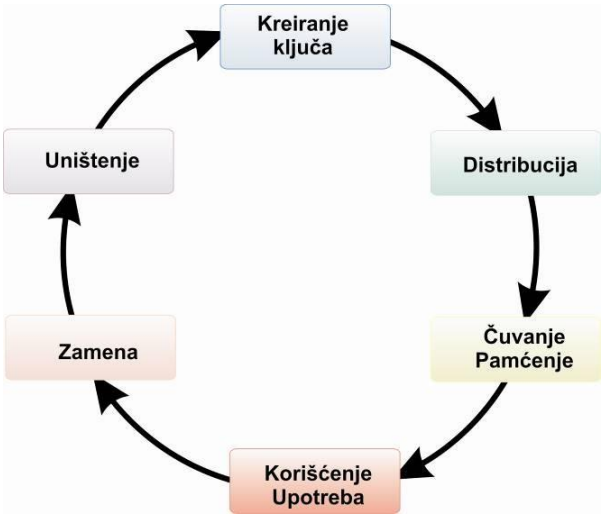
Osoba A pre nego što započne komunikaciju sa osobom B, obraća se serveru za autentifikaciju i od njega traži javni ključ osobe B (K_B). Kada server ustanovi identitet osobe A šalje joj poruku sa ključem K_B. Pošto je primila poruku, osoba A formira pseudo-slučajni broj R_A i zajedno sa svojim identitetom ga kriptuje pomoću javnog ključa K_B. Poruka je kriptovana pomoću javnog ključa osobe B što znači da je može dekriptovati jedino osoba B pomoću svog tajnog ključa. Osoba B, kada primi poruku dekriptuje je

sa svojim tajnim ključem, i traži od servera javni ključ osobe A, K_A. Server, takođe proverava identitet osobe B i ako je sve u redu šalje joj traženi ključ K_A. Koristeći javni ključ K_A, osoba B kriptuje poruku koja sadrži primljeni broj R_A, njen broj R_B i ključ sesije K_S. Ovako formiranu poruku osoba B šalje osobi A, i na taj način joj dokazuje da je uspešno došla do broja R_A i da je ona upravo ta osoba kojoj su poruke namenjene. Istovremeno formira ključ sesije K_S koji će kasnije služiti kao zajednički ključ za sigurnu razmenu podataka između ova dva entiteta. Na kraju, osoba A kada primi poruku dekriptuje je sa svojim tajnim ključem, uzima broj R_B i koristeći ključ sesije K_S kriptuje poruku. Tako kriptovanu poruku šalje nazad ka osobi B. Na ovaj način je osoba A dokazala osobi B da je uspešno dekriptovala poruku i da je primila ključ za obavljanje razmene podataka K_S [10], [28].

5 Upravljanje ključevima

Ukoliko neka osoba poseduje određeni ključ za dekriptovanje lako može pročitati odgovarajuću poruku, a ukoliko se ključ ne poseduje čitanje poruke je nemoguće. Upravo ta činjenica ukazuje na to da su ključevi najbitnije informacije u kriptografiji, a samim tim upravljanje ključevima je od suštinske važnosti u svim kriptosistemima.

Na Slici 5.1 je prikazan životni vek ključa. Svaki ključ započinje svoj život kreiranjem, a završava sa uništenjem. Pre upotrebe ključ mora da prođe kroz fazu distribucije do korisnika i čuvanje ključa do upotrebe. U dosta slučajeva se dešava da je potrebno da se ključ, usled nekog nedostatka, zameni odgovarajućim novim ključem.



Slika 5.1
Zivotni vek ključa

Zadatak upravljanja ključevima je da u svakom trenutku obezbedi tajnost i integritet ključeva. Upravljanje ključevima se ne odnosi samo na zaštitu ključeva tokom upotrebe, nego i na kreiranje sigurnog ključa, njegovu distribuciju do udaljenog korisnika, utvrđivanje korektnosti ključa i njegovo uništavanje u slučaju da se dovede u sumnju ili izgubi.

Veoma je važan odabir odgovarajućeg ključa za kriptovanje određene vrste podataka. Ranije je bilo govora o korišćenju simetrične i asimetrične kriptografije u zaštiti poruka i autentifikaciji, međutim nije bilo reči kako se vrši distribucija ključeva u okviru simetrične kriptografije ili kako se vrši provera ključeva u okviru asimetrične kriptografije. Distribucija i provera ključeva su takođe od ključne važnosti u sigurnosnim sistemima.

5.1 Kreiranje ključa

Pojedini ključevi mogu imati nezadovoljavajuće karakteristike ako se upotrebe u određenim algoritmima za kriptovanje. Na primer, ako se u okviru DES algoritma upotrebi ključ koji sadrži sve nule, po pitanju sigurnosti neće dati zadovoljavajuće rezultate. Isto tako, kada se koristi RSA algoritam brojevi p i q se moraju birati iz skupa prostih brojeva.

Većina sistema za kriptovanje poseduje više metoda za generisanje ključeva. Obično korisnik formira ključ izborom određene lozinke. Lozinka se posredstvom nekog algoritma transformiše u ključ. Na ovaj način se korisniku omogućava pamćenje jednostavne lozinke, umesto pamćenja ključa velike

dužine. Pametan izbor lozinke uključuje korišćenje brojeva i specijalnih karaktera. Treba napomenuti da je prostor predviđen za ukucavanje lozinke promenjive veličine i da je poželjno koristiti što veći broj karaktera. Ovo smanjuje mogućnost, a i produžava vreme koje je potrebno, za otkrivanje ključa. Neki ključevi se dobijaju uz pomoć pseudo-slučajnog broja. Za ovakav način dobijanja ključa potrebni su generatori slučajnih brojeva. Ukoliko generator generiše baš slučajne brojeve malo je verovatno da će napadač otkriti ključ predviđajući cifre koje ga čine.

Jedna od najvažnijih karakteristika koja se mora imati u vidu je dužina ključa. Neki od algoritama koriste ključeve fiksne dužine, kao što je DES algoritam sa ključem od 56 bita, međutim najbolja varijanta je kada korisnik može da vrši izbor dužine ključa. Ključ dužine 1024 bita u okviru RSA algoritma je sigurniji nego ključ od 512 bita kod istog algoritma.

5.2 Distribucija ključeva

Nakon generisanja ključeva, isti se mogu koristiti na različitim lokacijama i sa različitom opremom. Da bi se dopremili do udaljenih lokacija, ključevi se po pravilu transportuju preko nezaštićenih komunikacionih linija. Ako se ključevi tokom transporta ne zaštite, mogu biti „ukradeni“, što ceo sistem za kriptovanje čini nesigurnim. Preporučljivo rešenje je da se za prenos ključeva koriste sigurne linije, kao što je na primer dostavljanje ključa korisnicima putem pošte.

Isto tako, format zapisivanja ključa treba biti razumljiv i na neki način univerzalan, tj. mora biti prepoznatljiv za svakog korisnika ili mašinu. Zbog razlika u načinu funkcionisanja simetričnih i asimetričnih sistema nameće se različita upotreba ključeva što rezultuje različitim problemima prilikom distribucije istih.

5.2.1 Distribucija ključeva u simetričnim kriptografskim sistemima

Prilikom korišćenja simetričnih ključeva javljaju se tri osnovna problema: Prvi, ukoliko n osoba želi da ostvari neku komunikaciju, potrebno je da raspolaže sa $n(n-1)/2$ simetričnih ključeva. Treba imati u vidu to da svaki od n osoba može komunicirati sa $(n-1)$ drugih, što iziskuje upotrebu $n(n-1)$ ključeva. Zbog toga što se simetrično kriptovanje odnosi na komunikaciju između dve osobe, potrebno je $n(n-1)/2$ ključeva. Problem nastaje ukoliko je n veliki broj. Na primer, ako n iznosi 1 milion osoba za sigurnu komunikaciju je neophodno pola milijarde ključeva. Drugo, u grupi od n osoba, svaka osoba mora posedovati a i zapamtiti $(n-1)$ ključeva, po jedan za po svaku osobu u grupi. To znači da ukoliko 1 milion ljudi želi da komunicira sa ostalima, mora da zapamti oko 1 milion ključeva. Treće, ključevi se moraju prenositi preko sigurne komunikacione linije, nikako preko telefona ili interneta.

Razmatrajući navedene probleme, kao najbolje, nameće se rešenje da se ključevi kreiraju dinamički za svaku sesiju i nakon nje uništavaju. U tom slučaju se ne zahteva pamćenje ključeva od strane osoba koje komuniciraju.

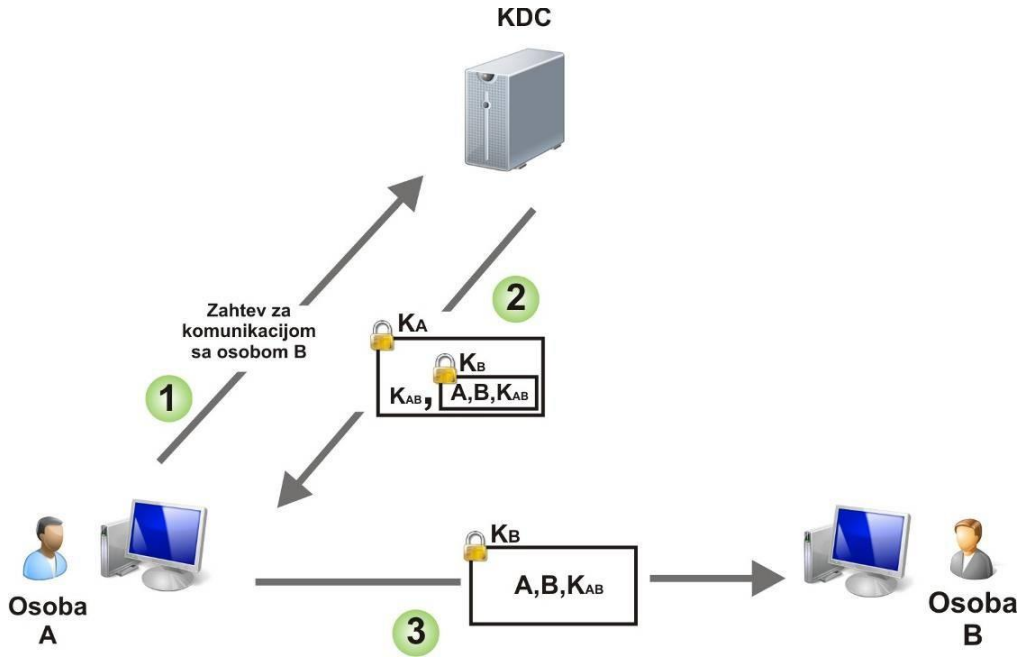
Kao najjednostavnija metoda za distribuciju ključa putem računarske mreže predstavlja ranije opisana *Diffie-Hellman* ili D-H metoda (videti poglavlje 2.7.2). Ova metoda se najčešće koristi za sesije sa ključevima za jednokratnu upotrebu (eng. *one-time key*). Dve strane u komunikaciji koriste ove ključeve za razmenu podataka bez pamćenja istih. Prilikom razmene ključeva ova metoda ne pruža mogućnost autentifikacije strana koje su u komunikaciji, što predstavlja veliki nedostatak po pitanju sigurnosti.

5.2.1.1 Centar za distribuciju ključeva (KDC)

Ranije opisana D-H metoda se zasniva na slanju nezaštićenih ključeva putem komunikacione linije, čime je ugrožena njihova bezbednost. Sa druge strane, da bi se uspostavila komunikacija između dva entiteta obe strane moraju posedovati isti par ključeva. U cilju sigurne razmene ključeva među entitetima

koristi se neki treći entitet kome veruju obe strane u komunikaciji. Ovaj entitet se naziva centar za distribuciju ključeva ili KDC (eng. *Key Distribution Center*).

Osobe A i B su klijenti KDC-a (Slika 5.2). Između osobe A i KDC-a se koristi jedan simetrični tajni ključ do koga osoba A dolazi na siguran način, na primer ličnim odlaskom u centar. K_A je tajni ključ osobe A, dok je K_B tajni ključ osobe B. Ove tajne ključeve mora posedovati svako ko želi komunicirati pomoću KDC-a [10].



Slika 5.2
Proces razmene ključeva pomoću KDC-a

Ključ K_{AB} je zajednički ključ za sesiju između osobe A i B. Postupak razmene ključa se obično obavlja u nekoliko koraka:

- Korak1. Osoba A šalje poruku KDC-u sa zahtevom za komunikacijom sa osobom B, a u cilju kreiranja zajedničkog ključa sesije K_{AB}. Poruka sadrži podatke o identitetu osobe A i identitetu osobe B. Treba napomenuti da ova poruka nije kriptovana.
- Korak2. KDC prima poruku i kreira takozvani tiket namenjen osobi B. Pošto KDC zna tajni ključ osobe B, koristi ga kako bi kriptovao pomenuti tiket. Tiket sadrži identitete osoba A i B i zajednički ključ K_{AB}. Ovako formiran tiket se upakuje zajedno sa zajedničkim ključem za jednu sesiju K_{AB}, kriptuje sa ključem K_A i na kraju pošalje osobi A, kao poruku. Nakon prijema poruke osoba A je dekriptuje, a zatim otpakuje zajednički ključ, K_{AB}. Pošto nema ključ K_B ne može da otpakuje tiket namenjen osobi B.
- Korak3. U okviru ovog koraka osoba A prosleđuje tiket namenjen osobi B. Osoba B pošto primi tiket, otpakuje ga i pamti ključ K_{AB} koji joj je neophodan radi razmene poruka sa osobom A [10].

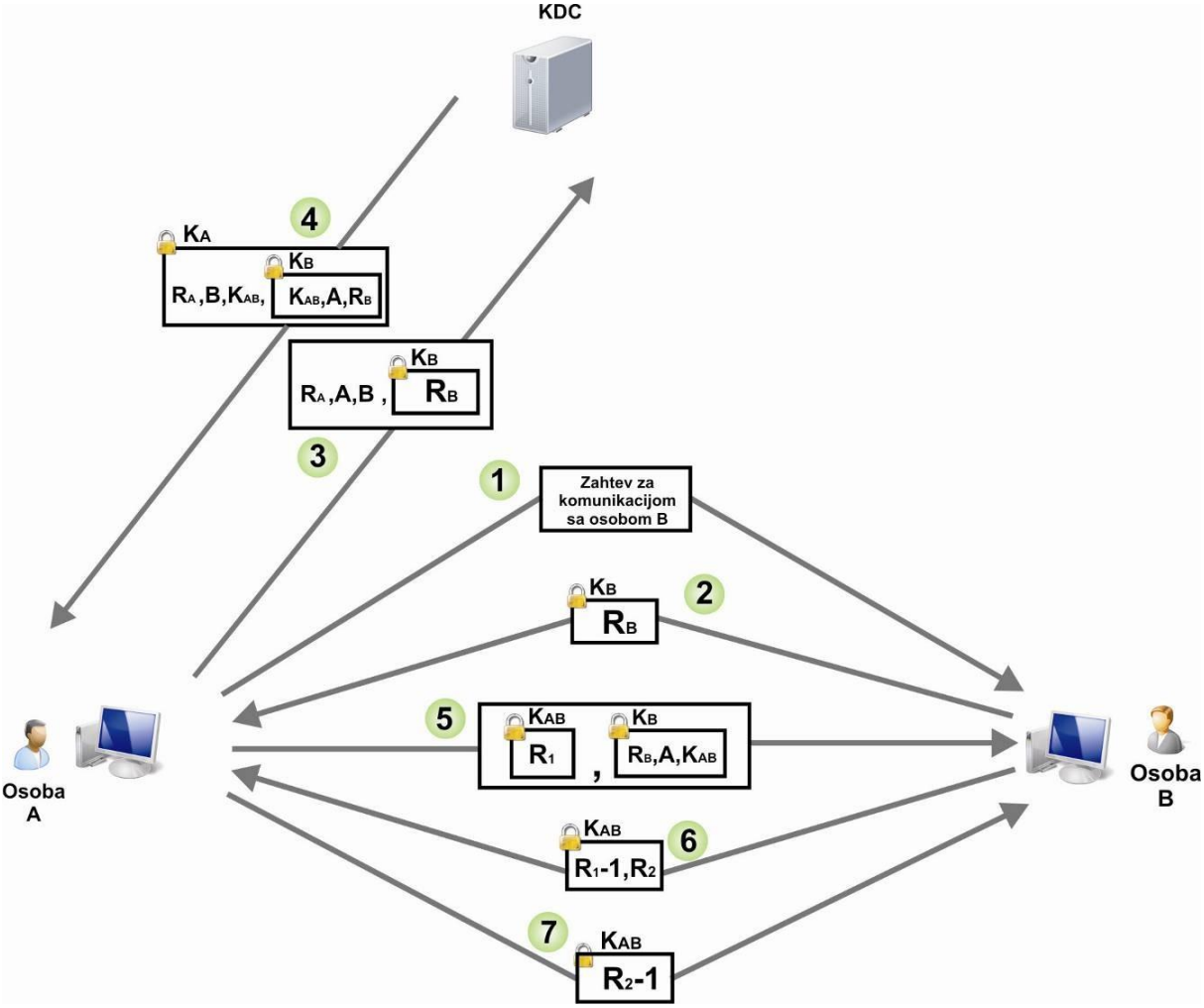
Nakon trećeg koraka, Osobe A i B mogu razmeniti podatke korišćenjem ključa K_{AB} kao ključa za *one-time* sesiju. Pored navedenog standardnog načina razmene ključeva sa KDC-om postoji nekoliko drugih varijanti odnosno protokola za razmenu ključeva.

5.2.1.2 *Needham-Schroeder* protokol

Kako bi postigao što sigurniju razmenu ključeva između entiteta ovaj protokol čini više naizmeničnih akcija gde se pod akcijom smatra razmena poruka tipa „izazov-odgovor između dva entiteta. Tokom akcija razmene ključeva *Needham-Schroeder* protokol koristi KDC i predstavlja odlično

rešenje po pitanju sigurnosti. Važi za fundamentalni protokol na osnovu koga su nastali mnogi drugi protokoli.

Prema poslednjoj verziji ovog protokola tokom akcija se koriste četiri slučajna broja: R_A, R_B, R_1 i R_2. Ovi brojevi se generišu samo za jednu sesiju. Na Slici 5.3 je prikazana razmena ključa pomoću *Needham-Schroeder* protokola [10].



Slika 5.3
Proces razmene ključa pomoću Needham-Schroeder protokola

Protokol se sastoji od sedam koraka:

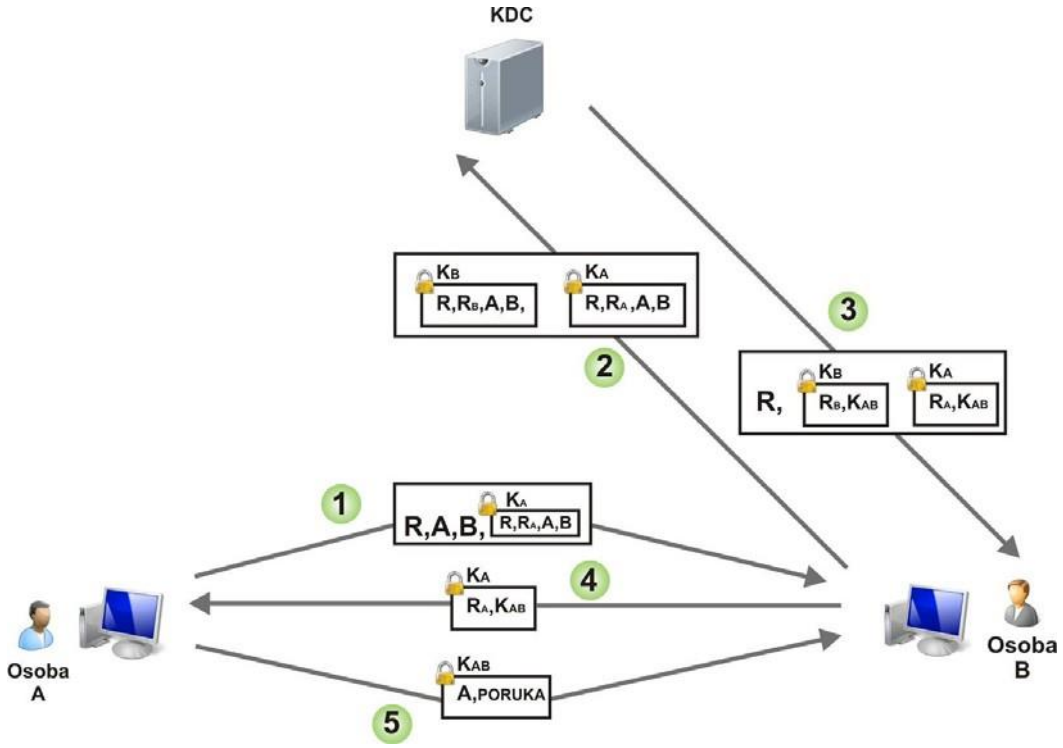
- Korak 1. Osoba A šalje svoj identitet osobi B, i na taj način joj saopštava da želi komunikaciju.
- Korak 2. Osoba B pomoću svog tajnog ključa K_B kriptuje predhodno generisani broj R_B koji namenjuje KDC-u, ali ga šalje osobi A. Kada primi poruku, osoba A će je proslediti do KDC-a u cilju dokazivanja da je osoba koja komunicira sa osobom B jedina koja će komunicirati sa KDC, odnosno da nije napadač.
- Korak 3. Poruka koju osoba A šalje ka KDC sadrži slučajan broj R_A, identitet osobe A, identitet osobe B i kriptovani broj R_B. Treba napomenuti da ova poruka nije kriptovana, i da predstavlja zahtev osobe A namenjen KDC-u u cilju obezbeđivanja tajnog ključa sesije K_{AB} za obavljanje sigurne razmene podataka između osoba A i B.
- Korak 4. KDC vraća osobi A poruku sa sledećim sadržajem: broj R_A, identitet osobe B, ključ sesije K_{AB} i kriptovani tiket koji je namenjen osobi B. Ovaj tiket je kriptovan tajnim ključem osobe B, a sadrži ključ sesije K_{AB}, identitet osobe A i broj koji je predhodno generisala osoba B,

R_B. Na taj način je osoba A dobila odziv od KDC-a koji sadrži sve neophodne podatke za bezbednu komunikaciju sa osobom B.

- Korak 5. Osoba A prosleđuje osobi B tiket dobijen od KDC i novi broj R_1 u cilju prozivanja i verifikacije osobe B.
- Korak 6. Osoba B odgovara na poziv osobe A slanjem kriptovanog broja $R_1 - 1$ i pozivom osobe A brojem R_2. Ako osoba A nakon dekriptovanja primljene poruke dobije očekivani broj R_1-1, to znači da je identitet osobe B verifikovan, jer je onaj ko je poslao ovu poruku bio u stanju da primenom K_B dođe do K_{AB}, a to može samo osoba B.
- Korak 7. U okviru poslednjeg koraka osoba A ponovo odgovara na poziv slanjem kriptovanog (pomoću ključa K_{AB}) broja $R_2 - 1$. Ovom porukom osoba A obaveštava osobu B da je primila verifikaciju i istovremeno verifikuje sebe osobi B [10].

5.2.1.3 *Otway-Rees* protokol

Otway-Rees protokol je takođe baziran na metodi razmene poruka pomoću KDC-a. Protokol čini nekoliko koraka koji su prikazani na Slici 5.4.



Slika 5.4
Razmena poruka pomoću Otway-Rees protokola

Prednost ovog protokola u odnosu na predhodnu metodu je u manjem broju poruka koje se razmenjuju, odnosno, pet poruka u odnosu na ranijih sedam. Tačnije, protokol čini sledećih pet koraka:

- Korak 1. Na početku, osoba A šalje poruku osobi B, koja sadrži slučajni broj R, identitete osoba A i B i kriptovani tiket namenjen KDC-u koji uključuje broj R_A, kopiju zajedničkog broja R i identitete osoba A i B.
- Korak 2. Osoba B kreira tiket istog tipa namenjen KDC-u, a čine ga broj R_B, broj R, identiteti osoba A i B. Ovaj tiket kriptuje pomoću ključa K_B i zajedno sa tiketom koji je primila od osobe A šalje ka KDC-u. Ova poruka se šalje u cilju dokazivanja identiteta KDC-u od strane osoba A i B.

- Korak 3. KDC kreira poruku koja sadrži zajednički broj R, tikete namenjene osobama A i B i tako kreiranu poruku šalje osobi B. Tiketi sadrže odgovarajuće brojeve R_A ili R_B i zajednički ključ sesije K_{AB}.
- Korak 4. Osoba B prosleđuje tiket namenjen osobi A.
- Korak 5. Osoba A šalje kriptovanu poruku sa ključem K_{AB}, koji je dobila od osobe B [10].

5.2.2 Distribucija ključeva u asimetričnim kriptografskim sistemima

U okviru asimetričnih kriptografskih sistema ne postoji potreba za razmenom tajnih ključeva. Ukoliko osoba A želi da pošalje poruku osobi B, potrebno je da zna samo javni ključ osobe B. Do ovog ključa se dolazi na jednostavan način i on je dostupan za bilo koga. Takođe, ako osoba B želi da pošalje poruku osobi A, mora da zna samo javni ključ osobe kojoj želi poslati poruku. Može se naglasiti prednost asimetričnih sistema za kriptovanje da svako može imati svačiji javni ključ.

Predhodna konstatacija nameće pitanje da li je kod asimetričnih sistema uopšte potrebna bezbedna razmena ključeva. Svako ko želi da primi neku poruku od nekoga, treba da objavi svoj javni ključ osobi koja šalje poruku. Upravo tu se javlja problem kako objaviti javni ključ, a da ga neka osoba C ili napadač ne može zloupotrebiti. Ako osoba B objavi javni ključ da bi ga osoba A iskoristila za slanje poruke osobi B, neka osoba C ga može uhvatiti. Nakon hvatanja, osoba C može poslati svoj javni ključ umesto javnog ključa osobe B. Neznajući od koga dolazi ključ osoba A prihvata podmetnuti ključ i kriptuje poruku pomoću njega a zatim je šalje osobi B. Sada opet nastupa osoba C koja hvata kriptovanu poruku umesto osobe B i lako je dekriptuje pomoću svog tajnog ključa. Iz navedenog primera se vidi da je i kod asimetričnih sistema za kriptografiju jako bitno da entiteti koji komuniciraju ili razmenjuju poruke potvrde identitete jedan drugom [10].

5.2.2.1 Potvrda identiteta

Za potvrdu identiteta osoba koje vrše razmenu podataka na Internetu se dodeljuju tkz. digitalni sertifikati. Digitalni sertifikati se često koriste prilikom elektronskog poslovanja, a mogu ih koristiti fizička i pravna lica, državne uprave, javne službe, preduzeća, organizacije i drugi. Digitalne sertifikate (ili elektronske sertifikate) izdaju organizacije pod nazivom sertifikaciona tela (eng. *Certification Authority*, CA). Za izdavanje digitalnih sertifikata, sertifikaciona tela koriste infrastrukturu javnih ključeva (eng. *Public Key Infrastructure*, *PKI*). PKI je složena infrastruktura koja se sastoji od kriptografskih protokola, standarda, procedura, servisa i aplikacija koje služe za izdavanje digitalnih sertifikata. Najzastupljeniji standard za izdavanje digitalnih sertifikata je X.509.

Digitalni sertifikat sadrži podatke o identitetu korisnika (ime i prezime, E-mail, adresa), korisnikov javni ključ i podatke o izdavaocu sertifikata odnosno sertifikacionom telu. Sertifikaciono telo koje izdaje sertifikat garantuje autentičnost podataka koji se nalaze u sertifikatu. Digitalni sertifikat je nemoguće falsifikovati jer je potpisan tajnim ključem sertifikacionog tela [30].

Postupak izdavanja sertifikata u praksi se može pojasniti na sledećem primeru: Osoba A podnosi zahtev za izdavanje sertifikata nekoj od CA organizacija. CA proverava njen identitet na osnovu uvida u lična dokumenta koje je osoba A prikazala pri podnošenju zahteva. Ako je sve u redu, osoba A prosleđuje svoj javni ključ CA organizaciji za koji ona kreira digitalni potpis i nakon toga izdaje sertifikat kojim se potvrđuje da taj javni ključ zaista pripada osobi A. Ako osoba A kasnije želi da komunicira sa nekim, šalje mu digitalni sertifikat i svoj javni ključ. S obzirom da primalac veruje CA organizaciji lako može proveriti validnost sertifikata odnosno identitet osobe A. Postoji više vrsta digitalnih sertifikata:

- Kvalifikovani sertifikat predstavlja osnovni vid digitalnog sertifikata u vidu digitalnog potpisa koji je istovetan običnom potpisu.
- Web sertifikat se koristi u okviru aplikacija za autentifikaciju, kriptovanje/dekriptovanje, potpis i verifikaciju datoteka, elektronske pošte i raznih transakcija.

- Sertifikat namenjen za Web servere, a koriste se za konfigurisanje SSL (eng. *Secure Sockets Layer*) i/ili TLS (eng. *Transport Layer Security*) protokola na Web serverima (*Microsoft* IIS, *Apache, Sun-Netscape iPlanet, Red Hat Stronghold, Sun ONE, IBM HTTP Server*,...). Namena SSL i TLS protokola je uspostavljanje zaštićenog komunikacionog kanala između Web servera i Web klijenata. Ovo praktično znači da ako se želi da se na nekoj Web prodavnici kupcima omogući plaćanje kreditnim karticama ili pružanje poverljivih informacija, neophodno je da server na kome se nalazi ta Web prezentacija radi kao Secure Web Server. Uslov koji mora da ispuni pomenuti Web server je da poseduje digitalni sertifikat od nekog CA [29].

Za čuvanje digitalnih sertifikata se koriste sledeći mediji: hard disk računara, CD, USB flash memorija, PKI smart kartica ili PKI USB smart token.

5.2.2.2 X.509 protokol

Iako sertifikaciono telo rešava problem zloupotrebe javnog ključa javljaja se jedan nedostatak. Ovaj nedostatak se odnosi na različitost formata koje sertifikati mogu imati. To znači da jedan sertifikat može imati javni ključ u jednom formatu, a drugi u drugom formatu. Javni ključ može biti u okviru prve linije nekog sertifikata, a kod nekog drugog može biti i u okviru treće linije.

Radi eliminisanja predhodno opisanog nedostatka, ustanovljen je protokol koji se naziva X.509. Nakon nekih izmena ovaj protokol je široko prihvaćen na Internetu. X.509 je postupak pomoću koga se opisuje sertifikat na strukturalnom nivou. X.509 sertifikat na strukturalnom nivou sadrži [31]:

- Verziju,
- Serijski broj,
- Algoritam za identifikaciju,
- Izdavalac,
- Validnost,
- Naslov,
- Informacije o javnom ključu,
- Jedinstveni identifikator izdavača sertifikata (opciono),
- Naslov jedinstvenog identifikatora (opciono),
- Ekstenzija (opciono),
- Algoritam potpisa sertifikata,
- Potpis sertifikata.

Na Slici 5.5 je prikazan sadržaj jednog X.509 sertifikata.

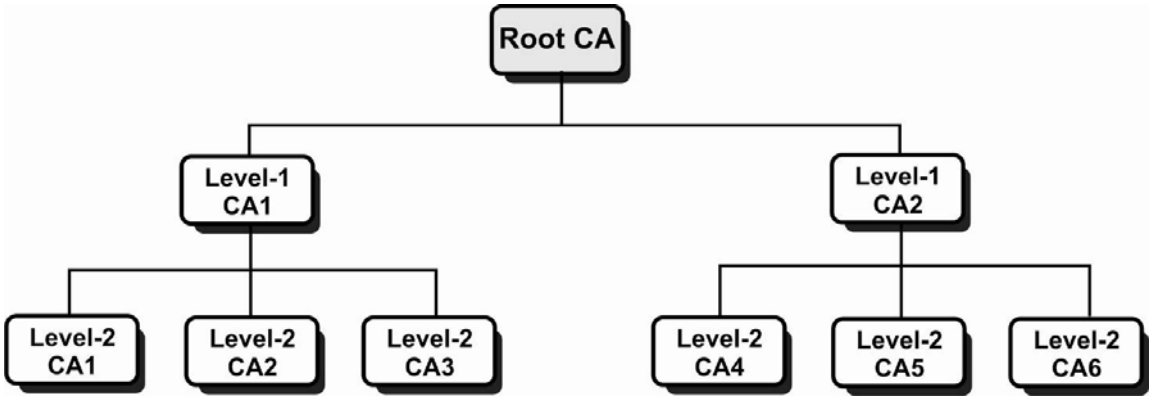
```
Certificate:
   Data:
       Version: 1 (0x0)
       Serial Number: 7829 (0x1e95)
       Signature Algorithm: md5WithRSAEncryption
       Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
               OU=Certification Services Division,
               CN=Thawte Server CA/emailAddress=server-certs@thawte.com
       Validity
           Not Before: Jul  9 16:04:02 1998 GMT
           Not After : Jul  9 16:04:02 1999 GMT
       Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,
                OU=FreeSoft, CN=www.freesoft.org/emailAddress=baccala@freesoft.org
       Subject Public Key Info:
           Public Key Algorithm: rsaEncryption
           RSA Public Key: (1024 bit)
               Modulus (1024 bit):
                   00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:
                   33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:
                   66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:
                   70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:
                   16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:
                   c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:
                   8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:
                   d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:
                   e8:35:1c:9e:27:52:7e:41:8f
               Exponent: 65537 (0x10001)
   Signature Algorithm: md5WithRSAEncryption
       93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:
       92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:
       ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:
       d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:
       0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:
       5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:
       8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:
       68:9f
```

Slika 5.5
Izgled X.509 sertifikata

5.2.2.3 Infrastruktura javnog ključa (*Public Key Infrastrukture*, PKI)

Upotrebom javnih ključeva se javlja problem sličan DNS-u (eng. *Domain Name System*). Problem se sastoji u tome da jedan DNS server ne može da opsluži sve zahteve na internetu. Iz tog razloga se uvodi više servera koji zajedno opslužuju zahteve, uz to došlo se do rešenja da je najbolje da se između servera uspostavi hijerarhijski odnos. To znači da ukoliko neka osoba A ima potrebu za IP adresom osobe B, osoba A šalje zahtev lokalnom serveru koji može a ne mora imati traženu IP adresu. Ukoliko ne poseduje IP adresu osobe B server prosleđuje zahtev ka drugom serveru koji se nalazi na višem nivou, i tako nadalje sve dok se ne pronađe zahtevana adresa [10].

Slično DNS-u, problem pribavljanja javnih ključeva se rešava upotrebom hijerarhijske strukture koja se naziva PKI (eng. *Public Key Infrastructure*). Primer jedne ovakve strukture je dat na Slici 5.6.



Slika 5.6
Hijerarhijska strruktura PKI-a

U okviru prvog nivoa strukture, nalazi se Root CA koji može da sertifikuje CA-ove u okviru sledećeg nižeg nivoa kao što su: Level-1 CA1 i Level-1 CA2. Ovi CA-ovi operišu nad velikim

geografskim i logičkim oblastima. Na nivou Level-2 se nalaze CA-ovi koji operišu nad manjim oblastima. CA-ove u okviru ovog nivoa mogu da sertifikuju CA-ovi koji se nalaze na nivou više.

Ovakvom hijerarhijom upravlja Root CA, tako da svi korisnici imaju poverenje u njega. Za razliku od Root CA ostalim CA-ima mogu, a i ne moraju verovati svi korisnici. Na primer, ako osoba A traži sertifikat osobe B može ga pribaviti od bilo kog CA, ali mu ne mora verovati. Ukoliko mu ne veruje osoba A može zatražiti sertifikat od CA koji se nalazi u nekom višem nivou. Traganje za validnim sertifikatom se može nastaviti u pravcu Root-a.

5.2.2.4 Kerberos

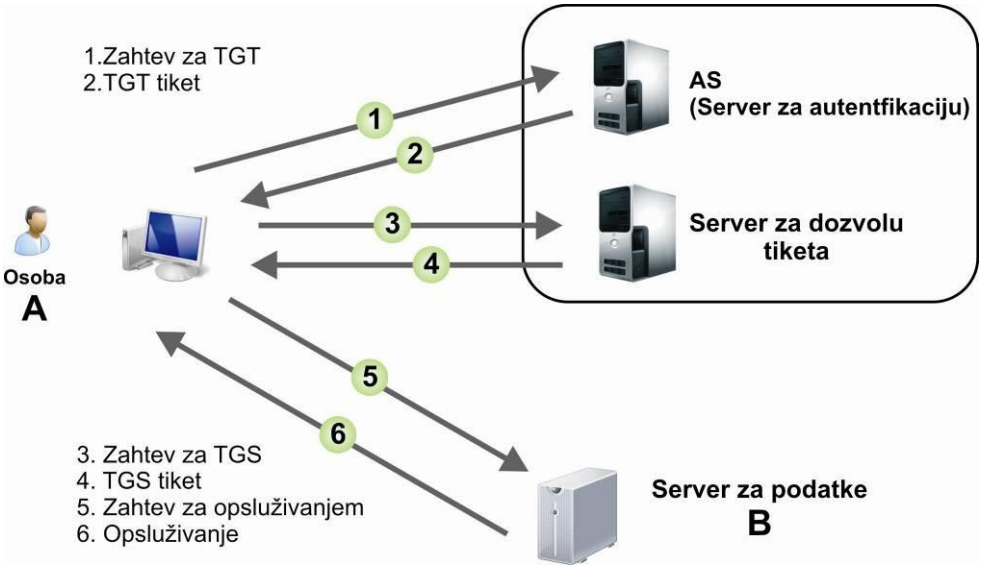
Kerberos je istovremeno autentifikacioni protokol i KDC. U poslednje vreme jako popularan, i koristi se u velikom broju aplikacija i operativnih sistema.

Kerberos se može opisati kao siguran protokol za autentifikaciju koji koristi prijavljivanje tipa „prijavi se samo jednom“ (eng. *Single Sign On*) što omogućava veliku efikasnost u radu. Korisnicima je omogućeno da se samo jednom prijave na sistem i da nakon toga, u skladu sa svojim ovlašćenjima, imaju pristup resursima u sistemu ili mreži [32].

Postoji više implementacija Kerberos protokola, koje su razvijane za različite namene:

- Osnovnu verziju kerberosa je formirao MIT (*Massachusetts Institute of Technology*) ranih devedesetih godina. Do danas se pojavilo više verzija ovog protokola, a najpopularnije su MIT verzija 4 i verzija 5.
- Heimdal kerberos je Švedska kopija MIT Kerberos-a sa kojim je u potpunosti kompatibilan.
- Aktivni direktorijum je razvio Microsoft i sam po sebi nije Kerberos ali sadrži neke njegove elemente u kombinaciji sa drugim servisima, kao što je LDAP (eng. *Lightweight Directory Access Protocol*). Treba napomenuti da Aktivni direktorijum nije kompatibilan sa MIT i Heimdal Kerberosom.
- *Trust Broker* predstavlja komercijalnu implementaciju Kerberos protokola. Ovu implementaciju podržava većina operativnih sistema. Uz to, *Trust Broker* je potpuno kompatibilna sa svim navedenim implementacijama Kerberos protokola, što je čini jako fleksibilnom.
- *Shishi* je GNU implementacija MIT Kerberosa verzije 5.

U cilju razumevanja redosleda operacija i načina rada Kerberos protokola, najpre je potrebno objasniti nekoliko pojmova vezanih za njega. Komunikacije između entiteta u okviru Kerberos protokola se baziraju na razmeni tiketa. Tiket predstavlja vrstu kriptovanih podataka koji se prenose putem mreže, i dostavljaju klijentu koji ih čuva i kasnije koristi kao propusnicu za uspostavljanje komunikacije sa odgovarajućim serverom. Prilikom kriptovanja poruka/tiketa Kerberos protokol koristi simetrični DES algoritam ili njegove varijante kao što je 3DES a Kerberos verzija 5 koristi isključivo AES algoritam za kriptovanje. Okruženje Kerberosa čine tri servera i to: server za autentifikaciju (eng. *Authentication server,* AS), server za izdavanje tiketa (eng. *Ticket-Granting Server*) i server za podatke koji se štiti opšteg tipa. Na Slici 1 je prikazan odnos između pomenutih servera [32],.

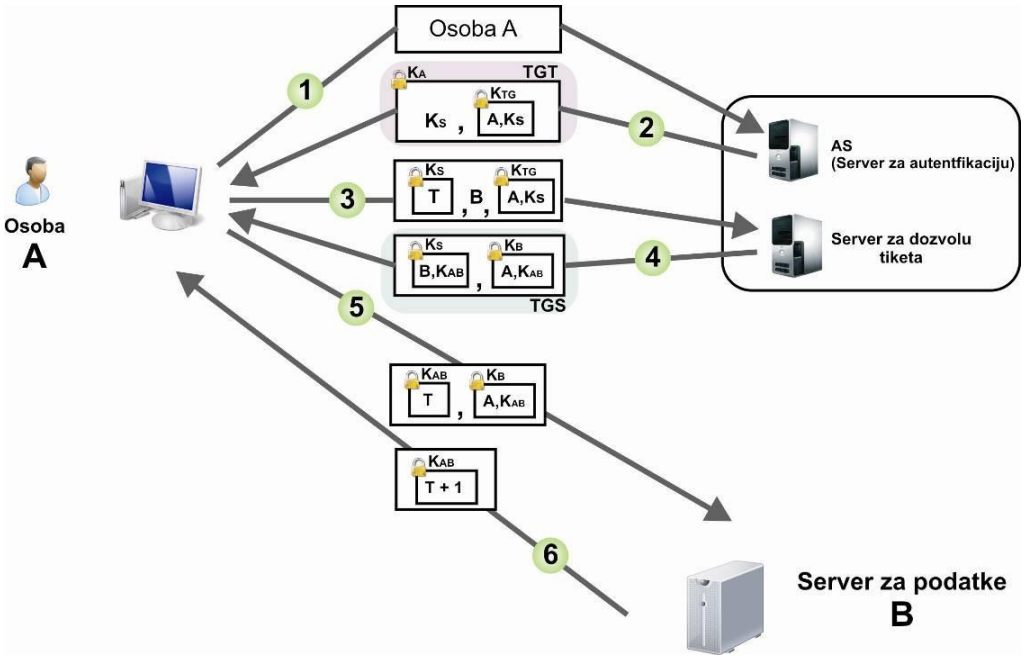


Slika 5.7
Serveri koji se koriste u okviru Kerberos protokola

Server za autentifikaciju (AS) koji se koristi u Kerberos protokolu ima ulogu KDC-a. Svaki entitet, ukoliko želi pristup serveru za podatke koji se štiti, mora se registrovati kod AS. AS poseduje bazu podataka u kojoj čuva identitete i odgovarajuće šifre za pristup serveru koji se štiti. Pošto dobije zahtev za pristupom od strane nekog entiteta (Osobe A), AS vrši utvrđivanje identiteta entiteta (osobe A), izdaje ključ koji se može koristiti za jednu sesiju između osobe A i servera za izdavanje tiketa i šalje tiket TGT (eng. *Ticket GrantingTicket*) ka osobi A, kao odgovor.

Server za izdavanje tiketa izdaje tiket TGS (eng. *Ticket Granting Service*) namenjen osobi A i serveru B a obezbeđuje ključ sesije K_{AB} koji se koristi u komunikaciji između osobe A i servera B. Kerberos razdvaja servise provere identiteta i izdavanja tiketa, što predstavlja pravo rešenje u slučaju kada neka osoba želi koristiti usluge više različitih servera. Dovoljno je da ta osoba izvrši verifikaciju svog identiteta kod AS samo jednom, onda se više puta obrati serveru za dozvolu tiketa u cilju dobijanja tiketa za pristup različitim serverima.

Svaki zaštićeni server pruža izvesne usluge entitetu, pri čemu se razmena podataka između entiteta i servera isključivo obavlja pomoću klijent-server aplikacija kao što je FTP (eng. *File Transfer Protocol*). Princip rada procesa pristupa klijenta serveru, pomoću Kerberos protokola, je prikazan na Slici 5.8.



Slika 5.8
Primer Kerberos protokola

Postupak pristupa klijenta (Osoba A) serveru B, u cilju bezbednog korišćenje njegovih usluga, se obavlja u okviru sledećih šest koraka:

- Korak 1. Osoba A šalje zahtev ka AS. Zahtev sadrži identitet osobe A u nekriptovanom obliku.
- Korak 2. Kada primi zahtev, AS šalje osobi A poruku, u vidu tiketa TGT, kriptovanu pomoću K_A odnosno simetričnog ključa osobe A. Poslati TGT se sastoji od dva podatka: Ključ sesije K_S koji će osoba A koristiti za komunikaciju sa serverom za dozvolu tiketa i tiket namenjen serveru za dozvolu tiketa koji je kriptovan sa simetričnim ključem K_{TG}. Osoba A ne mora da čuva svoj ključ K_A u originalnom obliku, ali ga po potrebi može formirati. Kada primi poruku od AS, osoba A unosi odgovarajuću šifru koja (ukoliko je korektna) uz pomoć odgovarajućeg algoritma kreira ključ K_A. Neposredno nakon uspešnog kreiranja ključa, uneta šifra se uklanja sa računara, radi sprečavanja eventualnih zloupotreba. Pomoću ključa K_A se uspešno dekriptuje poruka dobijena od strane AS-a čime osoba A dobija K_S i tiket namenjen serveru za dozvolu tiketa.
- Korak 3. Osoba A šalje ka serveru za dozvolu tiketa tri podatka: tiket primljen od AS-a, ime servera čije usluge želi koristiti (server B) i kriptovani (pomoću ključa K_S) vremenski zapis T, koji sadrži informacije o datumu i vremenu trajanja sesije. Vremenski zapis se koristi kao preventivna mera od eventualnog kasnijeg pristupa neke osobe C. Ovako upakovani podaci, namenjeni serveru za dozvolu tiketa, predstavljaju zahtev za izdavanje TGS tiketa.
- Korak 4. Pošto je primio zahtev, server za dozvolu tiketa šalje odgovor u vidu TGS tiketa ka osobi A. TGS tiket sadrži dva tiketa i to: jedan tiket namenjen upravo osobi A, a drugi namenjen serveru B. Tiket namenjen osobi A sadrži ključ sesije K_{AB} i identitet servera B, a kriptovan je pomoću ključa K_S. Tiket namenjen serveru B, takođe sadrži ključ sesije K_{AB}, identitet osobe A, a kriptovan je pomoću tajnog ključa servera B, K_B. Ukoliko neka osoba C dođe u posed ključa sesije K_{AB} može nesmetano obaviti razmenu podataka sa serverom B, međutim pošto ne poseduje tajne ključeve K_S ili K_B nije u mogućnosti da dekriptuje neki od tiketa koji sadrže ključ sesije.
- Korak 5. Osoba A šalje tiket namenjen serveru B i vremenski marker koji je kriptovan ključem sesije K_{AB}.
- Korak 6. Na kraju server vraća poruku osobi A, kao potvrdu komunikacije. Poruka se sastoji od primljenog vremenskog zapisa T uvećanog za 1 koji je kriptovan pomoću ključa sesije K_{AB} [10].

Ukoliko osoba A želi da koristi usluge nekog drugog servera, potrebno je da ponovi operacije samo u okviru poslednja četiri koraka. Prva dva koraka se odnose na proveru identiteta osobe A i nisu neophodna. Osoba A može zahtevati od servera za dozvolu tiketa da mu izda tikete za pristup različitim serverima.

Kerberos protokol predstavlja pouzdanu metodu za autentifikaciju u okviru javnih mreža kao što je na primer Internet. Na pouzdanost i opravdanost korišćenja utiču sledeće konstatacije:

- Kerberos se koristi već duže vreme, što ukazuje na činjenicu da je opstao i da je dosta testiran.
- Projektovan je da odgovori na strogo definisane zahteve po pitanju sigurne autentifikacije u otvorenim okruženjima koja koriste nezaštićene komunikacione linije.
- Sastoji se iz seta jednostavnih arhitekturalnih i funkcionalnih pojmova, koji omogućavaju jednostavnu integraciju sa drugim sistemima što ga čini jako fleksibilnim.
- Većina operativnih sistema i softverskih aplikacija ga već ima implementiranog u svom kodu, odnosno predstavlja deo IT infrastrukture današnjice [33].

5.3 Zamena ključa

Svi kriptografski sistemi moraju imati mogućnost zamene ključa. Može postojati više razloga za promenom, a te promene mogu biti planirane redovnim obnavljanjem ili usled sumnje u ranije korišćeni ključ. Ukoliko se radi o sumnji zamena bi trebala da bude izvedena što pre. Redovnim obnavljanjem ključa smanjuje se mogućnost zloupotrebe.

Mera uspešnosti napada direktno zavisi od vremena i truda koje napadač mora da uloži u cilju otkrivanja ključa. Na smanjenje mere uspešnosti napada naročito utiču sistemi koji za svaku transakciju koriste različite ključeve (eng. *one time key*). Kod ovakvih sistema napadač ulaže velike napore za otkrivanje ključa. Kad pređe na novu transakciju, sistem vrši zamenu ključa., čime odvraća napadača u daljim pokušajima, produžavajući vreme potrebno za otkrivanje novog ključa za novu transakciju. Kada poseduje jedan ključ napadač može dekriptovati samo jednu transakciju. Jedna transakcija ne znači cela sesija.

Ne postoji jasno pravilo o frekvenciji zamene ključeva. Jedno je sigurno, kada se neki ključ upotrebljava duži vremenski period preporučljiva je zamena. Za upravljanje rokom važnosti ključa odgovorno je sertifikaciono telo.

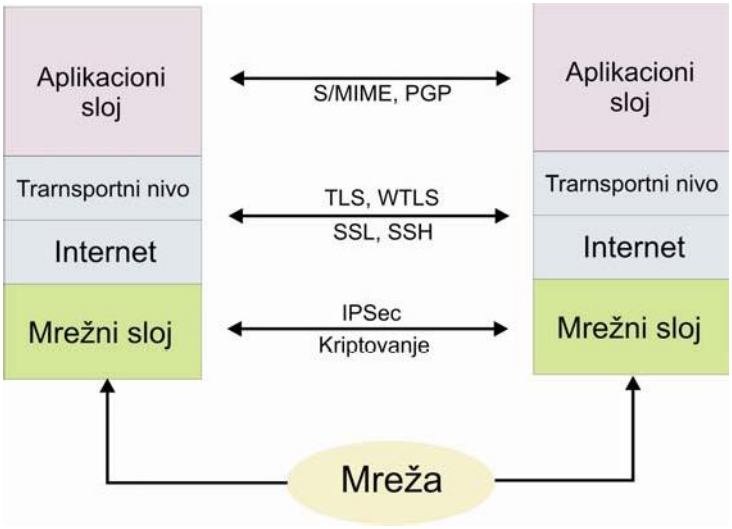
5.4 Uništenje ključa

Nakon formiranja ključevi imaju svoj vek trajanja. Često se upotrebljavaju ključevi namenjeni samo za jednu sesiju. Na kraju sesije ključ se uništava i ubuduće se ne može koristiti. Neki ključevi se mogu sertifikovati za određeni vremenski period. Javni ključevi se u proseku sertifikuju za period od jedne do dve godine.

Prilikom prenosa, ključ se može izgubiti ili oštetiti. Kada se to desi, vlasnik ključa mora obavestiti ostale korisnike da ključ više nije validan i da se ubuduće ne može koristiti. Ako se par ključeva ošteti ili kompromituje ne moraju se obaveštavati svi korisnici ključeva, naročito ako se ključevi čuvaju na serveru. Periodičnim pregledom podataka na serveru se može videti koji su ključevi van upotrebe. Kada se utvrdi koji ključevi nisu u upotrebi, pristupa se uništenju ali se informacija o uništenju ključa čuva na serveru sve dok ne istekne originalni sertifikat. Upravo ove karakteristike čine asimetrične sisteme pogodnim za korišćenje na Internetu.

6 Sigurnosni servisi u TCP/IP modelu

Savremene računarske mreže se uglavnom zasnivaju na TCP/IP protokolima. Prenos podataka pomoću TCP/IP protokola se vrši u obliku paketa te je relativno jednostavno ubacivanje poruka uz nemogućnost kasnijeg određivanja njihovog porekla i sadržaja. Stoga, ovi protokoli sa aspekta bezbednosti poseduju brojne slabosti, i kao takvi su nezamislivi bez upotrebe sigurnosnih protokola. Sigurnosni protokoli se baziraju na kriptotehnologijama i implementiraju se u arhitekturu TCP/IP modela u cilju unapređenja modela po pitanju bezbednosti. Na Slici 6.1 je prikazan TCP/IP model sa implementiranim sigurnosnim protokolima.



Slika 6.1
Sigurnosni protokoli u okviru TCP/IP modela

U okviru modela se može primetiti upotreba različitih sigurnosnih protokola za zaštitu podataka u zavisnosti o kom se TCP/IP sloju radi. Zaštita na mrežnom sloju obezbeđuje se između mrežnih čvorova pri čemu se vrši zaštita IP paketa. Zaštita IP paketa se ostvaruje primenom kriptografskih algoritama (simetrični i asimetrični algoritmi za kriptovanje, heš funkcije, digitalni potpis i drugi). Najpoznatiji sigurnosni protokol na ovom nivou je IPSec protokol.

Na transportnom sloju se ostvaruje zaštita tajnosti podataka primenom simetričnih kriptografskih algoritama i proverama identiteta (digitalni potpis, heš funkcije, digitalni sertifikati) strana koje komuniciraju . Najpoznatiji sigurnosni protokoli koji se koriste na ovom sloju su: SSL, SSH, TLS i WTLS.

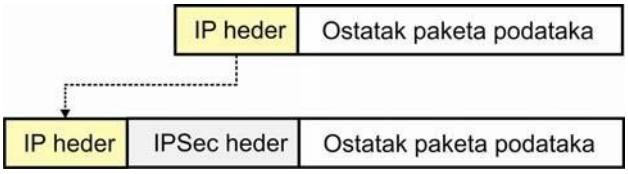
Na aplikacionom sloju primenjuju se tehnologije poput digitalnog potpisa i zaštita podataka primenom simetričnih kriptografskih algoritama. Najpoznatiji sigurnosni protokoli koji se primenjuju na ovom sloju su: S/MIME, Kerberos, PGP, i drugi.

6.1 IPSec

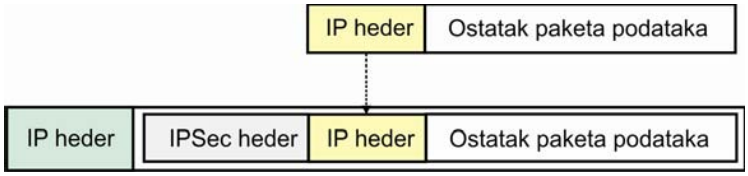
IPsec (eng. *IP security*) predstavlja skup sigurnosnih protokola namenjenih zaštiti paketa koji se prenose putem Interneta. Ovi protokoli funkcionišu na transportnom sloju TCP/IP modela i implementirani su u okviru IPv6 sistema, a mogu se opcionalno uključiti i unutar IPv4 sistema. Zbog činjenice da funkcioniše na transportnom sloju, IPsec pruža jednostavnu i efikasnu zaštitu i za TCP i za UDP protokole komunikacije preko računarske mreže. IPsec može da radi u dva načina rada:

- transportni način rada i
- tunelski način rada.

U okviru transportnog načina rada paketi se šalju između dva krajnja entiteta u mreži, pri čemu entitet koji prima paket izvršava sigurnosne provere pre isporučivanja paketa višim slojevima. U okviru ovog načina rada IPSec heder se umeće odmah nakon IP hedera, a pre ostatka paketa kao što je prikazano na Slici 6.2 (a). Kod tunelskog načina rada nekoliko entiteta ili cela jedna lokalna mreža sakriva se iza jednog čvora te je kao takva nevidljiva ostatku mreže, a samim tim i zaštićena od napada. Shodno tome, IPSec heder se dodaje ispred IP hedera poruke pa se ovako formiranoj poruci dodaje novi IP heder, shodno Slici 6.2(b).



(a)



(b)

Slika 6.2
Prikaz paketa prilikom: a) Transportnog načina rada; b) Tunelskog načina rada

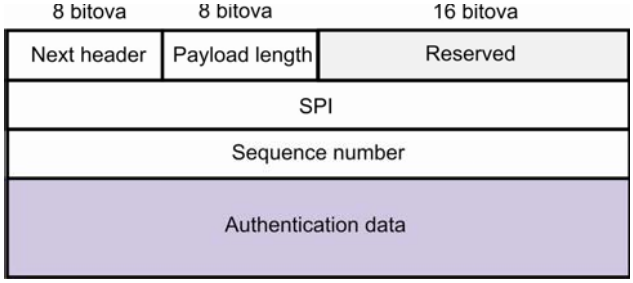
Kako standardni IP protokol nije pružao nikakve elemente zaštite, pred IPSec postavljeni su i sledeći zahtevi:

- kriptovanje podataka koji se prenose
- provera integriteta podataka
- autentificiranje entiteta kroz koje paket prolazi

U osnovi se IPsec deli na dva tipa protokola:

- AH protokol (eng. *Authentication Header*) i
- ESP protokol (eng. Encapsulating Security Payload),

***Authentication Header* (AH)** protokol je projektovan tako da garantuje autentičnost i integritet paketa podataka koji se prenose. Protokol korišćenjem neke od ranije navedenih heš funkcija (vidi poglavlje 3.3.) formira sažetak koji umeće u AH heder. AH heder se kasnije doda na odgovarajuću lokaciju u paketu, zavisno koji se način rada koristi (transportni ili tunelski). Izgled AH hedera je prikazan na Slici 6.3.



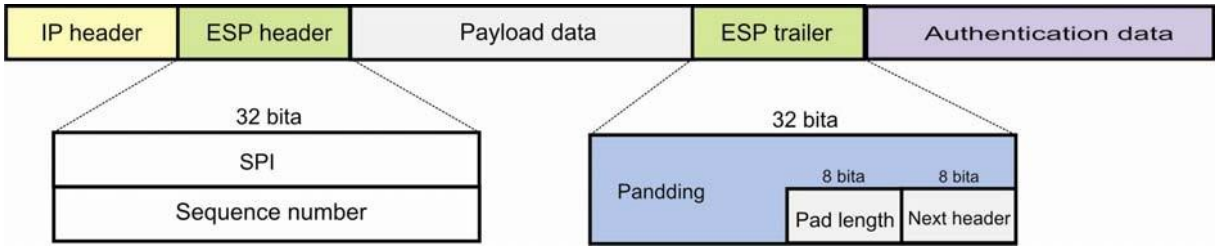
Slika 6.3
Izgled AH hedera

Značenje pojedinih polja u okviru AH je:

- *Next Header* – označava protokol koji se koristi za prenos podataka (TCP ili UDP)
- *Payload Length* – polje koje definiše dužinu AH
- *Reserved* – polje koje se još uvek ne koristi (svi bitovi se postavljaju na nulu)
- *Security Parameters Index* (SPI) – polje koje sadrži kombinacije sigurnosnih parametara
- *Sequence Number* – redni broj paketa, a koristi se za zaštitu od "replay" napada (radi se o napadima koji se zasnivaju na uzastopnom slanju istih paketa)
- *Authentication Data* – polje se koristi za umetanje sažetka poruke u cilju autentifikacije

AH protokol ne vrši autentifikaciju entiteta koji je poslao paket podataka, što predstavlja veliki nedostatak zbog koga je uveden alternativni protokol nazvan *Encapsulating Security Payload.*

Encapsulating Security Payload **(ESP)** protokol sadrži dva polja: ESP heder i ESP trailer. ESP heder se umeće odmah nakon IP hedera poruke a ESP trailer (veličine 32 bita) se umeće neposredno ispred podataka za autentifikaciju koji se nalaze na kraju poruke. Na Slici 6.4 je prikazana poruka koja sadrži pomenuta zaglavlja [34].



Slika 6.4
Prikaz paketa u okviru ESP protokola

Značenje pojedinačnih polja je:

- *Security Parameters Index* (SPI) - kombinacija sigurnosnih parametara slično kao i kod AH
- *Sequence Number* - redni broj paketa, koristi se za zaštitu od "replay" napada
- *Payload Data* – podaci koji se šalju
- *Padding* – koristi se za popunu podataka do veličine celog bloka (32 bita).
- *Pad Length* – polje označava dužinu *Padding* polja
- *Next Header* – označava protokol koji se koristi za prenos podataka (TCP ili UDP)
- *Authentication Data* – polje koje sadrži podatke potrebne za autentifikaciju entiteta i poruke.

Skup pravila koji definiše koje će strane u komunikaciji koristiti za zaštitu međusobnog saobraćaja je SA (eng. *Security Association*). SA sadrži sve sigurnosne parametre potrebne za siguran transport IPSec paketa kroz mrežu. SA su uvek jednosmerne, odnosno koriste se za zaštitu paketa u jednom smeru. Ukoliko je potrebno zaštititi saobraćaj u oba smera, potrebno je posedovati dve SA, i to po jedna na svakoj strani. SA se čuvaju u okviru SA baze podataka SADB (eng. *SA Data Base*). Skup pravila se čuva u SP (eng. *Security Policy*). Parametri koji čine SA su: 32-bitni sigurnosni parametar SPI, tip IPSec protokola (AH ili ESP) i izvorna IP adresa [10].

6.2 SSL/TLS

SSL (*Secure Socket Layer*) protokol koji je razvila firma *Netscape*, je trenutno najviše korišćen metod za obavljanje sigurnih transakcija na Internetu. Podržava ga većina Web servera kao i klijenata uključujući *Microsoft Internet Explorer, Netscape Navigator, Mozilla Firefox* i drugi. Posebna nezaštićena verzija SSL protokola 3.0 je poznata po imenu TLS (*Transport Layer Security*). Često se

SSL i TLS poistovećuju. WTLS (*Wap Transport Layer Security*) je specijalna verzija TLS protokola koja je razvijena za upotrebu u Wap aplikacijama [35].

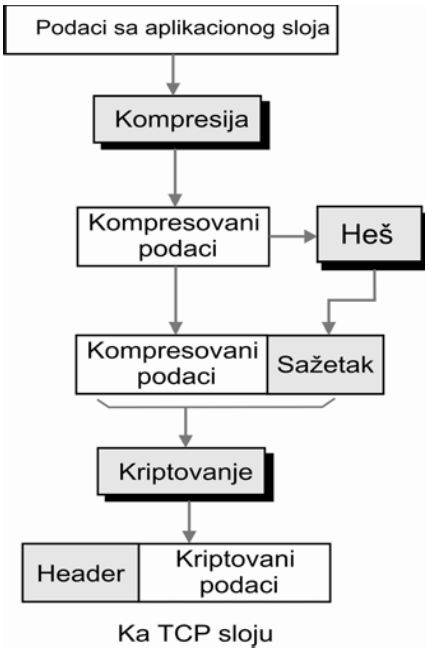
SSL obezbeđuje tajnost, integritet podataka i autentičnost pošiljalaca korišćenjem kombinacije kriptovanja javnim ključem (RSA), simetričnog kriptovanja (DES, RC5) i digitalnih sertifikata (SHA-1, MD5). Drugim rečima, SSL obezbeđuje mehanizme za identifikaciju servera, identifikaciju klijenta i kriptovanu razmenu podataka između njih. SSL je smešten iznad transportnog sloja, a ispod aplikativnog sloja od koga je nezavisan [36]. SSL se sastoji od dva protokola:

- SSL *Handshake* protokol
- SSL *Record* protokol

SSL *Handshake* protokol čini najsloženiji deo SSL protokola i omogućava uspostavljanje sesije. Tom prilikom SSL *Handshake* vrši proveru identiteta klijenta i identiteta servera, usaglašavanje algoritma za kriptovanje, izračunavanje sažetka i razmenu ključeva. Koristi se u fazi uspostavljanja konekcije pre bilo kakve razmene aplikacionih podataka. Ovaj protokol se sastoji od niza poruka koje se razmenjuju između starana u komunikaciji. Uspostavljanje veze se može prikazati kroz sledeće četiri faze:

- Postavljanje parametara veze
- Autentifikacija servera i razmena ključeva
- Autentifikacija klijenta i razmena ključeva
- Kraj uspostavljanja sigurne veze

Nakon uspostavljene veze otpočinje se sa razmenom kriptovanih aplikativnih podataka između klijenta i servera. SSL *Record* protokol koji se koristi na predajnoj strani vrši pripremu podataka za slanje (Slika 6.5). Prvo prihvata podatke od strane aplikativnog sloja, vrši njihovu podelu u blokove i ako je potrebno vrši komprimovanje. Nakon toga, u cilju autentifikacije vrši potpisivanje sažetka poruke pomoću neke od heš funkcija i poruci dodaje sažetak u vidu autentifikacionog koda (eng. *Message Authentation Code* – MAC). Tako formiranu poruku kriptuje, dodaje zaglavlje i predaje ih TCP sloju u cilju slanja. Na prijemnoj strani SSL *Record* protokol preuzima podatke od TCP nivoa, dekriptuje ih, vrši verifikaciju MAC-a, ukoliko je potrebno vrši dekompresiju podataka, spaja blokove kako bi ih u celini poslao aplikativnom sloju [10].



Slika 6.5
SSL Record protokol

SSL omogućava proveru identiteta pomoću sertifikata koji izdaje sertifikaciono telo CA. Ovaj vid SSL transakcije nudi zadovoljavajući nivo sigurnosti, te je relativno brzo postao standard za sigurnu komunikaciju. Upravo zbog ove činjenice, SSL protocol se najčešće koristi u aplikacijama za plaćanje kreditnim karticama.

7 Zaključak

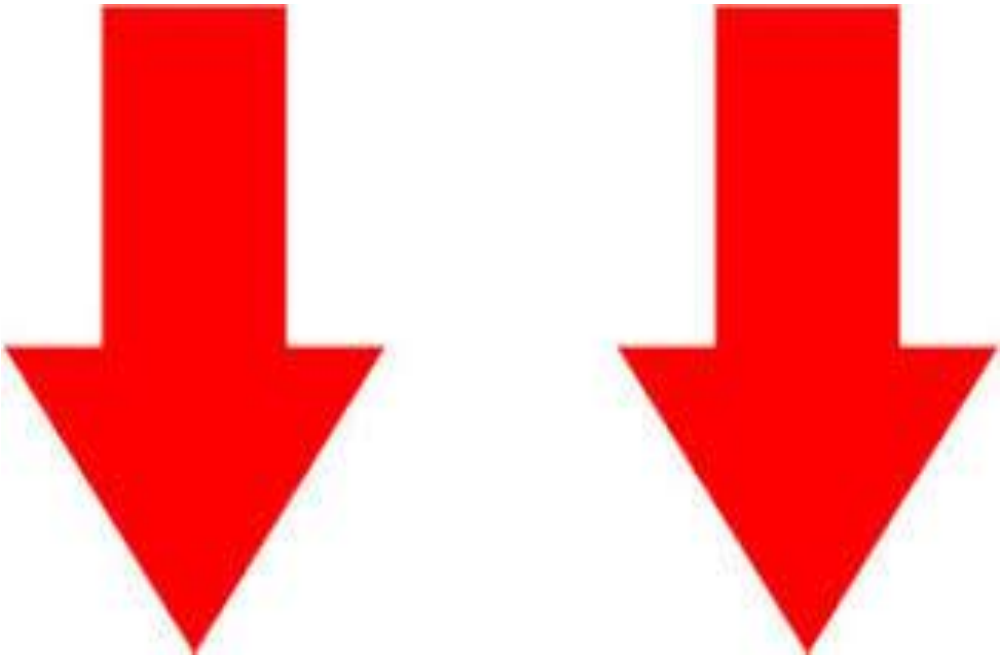
Nagli razvoj tehnologije računarskih sistema i mreža, doveo je do masovne razmene informacija i podataka. U toj razmeni pojavila se potreba za zaštitom poverljivih podataka od raznih krađa i zloupotreba, što je uslovilo pojavu novog pravca u oblasti bezbednosti podataka. Dalji razvoj tehnologije doprineo je poboljšanju karakteristika metoda koje doprinose bezbednosti podataka. Međutim, isti taj napredak doveo je do bržeg otkrivanja nedostataka i dodatno omogućio eventualne zloupotrebe. Tako su, tokom vremena, metode za zaštitu podataka postajale sve složenije i naprednije.

U osnovi priče o bezbednosti podataka u računarskim mrežama nalazi se Kriptografija. Kriptografija je veoma dinamična nauka koju odlikuje uska povezanost između teorije i prakse. Na ovo ukazuje činjenica da se napredak u teoriji brzo implementira u praksi, gde se još brže testira. Otkrivanjem nedostataka odmah se unapređuje teorijski rad i stečena iskustva se upotrebljavaju za izradu nove i bolje metode. Kriptografija je veoma široka oblast, a u praksi se bazira na upotrebi kriptosistema koji se sastoje od algoritama za kriptovanje, jednog ili više ključeva, sistema za upravljanje ključevima, podataka u vidu standardnog i kriptovanog teksta. Na složenost dodatno ukazuju i činjenice da se kriptosistemi mogu realizovati hardverski, softverski ili hardversko-softverski, kao i to da se prilikom realizacije moraju zadovoljiti osnovni sigurnosni servisi. Za realizaciju kriptografskih algoritma, koji su danas u upotrebi, koriste se složeni matematički izrazi kao i znanja iz elektronike i programiranja. Uprkos tome, napredak u postojećim kriptografskim algoritmima nastavlja se rastućim tempom, da bi zadovoljio potrebe širenja IT društva u kome živimo. Teško je zamisliti ozbiljnu PC aplikaciju koja u sebi nema implementiran neki sigurnosni algoritam, počev od bankarskih aplikacija, internet trgovine, pa sve do operativnih sistema. Takođe, kriptografski algoritmi za zaštitu su primenjivi u telekomunikacionim sistemima, televiziji, i drgim oblastima za koje često nismo ni svesni da ih čine ovi algoritami.

Imajući u vidu činjenice da je Kriptografija veoma dinamična oblast, da je aktuelna i da je veoma rasprostranjena, ovim radom su obuhvaćeni samo njeni osnovni koncepti. Razmatrane su kriptografske metode i algoritmi, bez upotrebe apstraktnih matematičkih izraza i teorija o brojevima. Ukazano je na osnovne sigurnosne servise koji se u cilju sigurnosti moraju zadovoljiti primenom kriptografskih metoda. Bilo je govora i o efikasnosti pojedinih algoritama, kao i o isplativosti njihove hardverske realizacije. Obrađeni su problemi kojii se javljaju tokom upotrebe i distribucije ključeva koji se koriste prilikom kriptovanja. Na kraju su dati primeri sigurnosnih servisa koji se upotrebljavaju za zaštitu podataka u okviru TCP/IP modela, a koji se baziraju na opisanim kriptografskim algoritmima.

Cilj ovog rada je da ukaže na nužnost zaštite podataka i prikaže osnovne mehanizme za njihovu zaštitu, koji se prenose putem računarske mreže. Zbog fizičkih i funkcionalnih karakteristika računarskih mreža, veoma često se javlja opasnost od ugrožavanja podataka prilikom prenosa. Predmet istraživanja rada je upoznavanje sa metodama koje se danas koriste da onemoguće zloupotrebu podataka. Pregledom najčešće korišćenih algoritama za kriptovanje, tehnika za autentifikaciju, problema prilikom upravljanja sa ključevima i sigurnosnim servisima u TCP/IP modelu, saznaju se osnove o zaštiti podataka. Uporednom analizom karakteristika određenih metoda za zaštitu ukazuje se na prednosti i/ili nedostatke. U cilju što efikasnijeg prikaza složenih procedura upotrebljen je veliki broj praktičnih primera i ilustracija.

Sigurno je da će se kriptografija razvijati sve bržim i bržim koracima. Upotreba kvantnih računara, računarskih gridova kao i sve masovnije upotrebe računarskih mreža samo su jedan od nagoveštaja budućnosti. U skladu sa tim će se postavljati sve veći sigurnosni zahtevi u smislu ranog otkrivanja i sprečavanja zloupotreba. Odgovor na ove zahteve je upravo upotreba savremenijih i sigurnijih kriptografskih algoritama, koji će prevazilaziti granice jednostavne bezbednosti.



Ima još jedna stranica dole hehe

Radio: Milutin Milosavljević